

概要

このドキュメントは、アプリケーション開発者を対象としています。STM32H562xx、STM32H563xx、および STM32H573xx マイクロコントローラのメモリとペリフェラルの使用方法について詳しく説明します。

注文情報、機械的および電氣的なデバイスの特性については、対応するデータシートを参照してください。

Arm[®] Cortex[®]-M33 コアについては、対応する Arm[®] テクニカルリファレンスマニュアルを参照してください (<http://infocenter.arm.com>)。

関連ドキュメント

- STM32H563/H573 および STM32H562 データシート
- STM32H563/H573 および STM32H562 エラッタ

目次

1	このマニュアルにおける表記の規則	100
1.1	一般情報	100
1.2	レジスタに関する略記	100
1.3	用語	101
2	メモリとバスのアーキテクチャ	102
2.1	システムアーキテクチャ	102
2.1.1	高速 C バス	103
2.1.2	低速 C バス	103
2.1.3	S- バス	103
2.1.4	DCache S バス	104
2.1.5	GPDMA1 および GPDMA2 バス	104
2.1.6	SDMMC1 および SDMMC2 コントローラ DMA バス	104
2.1.7	バスマトリックス	104
2.1.8	AHB/APB ブリッジ	104
2.1.9	Ethernet MAC	104
2.2	TrustZone セキュリティアーキテクチャ	105
2.2.1	デフォルトの TrustZone セキュリティ状態	105
2.2.2	TrustZone ペリフェラルの分類	106
2.3	メモリ構成	110
2.3.1	概要	110
2.3.2	メモリマップとレジスタ境界アドレス	111
2.3.3	内蔵 SRAM	117
2.3.4	Flash メモリの概要	117
2.3.5	ブートモード	117
3	システムセキュリティ	119
3.1	主なセキュリティ機能	119
3.2	セキュアインストール	120
3.3	セキュアブート	121
3.3.1	ユニークブートエントリ	121
3.3.2	システム Flash メモリの不変の信頼の起点	122
3.4	セキュア更新	122

3.5	非表示保護レベルを使用したリソース分離	123
3.6	TrustZone を使用したリソースの分離	123
3.6.1	TrustZone セキュリティアーキテクチャ	124
3.6.2	Cortex-M33 の Armv8-M セキュリティ拡張	124
3.6.3	IDAU/SAU を使用したメモリおよびペリフェラルの割り当て	125
3.6.4	GTZC を使用したメモリおよびペリフェラルの割り当て	126
3.6.5	TrustZone 対応ペリフェラルのセキュリティの管理	130
3.6.6	TrustZone セキュリティの有効化	136
3.6.7	TrustZone セキュリティの無効化	137
3.7	その他のリソース分離	137
3.7.1	セキュア非表示保護（HDP）を使用した時間的分離	137
3.7.2	Cortex 特権モードを使用したリソース分離	138
3.8	セキュアな実行	141
3.8.1	メモリ保護ユニット（MPU）	141
3.8.2	内蔵 Flash メモリ書き込み保護	142
3.8.3	タンパの検出と対応	142
3.9	セキュアストレージ	145
3.9.1	ハードウェア秘密キー管理	145
3.9.2	ユニーク ID	146
3.10	暗号化エンジン	146
3.10.1	暗号化エンジンの機能	146
3.10.2	セキュア AES コプロセッサ（SAES）	147
3.10.3	オンザフライ復号エンジン（OTFDEC）	148
3.11	製品ライフサイクル	149
3.11.1	製品の設定とセキュリティサービス	150
3.11.2	ライフサイクル管理	151
3.11.3	推奨されるプロトコル設定	154
3.12	ソフトウェア知的財産保護と共同開発	154
3.12.1	ソフトウェア知的財産保護	156
3.12.2	OTFDEC によるソフトウェア知的財産保護	156
3.12.3	その他のソフトウェア知的財産保護	158
4	ブートモード	159
4.1	STM32H562/H563 ブートモード	160
4.2	STM32H573x ブートモード	161

5	グローバル TrustZone® コントローラ (GTZC)	162
5.1	概要	162
5.2	GTZC の主な機能	162
5.3	GTZC の実装	164
5.4	GTZC の機能詳細	166
5.4.1	GTZC ブロック図	166
5.4.2	不正アクセスの定義	167
5.4.3	TrustZone セキュリティ・コントローラ (TZSC)	167
5.4.4	ブロックベースのメモリ保護コントローラ (MPCBB)	169
5.4.5	TrustZone 不正アクセスコントローラ (TZIC)	169
5.4.6	パワーオン/リセット状態	170
5.5	GTZC 割込み	170
5.6	GTZC1 TZSC レジスタ	171
5.6.1	GTZC1 TZSC 制御レジスタ (GTZC1_TZSC_CR)	171
5.6.2	GTZC1 TZSC セキュア設定レジスタ 1 (GTZC1_TZSC_SECCFGR1)	171
5.6.3	GTZC1 TZSC セキュア設定レジスタ 2 (GTZC1_TZSC_SECCFGR2)	174
5.6.4	GTZC1 TZSC セキュア設定レジスタ 3 (GTZC1_TZSC_SECCFGR3)	176
5.6.5	GTZC1 TZSC 特権設定レジスタ 1 (GTZC1_TZSC_PRIVCFGR1)	178
5.6.6	GTZC1 TZSC 特権設定レジスタ 2 (GTZC1_TZSC_PRIVCFGR2)	181
5.6.7	GTZC1 TZSC 特権設定レジスタ 3 (GTZC1_TZSC_PRIVCFGR3)	183
5.6.8	GTZC1 TZSC メモリ x サブ領域 z ウォーターマーク設定レジスタ (GTZC1_TZSC_MPCWMxzCFGR) (z = A から B)	185
5.6.9	GTZC1 TZSC メモリ x サブ領域 A ウォーターマーク・レジスタ (GTZC1_TZSC_MPCWMxAR)	186
5.6.10	GTZC1 TZSC メモリ x サブ領域 B ウォーターマーク・レジスタ (GTZC1_TZSC_MPCWMxBR)	187
5.6.11	GTZC1 TZSC レジスタのマップ	188
5.7	GTZC1 TZIC レジスタ	190
5.7.1	GTZC1 TZIC 割込み有効レジスタ 1 (GTZC1_TZIC_IER1)	190
5.7.2	GTZC1 TZIC 割込み有効レジスタ 2 (GTZC1_TZIC_IER2)	192
5.7.3	GTZC1 TZIC 割込み有効レジスタ 3 (GTZC1_TZIC_IER3)	194
5.7.4	GTZC1 TZIC 割込み有効レジスタ 4 (GTZC1_TZIC_IER4)	196
5.7.5	GTZC1 TZIC ステータス・レジスタ 1 (GTZC1_TZIC_SR1)	198
5.7.6	GTZC1 TZIC ステータス・レジスタ 2 (GTZC1_TZIC_SR2)	201
5.7.7	GTZC1 TZIC ステータス・レジスタ 3 (GTZC1_TZIC_SR3)	203
5.7.8	GTZC1 TZIC ステータス・レジスタ 4 (GTZC1_TZIC_SR4)	205
5.7.9	GTZC1 TZIC フラグ・クリア・レジスタ 1 (GTZC1_TZIC_FCR1)	207

5.7.10	GTZC1 TZIC フラグ・クリア・レジスタ 2 (GTZC1_TZIC_FCR2)	209
5.7.11	GTZC1 TZIC フラグ・クリア・レジスタ 3 (GTZC1_TZIC_FCR3)	211
5.7.12	GTZC1 TZIC フラグ・クリア・レジスタ 4 (GTZC1_TZIC_FCR4)	213
5.7.13	GTZC1 TZIC レジスタ・マップ	216
5.8	GTZC1 MPCBBz レジスタ (z = 1 から 3)	218
5.8.1	GTZC1 SRAMz MPCBB 制御レジスタ (GTZC1_MPCBBz_CR) (z = 1 から 3)	218
5.8.2	GTZC1 SRAMz MPCBB 設定ロック・レジスタ 1 (GTZC1_MPCBBz_CFGLOCK1) (z = 1 から 3)	219
5.8.3	スーパーブロック x レジスタの GTZC1 SRAMz MPCBB セキュリティ設定 (GTZC1_MPCBBz_SECCFGRx) (z = 1 から 3)	219
5.8.4	スーパーブロック x レジスタの GTZC1 SRAMz MPCBB 特権設定 (GTZC1_MPCBBz_PRIVCFGRx) (z = 1 から 3)	220
5.8.5	GTZC1 MPCBBz レジスタ・マップ (z = 1 から 3)	220
6	RAM 設定コントローラ (RAMCFG)	221
6.1	概要	221
6.2	RAMCFG の主な機能	221
6.3	RAMCFG の機能詳細	221
6.3.1	内部 SRAM 機能	221
6.3.2	エラーコード訂正 (SRAM2、SRAM3、BKPSRAM)	222
6.3.3	書き込み保護 (SRAM2)	224
6.3.4	ソフトウェア消去	224
6.4	RAMCFG 低電力モード	224
6.5	RAMCFG 割込み	224
6.6	RAMCFG レジスタ	225
6.6.1	RAMCFG メモリ x 制御レジスタ (RAMCFG_MxCR)	225
6.6.2	RAMCFG メモリ x 割込みイネーブルレジスタ (RAMCFG_MxIER)	226
6.6.3	RAMCFG メモリ割込みステータスレジスタ (RAMCFG_MxISR)	226
6.6.4	RAMCFG メモリ x ECC シングルエラーアドレスレジスタ (RAMCFG_MxSEAR)	227
6.6.5	RAMCFG メモリ x ECC ダブルエラーアドレスレジスタ (RAMCFG_MxDEAR)	227
6.6.6	RAMCFG メモリ x 割込みクリアレジスタ x (RAMCFG_MxICR)	228
6.6.7	RAMCFG メモリ 2 書き込み保護レジスタ 1 (RAMCFG_M2WPR1)	228
6.6.8	RAMCFG メモリ 2 書き込み保護レジスタ 2 (RAMCFG_M2WPR2)	229
6.6.9	RAMCFG メモリ x ECC キーレジスタ (RAMCFG_MxECCKEYR)	229
6.6.10	RAMCFG メモリ x 消去キーレジスタ (RAMCFG_MxERKEYR)	230

6.6.11	RAMCFG レジスタマップ	230
7	内蔵 Flash メモリ (FLASH)	233
7.1	概要	233
7.2	FLASH の主な機能	233
7.3	Flash の機能説明	234
7.3.1	FLASH のブロック図	234
7.3.2	FLASH 信号	235
7.3.3	Flash メモリのアーキテクチャおよび使用	236
7.3.4	FLASH 読出し操作	238
7.3.5	FLASH プログラム操作	240
7.3.6	FLASH 消去操作	243
7.3.7	FLASH 並行操作	247
7.3.8	FLASH エラー保護	247
7.3.9	OTP および RO メモリ・アクセス	247
7.3.10	Flash の高サイクル・データ	250
7.3.11	Flash バンク・スワッピング	251
7.3.12	FLASH のリセットとクロック	253
7.4	FLASH のオプション・バイト	255
7.4.1	オプション・バイトについて	255
7.4.2	オプション・バイトのローディング	255
7.4.3	オプション・バイトの変更	255
7.4.4	ユーザ・オプション・バイトとシステムオプション・バイトの説明	258
7.4.5	データ保護オプション・バイトの説明	259
7.4.6	ブートアドレスオプション・バイトの説明	260
7.4.7	オプション・バイト変更の特定の規則	260
7.5	オプション・バイトのキー・マネージメント	262
7.5.1	OBK のローディング	262
7.5.2	HDPL レベルごとの OBK アクセス	262
7.5.3	OBK プログラミング・シーケンス	263
7.5.4	OBK プログラミング有限ステート・マシン	265
7.5.5	OBK スワップ・セクタ	265
7.5.6	OBK 代替セクタ消去	268
7.6	FLASH のセキュリティと保護	268
7.6.1	TrustZone のセキュリティ保護	269
7.6.2	非表示保護 (HDP)	271

7.6.3	ブロックベースのセキュア Flash メモリの領域保護	272
7.6.4	ブロックベースの特権 Flash メモリの領域保護	273
7.6.5	Flash メモリ・レジスタの特権および非特権モード	275
7.6.6	バンク・スワップの際の Flash メモリ・バンク属性	275
7.6.7	Flash メモリ設定保護	277
7.6.8	書き込み保護	277
7.6.9	Flash の高サイクル・データ保護	277
7.6.10	ライフ・サイクル・マネージメント	279
7.6.11	製品状態の遷移	280
7.6.12	OBK 保護	282
7.6.13	一度だけプログラム可能な（ワンタイムプログラマブル）、 読出し専用メモリの保護	283
7.7	システムメモリ	284
7.7.1	概要	284
7.7.2	RSS ユーザ機能	284
7.8	FLASH 低電力モード	291
7.9	FLASH エラー管理	292
7.9.1	概要	292
7.9.2	非セキュア書き込み保護エラー（WRPERR）	292
7.9.3	セキュア書き込み保護エラー（WRPERR）	293
7.9.4	非セキュア・プログラミング・シーケンス・エラー（PGSERR）	294
7.9.5	セキュア・プログラミング・シーケンス・エラー（PGSERR）	295
7.9.6	非セキュア・ストローブ・エラー（STRBERR）	296
7.9.7	セキュア・ストローブ・エラー（STRBERR）	296
7.9.8	非セキュア不整合エラー（INCERR）	297
7.9.9	セキュア不整合エラー（INCERR）	297
7.9.10	エラー訂正コード・エラー（ECCC、ECCD）	298
7.9.11	不正アクセス（ILAFM/ILAP）	299
7.9.12	オプション・バイト変更エラー（OPTCHANGEERR）	299
7.9.13	その他のハードフォルトエラー	300
7.9.14	OBK エラー事例（OBKERR、OBKWERR）	300
7.10	FLASH 割込み	301
7.11	Flash レジスタ	303
7.11.1	Flash アクセス制御レジスタ（Flash_ACR）	303
7.11.2	Flash 非セキュア・キー・レジスタ（Flash_NSKEYR）	304
7.11.3	Flash セキュア・キー・レジスタ（Flash_SECKEYR）	304
7.11.4	Flash オプションキーレジスタ（Flash_OPTKEYR）	305

7.11.5	Flash 非セキュア・OBK キー・レジスタ (FLASH_NS0BKKEYR)	305
7.11.6	Flash セキュア OBK キー・レジスタ (FLASH_SECOBKKEYR)	306
7.11.7	Flash 操作ステータス・レジスタ (FLASH_OPSR)	306
7.11.8	Flash オプション制御レジスタ (Flash_OPTCR)	307
7.11.9	Flash 非セキュアステータスレジスタ (Flash_NSSR)	308
7.11.10	FLASH セキュア・ステータス・レジスタ (Flash_SECSR)	311
7.11.11	Flash 非セキュア制御レジスタ (Flash_NSCR)	313
7.11.12	Flash セキュア制御レジスタ (Flash_SECCR)	316
7.11.13	FLASH 非セキュア・クリア制御レジスタ (FLASH_NSCCR)	319
7.11.14	FLASH セキュア クリア 制御レジスタ (FLASH_SECCR)	319
7.11.15	Flash 特権設定レジスタ (FLASH_PRIVCFGR)	321
7.11.16	FLASH 非セキュア・OBK 設定・レジスタ (FLASH_NS0BKCFGR)	321
7.11.17	FLASH セキュア・OBK 設定・レジスタ (FLASH_SECOBKCFGR)	323
7.11.18	FLASH HDP 拡張レジスタ (FLASH_HDPEXTR)	324
7.11.19	FLASH オプションステータスレジスタ (FLASH_OPTSR_CUR)	324
7.11.20	FLASH オプション・ステータス・レジスタ (FLASH_OPTSR_PRG)	326
7.11.21	FLASH 非セキュア・エポック・タイム・レジスタ (FLASH_NSEPOCHR_CUR)	328
7.11.22	FLASH セキュア・エポック・タイム・レジスタ (FLASH_SECEPOCHR_CUR)	328
7.11.23	FLASH オプション・ステータス・レジスタ 2 (FLASH_OPTSR2_CUR)	329
7.11.24	FLASH オプション・ステータス・レジスタ 2 (FLASH_OPTSR2_PRG)	330
7.11.25	FLASH 非セキュア・ブート・レジスタ (FLASH_NSBOOTR_CUR)	331
7.11.26	FLASH 非セキュア・ブート・レジスタ (FLASH_NSBOOTR_PRG)	332
7.11.27	FLASH セキュア・ブート・レジスタ (FLASH_SECBOOTR_CUR)	332
7.11.28	FLASH セキュア・ブート・レジスタ (FLASH_BOOTR_PRG)	333
7.11.29	FLASH 非セキュア OTP ブロックのロック (FLASH_OTPBLR_CUR)	333
7.11.30	FLASH 非セキュア OTP ブロックのロック (FLASH_OTPBLR_PRG)	334
7.11.31	バンク 1 の FLASH セキュア・ブロック・ベースのレジスタ (FLASH_SECBB1Rx)	334
7.11.32	バンク 1 の FLASH 特権ブロック・ベースのレジスタ (FLASH_PRIVBB1Rx)	335
7.11.33	バンク 1 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM1R_CUR)	335
7.11.34	バンク 1 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM1R_PRG)	336
7.11.35	バンク 1 の FLASH 書き込みセクタ・グループ保護 (FLASH_WRP1R_CUR)	336
7.11.36	バンク 1 の FLASH 書き込みセクタ・グループ保護 (FLASH_WRP1R_PRG)	337
7.11.37	FLASH データ・セクタ設定、バンク 1 (FLASH_EDATA1R_CUR)	337

7.11.38	FLASH データ・セクタ設定、バンク 1 (FLASH_EDATA1R_PRG)	338
7.11.39	FLASH HDP バンク 1 設定 (FLASH_HDP1R_CUR)	339
7.11.40	FLASH HDP バンク 1 設定 (FLASH_HDP1R_PRG)	339
7.11.41	FLASH ECC 訂正レジスタ (FLASH_ECCCORR)	340
7.11.42	FLASH ECC 検出レジスタ (FLASH_ECCDETR)	341
7.11.43	FLASH ECC データ (FLASH_ECCDR)	342
7.11.44	バンク 2 の FLASH セキュア・ブロックベースのレジスタ (FLASH_SECBB2Rx)	342
7.11.45	バンク 2 の FLASH 特権ブロックベースのレジスタ (FLASH_PRIVBB2Rx)	343
7.11.46	バンク 2 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM2R_CUR)	343
7.11.47	バンク 2 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM2R_PRG)	344
7.11.48	バンク 2 の FLASH 書き込みセクタ・グループ保護 (FLASH_WRP2R_CUR) ...	344
7.11.49	バンク 2 の FLASH 書き込みセクタ・グループ保護 (FLASH_WRP2R_PRG) ...	345
7.11.50	FLASH データ・セクタ設定、バンク 2 (FLASH_EDATA2R_CUR)	345
7.11.51	FLASH データ・セクタ設定、バンク 2 (FLASH_EDATA2R_PRG)	346
7.11.52	FLASH HDP バンク 2 設定 (FLASH_HDP2R_CUR)	347
7.11.53	FLASH HDP バンク 2 設定 (FLASH_HDP2R_PRG)	347
7.12	Flash のレジスタ・マップとリセット値	348
8	命令キャッシュ (ICACHE)	353
8.1	ICACHE の概要	353
8.2	ICACHE の主な機能	353
8.3	ICACHE の実装	354
8.4	ICACHE の機能説明	354
8.4.1	ICACHE ブロック図	355
8.4.2	ICACHE のリセットおよびクロック	355
8.4.3	ICACHE TAG メモリ	356
8.4.4	ダイレクトマップ ICACHE (1 ウェイキャッシュ)	357
8.4.5	ICACHE 有効化	358
8.4.6	キャッシュ格納可および不可のトラフィック	358
8.4.7	アドレス再配置	359
8.4.8	キャッシュ可能アクセス	361
8.4.9	デュアルマスタキャッシュ	362
8.4.10	ICACHE のセキュリティ	362
8.4.11	ICACHE のメンテナンス	362

8.4.12	ICACHE 性能の監視	363
8.4.13	ICACHE ブート	363
8.5	ICACHE 低電力モード	363
8.6	ICACHE のエラー管理と割込み	364
8.7	ICACHE レジスタ	364
8.7.1	ICACHE 制御レジスタ (ICACHE_CR)	364
8.7.2	ICACHE ステータスレジスタ (ICACHE_SR)	365
8.7.3	ICACHE 割込み有効化レジスタ (ICACHE_IER)	366
8.7.4	ICACHE フラグクリアレジスタ (ICACHE_FCR)	366
8.7.5	ICACHE ヒットモニタレジスタ (ICACHE_HMONR)	367
8.7.6	ICACHE ミスモニタレジスタ (ICACHE_MMONR)	367
8.7.7	ICACHE 領域 x 設定レジスタ (ICACHE_CRRx)	368
8.7.8	ICACHE レジスタマップ	369
9	データキャッシュ (DCACHE)	370
9.1	DCACHE の概要	370
9.2	DCACHE の主な機能	370
9.3	DCACHE の実装	371
9.4	DCACHE の機能説明	371
9.4.1	DCACHE ブロック図	372
9.4.2	DCACHE のリセットおよびクロック	372
9.4.3	DCACHE TAG メモリ	373
9.4.4	DCACHE 有効化	375
9.4.5	キャッシュ格納可および不可のトラフィック	375
9.4.6	キャッシュ可能アクセス	376
9.4.7	DCACHE セキュリティ	378
9.4.8	DCACHE のメンテナンス	378
9.4.9	DCACHE 性能の監視	380
9.4.10	DCACHE ブート	380
9.5	DCACHE 低電力モード	380
9.6	DCACHE のエラー管理と割込み	380
9.7	DCACHE レジスタ	381
9.7.1	DCACHE 制御レジスタ (DCACHE_CR)	381
9.7.2	DCACHE ステータスレジスタ (DCACHE_SR)	383
9.7.3	DCACHE 割込みイネーブルレジスタ (DCACHE_IER)	384
9.7.4	DCACHE フラグクリアレジスタ (DCACHE_FCR)	384

9.7.5	DCACHE 読出しヒットモニタレジスタ (DCACHE_RHMONR)	385
9.7.6	DCACHE 読出しミスモニタレジスタ (DCACHE_RMMONR)	385
9.7.7	DCACHE 書込みヒットモニタレジスタ (DCACHE_WHMONR)	386
9.7.8	DCACHE 書込みミスモニタレジスタ (DCACHE_WMMONR)	386
9.7.9	DCACHE コマンド範囲開始アドレスレジスタ (DCACHE_CMDRSADDR)	386
9.7.10	DCACHE コマンド範囲終了アドレスレジスタ (DCACHE_CMDREADDR)	387
9.7.11	DCACHE レジスタマップ	387
10	電源制御 (PWR)	389
10.1	概要	389
10.2	PWR の主な機能	389
10.3	PWR ピンおよび内部信号	390
10.4	PWR 電源供給と電源ドメイン	391
10.4.1	外部電源	392
10.4.2	内部レギュレータ	393
10.4.3	パワーアップおよびパワーダウン電源シーケンス	395
10.4.4	独立したアナログペリフェラルの電源	395
10.4.5	独立 I/O 電源レール	396
10.4.6	独立した USB トランシーバの電源	396
10.4.7	バックアップドメイン	396
10.5	PWR システム電源電圧レギュレーション	398
10.5.1	SMPS および LDO 内蔵レギュレータ	398
10.5.2	V _{CORE} 電源対リセット、電圧スケーリング、および低電力モード	398
10.5.3	内蔵電圧レギュレータの動作モード	398
10.6	PWR 電源および温度監視	399
10.6.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	399
10.6.2	ブラウンアウトリセット (BOR)	399
10.6.3	プログラム可能な電圧検出器 (PVD)	400
10.6.4	アナログ電圧検出器 (AVD)	401
10.6.5	VDDIO2 電圧モニタ (IO2VM)	402
10.6.6	バックアップドメイン電圧監視	402
10.6.7	温度監視	403
10.7	PWR 管理	404
10.7.1	電圧スケーリング	404
10.7.2	電源管理の例	405

10.8	電源モード	406
10.8.1	システムクロックの低速化	410
10.8.2	ペリフェラルクロックゲーティング	410
10.8.3	低電力モード	410
10.8.4	SLEEP モード	411
10.8.5	STOP モード	412
10.8.6	STANDBY モード	414
10.8.7	電力モード出力ピン	417
10.9	PWR セキュリティおよび特権保護	417
10.9.1	PWR セキュリティ保護	417
10.9.2	PWR 特権保護	418
10.10	PWR 割込み	419
10.11	PWR レジスタ	420
10.11.1	PWR 電力モード制御レジスタ (PWR_PMCR)	420
10.11.2	PWR ステータスレジスタ (PWR_PMSR)	422
10.11.3	PWR 電圧スケーリング制御レジスタ (PWR_VOSCR)	422
10.11.4	PWR 電圧スケーリングステータスレジスタ (PWR_VOSSR)	423
10.11.5	PWR バックアップドメイン制御レジスタ (PWR_BDCR)	424
10.11.6	PWR 無効化バックアップドメイン制御レジスタ (PWR_DBPCR)	425
10.11.7	PWR バックアップドメインステータスレジスタ (PWR_BDSR)	425
10.11.8	PWR USB タイプ C パワーデリバリレジスタ (PWR_UCPDR)	426
10.11.9	PWR 電源設定制御レジスタ (PWR_SCCR)	427
10.11.10	PWR 電圧モニタ制御レジスタ (PWR_VMCR)	428
10.11.11	PWR USB 電源制御レジスタ (PWR_USBSCR)	429
10.11.12	PWR 電圧モニタステータスレジスタ (PWR_VMSR)	429
10.11.13	PWR ウェイクアップステータスクリアレジスタ (PWR_WUSCR)	430
10.11.14	PWR ウェイクアップステータスレジスタ (PWR_WUSR)	431
10.11.15	PWR ウェイクアップ設定レジスタ (PWR_WUCR)	431
10.11.16	PWR I/O 保持レジスタ (PWR_IORETR)	432
10.11.17	PWR セキュリティ設定レジスタ (PWR_SECCFGR)	433
10.11.18	PWR 特権設定レジスタ (PWR_PRIVCFGR)	434
10.11.19	PWR レジスタマップ	435
11	リセットおよびクロック制御 (RCC)	437
11.1	概要	437
11.2	RCC ピンおよび内部信号	437

11.3	RCC リセット機能の説明	437
11.3.1	電源リセット	437
11.3.2	システムリセット	438
11.3.3	バックアップドメインリセット	439
11.3.4	リセットソース識別	439
11.4	RCC クロック機能の説明	440
11.4.1	HSE クロック	442
11.4.2	HSI クロック	443
11.4.3	CSI オシレータ	444
11.4.4	HSI48 クロック	445
11.4.5	PLL の説明	445
11.4.6	LSE クロック	449
11.4.7	LSI クロック	450
11.4.8	システムクロック (SYSCLK) の選択	451
11.4.9	STOP モードおよび STANDBY モードでのクロックジェネレータの処理	451
11.4.10	クロックセキュリティシステム (CSS)	452
11.4.11	クロック出力の生成 (MCO1/MCO2)	454
11.4.12	カーネルクロックの選択	454
11.4.13	RTC および TAMP クロック	458
11.4.14	タイマクロック	458
11.4.15	ウォッチドッグクロック	458
11.4.16	ペリフェラルクロックゲーティングおよび自律モード	458
11.5	RCC セキュリティおよび特権機能の説明	459
11.5.1	RCC TrustZone セキュリティ保護モード	459
11.5.2	RCC 特権保護モード	462
11.6	RCC 低電力モード	462
11.7	RCC 割込み	464
11.8	RCC レジスタ	466
11.8.1	RCC クロック制御レジスタ (RCC_CR)	466
11.8.2	RCC HSI 較正レジスタ (RCC_HSI CFGR)	469
11.8.3	RCC クロックリカバリ RC レジスタ (RCC_CRR CR)	470
11.8.4	RCC CSI 較正レジスタ (RCC_CSI CFGR)	470
11.8.5	RCC クロック設定レジスタ 1 (RCC_CFGR1)	471
11.8.6	RCC CPU ドメインクロック設定レジスタ 2 (RCC_CFGR2)	474
11.8.7	RCC PLL1 クロックソース選択レジスタ (RCC_PLL1 CFGR)	476
11.8.8	RCC PLL2 クロックソース選択レジスタ (RCC_PLL2 CFGR)	478

11.8.9	RCC PLL3 クロックソース選択レジスタ (RCC_PLL3CFGR)	479
11.8.10	RCC PLL1 分周器レジスタ (RCC_PLL1DIVR)	481
11.8.11	RCC PLL1 分数分周器レジスタ (RCC_PLL1FRACR)	482
11.8.12	RCC PLL1 分周器レジスタ (RCC_PLL2DIVR)	483
11.8.13	RCC PLL2 小数分周器レジスタ (RCC_PLL2FRACR)	484
11.8.14	RCC PLL3 分周器レジスタ (RCC_PLL3DIVR)	485
11.8.15	RCC PLL3 小数分周器レジスタ (RCC_PLL3FRACR)	486
11.8.16	RCC クロックソース割込み有効レジスタ (RCC_CIER)	487
11.8.17	RCC クロックソース割込みフラグレジスタ (RCC_CIFR)	488
11.8.18	RCC クロックソース割込みクリアレジスタ (RCC_CICR)	490
11.8.19	RCC AHB1 リセットレジスタ (RCC_AHB1RSTR)	491
11.8.20	RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR)	492
11.8.21	RCC AHB4 ペリフェラルリセットレジスタ (RCC_AHB4RSTR)	494
11.8.22	RCC APB1 ペリフェラル低リセットレジスタ (RCC_APB1LRSTR)	495
11.8.23	RCC APB1 ペリフェラル高リセットレジスタ (RCC_APB1HRSTR)	498
11.8.24	RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)	499
11.8.25	RCC APB3 ペリフェラルリセットレジスタ (RCC_APB3RSTR)	500
11.8.26	RCC AHB1 ペリフェラルクロックレジスタ (RCC_AHB1ENR)	502
11.8.27	RCC AHB2 ペリフェラルクロックレジスタ (RCC_AHB2ENR)	504
11.8.28	RCC AHB4 ペリフェラルクロックレジスタ (RCC_AHB4ENR)	506
11.8.29	RCC APB1 ペリフェラルクロックレジスタ (RCC_APB1LENR)	507
11.8.30	RCC APB1 ペリフェラルクロックレジスタ (RCC_APB1HENR)	510
11.8.31	RCC APB2 ペリフェラルクロックレジスタ (RCC_APB2ENR)	511
11.8.32	RCC APB3 ペリフェラルクロックレジスタ (RCC_APB3ENR)	512
11.8.33	RCC AHB1 スリープクロックレジスタ (RCC_AHB1LPENR)	514
11.8.34	RCC AHB2 スリープクロックレジスタ (RCC_AHB2LPENR)	516
11.8.35	RCC AHB4 スリープクロックレジスタ (RCC_AHB4LPENR)	518
11.8.36	RCC APB1 スリープクロックレジスタ (RCC_APB1LLPENR)	519
11.8.37	RCC APB1 スリープクロックレジスタ (RCC_APB1HLPENR)	522
11.8.38	RCC APB2 スリープクロックレジスタ (RCC_APB2LPENR)	523
11.8.39	RCC APB3 スリープクロックレジスタ (RCC_APB3LPENR)	524
11.8.40	RCC カーネルクロック設定レジスタ 1 (RCC_CCIPR1)	526
11.8.41	RCC カーネルクロック設定レジスタ 2 (RCC_CCIPR2)	528
11.8.42	RCC カーネルクロック設定レジスタ 3 (RCC_CCIPR3)	530
11.8.43	RCC カーネルクロック設定レジスタ 4 (RCC_CCIPR4)	532
11.8.44	RCC カーネルクロック設定レジスタ 5 (RCC_CCIPR5)	533
11.8.45	RCC バックアップドメイン制御レジスタ (RCC_BDCR)	535

11.8.46	RCC リセットステータスレジスタ (RCC_RSR)	537
11.8.47	RCC セキュア設定レジスタ (RCC_SECCFGR)	539
11.8.48	RCC 特権設定レジスタ (RCC_PRIVCFGR)	541
11.9	RCC レジスタマップ	542
12	クロックリカバリシステム (CRS)	547
12.1	概要	547
12.2	CRS の主な機能	547
12.3	CRS の実装	547
12.4	CRS の機能説明	548
12.4.1	CRS ブロック図	548
12.4.2	同期入力	548
12.4.3	周波数誤差測定	549
12.4.4	周波数誤差評価および自動トリミング	550
12.4.5	CRS の初期化と設定	550
12.5	CRS 低電力モード	551
12.6	CRS 割込み	551
12.7	CRS レジスタ	552
12.7.1	CRS 制御レジスタ (CRS_CR)	552
12.7.2	CRS 設定レジスタ (CRS_CFGR)	553
12.7.3	CRS 割込みおよびステータスレジスタ (CRS_ISR)	554
12.7.4	CRS 割込みフラグクリアレジスタ (CRS_ICR)	556
12.7.5	CRS レジスタマップ	557
13	汎用 I/O (GPIO)	558
13.1	概要	558
13.2	GPIO の主な機能	558
13.3	GPIO の機能説明	558
13.3.1	汎用 I/O (GPIO)	560
13.3.2	I/O ピンオルタネート機能マルチプレクサと配置	561
13.3.3	I/O ポート制御レジスタ	561
13.3.4	I/O ポートデータレジスタ	562
13.3.5	I/O データのビット単位の操作	562
13.3.6	GPIO ロック機構	562
13.3.7	I/O オルタネート機能の入力/出力	562
13.3.8	外部割込み/ウェイクアップライン	563

13.3.9	入力設定	563
13.3.10	出力設定	564
13.3.11	オルタネート機能設定	564
13.3.12	アナログ設定	565
13.3.13	HSE または LSE オシレータのピンを GPIO として使用	565
13.3.14	GPIO ピンを RTC 供給ドメインで使用	566
13.3.15	STANDBY モード中の I/O 状態を維持	566
13.3.16	TrustZone セキュリティ	566
13.3.17	特権および非特権モード	567
13.3.18	ハイスピード低電圧モード (HSLV)	567
13.3.19	I/O 補正セル	568
13.4	GPIO レジスタ	568
13.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ I)	568
13.4.2	GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ I)	569
13.4.3	GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A ~ I)	569
13.4.4	GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ I)	570
13.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ I)	570
13.4.6	GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ I)	571
13.4.7	GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR) (x = A ~ I)	571
13.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ I)	572
13.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFR1) (x = A ~ I)	573
13.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A ~ H)	574
13.4.11	GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A ~ I)	575
13.4.12	GPIO ハイスピード低電圧レジスタ (GPIOx_HSLVR) (x = A ~ I)	575
13.4.13	GPIO セキュア設定レジスタ (GPIOx_SECCFGR) (x = A ~ I)	576
13.4.14	GPIO レジスタマップ	577
14	システム設定、ブート、およびセキュリティ (SBS)	579
14.1	SBS の概要	579
14.2	SBS の主な機能	579
14.3	SBS の機能詳細	580
14.3.1	SBS ブロック図	580
14.3.2	SBS 信号	581
14.3.3	SBS のリセットおよびクロック	581
14.3.4	SBS システム設定	582

14.3.5	SBS ブート制御	583
14.3.6	SBS デバッグ制御	586
14.3.7	SBS ハードウェア・セキュアストレージ制御	588
14.4	SBS 割込み	590
14.5	SBS レジスタ	590
14.5.1	SBS 時間的分離 (temporal isolation) 制御レジスタ (SBS_HDPLCR)	590
14.5.2	SBS 時間的分離ステータスレジスタ (SBS_HDPLSR)	591
14.5.3	SBS 次の HDPL 制御レジスタ (SBS_NEXTHDPLCR)	591
14.5.4	SBS デバッグ制御レジスタ (SBS_DBGCR)	592
14.5.5	SBS デバッグロックレジスタ (SBS_DBGLOCKR)	592
14.5.6	SBS RSS コマンドレジスタ (SBS_RSSCMDR)	593
14.5.7	SBS EPOCH 選択制御レジスタ (SBS_EPOCHSELCR)	593
14.5.8	SBS セキュリティモード設定制御レジスタ (SBS_SECCFGR)	594
14.5.9	SBS プロダクト・モードおよび設定レジスタ (SBS_PMCR)	595
14.5.10	SBS FPU 割込みマスクレジスタ (SBS_FPUIMR)	596
14.5.11	SBS メモリ消去ステータスレジスタ (SBS_MESR)	597
14.5.12	SBS I/O 制御およびステータスレジスタの補正セル (SBS_CCCSR)	597
14.5.13	SBS I/O 値レジスタの補正セル (SBS_CCVALR)	598
14.5.14	SBS I/O ソフトウェアコードレジスタの補正セル (SBS_CCSWCR)	599
14.5.15	SBS クラス B レジスタ (SBS_CFGR2)	600
14.5.16	SBS CPU 非セキュアロックレジスタ (SBS_CNSLCKR)	601
14.5.17	SBS CPU セキュアロックレジスタ (SBS_CSLCKR)	602
14.5.18	SBS flit ECC NMI マスクレジスタ (SBS_ECCNMIR)	603
14.5.19	SBS レジスタマップ	604
15	ペリフェラル相互接続マトリックス	606
15.1	相互接続マトリックスの概要	606
15.2	接続の一覧	607
15.3	相互接続の詳細	609
15.3.1	タイマのマスタからスレーブへの相互接続	609
15.3.2	ADC へのトリガ	609
15.3.3	タイマのトリガとしての ADC アナログウォッチドッグ	610
15.3.4	DAC へのトリガ	611
15.3.5	タイマへのクロックソース	611
15.3.6	低電力タイマへのトリガ	612
15.3.7	タイマへの入力としての RTC ウェイクアップ	613
15.3.8	タイマへのブレイク信号としてのシステムエラー	613

15.3.9	通信ペリフェラルのためのトリガ	614
15.3.10	GPDMA1/2 へのトリガ	614
15.3.11	アナログペリフェラルへの内部アナログ信号	615
15.3.12	DAC サンプルおよびホールドモードのクロックソース	615
15.3.13	内部タンパソース	615
15.3.14	タンパから RTC への出力	616
15.3.15	AES/SAES への暗号キー	616
16	汎用ダイレクト・メモリ・アクセス・コントローラ (GPDMA)	617
16.1	GPDMA の概要	617
16.2	GPDMA の主な機能	617
16.3	GPDMA の実装	618
16.3.1	GPDMA インスタンス	618
16.3.2	GPDMA チャネル	618
16.3.3	低電力モードにおける GPDMA 自律モード	619
16.3.4	GPDMA リクエスト	619
16.3.5	GPDMA ブロック・リクエスト	623
16.3.6	ペリフェラルの早期終了を伴う GPDMA チャネル	624
16.3.7	GPDMA トリガ	624
16.4	GPDMA の機能説明	626
16.4.1	GPDMA ブロック図	626
16.4.2	GPDMA チャネル状態とリンクリストを使用しない直接プログラミング	626
16.4.3	GPDMA チャネルのサスペンドとレジューム	627
16.4.4	GPDMA チャネルのアボートとリスタート	628
16.4.5	GPDMA のリンクリスト・データ構造	629
16.4.6	リンクリスト項目の転送実行	632
16.4.7	run-to-completion モードでの GPDMA チャネル状態および リンクリスト・プログラミング	632
16.4.8	リンク・ステップ・モードでの GPDMA チャネル状態および リンクリスト・プログラミング	636
16.4.9	GPDMA チャネル状態およびリンクリスト・プログラミング	643
16.4.10	GPDMA FIFO を使用した転送	645
16.4.11	GPDMA 転送リクエストとアービトレーション	652
16.4.12	GPDMA トリガ転送	656
16.4.13	リンクリスト・プログラミングでの GPDMA サークュラ・バッファリング	658
16.4.14	ペリフェラルのフロー制御モードでの GPDMA 転送	659
16.4.15	GPDMA セキュア／非セキュア・チャネル	660

16.4.16	GPDMA 特権／非特権チャネル	661
16.4.17	GPDMA エラー管理	661
16.4.18	GPDMA 自律モード	663
16.5	デバッグ・モードでの GPDMA	664
16.6	低電力モードの GPDMA	664
16.7	GPDMA 割込み	665
16.8	GPDMA レジスタ	666
16.8.1	GPDMA セキュア設定レジスタ (GPDMA_SECCFGR)	666
16.8.2	GPDMA 特権設定レジスタ (GPDMA_PRIVCFGR)	667
16.8.3	GPDMA 設定ロック・レジスタ (GPDMA_RCFGLOCKR)	667
16.8.4	GPDMA 非セキュア・マスク済み割込みステータス・レジスタ (GPDMA_MISR)	668
16.8.5	GPDMA セキュア・マスク済み割込みステータス・レジスタ (GPDMA_SMISR)	669
16.8.6	GPDMA チャネル x のリンクリスト・ベースのアドレス・レジスタ (GPDMA_CxLBAR)	670
16.8.7	GPDMA チャネル x フラグ・クリア・レジスタ (GPDMA_CxFCR)	671
16.8.8	GPDMA チャネル x ステータス・レジスタ (GPDMA_CxSR)	672
16.8.9	GPDMA チャネル x 制御レジスタ (GPDMA_CxCR)	673
16.8.10	GPDMA チャネル x 転送レジスタ 1 (GPDMA_CxTR1)	676
16.8.11	GPDMA チャネル x 転送レジスタ 2 (GPDMA_CxTR2)	679
16.8.12	GPDMA チャネル x ブロック・レジスタ 1 (GPDMA_CxBR1)	683
16.8.13	GPDMA チャネル x オルタネート・ブロック・レジスタ 1 (GPDMA_CxBR1)	684
16.8.14	GPDMA チャネル x 転送元アドレスレジスタ (GPDMA_CxSAR)	687
16.8.15	GPDMA チャネル x 転送先アドレス・レジスタ (GPDMA_CxDAR)	688
16.8.16	GPDMA チャネル x 転送レジスタ 3 (GPDMA_CxTR3)	689
16.8.17	GPDMA チャネル x ブロック・レジスタ 2 (GPDMA_CxBR2)	690
16.8.18	GPDMA チャネル x のリンクリスト・アドレス・レジスタ (GPDMA_CxLLR)	691
16.8.19	GPDMA チャネル x オルタネート・リンクリスト・アドレス・レジスタ (GPDMA_CxLLR)	692
16.8.20	GPDMA レジスタマップ	694
17	ネスト化されたベクタ割込みコントローラ (NVIC)	696
17.1	NVIC の主な機能	696
17.2	SysTick 較正值レジスタ	696
17.3	割込みベクタと例外ベクタ	697

18	拡張割込み／イベントコントローラ (EXTI)	702
18.1	EXTI の主な機能	702
18.2	EXTI ブロック図	703
18.2.1	ペリフェラルと CPU 間の EXTI 接続	704
18.2.2	EXTI 割込み／イベントのマッピング	704
18.3	EXTI の機能説明	706
18.3.1	EXTI の設定可能なイベント入力ウェイクアップ	706
18.3.2	EXTI のダイレクトイベント入力、ウェイクアップ	707
18.3.3	EXTI マルチプレクサの選択	708
18.4	EXTI の機能的挙動	708
18.5	EXTI イベント保護	709
18.5.1	EXTI セキュリティ保護	710
18.5.2	EXTI 特権保護	710
18.6	EXTI レジスタ	711
18.6.1	EXTI 立上りトリガ選択レジスタ 1 (EXTI_RTISR1)	711
18.6.2	EXTI 立下りトリガ選択レジスタ 1 (EXTI_FTSR1)	712
18.6.3	EXTI ソフトウェア割込みイベントレジスタ 1 (EXTI_SWIER1)	712
18.6.4	EXTI 立上りエッジペンディングレジスタ 1 (EXTI_RPR1)	713
18.6.5	EXTI 立下りエッジペンディングレジスタ 1 (EXTI_FPR1)	714
18.6.6	EXTI セキュリティ設定レジスタ 1 (EXTI_SECCFGR1)	714
18.6.7	EXTI 特権設定レジスタ 1 (EXTI_PRIVCFGR1)	715
18.6.8	EXTI 立上りトリガ選択レジスタ 2 (EXTI_RTISR2)	715
18.6.9	EXTI 立下りトリガ選択レジスタ 2 (EXTI_FTSR2)	716
18.6.10	EXTI ソフトウェア割込みイベント・レジスタ 2 (EXTI_SWIER2)	717
18.6.11	EXTI 立上りエッジ・ペンディング・レジスタ 2 (EXTI_RPR2)	719
18.6.12	EXTI 立下りエッジ・ペンディング・レジスタ 2 (EXTI_FPR2)	720
18.6.13	EXTI セキュリティ設定レジスタ 2 (EXTI_SECCFGR2)	721
18.6.14	EXTI 特権設定レジスタ 2 (EXTI_PRIVCFGR2)	722
18.6.15	EXTI 外部割込み選択レジスタ 1 (EXTI_EXTICR1)	723
18.6.16	EXTI 外部割込み選択レジスタ 2 (EXTI_EXTICR2)	725
18.6.17	EXTI 外部割込み選択レジスタ 3 (EXTI_EXTICR3)	727
18.6.18	EXTI 外部割込み選択レジスタ 4 (EXTI_EXTICR4)	729
18.6.19	EXTI ロックレジスタ (EXTI_LOCKR)	731
18.6.20	割込みマスクレジスタによる EXTI CPU ウェイクアップレジスタ 1 (EXTI_IMR1)	732
18.6.21	イベントマスクレジスタによる EXTI CPU ウェイクアップレジスタ 1 (EXTI_EMR1)	732

18.6.22	割込みマスク・レジスタ 2 による EXTI CPU ウェイクアップ (EXTI_IMR2) ..	733
18.6.23	イベント・マスク・レジスタ 2 による EXTI CPU ウェイクアップ (EXTI_EMR2) ..	734
18.6.24	EXTI レジスタマップ ..	735
19	巡回冗長検査計算ユニット (CRC) ..	737
19.1	概要 ..	737
19.2	CRC の主な機能 ..	737
19.3	CRC の機能説明 ..	738
19.3.1	CRC ブロック図 ..	738
19.3.2	CRC 内部信号 ..	738
19.3.3	CRC 操作 ..	738
19.4	CRC レジスタ ..	740
19.4.1	CRC データレジスタ (CRC_DR) ..	740
19.4.2	CRC 独立型データレジスタ (CRC_IDR) ..	740
19.4.3	CRC 制御レジスタ (CRC_CR) ..	741
19.4.4	CRC の初期値 (CRC_INIT) ..	742
19.4.5	CRC 多項式 (CRC_POL) ..	742
19.4.6	CRC レジスタマップ ..	743
20	CORDIC コプロセッサ (CORDIC) ..	744
20.1	CORDIC の概要 ..	744
20.2	CORDIC の主な機能 ..	744
20.3	CORDIC の機能詳細 ..	744
20.3.1	概要 ..	744
20.3.2	CORDIC 関数 ..	745
20.3.3	固定小数点表現 ..	751
20.3.4	スケーリングファクタ ..	751
20.3.5	Precision ..	751
20.3.6	ゼロオーバーヘッドモード ..	755
20.3.7	ポーリングモード ..	756
20.3.8	割込みモード ..	756
20.3.9	DMA モード ..	757
20.4	CORDIC レジスタ ..	758
20.4.1	CORDIC 制御/ステータスレジスタ (CORDIC_CSR) ..	758
20.4.2	CORDIC 引数レジスタ (CORDIC_WDATA) ..	760
20.4.3	CORDIC 結果レジスタ (CORDIC_RDATA) ..	761

20.4.4	CORDIC レジスタマップ	761
21	フィルタ演算アクセラレータ (FMAC)	762
21.1	FMAC の概要	762
21.2	FMAC の主な機能	762
21.3	FMAC の機能詳細	763
21.3.1	概要	763
21.3.2	ローカルメモリとバッファ	764
21.3.3	入力バッファ	765
21.3.4	出力バッファ	767
21.3.5	初期化機能	769
21.3.6	フィルタ機能	770
21.3.7	固定小数点表現	774
21.3.8	FMAC を用いた FIR フィルタの実装	774
21.3.9	FMAC を用いた IIR フィルタの実装	776
21.3.10	フィルタ初期化の例	778
21.3.11	フィルタ演算の例	779
21.3.12	フィルタ設計のヒント	781
21.4	FMAC レジスタ	782
21.4.1	FMAC X1 バッファ設定レジスタ (FMAC_X1BUFCFG)	782
21.4.2	FMAC X2 バッファ設定レジスタ (FMAC_X2BUFCFG)	782
21.4.3	FMAC Y バッファ設定レジスタ (FMAC_YBUFCFG)	783
21.4.4	FMAC パラメータ・レジスタ (FMAC_PARAM)	784
21.4.5	FMAC 制御レジスタ (FMAC_CR)	785
21.4.6	FMAC ステータス・レジスタ (FMAC_SR)	786
21.4.7	FMAC ライト・データ・レジスタ (FMAC_WDATA)	788
21.4.8	FMAC リード・データ・レジスタ (FMAC_RDATA)	788
21.4.9	FMAC レジスタマップ	789
22	フレキシブル・スタティック・メモリ・コントローラ (FSMC)	790
22.1	概要	790
22.2	FMC の主な特徴	790
22.3	FMC ブロック図	792
22.4	AHB インタフェース	793
22.4.1	サポートされるメモリおよびトランザクション	793
22.5	外部デバイスアドレスマッピング	794

22.5.1	NOR/PSRAM アドレスマッピング	795
22.5.2	NAND Flash メモリ・アドレス・マッピング	796
22.5.3	SDRAM アドレスマッピング	797
22.6	NOR 型 Flash / PSRAM コントローラ	799
22.6.1	外部メモリインタフェース信号	800
22.6.2	サポートされるメモリおよびトランザクション	802
22.6.3	一般的なタイミング規則	803
22.6.4	NOR Flash/PSRAM コントローラ非同期トランザクション	803
22.6.5	同期トランザクション	821
22.6.6	NOR/PSRAM コントローラレジスタ	827
22.7	NAND 型 Flash コントローラ	835
22.7.1	外部メモリインタフェース信号	836
22.7.2	NAND 型 Flash によってサポートされるメモリとトランザクション	837
22.7.3	NAND Flash メモリのタイミング図	837
22.7.4	NAND Flash 動作	838
22.7.5	NAND Flash のプリウエイト機能	839
22.7.6	NAND 型 Flash メモリ内のエラー訂正コード (ECC) の計算	840
22.7.7	NAND Flash コントローラ・レジスタ	841
22.8	SDRAM コントローラ	846
22.8.1	SDRAM コントローラの主な機能	846
22.8.2	SDRAM 外部メモリインタフェース信号	846
22.8.3	SDRAM コントローラの機能説明	847
22.8.4	低電力モード	853
22.8.5	SDRAM コントローラレジスタ	856
22.8.6	FMC レジスタマップ	863
23	Octo-SPI インタフェース (OCTOSPI)	865
23.1	概要	865
23.2	OCTOSPI の主な機能	865
23.3	OCTOSPI の実装	866
23.4	OCTOSPI の機能詳細	867
23.4.1	OCTOSPI ブロック図	867
23.4.2	OCTOSPI ピンおよび内部信号	868
23.4.3	OCTOSPI インタフェースを用いたメモリモード	869
23.4.4	OCTOSPI レギュラコマンドプロトコル	869
23.4.5	OCTOSPI レギュラコマンドプロトコル信号インタフェース	872

23.4.6	HyperBus プロトコル	876
23.4.7	特有の機能	880
23.4.8	OCTOSPI の動作モードの概要	881
23.4.9	OCTOSPI インダイレクトモード	881
23.4.10	OCTOSPI 自動ステータスポーリングモード	883
23.4.11	OCTOSPI メモリマップドモード	884
23.4.12	OCTOSPI の設定の概要	884
23.4.13	OCTOSPI のシステム設定	884
23.4.14	OCTOSPI のデバイス設定	885
23.4.15	OCTOSPI レギュラコマンドモードの設定	886
23.4.16	OCTOSPI HyperBus プロトコルの設定	888
23.4.17	OCTOSPI エラー管理	889
23.4.18	OCTOSPI の BUSY およびアボート	889
23.4.19	OCTOSPI の再設定または無効化	890
23.4.20	NCS の動作	890
23.5	アドレスのアライメントとデータ数	891
23.6	OCTOSPI 割込み	893
23.7	OCTOSPI レジスタ	893
23.7.1	OCTOSPI 制御レジスタ (OCTOSPI_CR)	893
23.7.2	OCTOSPI デバイス設定レジスタ 1 (OCTOSPI_DCR1)	896
23.7.3	OCTOSPI デバイス設定レジスタ 2 (OCTOSPI_DCR2)	897
23.7.4	OCTOSPI デバイス設定レジスタ 3 (OCTOSPI_DCR3)	898
23.7.5	OCTOSPI デバイス設定レジスタ 4 (OCTOSPI_DCR4)	899
23.7.6	OCTOSPI ステータスレジスタ (OCTOSPI_SR)	899
23.7.7	OCTOSPI フラグクリアレジスタ (OCTOSPI_FCR)	900
23.7.8	OCTOSPI データ長レジスタ (OCTOSPI_DLR)	901
23.7.9	OCTOSPI アドレスレジスタ (OCTOSPI_AR)	901
23.7.10	OCTOSPI データレジスタ (OCTOSPI_DR)	902
23.7.11	OCTOSPI ポーリングステータスマスクレジスタ (OCTOSPI_PSMKR)	903
23.7.12	OCTOSPI ポーリングステータス一致レジスタ (OCTOSPI_PSMAR)	903
23.7.13	OCTOSPI ポーリングインターバルレジスタ (OCTOSPI_PIR)	904
23.7.14	OCTOSPI 通信設定レジスタ (OCTOSPI_CCR)	904
23.7.15	OCTOSPI タイミング設定レジスタ (OCTOSPI_TCR)	906
23.7.16	OCTOSPI 命令レジスタ (OCTOSPI_IR)	907
23.7.17	OCTOSPI オルタネットバイトレジスタ (OCTOSPI_ABR)	907
23.7.18	OCTOSPI 低電力タイムアウトレジスタ (OCTOSPI_LPTR)	908
23.7.19	OCTOSPI ラップ通信設定レジスタ (OCTOSPI_WPCCR)	908

23.7.20	OCTOSPI ラップタイミング設定レジスタ (OCTOSPI_WPTCR)	910
23.7.21	OCTOSPI ラップ命令レジスタ (OCTOSPI_WPIR)	911
23.7.22	OCTOSPI ラップオルタネートバイトレジスタ (OCTOSPI_WPABR)	911
23.7.23	OCTOSPI 書き込み通信設定レジスタ (OCTOSPI_WCCR)	912
23.7.24	OCTOSPI 書き込みタイミング設定レジスタ (OCTOSPI_WTCR)	914
23.7.25	OCTOSPI 書き込み命令レジスタ (OCTOSPI_WIR)	914
23.7.26	OCTOSPI 書き込みオルタネートバイトレジスタ (OCTOSPI_WABR)	915
23.7.27	OCTOSPI HyperBus 遅延設定レジスタ (OCTOSPI_HLCR)	915
23.7.28	OCTOSPI レジスタマップ	916
24	セキュアデジタル入力 / 出力 MultiMediaCard インタフェース (SDMMC)	919
24.1	SDMMC の主な機能	919
24.2	SDMMC の実装	919
24.3	SDMMC バストポロジ	920
24.4	SDMMC 動作モード	922
24.5	SDMMC の機能説明	923
24.5.1	SDMMC ブロック図	923
24.5.2	SDMMC ピンおよび内部信号	923
24.5.3	概要	924
24.5.4	SDMMC アダプタ	926
24.5.5	SDMMC AHB スレーブインタフェース	946
24.5.6	SDMMC AHB マスタインタフェース	946
24.5.7	AHB と SDMMC_CK のクロック関係	949
24.6	カード機能詳細	950
24.6.1	SD I/O モード	950
24.6.2	CMD12 送信タイミング	958
24.6.3	SLEEP (CMD5)	961
24.6.4	割込みモード (Wait-IRQ)	962
24.6.5	ブート操作	963
24.6.6	レスポンス R1b の処理	966
24.6.7	リセットとカードサイクルパワー	967
24.7	ハードウェアフロー制御	968
24.8	超ハイスピードフェーズ I (UHS-I) の電圧スイッチ	969
24.9	SDMMC 割込み	972
24.10	SDMMC レジスタ	974

24.10.1	SDMMC 電源制御レジスタ (SDMMC_POWER)	974
24.10.2	SDMMC クロック制御レジスタ (SDMMC_CLKCR)	975
24.10.3	SDMMC 引数レジスタ (SDMMC_ARGR)	977
24.10.4	SDMMC コマンドレジスタ (SDMMC_CMDR)	977
24.10.5	SDMMC コマンドレスポンスレジスタ (SDMMC_RESPCMDR)	979
24.10.6	SDMMC レスポンス x レジスタ (SDMMC_RESPxR)	979
24.10.7	SDMMC データタイマレジスタ (SDMMC_DTIMER)	980
24.10.8	SDMMC データ長レジスタ (SDMMC_DLENR)	981
24.10.9	SDMMC データ制御レジスタ (SDMMC_DCTRL)	981
24.10.10	SDMMC データカウンタレジスタ (SDMMC_DCNTNR)	984
24.10.11	SDMMC ステータスレジスタ (SDMMC_STAR)	984
24.10.12	SDMMC 割込みクリアレジスタ (SDMMC_ICR)	987
24.10.13	SDMMC マスクレジスタ (SDMMC_MASKR)	989
24.10.14	SDMMC 確認応答タイマレジスタ (SDMMC_ACKTIMER)	992
24.10.15	SDMMC データ FIFO レジスタ x (SDMMC_FIFORx)	992
24.10.16	SDMMC DMA 制御レジスタ (SDMMC_IDMACTRLR)	993
24.10.17	SDMMC IDMA バッファサイズレジスタ (SDMMC_IDMABSIZER)	994
24.10.18	SDMMC IDMA バッファベースアドレスレジスタ (SDMMC_IDMABASER)	994
24.10.19	SDMMC IDMA リンクされたリストのアドレスレジスタ (SDMMC_IDMALAR)	995
24.10.20	SDMMC IDMA リンクされたリストメモリのベースレジスタ (SDMMC_IDMABAR)	996
24.10.21	SDMMC レジスタマップ	997
25	遅延ブロック (DLYB)	1000
25.1	概要	1000
25.2	DLYB の主な機能	1000
25.3	DLYB の機能説明	1000
25.3.1	DLYB 図	1000
25.3.2	DLYB ピンおよび内部信号	1001
25.3.3	概要	1001
25.3.4	遅延ライン長さ設定手順	1002
25.3.5	出カクロック位相の設定手順	1002
25.4	DLYB レジスタ	1003
25.4.1	DLYB 制御レジスタ (DLYB_CR)	1003
25.4.2	DLYB 設定レジスタ (DLYB_CFGR)	1004
25.4.3	DLYB レジスタマップ	1004

26	アナログデジタルコンバータ (ADC1/2)	1005
26.1	概要	1005
26.2	ADC の主な機能	1005
26.3	ADC の実装	1007
26.4	ADC の機能説明	1008
26.4.1	ADC のブロック図	1008
26.4.2	ADC ピンおよび内部信号	1009
26.4.3	ADC クロック	1011
26.4.4	ADC 接続性	1013
26.4.5	スレーブ AHB インタフェース	1015
26.4.6	ADC ディープパワーダウンモード (DEEPPWD) および ADC 電圧レギュレータ (ADVREGEN)	1015
26.4.7	シングルエンドおよび差動入力のチャンネル	1015
26.4.8	較正 (ADCAL、ADCALDIF、ADC_CALFACT)	1016
26.4.9	ADC オン / オフ制御 (ADEN、ADDIS、ADRDY)	1019
26.4.10	ADC 制御ビット書き込み時の制約	1020
26.4.11	チャンネルの選択 (SQRx、JSQRx)	1021
26.4.12	チャンネル単位でプログラム可能なサンプリング時間 (SMPR1、SMPR2) ..	1022
26.4.13	シングル変換モード (CONT = 0)	1023
26.4.14	連続変換モード (CONT = 1)	1024
26.4.15	変換の開始 (ADSTART、JADSTART)	1025
26.4.16	ADC のタイミング	1026
26.4.17	実行中の変換の停止 (ADSTP、JADSTP)	1026
26.4.18	外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN、JEXTSEL、JEXTEN)	1028
26.4.19	インジェクトチャンネルの管理	1029
26.4.20	不連続モード (DISCEN、DISCNUM、JDISCEN)	1031
26.4.21	インジェクト変換のコンテキストのキュー	1032
26.4.22	プログラム可能な分解能 (RES) - 高速変換モード	1040
26.4.23	変換の終了、サンプリングフェーズの終了 (EOC、JEOC、EOSMP)	1041
26.4.24	変換シーケンスの終了 (EOS、JEOS)	1041
26.4.25	タイミング図の例 (シングル / 連続モード、ハードウェア / ソフトウェアトリガ)	1042
26.4.26	データ管理	1043
26.4.27	動的低電力機能	1049
26.4.28	アナログウィンドウウォッチドッグ (AWD1EN、JAWD1EN、AWD1SGL、 AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx)	1054
26.4.29	オーバーサンプリング回路	1058

26.4.30	デュアル ADC モード	1064
26.4.31	温度センサ	1077
26.4.32	VBAT 電源監視	1079
26.4.33	内部電圧基準の監視	1080
26.4.34	電源電圧の監視	1081
26.5	ADC 割込み	1081
26.6	ADC レジスタ (ADC ごと)	1083
26.6.1	ADC 割込みおよびステータスレジスタ (ADC_ISR)	1083
26.6.2	ADC 割込み有効レジスタ (ADC_IER)	1085
26.6.3	ADC 制御レジスタ (ADC_CR)	1087
26.6.4	ADC 設定レジスタ (ADC_CFGR)	1090
26.6.5	ADC 設定レジスタ 2 (ADC_CFGR2)	1095
26.6.6	ADC サンプル時間レジスタ 1 (ADC_SMPR1)	1097
26.6.7	ADC サンプル時間レジスタ 2 (ADC_SMPR2)	1098
26.6.8	ADC ウォッチドッグ閾値レジスタ 1 (ADC_TR1)	1098
26.6.9	ADC ウォッチドッグ閾値レジスタ 2 (ADC_TR2)	1099
26.6.10	ADC ウォッチドッグ閾値レジスタ 3 (ADC_TR3)	1100
26.6.11	ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)	1100
26.6.12	ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)	1101
26.6.13	ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)	1102
26.6.14	ADC レギュラシーケンスレジスタ 4 (ADC_SQR4)	1103
26.6.15	ADC レギュラデータレジスタ (ADC_DR)	1104
26.6.16	ADC インジェクトシーケンスレジスタ (ADC_JSQR)	1104
26.6.17	ADC オフセット y レジスタ (ADC_OFRy)	1106
26.6.18	ADC インジェクトチャネル y データレジスタ (ADC_JDRy)	1107
26.6.19	ADC アナログウォッチドッグ 2 設定レジスタ (ADC_AWD2CR)	1108
26.6.20	ADC アナログウォッチドッグ 3 設定レジスタ (ADC_AWD3CR)	1108
26.6.21	ADC 差動モード選択レジスタ (ADC_DIFSEL)	1109
26.6.22	ADC 較正係数 (ADC_CALFACT)	1109
26.6.23	ADC オプションレジスタ (ADC_OR)	1110
26.7	ADC 共通レジスタ	1111
26.7.1	ADC 共通ステータスレジスタ (ADC_CSR)	1111
26.7.2	ADC 共通制御レジスタ (ADC_CCR)	1112
26.7.3	デュアルモード用 ADC 共通レギュラデータレジスタ (ADC_CDR)	1115
26.7.4	ADC ハードウェア設定レジスタ (ADC_HWCFGR0)	1116
26.7.5	ADC バージョンレジスタ (ADC_VERR)	1116
26.7.6	ADC 識別レジスタ (ADC_IPDR)	1117

26.7.7	ADC サイズ識別レジスタ (ADC_SIDR)	1117
26.8	ADC レジスタマップ	1118
27	デジタル温度センサ (DTS)	1121
27.1	概要	1121
27.2	DTS の主な機能	1121
27.3	DTS の機能詳細	1122
27.3.1	DTS ブロック図	1122
27.3.2	DTS 内部信号	1122
27.3.3	DTS ブロックの動作	1123
27.3.4	動作モード	1123
27.3.5	較正	1123
27.3.6	プリスケアラ	1123
27.3.7	温度測定の実理	1124
27.3.8	サンプリング時間	1125
27.3.9	クイック測定モード	1125
27.3.10	トリガ入力	1126
27.3.11	オン/オフ制御およびレディフラグ	1126
27.3.12	温度測定シーケンス	1127
27.4	DTS 低消費電力モード	1128
27.5	DTS 割込み	1128
27.5.1	温度ウィンドウコンパレータ	1128
27.5.2	同期割込み	1128
27.5.3	非同期ウェイクアップ	1128
27.6	DTS レジスタ	1129
27.6.1	温度センサ設定レジスタ 1 (DTS_CFGR1)	1129
27.6.2	温度センサ T0 値レジスタ 1 (DTS_T0VALR1)	1131
27.6.3	温度センサのランプ値レジスタ (DTS_RAMPVALR)	1131
27.6.4	温度センサ割込み閾値レジスタ 1 (DTS_ITR1)	1132
27.6.5	温度センサデータレジスタ (DTS_DR)	1132
27.6.6	温度センサステータスレジスタ (DTS_SR)	1133
27.6.7	温度センサ割込み有効レジスタ (DTS_ITENR)	1134
27.6.8	温度センサ割込みクリアフラグレジスタ (DTS_ICIFR)	1135
27.6.9	温度センサ・オプション・レジスタ (DTS_OR)	1136
27.6.10	DTS レジスタマップ	1137

28	DAC (DAC)	1138
28.1	概要	1138
28.2	DAC の主な機能	1138
28.3	DAC の実装	1139
28.4	DAC の機能説明	1140
28.4.1	DAC ブロック図	1140
28.4.2	DAC ピンおよび内部信号	1141
28.4.3	DAC クロック	1142
28.4.4	DAC チャネルイネーブル	1142
28.4.5	DAC データフォーマット	1142
28.4.6	DAC 変換	1144
28.4.7	DAC 出力電圧	1145
28.4.8	DAC トリガ選択	1145
28.4.9	DMA リクエスト	1146
28.4.10	ノイズ生成	1147
28.4.11	三角波生成	1148
28.4.12	DAC チャネルモード	1149
28.4.13	DAC チャネルバッファの較正	1152
28.4.14	デュアル DAC チャネル変換モード (デュアルチャネルを使用できる場合)	1153
28.5	低消費電力モードでの DAC	1157
28.6	DAC 割込み	1157
28.7	DAC レジスタ	1158
28.7.1	DAC 制御レジスタ (DAC_CR)	1158
28.7.2	DAC ソフトウェアトリガレジスタ (DAC_SWTRGR)	1161
28.7.3	DAC チャネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)	1162
28.7.4	DAC チャネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)	1162
28.7.5	DAC チャネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)	1163
28.7.6	DAC チャネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)	1163
28.7.7	DAC チャネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2)	1164
28.7.8	DAC チャネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2)	1164
28.7.9	デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)	1165

28.7.10	デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD) . . .	1165
28.7.11	デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)	1166
28.7.12	DAC チャンネル 1 データ出力レジスタ (DAC_DOR1)	1166
28.7.13	DAC チャンネル 2 データ出力レジスタ (DAC_DOR2)	1167
28.7.14	DAC ステータスレジスタ (DAC_SR)	1167
28.7.15	DAC 較正制御レジスタ (DAC_CCR)	1169
28.7.16	DAC モード制御レジスタ (DAC_MCR)	1169
28.7.17	DAC チャンネル 1 サンプルおよびホールドのサンプル時間レジスタ (DAC_SHSR1)	1171
28.7.18	DAC チャンネル 2 サンプルおよびホールドのサンプル時間レジスタ (DAC_SHSR2)	1171
28.7.19	DAC サンプルおよびホールド時間レジスタ (DAC_SHHR)	1172
28.7.20	DAC サンプルおよびホールドのリフレッシュ時間レジスタ (DAC_SHRR)	1172
28.7.21	DAC レジスタマップ	1173
29	基準電圧バッファ (VREFBUF)	1175
29.1	概要	1175
29.2	VREFBUF の実装	1175
29.3	VREFBUF の機能説明	1175
29.4	VREFBUF のトリミング	1176
29.5	VREFBUF レジスタ	1177
29.5.1	VREFBUF 制御およびステータスレジスタ (VREFBUF_CSR)	1177
29.5.2	VREFBUF 較正制御レジスタ (VREFBUF_CCR)	1178
29.5.3	VREFBUF レジスタマップ	1178
30	デジタルカメラインタフェース (DCMI)	1179
30.1	概要	1179
30.2	DCMI の主な機能	1179
30.3	DCMI の機能説明	1179
30.3.1	DCMI ブロック図	1180
30.3.2	DCMI ピンおよび内部信号	1180
30.3.3	DCMI クロック	1181
30.3.4	DCMI DMA インタフェース	1181
30.3.5	DCMI 物理インタフェース	1181
30.3.6	DCMI の同期	1183
30.3.7	DCMI キャプチャモード	1185

30.3.8	DCMI クロップ機能	1186
30.3.9	DCMI JPEG フォーマット	1187
30.3.10	DCMI の FIFO	1187
30.3.11	DCMI データフォーマットの説明	1187
30.4	DCMI 割込み	1190
30.5	DCMI レジスタ	1190
30.5.1	DCMI 制御レジスタ (DCMI_CR)	1190
30.5.2	DCMI ステータスレジスタ (DCMI_SR)	1192
30.5.3	DCMI raw 割込みステータスレジスタ (DCMI_RIS)	1193
30.5.4	DCMI 割込み有効レジスタ (DCMI_IER)	1194
30.5.5	DCMI マスク済み割込みステータスレジスタ (DCMI_MIS)	1195
30.5.6	DCMI 割込みクリアレジスタ (DCMI_ICR)	1196
30.5.7	DCMI 埋め込み同期コードレジスタ (DCMI_ESCR)	1197
30.5.8	DCMI 埋め込み同期マスク解除レジスタ (DCMI_ESUR)	1198
30.5.9	DCMI クロップウィンドウ開始 (DCMI_CWSTRT)	1199
30.5.10	DCMI クロップウィンドウサイズ (DCMI_CWSIZE)	1199
30.5.11	DCMI データレジスタ (DCMI_DR)	1200
30.5.12	DCMI レジスタマップ	1201
31	パラレル同期スレーブインタフェース (PSSI)	1202
31.1	概要	1202
31.2	PSSI の主な機能	1202
31.3	PSSI の機能説明	1202
31.3.1	PSSI ブロック図	1203
31.3.2	PSSI ピンおよび内部信号	1203
31.3.3	PSSI クロック	1204
31.3.4	PSSI データ管理	1204
31.3.5	PSSI オプション制御信号	1206
31.4	PSSI 割込み	1209
31.5	PSSI レジスタ	1210
31.5.1	PSSI 制御レジスタ (PSSI_CR)	1210
31.5.2	PSSI ステータスレジスタ (PSSI_SR)	1212
31.5.3	PSSI raw 割込みステータスレジスタ (PSSI_RIS)	1212
31.5.4	PSSI 割込み有効レジスタ (PSSI_IER)	1213
31.5.5	PSSI マスク済み割込みステータスレジスタ (PSSI_MIS)	1213
31.5.6	PSSI 割込みクリアレジスタ (PSSI_ICR)	1214

	31.5.7	PSSI データレジスタ (PSSI_DR)	1215
	31.5.8	PSSI レジスタマップ	1216
32		真性乱数発生器 (RNG)	1217
	32.1	概要	1217
	32.2	RNG の主な機能	1217
	32.3	RNG の機能説明	1218
	32.3.1	RNG ブロック図	1218
	32.3.2	RNG 内部信号	1218
	32.3.3	乱数の生成	1219
	32.3.4	RNG 初期化	1221
	32.3.5	RNG 操作	1223
	32.3.6	RNG クロック供給	1224
	32.3.7	エラー管理	1224
	32.3.8	RNG の低消費電力時の取り扱い	1225
	32.4	RNG 割込み	1226
	32.5	RNG 処理時間	1226
	32.6	RNG エントロピーソース検証	1227
	32.6.1	概要	1227
	32.6.2	検証条件	1227
	32.6.3	データ収集	1227
	32.7	RNG レジスタ	1228
	32.7.1	RNG 制御レジスタ (RNG_CR)	1228
	32.7.2	RNG ステータスレジスタ (RNG_SR)	1230
	32.7.3	RNG データレジスタ (RNG_DR)	1231
	32.7.4	RNG 健全性テスト制御レジスタ (RNG_HTCR)	1231
	32.7.5	RNG レジスタマップ	1232
33		AES ハードウェアアクセラレータ (AES)	1233
	33.1	概要	1233
	33.2	AES の主な特徴	1233
	33.3	AES の実装	1234
	33.4	AES 機能詳細	1234
	33.4.1	AES ブロック図	1234
	33.4.2	AES 内部信号	1235
	33.4.3	AES のリセットおよびクロック	1235

33.4.4	AES 対称暗号の実装	1235
33.4.5	AES の暗号化または復号の一般的な使用法	1236
33.4.6	AES の認証済み暗号化、復号、および暗号ベースのメッセージ認証	1238
33.4.7	AES の暗号文借用およびデータパディング	1238
33.4.8	AES のサスペンドおよびレジューム操作	1239
33.4.9	AES 基本連鎖モード (ECB、CBC)	1239
33.4.10	AES カウンタ (CTR) モード	1243
33.4.11	AES ガロア／カウンタモード (GCM)	1246
33.4.12	AES ガロアメッセージ認証コード (GMAC : Galois message authentication code)	1250
33.4.13	AESCBC-MAC 付きカウンタ (CCM)	1251
33.4.14	AES セキュア・コプロセッサとの AES キー共有	1256
33.4.15	AES データレジスタおよびデータスワッピング	1257
33.4.16	AES キーレジスタ	1259
33.4.17	AES 初期化ベクタレジスタ	1259
33.4.18	AES エラー管理	1260
33.5	AES 割込み	1261
33.6	AES DMA リクエスト	1261
33.7	AES の処理遅延時間	1262
33.8	AES レジスタ	1263
33.8.1	AES 制御レジスタ (AES_CR)	1263
33.8.2	AES ステータスレジスタ (AES_SR)	1265
33.8.3	AES データ入力レジスタ (AES_DINR)	1266
33.8.4	AES データ出力レジスタ (AES_DOUTR)	1267
33.8.5	AES キーレジスタ 0 (AES_KEYR0)	1267
33.8.6	AES キーレジスタ 1 (AES_KEYR1)	1268
33.8.7	AES キーレジスタ 2 (AES_KEYR2)	1268
33.8.8	AES キーレジスタ 3 (AES_KEYR3)	1268
33.8.9	AES 初期化ベクタレジスタ 0 (AES_IVR0)	1269
33.8.10	AES 初期化ベクタレジスタ 1 (AES_IVR1)	1269
33.8.11	AES 初期化ベクタレジスタ 2 (AES_IVR2)	1269
33.8.12	AES 初期化ベクタレジスタ 3 (AES_IVR3)	1270
33.8.13	AES キーレジスタ 4 (AES_KEYR4)	1270
33.8.14	AES キーレジスタ 5 (AES_KEYR5)	1270
33.8.15	AES キーレジスタ 6 (AES_KEYR6)	1271
33.8.16	AES キーレジスタ 7 (AES_KEYR7)	1271
33.8.17	AES サスペンドレジスタ (AES_SUSPRx)	1272

33.8.18	AES 割込み有効レジスタ (AES_IER)	1272
33.8.19	AES 割込みステータスレジスタ (AES_ISR)	1273
33.8.20	AES 割込みクリアレジスタ (AES_ICR)	1274
33.8.21	AES レジスタマップ	1274
34	セキュアな AES コプロセッサ (SAES)	1277
34.1	概要	1277
34.2	SAES の主な特徴	1278
34.3	SAES の実装	1279
34.4	SAES 機能詳細	1279
34.4.1	SAES ブロック図	1279
34.4.2	SAES 内部信号	1280
34.4.3	SAES のリセットおよびクロック	1280
34.4.4	SAES 対称暗号の実装	1280
34.4.5	SAES の暗号化または復号の一般的な使用法	1281
34.4.6	SAES の認証済み暗号化、復号、および暗号ベースのメッセージ認証	1283
34.4.7	SAES の暗号文借用およびデータパディング	1284
34.4.8	SAES のサスペンドおよびレジューム操作	1284
34.4.9	SAES 基本連鎖モード (ECB、CBC)	1285
34.4.10	SAES カウンタ (CTR) モード	1289
34.4.11	SAES ガロア/カウンタモード (GCM)	1291
34.4.12	SAES ガロアメッセージ認証コード (GMAC : Galois message authentication code)	1295
34.4.13	SAES の CBC-MAC 付きカウンタ (CCM)	1296
34.4.14	ラップキーを使った SAES の動作	1301
34.4.15	共有キーを使った SAES の動作	1305
34.4.16	SAES データレジスタおよびデータスワッピング	1307
34.4.17	SAES キーレジスタ	1309
34.4.18	SAES 初期化ベクタレジスタ	1311
34.4.19	SAES エラー管理	1311
34.5	SAES 割込み	1313
34.6	SAES DMA リクエスト	1314
34.7	SAES の処理遅延時間	1315
34.8	SAES レジスタ	1316
34.8.1	SAES 制御レジスタ (SAES_CR)	1316
34.8.2	SAES ステータスレジスタ (SAES_SR)	1319
34.8.3	SAES データ入力レジスタ (SAES_DINR)	1320

34.8.4	SAES データ出力レジスタ (SAES_DOUTR)	1321
34.8.5	SAES キーレジスタ 0 (SAES_KEYR0)	1321
34.8.6	SAES キーレジスタ 1 (SAES_KEYR1)	1322
34.8.7	SAES キーレジスタ 2 (SAES_KEYR2)	1322
34.8.8	SAES キーレジスタ 3 (SAES_KEYR3)	1322
34.8.9	SAES 初期化ベクタレジスタ 0 (SAES_IVR0)	1323
34.8.10	SAES 初期化ベクタレジスタ 1 (SAES_IVR1)	1323
34.8.11	SAES 初期化ベクタレジスタ 2 (SAES_IVR2)	1323
34.8.12	SAES 初期化ベクタレジスタ 3 (SAES_IVR3)	1324
34.8.13	SAES キーレジスタ 4 (SAES_KEYR4)	1324
34.8.14	SAES キーレジスタ 5 (SAES_KEYR5)	1324
34.8.15	SAES キーレジスタ 6 (SAES_KEYR6)	1325
34.8.16	SAES キーレジスタ 7 (SAES_KEYR7)	1325
34.8.17	SAES サスペンドレジスタ (SAES_SUSPRx)	1326
34.8.18	SAES 割込み有効レジスタ (SAES_IER)	1326
34.8.19	SAES 割込みステータスレジスタ (SAES_ISR)	1327
34.8.20	SAES 割込みクリアレジスタ (SAES_ICR)	1329
34.8.21	SAES レジスタマップ	1330

35 ハッシュプロセッサ (HASH) 1332

35.1	概要	1332
35.2	HASH の主な機能	1332
35.3	ハッシュの実装	1333
35.4	HASH の機能詳細	1333
35.4.1	HASH ブロック図	1333
35.4.2	HASH 内部信号	1333
35.4.3	セキュアハッシュアルゴリズムについて	1334
35.4.4	メッセージデータの供給	1334
35.4.5	メッセージダイジェストの計算	1335
35.4.6	メッセージのパディング	1337
35.4.7	HMAC 操作	1339
35.4.8	HASH サスペンド/レジューム操作	1341
35.4.9	HASH DMA インタフェース	1343
35.4.10	HASH エラー管理	1343
35.5	HASH 割込み	1344
35.6	HASH 処理時間	1344

35.7	HASH レジスタ	1345
35.7.1	HASH 制御レジスタ (HASH_CR)	1345
35.7.2	HASH データ入力レジスタ (HASH_DIN)	1347
35.7.3	HASH スタートレジスタ (HASH_STR)	1348
35.7.4	HASH ダイジェストレジスタ	1349
35.7.5	HASH 割込み有効レジスタ (HASH_IMR)	1351
35.7.6	HASH ステータスレジスタ (HASH_SR)	1351
35.7.7	HASH コンテキストスワップレジスタ	1352
35.7.8	HASH レジスタマップ	1354
36	公開鍵アクセラレータ (PKA)	1355
36.1	概要	1355
36.2	PKA の主な機能	1355
36.3	PKA の機能説明	1356
36.3.1	PKA ブロック図	1356
36.3.2	PKA 内部信号	1356
36.3.3	PKA のリセットおよびクロック	1356
36.3.4	PKA 公開鍵の高速化	1357
36.3.5	PKA の標準的なアプリケーション	1358
36.3.6	演算を行うための PKA の手順	1361
36.3.7	PKA エラー管理	1362
36.4	PKA 演算モード	1362
36.4.1	概要	1362
36.4.2	モンゴメリパラメータ計算	1363
36.4.3	剰余加算	1364
36.4.4	剰余減算	1364
36.4.5	剰余乗算とモンゴメリ乗算	1365
36.4.6	べき剰余	1366
36.4.7	モジュラ逆数	1367
36.4.8	モジュラリダクション	1368
36.4.9	算術加算	1368
36.4.10	算術減算	1368
36.4.11	算術乗算	1369
36.4.12	算術比較	1369
36.4.13	RSA CRT べき乗	1369
36.4.14	楕円曲線 Fp 上の点の確認	1370
36.4.15	ECC Fp スカラー乗算	1371

36.4.16	ECDSA 署名	1372
36.4.17	ECDSA 検証	1373
36.4.18	ECC 完全加算	1374
36.4.19	ECC ダブルベースラダー	1375
36.4.20	ECC 射影からアフィンへ	1376
36.5	設定と処理時間の例	1376
36.5.1	サポートされている楕円曲線	1376
36.5.2	計算時間	1378
36.6	PKA 割込み	1380
36.7	PKA レジスタ	1380
36.7.1	PKA 制御レジスタ (PKA_CR)	1380
36.7.2	PKA ステータスレジスタ (PKA_SR)	1382
36.7.3	PKA クリアフラグレジスタ (PKA_CLRFR)	1383
36.7.4	PKA RAM	1384
36.7.5	PKA レジスタマップ	1384
37	オンザフライ復号エンジン (OTFDEC)	1385
37.1	概要	1385
37.2	OTFDEC の主な機能	1385
37.3	OTFDEC の機能説明	1386
37.3.1	OTFDEC ブロック図	1386
37.3.2	OTFDEC 内部信号	1386
37.3.3	OTFDEC オンザフライ復号	1387
37.3.4	OTFDEC を使用したカウンタモードでの AES 復号	1388
37.3.5	フロー制御管理	1389
37.3.6	OTFDEC エラー管理	1389
37.4	OTFDEC 割込み	1390
37.5	OTFDEC アプリケーション情報	1391
37.5.1	OTFDEC 初期化处理	1391
37.5.2	OTFDEC および電力管理	1392
37.5.3	OTFDEC での暗号化	1392
37.5.4	OTFDEC キー CRC のソースコード	1394
37.6	OTFDEC レジスタ	1395
37.6.1	OTFDEC 制御レジスタ (OTFDEC_CR)	1395
37.6.2	OTFDEC 特権アクセス制御設定レジスタ (OTFDEC_PRIVCFGR)	1396
37.6.3	OTFDEC 領域 x 設定レジスタ (OTFDEC_RxCFGR)	1396

37.6.4	OTFDEC 領域 x 開始アドレスレジスタ (OTFDEC_RxSTARTADDR)	1398
37.6.5	OTFDEC 領域 x 終了アドレスレジスタ (OTFDEC_RxENDADDR)	1399
37.6.6	OTFDEC 領域 x ノンスレジスタ 0 (OTFDEC_RxNONCER0)	1399
37.6.7	OTFDEC 領域 x ノンスレジスタ 1 (OTFDEC_RxNONCER1)	1400
37.6.8	OTFDEC 領域 x キーレジスタ 0 (OTFDEC_RxKEYR0)	1400
37.6.9	OTFDEC 領域 x キーレジスタ 1 (OTFDEC_RxKEYR1)	1401
37.6.10	OTFDEC 領域 x キーレジスタ 2 (OTFDEC_RxKEYR2)	1401
37.6.11	OTFDEC 領域 x キーレジスタ 3 (OTFDEC_RxKEYR3)	1402
37.6.12	OTFDEC 割込みステータスレジスタ (OTFDEC_ISR)	1402
37.6.13	OTFDEC 割込みクリアレジスタ (OTFDEC_ICR)	1403
37.6.14	OTFDEC 割込み有効レジスタ (OTFDEC_IER)	1404
37.6.15	OTFDEC レジスタマップ	1405
38	高機能制御タイマ (TIM1/TIM8)	1408
38.1	TIM1/TIM8 の概要	1408
38.2	TIM1/TIM8 の主な特徴	1408
38.3	TIM1/TIM8 機能詳細	1409
38.3.1	ブロック図	1409
38.3.2	TIM1/TIM8 ピンおよび内部信号	1410
38.3.3	タイムベースユニット	1414
38.3.4	カウンタモード	1416
38.3.5	繰返しカウンタ	1428
38.3.6	外部トリガ入力	1430
38.3.7	クロック選択	1430
38.3.8	キャプチャ/比較チャネル	1434
38.3.9	入力キャプチャモード	1437
38.3.10	PWM 入力モード	1438
38.3.11	強制出力モード	1439
38.3.12	出力比較モード	1439
38.3.13	PWM モード	1440
38.3.14	非対称 PWM モード	1449
38.3.15	組合せ PWM モード	1450
38.3.16	組合せ 3 相 PWM モード	1451
38.3.17	相補出力とデッドタイム挿入	1452
38.3.18	ブレーク機能の使用	1455
38.3.19	双方向ブレーク入力	1461
38.3.20	外部イベントによる tim_ocxref 信号のクリア	1462

38.3.21	6 ステップ PWM 生成	1464
38.3.22	ワンパルスモード	1465
38.3.23	再トリガ可能なワンパルスモード	1466
38.3.24	比較モードでのパルス	1467
38.3.25	エンコーダインタフェースモード	1469
38.3.26	方向ビットの出力	1487
38.3.27	UIF ビットの再配置	1487
38.3.28	タイマ入力 XOR 機能	1487
38.3.29	ホールセンサとのインタフェース	1488
38.3.30	タイマの同期	1489
38.3.31	ADC トリガ	1493
38.3.32	DMA バーストモード	1494
38.3.33	TIM1/TIM8DMA リクエスト	1495
38.3.34	デバッグモード	1495
38.4	TIM1/TIM8 低電力モード	1495
38.5	TIM1/TIM8 割込み	1496
38.6	TIM1/TIM8 レジスタ	1497
38.6.1	TIMx 制御レジスタ 1 (TIMx_CR1) (x = 1、8)	1497
38.6.2	TIMx 制御レジスタ 2 (TIMx_CR2) (x = 1、8)	1499
38.6.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR) (x = 1、8)	1502
38.6.4	TIMx DMA / 割込み有効レジスタ (TIMx_DIER) (x = 1、8)	1506
38.6.5	TIMx ステータスレジスタ (TIMx_SR) (x = 1、8)	1507
38.6.6	TIMx イベント生成レジスタ (TIMx_EGR) (x = 1、8)	1510
38.6.7	TIMx キャプチャ / 比較モードレジスタ 1 (TIMx_CCMR1) (x = 1、8)	1511
38.6.8	TIMx キャプチャ / 比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 1、8)	1513
38.6.9	TIMx キャプチャ / 比較モードレジスタ 2 (TIMx_CCMR2) (x = 1、8)	1515
38.6.10	TIMx キャプチャ / 比較モードレジスタ 2 [オルタネート] (TIMx_CCMR2) (x = 1、8)	1516
38.6.11	TIMx キャプチャ / 比較有効レジスタ (TIMx_CCER) (x = 1、8)	1519
38.6.12	TIMx カウンタ (TIMx_CNT) (x = 1、8)	1523
38.6.13	TIMx プリスケアラ (TIMx_PSC) (x = 1、8)	1523
38.6.14	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 1、8)	1524
38.6.15	TIMx 繰返しカウンタレジスタ (TIMx_RCR) (x = 1、8)	1524
38.6.16	TIMx キャプチャ / 比較レジスタ 1 (TIMx_CCR1) (x = 1、8)	1525
38.6.17	TIMx キャプチャ / 比較レジスタ 2 (TIMx_CCR2) (x = 1、8)	1526
38.6.18	TIMx キャプチャ / 比較レジスタ 3 (TIMx_CCR3) (x = 1、8)	1527

38.6.19	TIMx キャプチャ／比較レジスタ 4 (TIMx_CCR4) (x = 1、8)	1528
38.6.20	TIMx ブレークおよびデッドタイムレジスタ (TIMx_BDTR) (x = 1、8)	1529
38.6.21	TIMx キャプチャ／比較レジスタ 5 (TIMx_CCR5) (x = 1、8)	1533
38.6.22	TIMx キャプチャ／比較レジスタ 6 (TIMx_CCR6) (x = 1、8)	1534
38.6.23	TIMx キャプチャ／比較モードレジスタ 3 (TIMx_CCMR3) (x = 1、8)	1535
38.6.24	TIMx タイマデッドタイムレジスタ 2 (TIMx_DTR2) (x = 1、8)	1536
38.6.25	TIMx タイマエンコーダ制御レジスタ (TIMx_ECR) (x = 1、8)	1537
38.6.26	TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 1、8)	1538
38.6.27	TIMx オルタネート機能オプションレジスタ 1 (TIMx_AF1) (x = 1、8)	1539
38.6.28	TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 1、8)	1542
38.6.29	TIMx DMA 制御レジスタ (TIMx_DCR) (x = 1、8)	1545
38.6.30	完全転送の TIMx DMA アドレス (TIMx_DMAR) (x = 1、8)	1546
38.6.31	TIMx レジスタマップ	1547
39	汎用タイマ (TIM2/TIM3/TIM4/TIM5)	1550
39.1	TIM2/TIM3/TIM4/TIM5 の概要	1550
39.2	TIM2/TIM3/TIM4/TIM5 の主な特徴	1550
39.3	TIM2/TIM3/TIM4/TIM5 の実装	1551
39.4	TIM2/TIM3/TIM4/TIM5 機能詳細	1552
39.4.1	ブロック図	1552
39.4.2	TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号	1553
39.4.3	タイムベースユニット	1556
39.4.4	カウンタモード	1558
39.4.5	クロック選択	1568
39.4.6	キャプチャ／比較チャネル	1572
39.4.7	入力キャプチャモード	1574
39.4.8	PWM 入力モード	1575
39.4.9	強制出力モード	1576
39.4.10	出力比較モード	1576
39.4.11	PWM モード	1578
39.4.12	非対称 PWM モード	1586
39.4.13	組合せ PWM モード	1587
39.4.14	外部イベントによる tim_ocxref 信号のクリア	1588
39.4.15	ワンパルスモード	1590
39.4.16	再トリガ可能なワンパルスモード	1591
39.4.17	比較モードでのパルス	1592
39.4.18	エンコーダインタフェースモード	1594

39.4.19	方向ビットの出力	1612
39.4.20	UIF ビットの再配置	1612
39.4.21	タイマ入力 XOR 機能	1612
39.4.22	タイマと外部トリガの同期	1612
39.4.23	タイマの同期	1616
39.4.24	ADC トリガ	1621
39.4.25	DMA バーストモード	1622
39.4.26	TIM2/TIM3/TIM4/TIM5 DMA リクエスト	1623
39.4.27	デバッグモード	1623
39.4.28	TIM2/TIM3/TIM4/TIM5 低電力モード	1623
39.4.29	TIM2/TIM3/TIM4/TIM5 割込み	1624
39.5	TIM2/TIM3/TIM4/TIM5 レジスタ	1625
39.5.1	TIMx 制御レジスタ 1 (TIMx_CR1) (x = 2 ~ 5)	1625
39.5.2	TIMx 制御レジスタ 2 (TIMx_CR2) (x = 2 ~ 5)	1627
39.5.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR) (x = 2 ~ 5)	1628
39.5.4	TIMx DMA / 割込み有効レジスタ (TIMx_DIER) (x = 2 ~ 5)	1631
39.5.5	TIMx ステータスレジスタ (TIMx_SR) (x = 2 ~ 5)	1632
39.5.6	TIMx イベント生成レジスタ (TIMx_EGR) (x = 2 ~ 5)	1634
39.5.7	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) (x = 2 ~ 5)	1635
39.5.8	TIMx キャプチャ/比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 2 ~ 5)	1637
39.5.9	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2) (x = 2 ~ 5)	1639
39.5.10	TIMx キャプチャ/比較モードレジスタ 2 [オルタネート] (TIMx_CCMR2) (x = 2 ~ 5)	1640
39.5.11	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 2 ~ 5)	1641
39.5.12	TIMx カウンタ (TIMx_CNT) (x = 3, 4)	1643
39.5.13	TIMx カウンタ (TIMx_CNT) (x = 2, 5)	1643
39.5.14	TIMx プリスケアラ (TIMx_PSC) (x = 2 ~ 5)	1644
39.5.15	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 3, 4)	1644
39.5.16	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 2, 5)	1645
39.5.17	TIMx キャプチャ/比較レジスタ 1 (TIMx_CCR1) (x = 3, 4)	1646
39.5.18	TIMx キャプチャ/比較レジスタ 1 (TIMx_CCR1) (x = 2, 5)	1647
39.5.19	TIMx キャプチャ/比較レジスタ 2 (TIMx_CCR2) (x = 3, 4)	1648
39.5.20	TIMx キャプチャ/比較レジスタ 2 (TIMx_CCR2) (x = 2, 5)	1649
39.5.21	TIMx キャプチャ/比較レジスタ 3 (TIMx_CCR3) (x = 3, 4)	1650
39.5.22	TIMx キャプチャ/比較レジスタ 3 (TIMx_CCR3) (x = 2, 5)	1651
39.5.23	TIMx キャプチャ/比較レジスタ 4 (TIMx_CCR4) (x = 3, 4)	1652
39.5.24	TIMx キャプチャ/比較レジスタ 4 (TIMx_CCR4) (x = 2, 5)	1653

39.5.25	TIMx タイマエンコーダ制御レジスタ (TIMx_ECR) (x = 2 ~ 5)	1654
39.5.26	TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 2 ~ 5)	1655
39.5.27	TIMx オルタネート機能レジスタ 1 (TIMx_AF1) (x = 2 ~ 5)	1656
39.5.28	TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 2 ~ 5)	1657
39.5.29	TIMx DMA 制御レジスタ (TIMx_DCR) (x = 2 ~ 5)	1658
39.5.30	完全転送の TIMx DMA アドレス (TIMx_DMAR) (x = 2 ~ 5)	1659
39.5.31	TIMx レジスタマップ	1660
40	基本タイマ (TIM6/TIM7)	1662
40.1	TIM6/TIM7 の概要	1662
40.2	TIM6/TIM7 の主な特徴	1662
40.3	TIM6/TIM7 機能詳細	1663
40.3.1	TIM6/TIM7 ブロック図	1663
40.3.2	TIM6/TIM7 内部信号	1663
40.3.3	TIM6/TIM7 クロック	1664
40.3.4	タイムベースユニット	1664
40.3.5	カウントモード	1666
40.3.6	UIF ビットの再配置	1673
40.3.7	ADC トリガ	1674
40.3.8	TIM6/TIM7 DMA リクエスト	1674
40.3.9	デバッグモード	1674
40.3.10	TIM6/TIM7 低消費電力モード	1674
40.3.11	TIM6/TIM7 割込み	1674
40.4	TIM6/TIM7 レジスタ	1675
40.4.1	TIMx 制御レジスタ 1 (TIMx_CR1) (x = 6 ~ 7)	1675
40.4.2	TIMx 制御レジスタ 2 (TIMx_CR2) (x = 6 ~ 7)	1677
40.4.3	TIMx DMA / 割込み有効レジスタ (TIMx_DIER) (x = 6 ~ 7)	1677
40.4.4	TIMx ステータスレジスタ (TIMx_SR) (x = 6 ~ 7)	1678
40.4.5	TIMx イベント生成レジスタ (TIMx_EGR) (x = 6 ~ 7)	1678
40.4.6	TIMx カウンタ (TIMx_CNT) (x = 6 ~ 7)	1679
40.4.7	TIMx プリスケアラ (TIMx_PSC) (x = 6 ~ 7)	1679
40.4.8	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 6 ~ 7)	1680
40.4.9	TIMx レジスタマップ	1681
41	汎用タイマ (TIM12/TIM13/TIM14)	1682
41.1	TIM12/TIM13/TIM14 の概要	1682
41.2	TIM12 の主な特徴	1682

41.3	TIM13/TIM14 の主な特徴	1683
41.4	TIM12/TIM13/TIM14 機能詳細	1684
41.4.1	ブロック図	1684
41.4.2	TIM12/TIM13/TIM14 ピンおよび内部信号	1685
41.4.3	タイムベースユニット	1687
41.4.4	カウンタモード	1689
41.4.5	クロック選択	1692
41.4.6	キャプチャ／比較チャネル	1695
41.4.7	入力キャプチャモード	1697
41.4.8	PWM 入力モード (TIM12 のみ)	1698
41.4.9	強制出力モード	1699
41.4.10	出力比較モード	1699
41.4.11	PWM モード	1701
41.4.12	組合せ PWM モード (TIM12 のみ)	1706
41.4.13	ワンパルスモード	1707
41.4.14	再トリガ可能なワンパルスモード (TIM12 のみ)	1709
41.4.15	UIF ビットの再配置	1710
41.4.16	タイマ入力 XOR 機能	1710
41.4.17	TIM12 外部トリガ同期	1710
41.4.18	スレーブモード - リセットモードとトリガモードの組合せ	1713
41.4.19	スレーブモード - リセットモードとゲートモードの組合せ	1713
41.4.20	タイマ同期 (TIM12 のみ)	1713
41.4.21	タイマ出力を他のタイマのトリガとして使用 (TIM13/TIM14 のみ)	1713
41.4.22	ADC トリガ (TIM12 のみ)	1713
41.4.23	デバッグモード	1713
41.5	TIM12/TIM13/TIM14 低消費電力モード	1714
41.6	TIM12/TIM13/TIM14 割込み	1714
41.7	TIM12 レジスタ	1715
41.7.1	TIM12 制御レジスタ 1 (TIM12_CR1)	1715
41.7.2	TIM12 制御レジスタ 2 (TIM12_CR2)	1716
41.7.3	TIM12 スレーブモード制御レジスタ (TIM12_SMCR)	1717
41.7.4	TIM12 割込み有効レジスタ (TIM12_DIER)	1718
41.7.5	TIM12 ステータスレジスタ (TIM12_SR)	1719
41.7.6	TIM12 イベント生成レジスタ (TIM12_EGR)	1720
41.7.7	TIM12 キャプチャ／比較モードレジスタ 1 (TIM12_CCMR1)	1721
41.7.8	TIM12 キャプチャ／比較モードレジスタ 1 [オルタネート] (TIM12_CCMR1)	1722

41.7.9	TIM12 キャプチャ／比較有効レジスタ (TIM12_CCER)	1724
41.7.10	TIM12 カウンタ (TIM12_CNT)	1726
41.7.11	TIM12 プリスケラ (TIM12_PSC)	1726
41.7.12	TIM12 自動再ロードレジスタ (TIM12_ARR)	1727
41.7.13	TIM12 キャプチャ／比較レジスタ 1 (TIM12_CCR1)	1728
41.7.14	TIM12 キャプチャ／比較レジスタ 2 (TIM12_CCR2)	1729
41.7.15	TIM12 タイマ入力選択レジスタ (TIM12_TISEL)	1730
41.7.16	TIM12 レジスタマップ	1731
41.8	TIM13/TIM14 レジスタ	1733
41.8.1	TIMx 制御レジスタ 1 (TIMx_CR1) (x = 13、14)	1733
41.8.2	TIMx 割込み有効レジスタ (TIMx_DIER) (x = 13、14)	1734
41.8.3	TIMx ステータスレジスタ (TIMx_SR) (x = 13、14)	1735
41.8.4	TIMx イベント生成レジスタ (TIMx_EGR) (x = 13、14)	1736
41.8.5	TIMx キャプチャ／比較モードレジスタ 1 (TIMx_CCMR1) (x = 13、14)	1736
41.8.6	TIMx キャプチャ／比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 13、14)	1738
41.8.7	TIMx キャプチャ／比較有効レジスタ (TIMx_CCER) (x = 13、14)	1739
41.8.8	TIMx カウンタ (TIMx_CNT) (x = 13、14)	1740
41.8.9	TIMx プリスケラ (TIMx_PSC) (x = 13、14)	1741
41.8.10	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 13、14)	1741
41.8.11	TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 13、14)	1742
41.8.12	TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 13、14)	1743
41.8.13	TIM13/TIM14 レジスタマップ	1743
42	汎用タイマ (TIM15/TIM16/TIM17)	1745
42.1	TIM15/TIM16/TIM17 の概要	1745
42.2	TIM15 の主な機能	1745
42.3	TIM16/TIM17 の主な特徴	1746
42.4	TIM15/TIM16/TIM17 機能詳細	1747
42.4.1	ブロック図	1747
42.4.2	TIM15/TIM16/TIM17 ピンおよび内部信号	1748
42.4.3	タイムベースユニット	1751
42.4.4	カウンタモード	1753
42.4.5	繰返しカウンタ	1757
42.4.6	クロック選択	1758
42.4.7	キャプチャ／比較チャンネル	1760

42.4.8	入力キャプチャモード	1763
42.4.9	PWM 入力モード (TIM15 の場合のみ)	1764
42.4.10	強制出力モード	1765
42.4.11	出力比較モード	1765
42.4.12	PWM モード	1767
42.4.13	組合せ PWM モード (TIM15 のみ)	1772
42.4.14	相補出力とデッドタイム挿入	1773
42.4.15	ブレーク機能の使用	1776
42.4.16	双方向ブレーク入力	1780
42.4.17	外部イベントによる tim_ocxref 信号のクリア	1782
42.4.18	6 ステップ PWM 生成	1782
42.4.19	ワンパルスモード	1784
42.4.20	再トリガ可能なワンパルスモード (TIM15 のみ)	1785
42.4.21	UIF ビットの再配置	1786
42.4.22	タイマ入力 XOR 機能 (TIM15 のみ)	1786
42.4.23	外部トリガ同期 (TIM15 のみ)	1787
42.4.24	スレーブモード - リセットモードとトリガモードの組合せ (TIM15 のみ)	1789
42.4.25	スレーブモード - リセットモードとゲートモードの組合せ (TIM15 のみ)	1790
42.4.26	タイマ同期 (TIM15 のみ)	1790
42.4.27	タイマ出力を他のタイマのトリガとして使用 (TIM16/TIM17 のみ)	1790
42.4.28	ADC トリガ (TIM15 のみ)	1790
42.4.29	DMA パーストモード	1790
42.4.30	TIM15/TIM16/TIM17 DMA リクエスト	1791
42.4.31	デバッグモード	1792
42.5	TIM15/TIM16/TIM17 低消費電力モード	1792
42.6	TIM15/TIM16/TIM17 割込み	1792
42.7	TIM15 レジスタ	1793
42.7.1	TIM15 制御レジスタ 1 (TIM15_CR1)	1793
42.7.2	TIM15 制御レジスタ 2 (TIM15_CR2)	1794
42.7.3	TIM15 スレーブモード制御レジスタ (TIM15_SMCR)	1796
42.7.4	TIM15 DMA / 割込み有効レジスタ (TIM15_DIER)	1798
42.7.5	TIM15 ステータスレジスタ (TIM15_SR)	1799
42.7.6	TIM15 イベント生成レジスタ (TIM15_EGR)	1801
42.7.7	TIM15 キャプチャ / 比較モードレジスタ 1 (TIM15_CCMR1)	1802
42.7.8	TIM15 キャプチャ / 比較モードレジスタ 1 [オルタネート] (TIM15_CCMR1)	1803
42.7.9	TIM15 キャプチャ / 比較有効レジスタ (TIM15_CCER)	1806

42.7.10	TIM15 カウンタ (TIM15_CNT)	1809
42.7.11	TIM15 プリスケアラ (TIM15_PSC)	1809
42.7.12	TIM15 の自動再ロードレジスタ (TIM15_ARR)	1810
42.7.13	TIM15 繰返しカウンタレジスタ (TIM15_RCR)	1810
42.7.14	TIM15 のキャプチャ/比較レジスタ 1 (TIM15_CCR1)	1811
42.7.15	TIM15 のキャプチャ/比較レジスタ 2 (TIM15_CCR2)	1812
42.7.16	TIM15 ブレークおよびデッドタイムレジスタ (TIM15_BDTR)	1813
42.7.17	TIM15 タイマデッドタイムレジスタ 2 (TIM15_DTR2)	1816
42.7.18	TIM15 入力選択レジスタ (TIM15_TISEL)	1817
42.7.19	TIM15 オルタネート機能レジスタ 1 (TIM15_AF1)	1818
42.7.20	TIM15 オルタネート機能レジスタ 2 (TIM15_AF2)	1820
42.7.21	TIM15 DMA 制御レジスタ (TIM15_DCR)	1821
42.7.22	完全転送の TIM15 DMA アドレス (TIM15_DMAR)	1822
42.7.23	TIM15 レジスタマップ	1822
42.8	TIM16/TIM17 レジスタ	1825
42.8.1	TIMx 制御レジスタ 1 (TIMx_CR1) (x = 16 ~ 17)	1825
42.8.2	TIMx 制御レジスタ 2 (TIMx_CR2) (x = 16 ~ 17)	1826
42.8.3	TIMx DMA / 割込み有効レジスタ (TIMx_DIER) (x = 16 ~ 17)	1827
42.8.4	TIMx ステータスレジスタ (TIMx_SR) (x = 16 ~ 17)	1828
42.8.5	TIMx イベント生成レジスタ (TIMx_EGR) (x = 16 ~ 17)	1829
42.8.6	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) (x = 16 ~ 17)	1830
42.8.7	TIMx キャプチャ/比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 16 ~ 17)	1831
42.8.8	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 16 ~ 17)	1833
42.8.9	TIMx カウンタ (TIMx_CNT) (x = 16 ~ 17)	1835
42.8.10	TIMx プリスケアラ (TIMx_PSC) (x = 16 ~ 17)	1835
42.8.11	TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 16 ~ 17)	1836
42.8.12	TIMx 繰返しカウンタレジスタ (TIMx_RCR) (x = 16 ~ 17)	1836
42.8.13	TIMx キャプチャ/比較レジスタ 1 (TIMx_CCR1) (x = 16 ~ 17)	1837
42.8.14	TIMx ブレークおよびデッドタイムレジスタ (TIMx_BDTR) (x = 16 ~ 17)	1838
42.8.15	TIMx タイマデッドタイムレジスタ 2 (TIMx_DTR2) (x = 16 ~ 17)	1841
42.8.16	TIMx 入力選択レジスタ (TIMx_TISEL) (x = 16 ~ 17)	1842
42.8.17	TIMx オルタネート機能レジスタ 1 (TIMx_AF1) (x = 16 ~ 17)	1842
42.8.18	TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 16 ~ 17)	1845
42.8.19	TIMx DMA 制御レジスタ (TIMx_DCR) (x = 16 ~ 17)	1845
42.8.20	完全転送の TIM16/TIM17 DMA アドレス (TIMx_DMAR) (x = 16 ~ 17) ...	1846

42.8.21	TIM16/TIM17 レジスタマップ	1847
43	低消費電力タイマ (LPTIM)	1849
43.1	概要	1849
43.2	LPTIM の主な機能	1849
43.3	LPTIM の実装	1850
43.4	LPTIM の機能説明	1851
43.4.1	LPTIM ブロック図	1851
43.4.2	LPTIM ピンおよび内部信号	1852
43.4.3	LPTIM 入力およびトリガマッピング	1854
43.4.4	LPTIM のリセットとクロック	1855
43.4.5	グリッチフィルタ	1856
43.4.6	プリスケアラ	1856
43.4.7	トリガマルチプレクサ	1857
43.4.8	動作モード	1857
43.4.9	タイムアウト機能	1859
43.4.10	波形生成	1859
43.4.11	レジスタの更新	1860
43.4.12	カウンタモード	1861
43.4.13	タイマ有効	1861
43.4.14	タイマカウンタのリセット	1862
43.4.15	エンコーダモード	1862
43.4.16	繰返しカウンタ	1864
43.4.17	キャプチャ/比較チャネル	1865
43.4.18	入力キャプチャモード	1866
43.4.19	PWM モード	1867
43.4.20	DMA リクエスト	1869
43.4.21	デバッグモード	1870
43.5	LPTIM 低消費電力モード	1870
43.6	LPTIM 割込み	1871
43.7	LPTIM レジスタ	1872
43.7.1	LPTIM4 割込みおよびステータスレジスタ (LPTIM4_ISR)	1872
43.7.2	LPTIMx 割込みおよびステータスレジスタ [オルタネート] (LPTIMx_ISR) (x = 1 ~ 3、5、6)	1873
43.7.3	LPTIMx 割込みおよびステータスレジスタ [オルタネート] (LPTIMx_ISR) (x = 1 ~ 3、5、6)	1875
43.7.4	LPTIM4 割込みクリアレジスタ (LPTIM4_ICR)	1877

43.7.5	LPTIMx 割込みクリアレジスタ [オルタネート] (LPTIMx_ICR) (x = 1 ~ 3、5、6)	1878
43.7.6	LPTIMx 割込みクリアレジスタ [オルタネート] (LPTIMx_ICR) (x = 1 ~ 3、5、6)	1880
43.7.7	LPTIM4 割込み有効レジスタ (LPTIM4_DIER)	1881
43.7.8	LPTIMx 割込み有効レジスタ [オルタネート] (LPTIMx_DIER) (x = 1 ~ 3、5、6)	1882
43.7.9	LPTIMx 割込み有効レジスタ [オルタネート] (LPTIMx_DIER) (x = 1 ~ 3、5、6)	1884
43.7.10	LPTIM 設定レジスタ (LPTIM_CFGR)	1886
43.7.11	LPTIM 制御レジスタ (LPTIM_CR)	1889
43.7.12	LPTIM 比較レジスタ 1 (LPTIM_CCR1)	1890
43.7.13	LPTIM 自動再ロードレジスタ (LPTIM_ARR)	1891
43.7.14	LPTIM カウンタレジスタ (LPTIM_CNT)	1891
43.7.15	LPTIM 設定レジスタ 2 (LPTIM_CFGR2)	1892
43.7.16	LPTIM 繰返しレジスタ (LPTIM_RCR)	1893
43.7.17	LPTIM キャプチャ/比較モードレジスタ 1 (LPTIM_CCMR1)	1893
43.7.18	LPTIM 比較レジスタ 2 (LPTIM_CCR2)	1896
43.7.19	LPTIM レジスタマップ	1897
44	独立型ウォッチドッグ (IWDG)	1899
44.1	概要	1899
44.2	IWDG の主な機能	1899
44.3	IWDG の実装	1899
44.4	IWDG の機能説明	1900
44.4.1	IWDG ブロック図	1900
44.4.2	IWDG 内部信号	1900
44.4.3	ソフトウェアおよびハードウェアウォッチドッグモード	1901
44.4.4	ウィンドウオプション	1902
44.4.5	デバッグ	1905
44.4.6	レジスタのアクセス保護	1905
44.5	IWDG 低電力モード	1906
44.6	IWDG 割込み	1906
44.7	IWDG レジスタ	1908
44.7.1	IWDG キーレジスタ (IWDG_KR)	1909
44.7.2	IWDG プリスケアラレジスタ (IWDG_PR)	1910
44.7.3	IWDG 再ロードレジスタ (IWDG_RLR)	1911
44.7.4	IWDG ステータスレジスタ (IWDG_SR)	1911

44.7.5	IWDG ウィンドウレジスタ (IWDG_WINR)	1913
44.7.6	IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)	1913
44.7.7	IWDG レジスタマップ	1915
45	システムウィンドウ型ウォッチドッグ (WWDG)	1916
45.1	概要	1916
45.2	WWDG の主な機能	1916
45.3	WWDG の実装	1916
45.4	WWDG の機能説明	1917
45.4.1	WWDG ブロック図	1917
45.4.2	WWDG 内部信号	1917
45.4.3	ウォッチドッグの有効化	1918
45.4.4	ダウンカウンタの制御	1918
45.4.5	ウォッチドッグタイムアウトをプログラムする方法	1918
45.4.6	デバッグモード	1919
45.5	WWDG 割込み	1920
45.6	WWDG レジスタ	1921
45.6.1	WWDG 制御レジスタ (WWDG_CR)	1921
45.6.2	WWDG 設定レジスタ (WWDG_CFR)	1922
45.6.3	WWDG ステータスレジスタ (WWDG_SR)	1922
45.6.4	WWDG レジスタマップ	1923
46	リアルタイムクロック (RTC)	1924
46.1	概要	1924
46.2	RTC の主な機能	1924
46.3	RTC の機能説明	1925
46.3.1	RTC ブロック図	1925
46.3.2	RTC ピンおよび内部信号	1926
46.3.3	RTC および TAMP によって制御される GPIO	1927
46.3.4	RTC セキュア保護モード	1931
46.3.5	RTC 特権保護モード	1932
46.3.6	クロックとプリスケアラ	1934
46.3.7	リアルタイムクロックとカレンダー	1935
46.3.8	カレンダー超低消費電力モード	1935
46.3.9	プログラム可能なアラーム	1935
46.3.10	周期的自動ウェイクアップ	1936

46.3.11	RTC の初期化と設定	1937
46.3.12	カレンダーの読出し	1939
46.3.13	RTC のリセット	1940
46.3.14	RTC の同期	1941
46.3.15	RTC リファレンスクロック検出	1941
46.3.16	RTC の高精度デジタル較正	1942
46.3.17	タイムスタンプ機能	1944
46.3.18	較正クロック出力	1945
46.3.19	タンパおよびアラーム出力	1945
46.4	RTC 低消費電力モード	1946
46.5	RTC 割込み	1947
46.6	RTC レジスタ	1948
46.6.1	RTC 時刻レジスタ (RTC_TR)	1948
46.6.2	RTC 日付レジスタ (RTC_DR)	1949
46.6.3	RTC サブセカンドレジスタ (RTC_SSR)	1950
46.6.4	RTC 初期化制御およびステータスレジスタ (RTC_ICSR)	1951
46.6.5	RTC プリスケアラレジスタ (RTC_PRER)	1953
46.6.6	RTC ウェイクアップタイマレジスタ (RTC_WUTR)	1954
46.6.7	RTC 制御レジスタ (RTC_CR)	1955
46.6.8	RTC 特権モード制御レジスタ (RTC_PRIVCFGR)	1958
46.6.9	RTC セキュア設定レジスタ (RCC_SECCFGR)	1959
46.6.10	RTC 書込み保護レジスタ (RTC_WPR)	1960
46.6.11	RTC 較正レジスタ (RTC_CALR)	1961
46.6.12	RTC シフト制御レジスタ (RTC_SHIFTR)	1962
46.6.13	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	1963
46.6.14	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	1964
46.6.15	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)	1965
46.6.16	RTC アラーム A レジスタ (RTC_ALRMAR)	1965
46.6.17	RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)	1967
46.6.18	RTC アラーム B レジスタ (RTC_ALRMBR)	1968
46.6.19	RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)	1969
46.6.20	RTC ステータスレジスタ (RTC_SR)	1970
46.6.21	RTC 非セキュアマスク済み割込みステータスレジスタ (RTC_MISR)	1971
46.6.22	RTC セキュアマスク済み割込みステータスレジスタ (RTC_SMISR)	1972
46.6.23	RTC ステータスクリアレジスタ (RTC_SCR)	1973
46.6.24	RTC オプションレジスタ (RTC_OR)	1974
46.6.25	RTC アラーム A バイナリモードレジスタ (RTC_ALRABINR)	1974

46.6.26	RTC アラーム B バイナリモードレジスタ (RTC_ALRBBINR)	1975
46.6.27	RTC レジスタマップ	1976
47	タンパおよびバックアップレジスタ (TAMP)	1978
47.1	概要	1978
47.2	TAMP の主な機能	1978
47.3	TAMP の機能説明	1979
47.3.1	TAMP ブロック図	1979
47.3.2	TAMP ピンおよび内部信号	1980
47.3.3	RTC および TAMP によって制御される GPIO	1982
47.3.4	TAMP レジスタ書込み保護	1982
47.3.5	TAMP セキュア保護モード	1983
47.3.6	バックアップレジスタ保護ゾーン	1983
47.3.7	TAMP 特権保護モード	1984
47.3.8	ブート・ハードウェア・キー (BHK)	1984
47.3.9	タンパ検出	1984
47.3.10	TAMP バックアップレジスタとその他のデバイス機密情報の消去	1985
47.3.11	タンパ検出設定と初期化	1986
47.4	TAMP 低消費電力モード	1992
47.5	TAMP 割込み	1993
47.6	TAMP レジスタ	1993
47.6.1	TAMP 制御レジスタ 1 (TAMP_CR1)	1993
47.6.2	TAMP 制御レジスタ 2 (TAMP_CR2)	1995
47.6.3	TAMP 制御レジスタ 3 (TAMP_CR3)	1998
47.6.4	TAMP フィルタ制御レジスタ (TAMP_FLTCR)	1999
47.6.5	TAMP アクティブタンパ制御レジスタ 1 (TAMP_ATCR1)	2001
47.6.6	TAMP アクティブタンパシードレジスタ (TAMP_ATSEEDR)	2003
47.6.7	TAMP アクティブタンパ出力レジスタ (TAMP_ATOR)	2004
47.6.8	TAMP アクティブタンパ制御レジスタ 2 (TAMP_ATCR2)	2005
47.6.9	TAMP セキュア設定レジスタ (TAMP_SECCFGR)	2008
47.6.10	TAMP 特権設定レジスタ (TAMP_PRIVCFGR)	2010
47.6.11	TAMP 割込み有効レジスタ (TAMP_IER)	2011
47.6.12	TAMP ステータスレジスタ (TAMP_SR)	2013
47.6.13	TAMP 非セキュアマスク済み割込みステータスレジスタ (TAMP_MISR)	2015
47.6.14	TAMP セキュアマスク済み割込みステータスレジスタ (TAMP_SMISR)	2016
47.6.15	TAMP ステータスクリアレジスタ (TAMP_SCR)	2018

47.6.16	TAMP 単調カウンタ 1 レジスタ (TAMP_COUNT1R)	2020
47.6.17	TAMP オプションレジスタ (TAMP_OR)	2020
47.6.18	TAMP リソース保護設定レジスタ (TAMP_RPCFGR)	2021
47.6.19	TAMP バックアップ x レジスタ (TAMP_BKPxR)	2022
47.6.20	TAMP レジスタマップ	2023
48	I²C (Inter-integrated circuit) インタフェース	2025
48.1	概要	2025
48.2	I ² C の主な機能	2025
48.3	I ² C の実装	2026
48.4	I ² C の機能詳細	2026
48.4.1	I ² C ブロック図	2027
48.4.2	I ² C ピンおよび内部信号	2028
48.4.3	I ² C クロックの要件	2028
48.4.4	モード選択	2028
48.4.5	I ² C の初期化	2029
48.4.6	ソフトウェアリセット	2034
48.4.7	データ転送	2034
48.4.8	I ² C スレーブモード	2036
48.4.9	I ² C マスタモード	2045
48.4.10	I2C_TIMINGR レジスタの設定例	2057
48.4.11	SMBus 固有の機能	2058
48.4.12	SMBus 初期化	2060
48.4.13	SMBus : I2C_TIMEOCTR レジスタの設定例	2062
48.4.14	SMBus スレーブモード	2063
48.4.15	アドレス一致時に STOP モードからウェイクアップ	2070
48.4.16	エラー条件	2070
48.4.17	DMA リクエスト	2072
48.4.18	デバッグモード	2073
48.5	I ² C 低電力モード	2073
48.6	I ² C 割込み	2074
48.7	I ² C レジスタ	2075
48.7.1	I ² C 制御レジスタ 1 (I2C_CR1)	2075
48.7.2	I ² C 制御レジスタ 2 (I2C_CR2)	2078
48.7.3	I ² C Own Address 1 レジスタ (I2C_OAR1)	2080
48.7.4	I ² C Own Address 2 レジスタ (I2C_OAR2)	2081

48.7.5	I ² C タイミングレジスタ (I2C_TIMINGR)	2082
48.7.6	I ² C タイムアウトレジスタ (I2C_TIMEOUTR)	2083
48.7.7	I ² C 割込みおよびステータスレジスタ (I2C_ISR)	2084
48.7.8	I ² C 割込みクリアレジスタ (I2C_ICR)	2086
48.7.9	I ² C PEC レジスタ (I2C_PECR)	2087
48.7.10	I ² C 受信データレジスタ (I2C_RXDR)	2088
48.7.11	I ² C 送信データレジスタ (I2C_TXDR)	2088
48.7.12	I ² C レジスタマップ	2089
49	I3C (Improved inter-integrated circuit).....	2090
49.1	概要	2090
49.2	I3C の主な機能	2090
49.3	I3C の実装	2092
49.3.1	I3C のインスタンス化	2092
49.3.2	低電力モードからの I3C のウェイクアップ	2092
49.3.3	I3C の FIFO	2092
49.3.4	I3C トリガ	2092
49.3.5	I3C 割込み	2092
49.3.6	I3C MIPI® サポート	2093
49.4	I3C ブロック図	2094
49.5	I3C ピンおよび内部信号	2094
49.6	I3C のリセットおよびクロック	2095
49.6.1	I3C リセット	2095
49.6.2	I3C のクロックおよび要件	2095
49.7	I3C ペリフェラルの状態とプログラミング	2097
49.7.1	I3C ペリフェラルの状態	2097
49.7.2	I3C コントローラの状態とプログラミングシーケンス	2098
49.7.3	I3C ターゲットの状態とプログラミングシーケンス	2102
49.8	I3C レジスタおよびプログラミング	2106
49.8.1	I3C レジスタのセット (コントローラ / ターゲットの場合)	2106
49.8.2	コントローラとしての I3C レジスタとフィールドの使用方法和 ペリフェラルの状態	2107
49.8.3	I3C レジスタとフィールドの使用方法和ペリフェラルの状態 (ターゲットの場合)	2109
49.9	I3C バス転送およびプログラミング	2111
49.9.1	I3C コマンド・セット (CCC) (コントローラ / ターゲットの場合)	2111

49.9.2	I3C ブロードキャスト / 直接 CCC 転送 (ENTDAA、RSTACT を除く) (コントローラの場合)	2115
49.9.3	I3C ブロードキャスト ENTDAA CCC 転送 (コントローラの場合)	2117
49.9.4	I3C ブロードキャスト / 直接 RSTACT CCC 転送 (コントローラの場合) ...	2117
49.9.5	I3C ブロードキャスト / 直接 CCC 転送 (ENTDAA、DEFTGTS、DEFGRPA を除く) (ターゲットの場合)	2119
49.9.6	I3C ブロードキャスト ENTDAA CCC 転送 (ターゲットの場合)	2121
49.9.7	I3C ブロードキャスト DEFTGTS CCC 転送 (ターゲットの場合)	2122
49.9.8	I3C ブロードキャスト DEFGRPA CCC 転送 (ターゲットの場合)	2123
49.9.9	I3C 直接 GETSTATUS CCC 応答 (ターゲットの場合)	2123
49.9.10	I3C プライベート読出し / 書込み転送 (コントローラの場合)	2125
49.9.11	I3C プライベート読出し / 書込み転送 (ターゲットの場合)	2126
49.9.12	従来の I2C 読出し / 書込み転送 (コントローラの場合)	2127
49.9.13	I3C IBI 転送 (コントローラ / ターゲットの場合)	2128
49.9.14	I3C ホットジョイン・リクエスト転送 (コントローラ / ターゲットの場合)	2130
49.9.15	I3C コントローラ機能リクエスト転送 (コントローラ / ターゲットの場合)	2131
49.10	I3C FIFO 管理 (コントローラの場合)	2132
49.10.1	C-FIFO 管理 (コントローラの場合)	2132
49.10.2	TX-FIFO 管理 (コントローラの場合)	2133
49.10.3	RX-FIFO 管理 (コントローラの場合)	2136
49.10.4	S-FIFO 管理 (コントローラの場合)	2138
49.11	I3C FIFO 管理 (ターゲットの場合)	2140
49.11.1	RX-FIFO 管理 (ターゲットの場合)	2140
49.11.2	TX-FIFO 管理 (ターゲットの場合)	2141
49.12	I3C エラー管理	2145
49.12.1	コントローラのエラー管理	2145
49.12.2	ターゲットのエラー管理	2147
49.13	低電力モードからの I3C のウェイクアップ	2148
49.13.1	STOP からのウェイクアップ	2148
49.14	低電力モードの I3C	2150
49.15	I3C 割込み	2151
49.16	I3C レジスタ	2152
49.16.1	I3C メッセージ制御レジスタ (I3C_CR)	2152
49.16.2	I3C メッセージ制御レジスタ [オルタネート] (I3C_CR)	2154
49.16.3	I3C 設定レジスタ (I3C_CFGR)	2156

49.16.4	I3C 受信データバイトレジスタ (I3C_RDR)	2161
49.16.5	I3C 受信データワードレジスタ (I3C_RDWR)	2161
49.16.6	I3C 送信データバイトレジスタ (I3C_TDR)	2162
49.16.7	I3C 送信データワードレジスタ (I3C_TDWR)	2163
49.16.8	I3C IBI ペイロード・データ・レジスタ (I3C_IBIDR)	2165
49.16.9	I3C ターゲット送信設定レジスタ (I3C_TGTTDR)	2166
49.16.10	I3C ステータスレジスタ (I3C_SR)	2167
49.16.11	I3C ステータス・エラー・レジスタ (I3C_SER)	2168
49.16.12	I3C 受信メッセージレジスタ (I3C_RMR)	2170
49.16.13	I3C イベントレジスタ (I3C_EVR)	2171
49.16.14	I3C 割込み有効レジスタ (I3C_IER)	2175
49.16.15	I3C クリアイベントレジスタ (I3C_CEV)	2177
49.16.16	I3C 独自デバイス特性レジスタ (I3C_DEVR0)	2179
49.16.17	I3C デバイス x 特性レジスタ (I3C_DEVRx)	2181
49.16.18	I3C 最大読出し長レジスタ (I3C_MAXRLR)	2183
49.16.19	I3C 最大書込み長レジスタ (I3C_MAXWLR)	2184
49.16.20	I3C タイミングレジスタ 0 (I3C_TIMINGR0)	2185
49.16.21	I3C タイミングレジスタ 1 (I3C_TIMINGR1)	2186
49.16.22	I3C タイミングレジスタ 2 (I3C_TIMINGR2)	2188
49.16.23	I3C バス特性レジスタ (I3C_BCR)	2189
49.16.24	I3C デバイス特性レジスタ (I3C_DCR)	2190
49.16.25	I3C 取得のケーパビリティ・レジスタ (I3C_GETCAPR)	2191
49.16.26	I3C コントローラ機能のケーパビリティ・レジスタ (I3C_CRCAPR)	2192
49.16.27	I3C 最大データ速度の取得レジスタ (I3C_GETMXDSR)	2193
49.16.28	I3C 拡張された専用 ID レジスタ (I3C_EPIDR)	2195
49.16.29	I3C レジスタマップ	2196

50	ユニバーサル同期／非同期レシーバトランスミッタ (USART/UART)	2199
50.1	概要	2199
50.2	USART の主な機能	2199
50.3	USART の拡張機能	2200
50.4	USART の実装	2200
50.5	USART の機能説明	2202
50.5.1	USART ブロック図	2202
50.5.2	USART ピンおよび内部信号	2202
50.5.3	USART クロック	2204

50.5.4	USART キャラクタの説明	2205
50.5.5	USART の FIFO と閾値	2207
50.5.6	USART トランスミッタ	2207
50.5.7	USART レシーバ	2211
50.5.8	USART ボーレート生成	2218
50.5.9	クロック偏差に対する USART レシーバの許容誤差	2219
50.5.10	USART 自動ボーレート検出	2220
50.5.11	USART マルチプロセッサ通信	2222
50.5.12	USART Modbus 通信	2224
50.5.13	USART パリティ制御	2224
50.5.14	USART LIN (Local Interconnection Network) モード	2225
50.5.15	USART 同期モード	2227
50.5.16	USART 単線半二重通信	2231
50.5.17	USART レシーバタイムアウト	2231
50.5.18	USART スマートカードモード	2232
50.5.19	USART IrDA SIR ENDEC ブロック	2236
50.5.20	USART および DMA を使用した連続通信	2238
50.5.21	RS232 ハードウェアフロー制御および RS485 ドライバ有効	2241
50.5.22	USART 低消費電力管理	2243
50.6	低消費電力モードの USART	2246
50.7	USART 割込み	2247
50.8	USART レジスタ	2249
50.8.1	USART 制御レジスタ 1 (USART_CR1)	2249
50.8.2	USART 制御レジスタ 1[オルタネート] (USART_CR1)	2253
50.8.3	USART 制御レジスタ 2 (USART_CR2)	2256
50.8.4	USART 制御レジスタ 3 (USART_CR3)	2260
50.8.5	USART ボーレートレジスタ (USART_BRR)	2264
50.8.6	USART ガード時間およびプリスケアラレジスタ (USART_GTPR)	2265
50.8.7	USART レシーバタイムアウトレジスタ (USART_RTOR)	2266
50.8.8	USART リクエストレジスタ (USART_RQR)	2267
50.8.9	USART 割込みおよびステータスレジスタ (USART_ISR)	2268
50.8.10	USART 割込みおよびステータスレジスタ [オルタネート] (USART_ISR)	2274
50.8.11	USART 割込みフラグクリアレジスタ (USART_ICR)	2279
50.8.12	USART レシーバデータレジスタ (USART_RDR)	2281
50.8.13	USART トランスミッタデータレジスタ (USART_TDR)	2281
50.8.14	USART プリスケアラレジスタ (USART_PRESC)	2282

50.8.15	USART レジスタマップ	2283
51	低消費電力ユニバーサル非同期レシーバトランスミッタ (LPUART)	2285
51.1	概要	2285
51.2	LPUART の主な機能	2285
51.3	LPUART の実装	2286
51.4	LPUART の機能詳細	2288
51.4.1	LPUART ブロック図	2288
51.4.2	LPUART ピンおよび内部信号	2289
51.4.3	LPUART クロック	2290
51.4.4	LPUART キャラクタの説明	2290
51.4.5	LPUART の FIFO と閾値	2292
51.4.6	LPUART トランスミッタ	2292
51.4.7	LPUART レシーバ	2296
51.4.8	LPUART ボーレート生成	2299
51.4.9	クロック偏差に対する LPUART レシーバの許容誤差	2301
51.4.10	LPUART マルチプロセッサ通信	2302
51.4.11	LPUART パリティ制御	2304
51.4.12	LPUART 単線半二重通信	2305
51.4.13	DMA および LPUART を使用した連続通信	2305
51.4.14	RS232 ハードウェアフロー制御および RS485 ドライバ有効	2308
51.4.15	LPUART 低消費電力管理	2310
51.5	低消費電力モードの LPUART	2313
51.6	LPUART 割込み	2314
51.7	LPUART レジスタ	2315
51.7.1	LPUART 制御レジスタ 1 (LPUART_CR1)	2315
51.7.2	LPUART 制御レジスタ 1[オルタネート] (LPUART_CR1)	2318
51.7.3	LPUART 制御レジスタ 2 (LPUART_CR2)	2321
51.7.4	LPUART 制御レジスタ 3 (LPUART_CR3)	2323
51.7.5	LPUART ボーレートレジスタ (LPUART_BRR)	2326
51.7.6	LPUART リクエストレジスタ (LPUART_RQR)	2326
51.7.7	LPUART 割込みおよびステータスレジスタ (LPUART_ISR)	2327
51.7.8	LPUART 割込みとステータスレジスタ [オルタネート] (LPUART_ISR) ..	2331
51.7.9	LPUART 割込みフラグクリアレジスタ (LPUART_ICR)	2335
51.7.10	LPUART 受信データレジスタ (LPUART_RDR)	2336
51.7.11	LPUART 送信データレジスタ (LPUART_TDR)	2336

51.7.12	LPUART プリスケアラレジスタ (LPUART_PRESC)	2337
51.7.13	LPUART レジスタマップ	2338
52	シリアルペリフェラルインタフェース (SPI)	2340
52.1	概要	2340
52.2	SPI の主な機能	2340
52.3	SPI の実装	2341
52.4	SPI の機能説明	2342
52.4.1	SPI ブロック図	2342
52.4.2	SPI ピンおよび内部信号	2343
52.4.3	SPI 通信の一般的側面	2344
52.4.4	マスタとスレーブの 1 対 1 の通信	2345
52.4.5	標準のマルチスレーブ通信	2347
52.4.6	マルチマスタ通信	2350
52.4.7	スレーブ選択 (SS) ピンの管理	2350
52.4.8	レディピン (RDY) の管理	2354
52.4.9	通信フォーマット	2354
52.4.10	SPI の設定	2356
52.4.11	SPI の有効化	2357
52.4.12	SPI のデータ送受信手順	2357
52.4.13	SPI の無効化	2362
52.4.14	データパッキング	2363
52.4.15	DMA (ダイレクトメモリアクセス) を使用する通信	2365
52.5	SPI の特定モードおよび制御	2367
52.5.1	TI モード	2367
52.5.2	SPI エラーフラグ	2368
52.5.3	CRC 計算	2370
52.6	SPI 低電力モード	2372
52.7	SPI 割込み	2372
52.8	I ² S の主な機能	2374
52.9	I ² S の機能詳細	2374
52.9.1	I ² S の概要	2374
52.9.2	SPI 機能とのピンの共用	2375
52.9.3	I ² S/PCM モードで使用できるビットフィールド	2375
52.9.4	スレーブおよびマスタモード	2376
52.9.5	サポートされるオーディオプロトコル	2376

52.9.6	シリアルインタフェースの付加的な柔軟性	2383
52.9.7	起動シーケンス	2385
52.9.8	停止シーケンス	2387
52.9.9	クロックジェネレータ	2388
52.9.10	内部 FIFO	2390
52.9.11	FIFO ステータスフラグ	2391
52.9.12	アンダーラン状況の取り扱い	2392
52.9.13	オーバーラン状況の取り扱い	2393
52.9.14	フレームエラー検出	2393
52.9.15	DMA インタフェース	2395
52.9.16	プログラミングの例	2395
52.10	I ² S 割込み	2398
52.11	SPI/I2S レジスタ	2399
52.11.1	SPI/I2S 制御レジスタ 1 (SPI_CR1)	2399
52.11.2	SPI/I2S 制御レジスタ 2 (SPI_CR2)	2401
52.11.3	SPI/I2S 設定レジスタ 1 (SPI_CFG1)	2401
52.11.4	SPI/I2S 設定レジスタ 2 (SPI_CFG2)	2404
52.11.5	SPI/I2S 割込み有効レジスタ (SPI_IER)	2406
52.11.6	SPI/I2S ステータスレジスタ (SPI_SR)	2407
52.11.7	SPI/I2S 割込み / ステータスフラグクリアレジスタ (SPI_IFCR)	2410
52.11.8	SPI/I2S 送信データレジスタ (SPI_TXDR)	2411
52.11.9	SPI/I2S 受信データレジスタ (SPI_RXDR)	2411
52.11.10	SPI/I2S 多項式レジスタ (SPI_CRCPOLY)	2412
52.11.11	SPI/I2S トランスミッタ CRC レジスタ (SPI_TXCRC)	2412
52.11.12	SPI/I2S レシーバ CRC レジスタ (SPI_RXCRC)	2413
52.11.13	SPI/I2S アンダーランデータレジスタ (SPI_UDRDR)	2413
52.11.14	SPI/I2S 設定レジスタ (SPI_I2SCFGR)	2414
52.11.15	SPI/I2S レジスタマップ	2416
53	シリアルオーディオインタフェース (SAI)	2417
53.1	概要	2417
53.2	SAI の主な機能	2417
53.3	SAI の実装	2418
53.4	SAI の機能説明	2419
53.4.1	SAI ブロック図	2419
53.4.2	SAI ピンおよび内部信号	2420

53.4.3	主要な SAI モード	2421
53.4.4	SAI 同期モード	2422
53.4.5	オーディオデータサイズ	2423
53.4.6	フレーム同期	2423
53.4.7	スロットの設定	2426
53.4.8	SAI クロックジェネレータ	2428
53.4.9	内部 FIFO	2431
53.4.10	PDM インタフェース	2433
53.4.11	AC'97 リンクコントローラ	2441
53.4.12	SPDIF 出力	2443
53.4.13	特有の機能	2445
53.4.14	エラーフラグ	2449
53.4.15	SAI の無効化	2452
53.4.16	SAI DMA インタフェース	2452
53.5	SAI 割込み	2453
53.6	SAI レジスタ	2454
53.6.1	SAI グローバル設定レジスタ (SAI_GCR)	2454
53.6.2	SAI 設定レジスタ 1 (SAI_ACR1)	2455
53.6.3	SAI 設定レジスタ 1 (SAI_BCR1)	2458
53.6.4	SAI 設定レジスタ 2 (SAI_ACR2)	2461
53.6.5	SAI 設定レジスタ 2 (SAI_BCR2)	2463
53.6.6	SAI フレーム設定レジスタ (SAI_AFRCCR)	2465
53.6.7	SAI フレーム設定レジスタ (SAI_BFRCCR)	2466
53.6.8	SAI スロットレジスタ (SAI_ASLOTR)	2467
53.6.9	SAI スロットレジスタ (SAI_BSLOTR)	2468
53.6.10	SAI 割込みマスクレジスタ (SAI_AIM)	2469
53.6.11	SAI 割込みマスクレジスタ (SAI_BIM)	2471
53.6.12	SAI ステータスレジスタ (SAI_ASR)	2472
53.6.13	SAI ステータスレジスタ (SAI_BSR)	2474
53.6.14	SAI クリアフラグレジスタ (SAI_ACLRFR)	2476
53.6.15	SAI クリアフラグレジスタ (SAI_BCLRFR)	2477
53.6.16	SAI データレジスタ (SAI_ADR)	2478
53.6.17	SAI データレジスタ (SAI_BDR)	2479
53.6.18	SAI PDM 制御レジスタ (SAI_PDMCR)	2479
53.6.19	SAI PDM ディレイレジスタ (SAI_PDMDLY)	2480
53.6.20	SAI レジスタマップ	2483

54	FD コントローラエリアネットワーク (FDCAN)	2485
54.1	概要	2485
54.2	FDCAN の主な機能	2487
54.3	FDCAN の機能詳細	2488
54.3.1	ビットタイミング	2489
54.3.2	動作モード	2490
54.3.3	メッセージ RAM	2498
54.3.4	FIFO 確認応答の処理	2506
54.3.5	FDCAN Rx FIFO 要素	2507
54.3.6	FDCAN の Tx バッファ要素	2509
54.3.7	FDCAN Tx イベント FIFO 要素	2511
54.3.8	FDCAN 標準メッセージ ID フィルタ要素	2512
54.3.9	FDCAN 拡張メッセージ ID フィルタ要素	2513
54.4	FDCAN レジスタ	2514
54.4.1	FDCAN コア開放レジスタ (FDCAN_CREL)	2514
54.4.2	FDCAN エンディアンレジスタ (FDCAN_ENDN)	2514
54.4.3	FDCAN データビットタイミングおよびプリスケアラレジスタ (FDCAN_DBTP)	2515
54.4.4	FDCAN テストレジスタ (FDCAN_TEST)	2516
54.4.5	FDCAN RAM ウォッチドッグレジスタ (FDCAN_RWD)	2517
54.4.6	FDCAN CC 制御レジスタ (FDCAN_CCCR)	2517
54.4.7	FDCAN 公称ビットタイミングおよびプリスケアラレジスタ (FDCAN_NBTP)	2520
54.4.8	FDCAN タイムスタンプカウンタ設定レジスタ (FDCAN_TSCC)	2521
54.4.9	FDCAN タイムスタンプカウンタ値レジスタ (FDCAN_TSCV)	2521
54.4.10	FDCAN タイムアウトカウンタ設定レジスタ (FDCAN_TOCC)	2522
54.4.11	FDCAN タイムアウトカウンタ値レジスタ (FDCAN_TOCV)	2523
54.4.12	FDCAN エラーカウンタレジスタ (FDCAN_ECR)	2523
54.4.13	FDCAN プロトコルステータスレジスタ (FDCAN_PSR)	2524
54.4.14	FDCAN トランスミッタ遅延補正レジスタ (FDCAN_TDCR)	2526
54.4.15	FDCAN 割込みレジスタ (FDCAN_IR)	2526
54.4.16	FDCAN 割込み有効レジスタ (FDCAN_IE)	2529
54.4.17	FDCAN 割込みライン選択レジスタ (FDCAN_ILS)	2531
54.4.18	FDCAN 割込みライン有効レジスタ (FDCAN_ILE)	2532
54.4.19	FDCAN グローバルフィルタ設定レジスタ (FDCAN_RXGFC)	2532
54.4.20	FDCAN 拡張 ID およびマスクレジスタ (FDCAN_XIDAM)	2534
54.4.21	FDCAN 高優先メッセージステータスレジスタ (FDCAN_HPMS)	2534

54.4.22	FDCAN Rx FIFO 0 ステータスレジスタ (FDCAN_RXF0S)	2535
54.4.23	CAN Rx FIFO 0 確認応答レジスタ (FDCAN_RXF0A)	2536
54.4.24	FDCAN Rx FIFO 1 ステータスレジスタ (FDCAN_RXF1S)	2536
54.4.25	FDCAN Rx FIFO 1 確認応答レジスタ (FDCAN_RXF1A)	2537
54.4.26	FDCAN Tx バッファ設定レジスタ (FDCAN_TXBC)	2537
54.4.27	FDCAN Tx FIFO / キューステータスレジスタ (FDCAN_TXFQS)	2538
54.4.28	FDCAN Tx バッファリクエストペンディングレジスタ (FDCAN_TXBRP)	2538
54.4.29	FDCAN Tx バッファ追加リクエストレジスタ (FDCAN_TXBAR)	2539
54.4.30	FDCAN Tx バッファキャンセルリクエストレジスタ (FDCAN_TXBCR)	2540
54.4.31	FDCAN Tx バッファ送信発生レジスタ (FDCAN_TXBTO)	2540
54.4.32	FDCAN Tx バッファキャンセル終了レジスタ (FDCAN_TXBCF)	2541
54.4.33	FDCAN Tx バッファ送信割込み有効レジスタ (FDCAN_TXBTIE)	2541
54.4.34	FDCAN Tx バッファキャンセル終了割込み有効レジスタ (FDCAN_TXBCIE)	2542
54.4.35	FDCAN Tx イベント FIFO ステータスレジスタ (FDCAN_TXEFS)	2542
54.4.36	FDCAN Tx イベント FIFO 確認応答レジスタ (FDCAN_TXEFA)	2543
54.4.37	FDCAN CFG クロック分周回路レジスタ (FDCAN_CKDIV)	2544
54.4.38	FDCAN レジスタマップ	2545

55	ユニバーサルシリアルバスフルスピードホスト / デバイスインタフェース (USB)	2548
55.1	概要	2548
55.2	USB の主な機能	2548
55.3	USB の実装	2548
55.4	USB の機能説明	2549
55.4.1	デバイスモードとホストモードの両方で使用される USB ブロックの説明	2551
55.4.2	ホストモードに固有のホストフレームスケジューラ (HFS) の説明	2552
55.5	デバイスモードおよびホストモードでのプログラミングの考慮事項	2553
55.5.1	汎用 USB デバイスのプログラミング	2553
55.5.2	システムリセットとパワーオンリセット	2553
55.5.3	ダブルバッファエンドポイントとデバイスモードでの使用	2559
55.5.4	ダブルバッファチャネル：ホストモードでの使用	2562
55.5.5	デバイスモードでのアイソクロナス転送	2563
55.5.6	ホストモードでのアイソクロナス転送	2564
55.5.7	サスペンド / レジュームイベント	2565

55.6	USB および USB SRAM レジスタ	2569
55.6.1	共通レジスタ	2569
55.6.2	バッファディスクリプタテーブル	2588
55.6.3	USB レジスタマップ	2592
56	USB Type-C®/USB Power Delivery インタフェース (UCPD)	2594
56.1	概要	2594
56.2	UCPD の主な機能	2594
56.3	UCPD の実装	2595
56.4	UCPD の機能詳細	2595
56.4.1	UCPD ブロック図	2596
56.4.2	UCPD のリセットおよびクロック	2597
56.4.3	物理層のプロトコル	2598
56.4.4	UCPD BMC トランスミッタ	2605
56.4.5	UCPD BMC レシーバ	2606
56.4.6	UCPD Type-C のプルアップ (Rp) およびプルダウン (Rd)	2608
56.4.7	UCPD Type-C 電圧監視とデバウンス	2608
56.4.8	UCPD ファストロールスワップ (FRS)	2609
56.4.9	UCPD DMA インタフェース	2609
56.4.10	STOP モードからのウェイクアップ	2609
56.5	UCPD プログラミングシーケンス	2609
56.5.1	初期化フェーズ	2610
56.5.2	Type-C ステートマシンの処理	2610
56.5.3	USB PD の送信	2611
56.5.4	USB PD の受信	2612
56.5.5	UCPD ソフトウェアのトリミング	2613
56.6	UCPD 低電力モード	2613
56.7	UCPD 割込み	2614
56.8	UCPD レジスタ	2615
56.8.1	UCPD 設定レジスタ 1 (UCPD_CFGR1)	2615
56.8.2	UCPD 設定レジスタ 2 (UCPD_CFGR2)	2617
56.8.3	UCPD 設定レジスタ 3 (UCPD_CFGR3)	2618
56.8.4	UCPD 制御レジスタ (UCPD_CR)	2618
56.8.5	UCPD 割込みマスクレジスタ (UCPD_IMR)	2621
56.8.6	UCPD ステータスレジスタ (UCPD_SR)	2622
56.8.7	UCPD 割込みクリアレジスタ (UCPD_ICR)	2625

56.8.8	UCPD Tx 順序集合タイプレジスタ (UCPD_TX_ORDSETR)	2626
56.8.9	UCPD Tx ペイロードサイズレジスタ (UCPD_TX_PAYSZR)	2627
56.8.10	UCPD Tx データレジスタ (UCPD_TXDR)	2627
56.8.11	UCPD Rx 順序集合レジスタ (UCPD_RX_ORDSETR)	2628
56.8.12	UCPD Rx ペイロードサイズレジスタ (UCPD_RX_PAYSZR)	2629
56.8.13	UCPD 受信データレジスタ (UCPD_RXDR)	2629
56.8.14	UCPD Rx 順序集合拡張レジスタ 1 (UCPD_RX_ORDEXTR1)	2630
56.8.15	UCPD Rx 順序集合拡張レジスタ 2 (UCPD_RX_ORDEXTR2)	2630
56.8.16	UCPD レジスタマップ	2631

57	Ethernet (ETH) : メディアアクセス制御 (MAC)、 DMA コントローラ付き	2633
57.1	Ethernet の概要	2633
57.2	Ethernet の主な機能	2633
57.2.1	準拠規格	2633
57.2.2	MAC の機能	2633
57.2.3	トランザクションレイヤ (MTL) の機能	2635
57.2.4	DMA ブロックの機能	2636
57.2.5	バスインタフェースの機能	2636
57.3	Ethernet ピンおよび内部信号	2637
57.4	Ethernet アーキテクチャ	2638
57.4.1	DMA コントローラ	2639
57.4.2	MTL	2648
57.4.3	MAC	2648
57.5)Ethernet 機能の説明 : MAC	2653
57.5.1	ダブル VLAN 処理	2653
57.5.2	ソースアドレスおよび VLAN の挿入、書換え、または削除	2654
57.5.3	パケットフィルタ	2655
57.5.4	IEEE 1588 タイムスタンプサポート	2662
57.5.5	チェックサムオフロードエンジン	2685
57.5.6	TCP セグメンテーションオフロード	2691
57.5.7	IPv4 ARP オフロード	2697
57.5.8	ループバック	2698
57.5.9	フロー制御	2699
57.5.10	MAC 管理カウンタ	2702
57.5.11	MAC によって生成される割込み	2704
57.5.12	MAC および MMC レジスタの説明	2704

57.6	Ethernet 機能の説明 : PHY インタフェース	2705
57.6.1	ステーション管理エージェント (SMA)	2705
57.6.2	メディア独立インタフェース (MII)	2711
57.6.3	減少メディア独立インタフェース (RMII)	2712
57.7	Ethernet 低電力モード	2716
57.7.1	低電力管理	2716
57.7.2	省電力型 Ethernet(EEE)	2722
57.8	Ethernet 割込み	2727
57.8.1	DMA 割込み	2727
57.8.2	MTL 割込み	2728
57.8.3	MAC 割込み	2729
57.9	Ethernet のプログラミングモデル	2730
57.9.1	DMA の初期化	2730
57.9.2	MTL の初期化	2731
57.9.3	MAC の初期化	2731
57.9.4	通常受信および送信動作の実行	2732
57.9.5	送信の停止および開始	2733
57.9.6	RxDMA での新しいディスクリプタリストへの切り替えのための プログラミングガイドライン	2733
57.9.7	AHB クロック周波数を切り替えるためのプログラミングガイドライン	2733
57.9.8	MII リンク状態遷移に関するプログラミングガイドライン	2734
57.9.9	IEEE 1588 タイムスタンプに関するプログラミングガイドライン	2735
57.9.10	PTP オフロード機能に関するプログラミングガイドライン	2736
57.9.11	省電力型 Ethernet (EEE) に関するプログラミングガイドライン	2740
57.9.12	柔軟性の高い Pulse-Per-Second (PPS) 出力に関するプログラミング ガイドライン	2742
57.9.13	TSO に関するプログラミングガイドライン	2744
57.9.14	受信時の VLAN フィルタに関するプログラミングガイドライン	2745
57.10	ディスクリプタ	2745
57.10.1	ディスクリプタの概要	2745
57.10.2	ディスクリプタの構造	2746
57.10.3	送信ディスクリプタ	2748
57.10.4	受信ディスクリプタ	2760
57.11	Ethernet レジスタ	2773
57.11.1	Ethernet レジスタマップ	2773
57.11.2	Ethernet DMA レジスタ	2773
57.11.3	Ethernet MTL レジスタ	2798

57.11.4	Ethernet MAC および MMC レジスタ	2809
58	デバッグサポート (DBG)	2904
58.1	概要	2904
58.2	DBG の機能説明	2905
58.2.1	DBG ブロック図	2905
58.2.2	DBG ピンおよび内部信号	2905
58.2.3	DBG のリセットおよびクロック	2906
58.2.4	DBG 電力ドメイン	2906
58.2.5	デバッグおよび低電力モード	2906
58.2.6	セキュリティ	2907
58.2.7	デバッグ認証	2909
58.3	シリアルワイヤ/JTAG デバッグポート (SWJ-DP)	2911
58.3.1	JTAG デバッグポート	2912
58.3.2	シリアルワイヤデバッグポート	2915
58.3.3	デバッグポートレジスタ	2916
58.3.4	デバッグポートのレジスタマップとリセット値	2923
58.4	アクセスポート	2924
58.4.1	アクセスポートレジスタ	2924
58.4.2	アクセスポートレジスタマップ	2931
58.5	ROM テーブル	2932
58.5.1	システム ROM テーブルレジスタ	2935
58.5.2	システム ROM テーブルレジスタマップ	2940
58.5.3	MCU ROM テーブルレジスタ	2941
58.5.4	MCU ROM テーブルレジスタマップ	2946
58.5.5	プロセッサ ROM テーブルレジスタ	2947
58.5.6	プロセッサ ROM テーブルレジスタマップ	2952
58.6	データウォッチポイントおよびトレースユニット (DWT)	2953
58.6.1	DWT レジスタ	2954
58.6.2	DWT レジスタマップ	2969
58.7	計装トレースマクロセル (ITM)	2971
58.7.1	ITM レジスタ	2972
58.7.2	ITM レジスタマップ	2981
58.8	ブレイクポイントユニット (BPU)	2982
58.8.1	BPU レジスタ	2982
58.8.2	BPU レジスタマップ	2989

58.9	組み込みトレースマクロセル (ETM)	2990
58.9.1	ETM レジスタ	2990
58.9.2	ETM レジスタマップ	3016
58.10	トレースポートインタフェースユニット (TPIU)	3020
58.10.1	TPIU レジスタ	3021
58.10.2	TPIU レジスタマップ	3032
58.11	クロストリガインタフェース (CTI)	3033
58.11.1	CTI レジスタ	3035
58.11.2	CTI レジスタマップ	3047
58.12	マイクロコントローラデバッグユニット (DBGMCU)	3048
58.12.1	デバイス ID	3048
58.12.2	低電力モードエミュレーション	3049
58.12.3	ペリフェラルクロックフリーズ	3049
58.12.4	DBGMCU レジスタ	3051
58.12.5	DBGMCU レジスタマップ	3066
58.13	参照	3068
59	デバイス電子署名	3069
59.1	ユニークデバイス ID レジスタ (96 ビット)	3069
59.2	Flash サイズデータレジスタ	3070
59.3	パッケージデータレジスタ	3071
60	セキュリティ上の重要通知	3072
61	改版履歴	3073

表の一覧

表 1.	TZSC によってセキュリティ保護可能なペリフェラル	106
表 2.	TrustZone 対応ペリフェラル	109
表 3.	メモリマップとペリフェラルレジスタ境界アドレス	112
表 4.	IDAU および SAU によるセキュリティ属性の設定	126
表 5.	MPCWMx リソース	128
表 6.	MPCBBx リソース	128
表 7.	DMA チャンネルの使用 (セキュリティ)	131
表 8.	ペリフェラルと割り当てられた I/O 間のセキュアオルタネート機能	134
表 9.	セキュア I/O に接続できない非セキュアペリフェラル機能	134
表 10.	セキュア I/O に接続できる非セキュアペリフェラル機能	135
表 11.	TrustZone 対応 DBGMCU の非セキュアアクセス管理	136
表 12.	DMA チャンネルの使用 (特権)	139
表 13.	TAMP の内部タンパ	143
表 14.	低消費電力モードが TAMP に与える影響	144
表 15.	加速された暗号操作	147
表 16.	主な製品ライフサイクル遷移	150
表 17.	一般的な製品ライフサイクルの フェーズ	151
表 18.	PRODUCT_STAT によるソフトウェア知的財産保護	156
表 19.	TrustZone が無効な場合 (TZEN = 0xC3) の STM32H562/H563 のブートモード	160
表 20.	TrustZone が有効な場合 (TZEN = 0xB4) の STM32H562/H563 のブートモード	160
表 21.	TrustZone が無効な場合 (TZEN = 0xC3) の STM32H573x のブートモード	161
表 22.	TrustZone が有効な場合 (TZEN = 0xB4) の STM32H573x のブートモード	161
表 23.	GTZC の機能	164
表 24.	GTZC1 サブブロック・アドレスオフセット	165
表 25.	MPCWM リソースの割当	165
表 26.	MPCBB リソースの割当	165
表 27.	サブ領域 A と B のセキュア・プロパティ	168
表 28.	サブ領域 A と B の特権プロパティ	168
表 29.	GTZC 割込みリクエスト	170
表 30.	GTZC1 TZSC レジスタのマップとリセット値	188
表 31.	GTZC1 TZIC レジスタのマップとリセット値	216
表 32.	GTZC1 MPCBBz レジスタのマップとリセット値 (z = 1 から 3)	220
表 33.	内部 SRAM 機能	222
表 34.	低消費電力モードが RAMCFG に与える影響	224
表 35.	RAMCFG 割込みリクエスト	224
表 36.	RAMCFG レジスタのマップとリセット値	230
表 37.	FLASH 推奨ウェイトステート数とプログラミング遅延	239
表 38.	Flash メモリ OTP 構成	248
表 39.	読出し専用公開データ構成	249
表 40.	メモリ・マップとスワップ・オプション	251
表 41.	FLASH_OPSR コンテンツへの推奨対応	254
表 42.	オプション・バイトの構成	257
表 43.	OB 変更の特定の規則	260
表 44.	閉じた製品で変更可能な OB	261
表 45.	オプション・バイトのキー領域	263
表 46.	デフォルトのセキュア・ウォーターマーク	269
表 47.	Flash メモリ TZ 保護のまとめ	269
表 48.	TZ 保護とバンクまたは全体消去のまとめ	270
表 49.	ウォーターマークベースセキュア領域	270

表 50.	セキュア非表示保護	272
表 51.	HDP 保護のまとめ	272
表 52.	セキュア設定、ブロック・ベースのレジスタ・アクセス条件	273
表 53.	特権保護のまとめ	274
表 54.	特権および全体またはバンク消去	274
表 55.	特権設定レジスタ・アクセス条件 (TZ 有効)	274
表 56.	特権設定レジスタ・アクセス条件 (TZ 無効)	274
表 57.	Flash メモリ・レジスタへのアクセス	275
表 58.	Flash インタフェースレジスタ保護の概要	277
表 59.	高サイクル領域保護のまとめ: データ領域のアドレス範囲へのアクセス	278
表 60.	HDP 保護の定義	278
表 61.	特権セクタとデータ領域: データ領域のアドレス範囲へのアクセス	278
表 62.	製品状態、デバッグ状態、デバッグ・ポリシー	279
表 63.	PRODUCT_STATE 遷移	281
表 64.	TZ OBK 保護のまとめ	282
表 65.	TZ 無効時の OBK 保護のまとめ	282
表 66.	セキュアな制御レジスタへのアクセス条件	283
表 67.	非セキュアな制御レジスタへのアクセス条件	283
表 68.	OTP/RO アクセスの制約	284
表 69.	RSS サービス向け Macros	285
表 70.	RSS ライブラリ・インタフェース機能	285
表 71.	NSS ライブラリ・インタフェース機能	289
表 72.	低電力モードが 内蔵 Flash メモリ に与える影響	291
表 73.	ECC 障害の場所	298
表 74.	Flash 割込みリクエスト	301
表 75.	レジスタマップとリセット値の表	348
表 76.	ICACHE の機能	354
表 77.	n ウェイセッタソシアティブ動作モード (デフォルト) の TAG メモリディメンショニング パラメータ	356
表 78.	ダイレクトマップキャッシュモードの TAG メモリディメンショニングパラメータ	357
表 79.	AHB トランザクションの ICACHE のキャッシュ可能性	359
表 80.	メモリ設定	359
表 81.	ICACHE 再配置領域のサイズ、ベースアドレス、および再配置アドレス	360
表 82.	ICACHE 割込み	364
表 83.	ICACHE レジスタのマップとリセット値	369
表 84.	DCACHE の機能	371
表 85.	TAG メモリディメンショニングパラメータ	374
表 86.	AHB トランザクションの DCACHE のキャッシュ可能性	376
表 87.	DCACHE 割込み	381
表 88.	DCACHE レジスタのマップとリセット値	387
表 89.	PWR の入出力ピン	390
表 90.	PWR 内部入力/出力信号	390
表 91.	低消費電力モードの概要	407
表 92.	動作モードに応じた機能	407
表 93.	SLEEP モード	412
表 94.	メモリシャットオフブロックの選択	412
表 95.	STOP モード	414
表 96.	STANDBY モード	416
表 97.	電力モード出力状態対 MCU 電力モード	417
表 98.	PWR セキュリティ設定一覧	418
表 99.	PWR 割込みリクエスト	419
表 100.	PWR レジスタマップとリセット値	435

表 101.	パッケージのピンまたはボールに接続された RCC スレーブ入力／出力信号	437
表 102.	リセットソース識別 (RCC_RSR)	439
表 103.	STOPWUCK と STOPKERWUCK の説明	452
表 104.	HSIKERON および CSIKERON の動作	452
表 105.	カーネルクロック分配の概要	454
表 106.	RCC セキュリティ設定一覧	460
表 107.	割込みソースとその制御	464
表 108.	RCC レジスタのマップとリセット値	542
表 109.	CRS の機能	547
表 110.	CRS 内部入力 / 出力信号	547
表 111.	低電力モードが CRS に与える影響	551
表 112.	割込み制御ビット	551
表 113.	CRS レジスタのマップとリセット値	557
表 114.	ポートビット設定	559
表 115.	GPIO セキュアビット	567
表 116.	GPIO レジスタのマップとリセット値	577
表 117.	SBS 内部入力／出力信号	581
表 118.	HDPL エンコード値	585
表 119.	SBS ブートロジック	585
表 120.	OBK-HDPL ロジック	590
表 121.	SBS レジスタのマップとリセット値	604
表 122.	ペリフェラル相互接続マトリックス	607
表 123.	GPDMA1/2 チャンネル実装	619
表 124.	低電力モードでの GPDMA1/2 自律モードとウェイクアップ	619
表 125.	プログラムされた GPDMA1/2 リクエスト	619
表 126.	ブロック・リクエストとしてプログラムされた GPDMA1/2 リクエスト	623
表 127.	ペリフェラルの早期終了を伴う GPDMA1/2 チャンネル	624
表 128.	ペリフェラルの早期終了を伴う、プログラムされた GPDMA1/2 リクエスト	624
表 129.	プログラムされた GPDMA1/2 トリガ	624
表 130.	プログラムされた GPDMA ソース／デスティネーション・バースト	645
表 131.	プログラムされたデータ・ハンドリング	650
表 132.	低消費電力モードが GPDMA に与える影響	664
表 133.	GPDMA 割込みリクエスト	665
表 134.	GPDMA レジスタのマップとリセット値	694
表 135.	STM32H563/H573 および STM32H562 ベクタテーブル	697
表 136.	EXTI 信号	703
表 137.	EVG 信号	704
表 138.	EXTI ラインの接続	704
表 139.	マスキングの機能	708
表 140.	レジスタ保護の概要	709
表 141.	EXTI レジスタマップセクション	711
表 142.	EXTI レジスタのマップとリセット値	735
表 143.	CRC 内部入力／出力信号	738
表 144.	CRC レジスタマップとリセット値	743
表 145.	CORDIC 関数	745
表 146.	余弦パラメータ	745
表 147.	正弦パラメータ	746
表 148.	位相パラメータ	746
表 149.	モジュラス・パラメータ	747
表 150.	逆正接パラメータ	748
表 151.	双曲線余弦パラメータ	748
表 152.	双曲線正弦パラメータ	749

表 153.	双曲線逆正接パラメータ	749
表 154.	自然対数パラメータ	750
表 155.	自然対数のスケーリングファクタと対応範囲	750
表 156.	平方根パラメータ	750
表 157.	平方根のスケーリング・ファクタと対応範囲	751
表 158.	精度対繰り返し回数	754
表 159.	CORDIC レジスタのマップとリセット値	761
表 160.	読出し手法と書込み手法の有効な組み合わせ	775
表 161.	FMAC レジスタのマップとリセット値	789
表 162.	NOR/PSRAM バンク選択	795
表 163.	NOR/PSRAM 外部メモリアドレス	796
表 164.	NAND メモリマッピングおよびタイミングレジスタ	796
表 165.	NAND バンク選択	796
表 166.	SDRAM バンク選択	797
表 167.	SDRAM アドレスマッピング	797
表 168.	8 ビットデータバス幅での SDRAM アドレスマッピング	797
表 169.	16 ビットデータバス幅での SDRAM アドレスマッピング	798
表 170.	プログラム可能な NOR/PSRAM のアクセスパラメータ	800
表 171.	非マルチプレクス I/O NOR Flash メモリ	800
表 172.	16 bit マルチプレクス I/O NOR Flash メモリ	801
表 173.	非マルチプレクス I/O PSRAM/SRAM	801
表 174.	16 ビットマルチプレクス I/O PSRAM	802
表 175.	NOR Flash / PSRAM : サポートされているメモリおよびトランザクションの例	802
表 176.	FMC_BCRx ビットフィールド (モード 1)	805
表 177.	FMC_BTRx ビットフィールド (モード 1)	806
表 178.	FMC_BCRx ビットフィールド (モード A)	807
表 179.	FMC_BTRx ビットフィールド (モード A)	808
表 180.	FMC_BWTRx ビットフィールド (モード A)	808
表 181.	FMC_BCRx ビットフィールド (モード 2/B)	810
表 182.	FMC_BTRx ビットフィールド (モード 2/B)	811
表 183.	FMC_BWTRx ビットフィールド (モード 2/B)	811
表 184.	FMC_BCRx ビットフィールド (モード C)	813
表 185.	FMC_BTRx ビットフィールド (モード C)	813
表 186.	FMC_BWTRx ビットフィールド (モード C)	814
表 187.	FMC_BCRx ビットフィールド (モード D)	815
表 188.	FMC_BTRx ビットフィールド (モード D)	816
表 189.	FMC_BWTRx ビットフィールド (モード D)	816
表 190.	FMC_BCRx ビットフィールド (マルチプレクス・モード)	818
表 191.	FMC_BTRx ビットフィールド (マルチプレクス・モード)	819
表 192.	FMC_BCRx ビットフィールド (同期マルチプレクス読出しモード)	824
表 193.	FMC_BTRx ビットフィールド (同期マルチプレクス読出しモード)	824
表 194.	FMC_BCRx ビットフィールド (同期マルチプレクス書込みモード)	825
表 195.	FMC_BTRx ビットフィールド (同期マルチプレクス書込みモード)	826
表 196.	プログラム可能な NAND Flash のアクセス・パラメータ	835
表 197.	8 ビット NAND 型 Flash	836
表 198.	16 ビット NAND 型 Flash	836
表 199.	サポートされるメモリおよびトランザクション	837
表 200.	ECC 結果関連ビット	845
表 201.	SDRAM 信号	846
表 202.	FMC レジスタマップとリセット値	863
表 203.	OCTOSPI の実装	866
表 204.	OCTOSPI の入出力ピン	868

表 205.	OCTOSPI 内部信号	868
表 206.	コマンド / アドレスフェーズの説明	877
表 207.	アドレスアライメントのケース	891
表 208.	OCTOSPI の割込みリクエスト	893
表 209.	OCTOSPI レジスタマップとリセット値	916
表 210.	SDMMC の機能	919
表 211.	SDMMC 動作モード SD および SDIO	922
表 212.	SDMMC 動作モード eMMC	922
表 213.	SDMMC 内部入力 / 出力信号	923
表 214.	SDMMC ピン	924
表 215.	SDMMC コマンドおよびデータの位相選択	925
表 216.	コマンドトークンフォーマット	931
表 217.	CRC ありのショートレスポンスのトークンフォーマット	932
表 218.	CRC なしのショートレスポンスのトークンフォーマット	932
表 219.	CRC ありのロングレスポンスのトークンフォーマット	932
表 220.	固有コマンドの概要	933
表 221.	コマンドパスステータスフラグ	933
表 222.	コマンドパスのエラー処理	934
表 223.	データトークンフォーマット	941
表 224.	データパスステータスフラグとクリアビット	941
表 225.	データパスエラー処理	942
表 226.	データ FIFO アクセス	943
表 227.	送信 FIFO ステータスフラグ	944
表 228.	受信 FIFO ステータスフラグ	945
表 229.	AHB と SDMMC_CLK のクロック周波数の関係	949
表 230.	SDIO 特殊動作制御	950
表 231.	4 ビットモードのスタート、割込み、CRC ステータスの信号検出	954
表 232.	CMD12 の使用事例	958
表 233.	SDMMC 割込み	972
表 234.	レスポンスタイプと SDMMC_RESPxR レジスタ	980
表 235.	SDMMC レジスタマップ	997
表 236.	DLYB 内部入力 / 出力信号	1001
表 237.	遅延ブロック制御	1001
表 238.	DLYB レジスタのマップとリセット値	1004
表 239.	ADC の機能	1007
表 240.	ADC の入出力ピン	1009
表 241.	ADC 内部入力 / 出力信号	1009
表 242.	ADC 相互接続	1009
表 243.	レギュラ外部トリガのトリガ極性の設定	1028
表 244.	インジェクト外部トリガのトリガ極性の設定	1028
表 245.	TSAR タイミングは分解能に依存	1041
表 246.	オフセット計算対データ分解能	1044
表 247.	アナログウォッチドッグチャンネル選択	1054
表 248.	アナログウォッチドッグ 1 の比較	1055
表 249.	アナログウォッチドッグ 2 および 3 の比較	1055
表 250.	最大出力結果対 N と M (灰色のセルは切詰め)	1059
表 251.	オーバーサンプリング動作モードの概要	1063
表 252.	ADC 割込み	1082
表 253.	DELAY ビット対 ADC 分解能	1114
表 254.	ADC グローバルレジスタマップ	1118
表 255.	ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100)	1118

表 256.	ADC レジスタマップとリセット値（マスタおよびスレーブ ADC 共通レジスタ）	1120
表 257.	DTS 内部入力／出力信号	1122
表 258.	サンプリング時間の設定	1125
表 259.	トリガの設定	1126
表 260.	低消費電力モードでの温度センサの動作	1128
表 261.	割込み制御ビット	1129
表 262.	DTS レジスタのマップとリセット値	1137
表 263.	DAC の機能	1139
表 264.	DAC の入出力ピン	1141
表 265.	DAC 内部入力／出力信号	1141
表 266.	DAC の相互接続	1141
表 267.	データフォーマット（12 ビットデータの場合）	1144
表 268.	HFSEL の説明	1144
表 269.	サンプルおよびリフレッシュタイミング	1150
表 270.	チャンネル出力モードの概要	1151
表 271.	低消費電力モードが DAC に与える影響	1157
表 272.	DAC 割込み	1157
表 273.	DAC レジスタのマップとリセット値	1173
表 274.	VREFBUF の標準値	1175
表 275.	VREF バッファモード	1176
表 276.	VREFBUF レジスタマップとリセット値	1178
表 277.	DCMI の入出力ピン	1180
表 278.	DCMI 内部入力／出力信号	1180
表 279.	32 ビットワード（8 ビット幅）でキャプチャされたデータバイトの配置	1182
表 280.	32 ビットワード（10 ビット幅）でキャプチャされたデータバイトの配置	1182
表 281.	32 ビットワード（12 ビット幅）でキャプチャされたデータバイトの配置	1182
表 282.	32 ビットワード（14 ビット幅）でキャプチャされたデータバイトの配置	1183
表 283.	モノクロプログレッシブビデオフォーマットでのデータ保存	1188
表 284.	RGB プログレッシブビデオフォーマットでのデータ保存	1189
表 285.	YCbCr プログレッシブビデオフォーマットでのデータ保存	1189
表 286.	YCbCr プログレッシブビデオフォーマットでのデータ保存 - Y 抽出モード	1189
表 287.	DCMI 割込み	1190
表 288.	DCMI レジスタのマップとリセット値	1201
表 289.	PSSI の入出力ピン	1204
表 290.	PSSI 内部入力／出力信号	1204
表 291.	32 ビットワード（8 ビット幅）でキャプチャされたデータバイトの配置	1205
表 292.	32 ビットワード（16 ビット幅）でキャプチャされたデータバイトの配置	1206
表 293.	PSSI 割込みリクエスト	1209
表 294.	PSSI レジスタのマップとリセット値	1216
表 295.	RNG 内部入力／出力信号	1218
表 296.	RNG 割込みリクエスト	1226
表 297.	RNG の設定	1227
表 298.	RNG レジスタマップとリセット値	1232
表 299.	AES および SAES の機能	1234
表 300.	AES 内部入力／出力信号	1235
表 301.	AES の承認済みの対称キー機能	1235
表 302.	カウンタモードの初期化ベクタの定義	1244
表 303.	GCM モードでの IV レジスタの初期化	1247
表 304.	GCM 最終ブロックの定義	1247
表 305.	CCM モードでの IV レジスタの初期化	1254
表 306.	AES データスワッピングの例	1257
表 307.	AES_KEYRx レジスタでのキーエンディアン（128/256 ビットのキー）	1259

表 308.	AES_IVRx レジスタ上に配置される IVI ビットフィールド	1260
表 309.	AES 割込みリクエスト	1261
表 310.	ECB、CBC および CTR での処理遅延時間	1262
表 311.	GCM および CCM での処理遅延時間（クロックサイクル数）	1262
表 312.	AES レジスタマップとリセット値	1274
表 313.	AES および SAES の機能	1279
表 314.	SAES 内部入力／出力信号	1280
表 315.	SAES の承認済みの対称キー機能	1281
表 316.	カウンタモードの初期化ベクタの定義	1290
表 317.	GCM モードでの IV レジスタの初期化	1293
表 318.	GCM 最終ブロックの定義	1293
表 319.	CCM モードでの IV レジスタの初期化	1299
表 320.	AES データスワッピングの例	1307
表 321.	SAES_KEYRx レジスタでのキーエンディアン（128/256 ビットのキー）	1309
表 322.	SAES_IVRx レジスタ上に配置される IVI ビットフィールド	1311
表 323.	SAES 割込みリクエスト	1313
表 324.	ECB、CBC および CTR での処理遅延時間	1315
表 325.	GCM および CCM での処理遅延時間（SAES カーネルクロックサイクル数）	1315
表 326.	SAES レジスタマップとリセット値	1330
表 327.	HASH 内部入力／出力信号	1333
表 328.	サポートされているハッシュアルゴリズムの情報	1334
表 329.	ハッシュプロセッサ出力	1337
表 330.	HASH 割込みリクエスト	1344
表 331.	処理時間（クロックサイクル数）	1344
表 332.	HASH1 レジスタマップとリセット値	1354
表 333.	内部入力／出力信号	1356
表 334.	PKA 整数算術機能リスト	1357
表 335.	PKA 素体（Fp）楕円曲線機能リスト	1358
表 336.	モンゴメリパラメータ計算	1364
表 337.	剰余加算	1364
表 338.	剰余減算	1364
表 339.	モンゴメリ乗算	1365
表 340.	べき剰余（通常モード）	1366
表 341.	べき剰余（高速モード）	1366
表 342.	べき剰余（保護モード）	1367
表 343.	モジュラ逆数	1367
表 344.	モジュラリダクション	1368
表 345.	算術加算	1368
表 346.	算術減算	1368
表 347.	算術乗算	1369
表 348.	算術比較	1369
表 349.	CRT べき乗	1370
表 350.	楕円曲線 Fp 上の点の確認	1370
表 351.	ECC Fp スカラー乗算	1371
表 352.	ECDSA 署名 - 入力	1372
表 353.	ECDSA 署名 - 出力	1372
表 354.	拡張 ECDSA 署名 - 追加出力	1373
表 355.	ECDSA 検証 - 入力	1373
表 356.	ECDSA 検証 - 出力	1374
表 357.	ECC 完全加算	1374
表 358.	ECC ダブルベースラダー	1375
表 359.	ECC 射影からアフィンへ	1376

表 360.	ECC 演算でサポートされる曲線群	1377
表 361.	べき剰余	1378
表 362.	ECC スカラー乗算	1378
表 363.	ECDSA 署名の平均計算時間	1378
表 364.	ECDSA 検証の平均計算時間	1379
表 365.	ECC ダブルベースラダーの平均計算時間	1379
表 366.	ECC 射影からアフィンへの平均計算時間	1379
表 367.	ECC 完全加算の平均計算時間	1379
表 368.	楕円曲線 Fp 上の点の確認の平均計算時間	1379
表 369.	モンゴメリパラメータの平均計算時間	1379
表 370.	PKA 割込みリクエスト	1380
表 371.	PKA レジスタのマップとリセット値	1384
表 372.	OTFDEC 内部入力／出力信号	1386
表 373.	OTFDEC 割込みリクエスト	1390
表 374.	OTFDEC レジスタのマップとリセット値	1405
表 375.	TIM の入出力ピン	1410
表 376.	TIM 内部入力／出力信号	1410
表 377.	tim_ti1 入力マルチプレクサへの相互接続	1412
表 378.	tim_ti2 入力マルチプレクサへの相互接続	1412
表 379.	tim_ti3 入力マルチプレクサへの相互接続	1412
表 380.	tim_ti4 入力マルチプレクサへの相互接続	1412
表 381.	内部トリガ接続	1412
表 382.	tim_etr 入力マルチプレクサへの相互接続	1413
表 383.	タイマブレーク相互接続	1413
表 384.	タイマ break2 相互接続	1413
表 385.	システムブレーク相互接続	1413
表 386.	CCR および ARR レジスタの変更のディザリングパターン	1447
表 387.	センターアライン PWM モードでの CCR レジスタ変更のディザリングパターン	1448
表 388.	タイマ出力と tim_brk/tim_brk2 入力の動作	1460
表 389.	ブレーク保護解除条件	1461
表 390.	カウント方向とエンコーダ信号 (CC1P = CC2P = 0)	1470
表 391.	カウント方向 対 エンコーダ信号および極性設定	1474
表 392.	DMA リクエスト	1495
表 393.	低消費電力モードが TIM1/TIM8 に与える影響	1495
表 394.	割込みリクエスト	1496
表 395.	ブレーク機能を持つ相補 tim_ocx および tim_ocxn チャネルの出力制御ビット	1522
表 396.	TIMx レジスタマップとリセット値	1547
表 397.	STM32H563/H573 および STM32H562 汎用タイマ	1551
表 398.	TIM の入出力ピン	1553
表 399.	TIM 内部入力／出力信号	1553
表 400.	tim_ti1 入力マルチプレクサへの相互接続	1554
表 401.	tim_ti2 入力マルチプレクサへの相互接続	1554
表 402.	tim_ti3 入力マルチプレクサへの相互接続	1554
表 403.	tim_ti4 入力マルチプレクサへの相互接続	1554
表 404.	TIMx 内部トリガ接続	1555
表 405.	tim_etr 入力マルチプレクサへの相互接続	1555
表 406.	CCR および ARR レジスタの変更のディザリングパターン	1585
表 407.	センターアライン PWM モードでの CCR レジスタ変更のディザリングパターン	1586
表 408.	カウント方向とエンコーダ信号 (CC1P = CC2P = 0)	1595
表 409.	カウント方向 対 エンコーダ信号および極性設定	1599
表 410.	DMA リクエスト	1623
表 411.	低消費電力モードが TIM2/TIM3/TIM4/TIM5 に与える影響	1623

表 412.	割込みリクエスト	1624
表 413.	標準 tim_ocx チャンネルの出力制御ビット	1642
表 414.	TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値	1660
表 415.	TIM 内部入力／出力信号	1663
表 416.	TIMx_ARR レジスタの変更のディザリングパターン	1673
表 417.	DMA リクエスト	1674
表 418.	低消費電力モードが TIM6/TIM7 に与える影響	1674
表 419.	割込みリクエスト	1674
表 420.	TIMx レジスタマップとリセット値	1681
表 421.	TIM の入出力ピン	1685
表 422.	TIM 内部入力／出力信号	1685
表 423.	tim_ti1 入力マルチプレクサへの相互接続	1686
表 424.	tim_ti2 入力マルチプレクサへの相互接続	1686
表 425.	TIMx 内部トリガ接続	1686
表 426.	CCR および ARR レジスタの変更のディザリングパターン	1705
表 427.	低消費電力モードが TIM12/TIM13/TIM14 に与える影響	1714
表 428.	割込みリクエスト	1714
表 429.	標準 tim_ocx チャンネルの出力制御ビット	1725
表 430.	TIM12 レジスタマップとリセット値	1731
表 431.	標準 tim_ocx チャンネルの出力制御ビット	1740
表 432.	TIM13/TIM14 レジスタマップとリセット値	1743
表 433.	TIM の入出力ピン	1748
表 434.	TIM 内部入力／出力信号	1749
表 435.	tim_ti1 入力マルチプレクサへの相互接続	1750
表 436.	tim_ti2 入力マルチプレクサへの相互接続	1750
表 437.	TIMx 内部トリガ接続	1750
表 438.	タイマブレーク相互接続	1751
表 439.	システムブレーク相互接続	1751
表 440.	CCR および ARR レジスタの変更のディザリングパターン	1771
表 441.	ブレーク保護解除条件	1780
表 442.	DMA リクエスト	1791
表 443.	低消費電力モードが TIM15/TIM16/TIM17 に与える影響	1792
表 444.	割込みリクエスト	1792
表 445.	ブレーク機能を持つ相補 tim_ocx および tim_ocxn チャンネルの出力制御ビット (TIM15)	1808
表 446.	TIM15 レジスタマップとリセット値	1822
表 447.	ブレーク機能を持つ相補 tim_oc1 および tim_oc1n チャンネルの出力制御ビット (TIM16/TIM17)	1834
表 448.	TIM16/TIM17 レジスタマップとリセット値	1847
表 449.	STM32H563/H573 および STM32H562 LPTIM 機能	1850
表 450.	LPTIM1/2/3/5/6 の入出力ピン	1852
表 451.	LPTIM4 の入出力ピン	1852
表 452.	LPTIM1/2/3/5/6 の内部信号	1853
表 453.	LPTIM4 内部信号	1853
表 454.	LPTIM1/2/3/4/5/6 外部トリガ接続	1854
表 455.	LPTIM1/2/3/5/6 の入力 1 の接続	1854
表 456.	LPTIM1/2/3/5/6 の入力 2 の接続	1854
表 457.	LPTIM1/2/3/5/6 の入力キャプチャ 1 の接続	1854
表 458.	LPTIM1 入力キャプチャ 2 の接続	1855
表 459.	LPTIM2 入力キャプチャ 2 の接続	1855
表 460.	LPTIM3/5/6 入力キャプチャ 2 の接続	1855
表 461.	プリスケアラ分周比	1856
表 462.	エンコーダのカウントシナリオ	1863

表 463.	入力キャプチャのグリッチフィルタの遅延 (カウンタステップ単位)	1866
表 464.	低消費電力モードが LPTIM に与える影響	1870
表 465.	割込みイベント	1871
表 466.	LPTIM レジスタのマップとリセット値	1897
表 467.	IWDG の機能	1899
表 468.	IWDG 内部入力/出力信号	1900
表 469.	低消費電力モードが IWDG に与える影響	1906
表 470.	IWDG 割込みリクエスト	1908
表 471.	IWDG レジスタのマップとリセット値	1915
表 472.	WWDG の機能	1916
表 473.	WWDG 内部入力/出力信号	1917
表 474.	WWDG 割込みリクエスト	1920
表 475.	WWDG レジスタのマップとリセット値	1923
表 476.	RTC の入出力ピン	1926
表 477.	RTC 内部入力/出力信号	1926
表 478.	RTC 相互接続	1927
表 479.	RTC ピン PC13 の設定	1928
表 480.	PI8 設定	1930
表 481.	RTC_OUT の配置	1931
表 482.	低消費電力モードが RTC に与える影響	1946
表 483.	RTC ピン機能 (モード共通)	1946
表 484.	非セキュア割込みリクエスト	1947
表 485.	セキュア割込みリクエスト	1947
表 486.	RTC レジスタのマップとリセット値	1976
表 487.	TAMP の入出力ピン	1980
表 488.	TAMP 内部入力/出力信号	1980
表 489.	TAMP 相互接続	1981
表 490.	デバイスリソース x のタンパ保護	1986
表 491.	アクティブタンパ出力の変更周期	1988
表 492.	最低 ATPER 値	1989
表 493.	アクティブタンパのフィルタパルス時間	1990
表 494.	低消費電力モードが TAMP に与える影響	1992
表 495.	TAMP ピン機能 (モード共通)	1992
表 496.	割込みリクエスト	1993
表 497.	TAMP レジスタのマップとリセット値	2023
表 498.	STM32H563/H573 および STM32H562I2C の実装	2026
表 499.	I ² C の入出力ピン	2028
表 500.	I ² C 内部入力/出力信号	2028
表 501.	アナログフィルタとデジタルフィルタの比較	2030
表 502.	I2C-SMBus 仕様のデータのセットアップおよびホールド時間	2032
表 503.	I2C 設定	2036
表 504.	I2C-SMBus 仕様のクロックタイミング	2047
表 505.	fI2CCLK = 8 MHz でのタイミング設定の例	2057
表 506.	fI2CCLK = 16 MHz でのタイミング設定の例	2057
表 507.	SMBus タイムアウト仕様	2059
表 508.	SMBus の PEC 設定	2061
表 509.	TIMEOUTA の設定例 (最大 t _{TIMEOUT} = 25 ms)	2062
表 510.	TIMEOUTB の設定例	2062
表 511.	TIMEOUTA の設定例 (最大 t _{IDLE} = 50 μs)	2062
表 512.	低電力モードが I ² C に与える影響	2073
表 513.	I ² C 割込みリクエスト	2074
表 514.	I ² C レジスタマップとリセット値	2089

表 515.	I3C のウェイクアップ	2092
表 516.	I3C FIFO の実装	2092
表 517.	I3C 割込み	2092
表 518.	I3C ペリフェラルのコントローラ / ターゲット機能と MIPI v1.1	2093
表 519.	I3C の入出力ピン	2094
表 520.	I3C 内部入力 / 出力信号	2094
表 521.	I3C レジスタの使用方法	2106
表 522.	I3C レジスタ / フィールドの使用方法とコントローラの状態	2107
表 523.	I3C レジスタ / フィールドの使用方法とターゲットの状態	2109
表 524.	サポートされている I3C CCC のリスト (コントローラ / ターゲットの場合)	2112
表 525.	I3C コントローラのエラー管理	2145
表 526.	I3C ターゲットのエラー管理	2147
表 527.	低電力モードの影響	2150
表 528.	I3C 割込みリクエスト	2151
表 529.	I3C レジスタマップとリセット値	2196
表 530.	STM32H563/H573 および STM32H562 機能	2200
表 531.	USART/LPUART の機能	2201
表 532.	USART の入出力ピン	2203
表 533.	USART 内部入力 / 出力信号	2204
表 534.	サンプリングされたデータからのノイズ検出	2216
表 535.	BRR [3:0] = 0000 のときの USART レシーバの許容誤差	2220
表 536.	BRR [3:0] が 0000 でないときの USART レシーバの許容誤差	2220
表 537.	USART フレームのフォーマット	2224
表 538.	低消費電力モードが USART に与える影響	2246
表 539.	USART 割込みリクエスト	2247
表 540.	USART レジスタのマップとリセット値	2283
表 541.	STM32H563/H573 および STM32H562 機能	2286
表 542.	USART/LPUART の機能	2286
表 543.	LPUART の入出力ピン	2289
表 544.	LPUART 内部入力 / 出力信号	2289
表 545.	lpuart_ker_ck_pres = 32.768 kHz でプログラムされたボーレートのエラー計算	2300
表 546.	fCK = 100 MHz でプログラムされたボーレートのエラー計算	2300
表 547.	LPUART レシーバの許容誤差	2301
表 548.	LPUART フレームのフォーマット	2304
表 549.	低消費電力モードが LPUART に与える影響	2313
表 550.	LPUART 割込みリクエスト	2314
表 551.	LPUART レジスタのマップとリセット値	2338
表 552.	SPI の機能	2341
表 553.	SPI/I2S の入出力ピン	2344
表 554.	SPI 内部入力 / 出力信号	2344
表 555.	低電力モードが SPI に与える影響	2372
表 556.	SPI のウェイクアップおよび割込みリクエスト	2373
表 557.	PCM/I2S モードで利用できるビットフィールド	2375
表 558.	AFCNTR = 1 のとき、SPI/I2S が有効化される前の WS および CK のレベル	2385
表 559.	シリアルデータラインのスワッピング	2385
表 560.	通常の I ² S 周波数での CLKGEN のプログラミング例	2389
表 561.	I2S 割込みリクエスト	2398
表 562.	SPI レジスタのマップとリセット値	2416
表 563.	STM32H563/H573 および STM32H562 SAI の機能	2418
表 564.	SAI 内部入力 / 出力信号	2420
表 565.	SAI の入出力ピン	2420
表 566.	外部同期の選択	2422

表 567.	MCLK_x の有効化条件	2428
表 568.	クロックジェネレータのプログラミング例	2431
表 569.	TDM 設定	2438
表 570.	TDM フレーム設定の例	2440
表 571.	SOPD パターン	2444
表 572.	パリティビットの計算	2444
表 573.	オーディオサンプリング周波数とシンボルレート	2445
表 574.	SAI 割込みソース	2453
表 575.	SAI レジスタのマッピングとリセット値	2483
表 576.	CAN サブシステム I/O 信号	2485
表 577.	FDCAN での DLC コーディング	2492
表 578.	フレーム送信の可能な設定	2504
表 579.	Rx FIFO 要素	2507
表 580.	Rx FIFO 要素の説明	2507
表 581.	Tx バッファおよび FIFO 要素	2509
表 582.	Tx バッファ要素の説明	2509
表 583.	Tx イベント FIFO 要素	2511
表 584.	Tx イベント FIFO 要素の説明	2511
表 585.	標準メッセージ ID フィルタ要素	2512
表 586.	標準メッセージ ID フィルタ要素フィールドの説明	2512
表 587.	拡張メッセージ ID フィルタ要素	2513
表 588.	拡張メッセージ ID フィルタ要素フィールドの説明	2513
表 589.	FDCAN レジスタのマッピングとリセット値	2545
表 590.	STM32H563/H573 および STM32H562 USB の実装	2548
表 591.	ダブルバッファリングバッファフラグの定義	2560
表 592.	バルクダブルバッファリングメモリバッファの用途 (デバイスモード)	2561
表 593.	バルクダブルバッファリングメモリバッファの用途 (ホストモード)	2562
表 594.	アイソクロナスメモリバッファの使用	2564
表 595.	アイソクロナスメモリバッファの使用	2565
表 596.	レジュームイベント検出	2567
表 597.	ホスト用のレジュームイベント検出	2568
表 598.	受信ステータスエンコード	2586
表 599.	エンドポイント / チャネルタイプのエンコード	2586
表 600.	エンドポイント / チャネルの種類の意味	2586
表 601.	送信ステータスエンコード	2586
表 602.	割り当てられるバッファメモリの定義	2589
表 603.	USB レジスタのマッピングとリセット値	2592
表 604.	UCPD の実装	2595
表 605.	UCPD ソフトウェアの trim データ	2595
表 606.	ピンの UCPD 信号	2596
表 607.	UCPD 内部信号	2597
表 608.	4b5b 記号エンコード表	2599
表 609.	順序集合	2600
表 610.	順序集合の検証	2601
表 611.	データサイズ	2601
表 612.	ANAMODE と ANASUBMODE のコーディングおよび TYPEC_VSTATE_CCx とのリンク	2610
表 613.	Type-C シーケンス (転送元: 3A)、ケーブル / シンク接続済み (CC1 は Rd、CC2 は Ra)	2611
表 614.	低電力モードが UCPD に与える影響	2613
表 615.	UCPD 割込みリクエスト	2614
表 616.	UCPD レジスタのマッピングとリセット値	2631
表 617.	Ethernet ペリフェラルピン	2637
表 618.	Ethernet 内部入力 / 出力信号	2638

表 619.	Tx DMA および Rx DMA 優先度スキーム	2647
表 620.	Tx パスでのダブル VLAN 処理機能	2653
表 621.	Rx パスでのダブル VLAN 処理	2654
表 622.	VLTi ビットに基づいた VLAN の挿入または書換え	2655
表 623.	デスティネーションアドレスフィルタリング	2658
表 624.	ソースアドレスフィルタリング	2659
表 625.	VLAN 一致ステータス	2660
表 626.	通常クロック : スナップショット用 PTP メッセージ	2663
表 627.	エンドツーエンドトランスペアレントクロック : スナップショット用 PTP メッセージ	2664
表 628.	ピアツーピアトランスペアレントクロック : スナップショット用 PTP メッセージ	2664
表 629.	PHY インタフェース用の出力遅延と入力遅延	2668
表 630.	最小 PTP クロック周波数例	2669
表 631.	IEEE 1588-2008 で定義されたメッセージ形式	2669
表 632.	IEEE 1588-2008 で定義されたメッセージ形式	2670
表 633.	制御とステータスに必要な IPv6-UDP PTP パケットフィールド	2671
表 634.	制御とステータスに必要な Ethernet PTP パケットフィールド	2672
表 635.	ETH_MACTSCR ビットへのタイムスタンプスナップショットの依存	2673
表 636.	PTP メッセージ生成の基準	2679
表 637.	共通の PTP メッセージヘッダフィールド	2681
表 638.	MAC 送信 PTP モードとワンステップタイムスタンプ動作	2684
表 639.	異なるパケットタイプの送信チェックサムオフロードエンジンの機能	2687
表 640.	異なるパケットタイプの受信チェックサムオフロードエンジンの機能	2690
表 641.	TSO : TCP ヘッダおよび IP ヘッダフィールド	2694
表 642.	ポーズパケットのフィールド	2699
表 643.	Tx MAC フロー制御	2700
表 644.	Rx MAC フロー制御	2700
表 645.	最大受信パケットのサイズ	2703
表 646.	MCD クロックの選択	2705
表 647.	MDIO Clause 45 のフレーム構造	2706
表 648.	MDIO Clause 22 のフレーム構造	2707
表 649.	リモートウェイクアップパケットフィルタレジスタ	2718
表 650.	リモートウェイクアップフィルタフィールドの説明	2719
表 651.	リモートウェイクアップパケットおよび PMT 割込みの生成	2720
表 652.	転送完了割込みの動作	2728
表 653.	TDES0 通常ディスクリプタ (読み出しフォーマット)	2748
表 654.	TDES1 通常ディスクリプタ (読み出しフォーマット)	2749
表 655.	TDES2 通常ディスクリプタ (読み出しフォーマット)	2749
表 656.	TDES3 通常ディスクリプタ (読み出しフォーマット)	2750
表 657.	TDES0 通常ディスクリプタ (書き戻しフォーマット)	2753
表 658.	TDES1 通常ディスクリプタ (書き戻しフォーマット)	2753
表 659.	TDES2 通常ディスクリプタ (書き戻しフォーマット)	2754
表 660.	TDES3 通常ディスクリプタ (書き戻しフォーマット)	2754
表 661.	TDES0 コンテキストディスクリプタ	2757
表 662.	TDES1 コンテキストディスクリプタ	2757
表 663.	TDES2 コンテキストディスクリプタ	2758
表 664.	TDES3 コンテキストディスクリプタ	2758
表 665.	RDES0 通常ディスクリプタ (読み出しフォーマット)	2761
表 666.	RDES1 通常ディスクリプタ (読み出しフォーマット)	2761
表 667.	RDES2 通常ディスクリプタ (読み出しフォーマット)	2761
表 668.	RDES3 通常ディスクリプタ (読み出しフォーマット)	2762
表 669.	RDES0 通常ディスクリプタ (書き戻しフォーマット)	2763
表 670.	RDES1 通常ディスクリプタ (書き戻しフォーマット)	2764

表 671.	RDES2 通常ディスクリプタ（書き戻しフォーマット）	2766
表 672.	RDES3 通常ディスクリプタ（書き戻しフォーマット）	2768
表 673.	RDES0 コンテキストディスクリプタ	2771
表 674.	RDES1 コンテキストディスクリプタ	2772
表 675.	RDES2 コンテキストディスクリプタ	2772
表 676.	RDES3 コンテキストディスクリプタ	2772
表 677.	ETH_DMA 共通レジスタマップとリセット値	2795
表 678.	ETH_DMA_CH レジスタマップとリセット値	2795
表 679.	ETH_MTL レジスタマップとリセット値	2807
表 680.	S2KP および JE ビットに基づいたジャイアントパケットステータス	2813
表 681.	CST および ACS ビットに基づいたパケット長	2813
表 682.	Ethernet MAC レジスタマップとリセット値	2893
表 683.	JTAG /シリアルワイヤデバッグポートのピン	2905
表 684.	トレースポートピン	2905
表 685.	シングルワイヤトレースポートのピン	2906
表 686.	TrustZone が有効な状態（TZEN = 0xB4）での認証信号の状態	2907
表 687.	TrustZone が無効な状態（TZEN = 0xC3）での認証信号の状態	2908
表 688.	ライフサイクル状態とデバッグ状態	2908
表 689.	プロビジョニングするデータの定義	2911
表 690.	許可マスク（エンディアン形式：リトルエンディアン）	2911
表 691.	JTAG-DP データレジスタ	2913
表 692.	パケットリクエスト	2915
表 693.	ACK 応答	2915
表 694.	データ転送	2915
表 695.	デバッグポートレジスタ	2916
表 696.	デバッグポートのレジスタマップとリセット値	2923
表 697.	MEM-AP レジスタ	2925
表 698.	アクセスポートのレジスタマップとリセット値	2931
表 699.	システム ROM テーブル	2932
表 700.	MCU ROM テーブル	2932
表 701.	プロセッサ ROM テーブル	2933
表 702.	システム ROM テーブルレジスタのマップとリセット値	2940
表 703.	MCU ROM テーブルレジスタのマップとリセット値	2946
表 704.	CPU ROM テーブルレジスタのマップとリセット値	2952
表 705.	DWT レジスタのマップとリセット値	2969
表 706.	ITM レジスタのマップとリセット値	2981
表 707.	BPU レジスタのマップとリセット値	2989
表 708.	ETM レジスタのマップとリセット値	3016
表 709.	TPIU レジスタのマップとリセット値	3032
表 710.	CTI 入力	3034
表 711.	CTI 出力	3034
表 712.	CTI レジスタのマップとリセット値	3047
表 713.	ペリフェラルクロックフリーズ制御ビット	3049
表 714.	デバッグモードでのペリフェラルの動作	3050
表 715.	DBGMCU レジスタのマップとリセット値	3066
表 716.	文書改版履歴	3073
表 717.	日本語版文書改版履歴	3073

図の一覧

図 1.	システムアーキテクチャ	103
図 2.	IDAU マッピングに基づくメモリマップ	111
図 3.	TrustZone テクノロジーを使用したセキュア / 非セキュアパーティショニング	123
図 4.	セキュア状態の CPU と非セキュア状態の CPU 間のメモリマップの共有	125
図 5.	セキュアワールド遷移とメモリパーティショニング	126
図 6.	グローバル TrustZone フレームワークと TrustZone 対応	127
図 7.	Flash メモリの TrustZone 保護	130
図 8.	Flash メモリのセキュア HDP 領域	137
図 9.	キー管理原則	145
図 10.	デバイスライフサイクルセキュリティ	149
図 11.	PRODUCT_STATE (簡略化した TrustZone 有効ビュー)	152
図 12.	PRODUCT_STATE (完全な TrustZone 有効ビュー)	153
図 13.	共同開発の原則	155
図 14.	SFI を使用した外部 Flash メモリの保護	157
図 15.	ArmV8-M サブシステム内の GTZC のブロック図	164
図 16.	GTZC ブロック図	166
図 17.	ウォーターマークベースのメモリ保護コントローラ (領域 x / サブ領域 A および B)	168
図 18.	MPCBB ブロック図	169
図 19.	SRAM1、ECC 付きの SRAM2、および ECC 付きの SRAM3 メモリマップ	223
図 20.	FLASH のブロック図 (簡略化)	234
図 21.	内蔵 Flash メモリの構成 (2-MB デバイス)	236
図 22.	内蔵 Flash メモリの使用	237
図 23.	2 MB デバイスの Flash メモリの高サイクル・データ・メモリ・マップ	250
図 24.	1 MB デバイスの Flash メモリの高サイクル・データ・メモリ・マップ	251
図 25.	Flash バンクスワッピングシーケンス	253
図 26.	OBK 保護確認	264
図 27.	キー書き込みフロー	265
図 28.	スワップのワークフロー	267
図 29.	ユーザ Flash メモリの HDP	271
図 30.	バンク・スワップの際の保護属性の図	276
図 31.	ICACHE ブロック図	355
図 32.	ICACHE TAG およびデータメモリの機能図	357
図 33.	ICACHE 再配置アドレスのメカニズム	360
図 34.	DCACHE ブロック図	372
図 35.	DCACHE TAG およびデータメモリの機能図	375
図 36.	SMPS 付き電源装置	391
図 37.	LDO 付き電源装置	392
図 38.	システム電源設定	394
図 39.	パワーオン (POR) / パワーダウン (PDR) リセットの波形	399
図 40.	BOR の閾値	400
図 41.	PVD の閾値	401
図 42.	AVD の閾値	402
図 43.	VBAT の閾値	403
図 44.	温度の閾値	404
図 45.	RUN モードでのダイナミック電圧スケーリング	405
図 46.	STANDBY モードにおける I/O の状態	415
図 47.	簡略化されたリセット回路図	438
図 48.	クロックツリー	441
図 49.	HSE/LSE クロックソース	442

図 50.	CSI 較正フロー	445
図 51.	PLL ブロック図	446
図 52.	PLL 初期化フロー	449
図 53.	CRS のブロック図	548
図 54.	CRS カウンタの動作	549
図 55.	3 V または 5 V トレラント GPIO (TT または FT) の構造	559
図 56.	入力フローティング/プルアップ/プルダウン設定	563
図 57.	出力設定	564
図 58.	オルタネート機能設定	565
図 59.	ハイインピーダンスアナログ設定	565
図 60.	SBS ブロック図	580
図 61.	補正セルの管理	582
図 62.	補正セルの使用	583
図 63.	SBS ブート制御	584
図 64.	SBS デバッグ制御	587
図 65.	SBS ハードウェア・セキュアストレージ制御	589
図 66.	GPDMA ブロック図	626
図 67.	リンクリストを用いない GPDMA チャンネル・ダイレクト・プログラミング (GPDMA_CxLLR = 0)	627
図 68.	GPDMA チャンネルのサスペンドとレジュームのシーケンス	628
図 69.	GPDMA チャンネルのアボートとリスタートのシーケンス	629
図 70.	チャンネル x のリニア・アドレス指定の静的リンクリスト・データ構造 (すべて Uxx = 1)	630
図 71.	チャンネル x の 2D・アドレス指定の静的リンクリスト・データ構造 (すべて Uxx = 1)	631
図 72.	チャンネル x の リニア・アドレス指定の GPDMA 動的リンクリスト・データ構造	631
図 73.	チャンネル x の 2D・アドレス指定の GPDMA 動的リンクリスト・データ構造	632
図 74.	run-to-completion モード (GPDMA_CxCR.LSM = 0) での GPDMA チャンネル実行および リンクリスト・プログラミング	634
図 75.	補助 GPDMA チャンネル y による LLIn の挿入	636
図 76.	リンク・ステップ・モード (GPDMA_CxCR.LSM = 1) での GPDMA チャンネル実行および リンクリスト・プログラミング	638
図 77.	LLIn+1 の設定: リンク・ステップ・モードでの GPDMA 動的リンクリスト	639
図 78.	リンク・ステップ・モードでのレジスタファイルの LLn</Subscript>	640
図 79.	リンク・ステップ・モードでの新しい LLIn' および LLIn+1' の置き換え (オプション 1)	641
図 80.	リンク・ステップ・モードでメモリの LLIn を新しい LLIn' への置き換え (オプション 2)	642
図 81.	GPDMA チャンネル実行およびリンクリスト・プログラミング	644
図 82.	プログラムされた 2D アドレス指定	647
図 83.	GPDMA アービトレーション・ポリシー	654
図 84.	トリガのヒット、メモリ書込み、オーバーラン波形	657
図 85.	GPDMA サーキュラ・バッファ・プログラミング: リニア・アドレス指定による メモリ開始アドレスの更新	658
図 86.	サーキュラ・バッファのある共有 GPDMA チャンネル: リニア・アドレス指定による メモリ開始アドレスの更新	659
図 87.	EXTI ブロック図	703
図 88.	設定可能なイベントのトリガロジック、CPU ウェイクアップ	706
図 89.	EXTI ダイレクト・イベント	707
図 90.	EXTI マルチプレクサの GPIO 選択	708
図 91.	CRC 計算ユニットのブロック図	738
図 92.	三角関数に対する CORDIC の収束	752
図 93.	双曲線関数に対する CORDIC の収束	753
図 94.	平方根に対する CORDIC の収束	754
図 95.	ブロック図	763
図 96.	入力バッファ領域	765

図 97.	サーキュラ入力バッファ	766
図 98.	サーキュラ入力バッファの操作	767
図 99.	サーキュラ出力バッファ	768
図 100.	サーキュラ出力バッファの操作	769
図 101.	FIR フィルタの構造	771
図 102.	IIR フィルタの構造 (直接型 1)	773
図 103.	X1 バッファの初期化	778
図 104.	フィルタ例 1	779
図 105.	フィルタ例 2	780
図 106.	FMC ブロック図	792
図 107.	FMC メモリバンク	795
図 108.	モード 1 読出しアクセス波形	804
図 109.	モード 1 書込みアクセス波形	805
図 110.	モード A 読出しアクセス波形	806
図 111.	モード A 書込みアクセス波形	807
図 112.	モード 2 およびモード B 読出しアクセス波形	809
図 113.	モード 2 書込みアクセス波形	809
図 114.	モード B 書込みアクセス波形	810
図 115.	モード C 読出しアクセス波形	812
図 116.	モード C 書込みアクセス波形	812
図 117.	モード D 読出しアクセス波形	814
図 118.	モード D 書込みアクセス波形	815
図 119.	マルチプレクス読出しアクセス波形	817
図 120.	マルチプレクス書込みアクセス波形	818
図 121.	読出しアクセス中の非同期ウェイト波形	820
図 122.	書込みアクセス中の非同期ウェイト波形	821
図 123.	待ち設定波形	823
図 124.	同期マルチプレクス読出しモード波形 - NOR、PSRAM (CRAM)	823
図 125.	同期マルチプレクス書込みモード波形 - PSRAM (CRAM)	825
図 126.	共通メモリ・アクセスの NAND Flash コントローラ波形	838
図 127.	「CE don't care」でない NAND Flash へのアクセス	839
図 128.	バースト書込み SDRAM アクセス波形	848
図 129.	バースト読み出し SDRAM アクセス	849
図 130.	RBURST ビットセット時の読出しアクセスの論理図 (CAS=1、RPIPE=0)	850
図 131.	Row 境界をまたぐ読出しアクセス	852
図 132.	Row 境界をまたぐ書込みアクセス	852
図 133.	セルフリフレッシュモード	854
図 134.	パワーダウンモード	855
図 135.	OCTOSPI ブロック図 (オクタル構成)	867
図 136.	OCTOSPI ブロック図 (クワッド構成)	867
図 137.	OCTOSPI ブロック図 (デュアルクワッド構成)	868
図 138.	オクタル構成における SDR 読出しコマンド	869
図 139.	Octal-SPI モードにおける DQS による DTR 読出し (Macronix モード) の例	872
図 140.	Octo-SPI モードにおける SDR 書込みコマンドの例	874
図 141.	Octal-SPI モードにおける DTR 書込み (Macronix モード) の例	875
図 142.	HyperBus 読出し動作の例	876
図 143.	初期遅延を設けた HyperBus 書込み動作	878
図 144.	追加の遅延を設けた HyperBus 読出し動作	878
図 145.	追加の遅延を設けた HyperBus 書込み動作	879
図 146.	遅延なしの HyperBus 書込み動作 (レジスタ書き込み)	879
図 147.	遅延を設けた HyperBus のページまたがり時の読出し動作	880
図 148.	CKMODE = 0 のときの NCS (T = CLK 周期)	890

図 149.	CKMODE = 1 で、SDR モードのときの NCS (T = CLK 周期)	890
図 150.	CKMODE = 1 で、DTR モードのときの NCS (T = CLK 周期)	891
図 151.	CKMODE = 1 で、アボートが発生したときの NCS (T = CLK 周期)	891
図 152.	SDMMC 「レスポンス無し」と「データなし」操作	920
図 153.	SDMMC (複数) ブロック読出し操作	920
図 154.	SDMMC (複数) ブロック書き込み操作	921
図 155.	SDMMC (シーケンシャル) ストリーム読出し操作	921
図 156.	SDMMC (シーケンシャル) ストリーム書き込み操作	921
図 157.	SDMMC のブロック図	923
図 158.	SDMMC コマンドおよびデータの位相関係	925
図 159.	制御ユニット	927
図 160.	コマンド / レスポンスパス	928
図 161.	コマンドパスステートマシン (CPSM)	929
図 162.	データパス	934
図 163.	DDR モードのデータパケットクロック供給	935
図 164.	DDR モードの CRC ステータス / ブート確認応答クロック供給	935
図 165.	データパスステートマシン (DPSM)	936
図 166.	CLKMUX ユニット	946
図 167.	リンクされたリストの構造	948
図 168.	非同期割込みの生成	951
図 169.	同期割込み期間のデータ読出し	952
図 170.	同期割込み期間のデータ書き込み	952
図 171.	非同期割込み期間のデータ読み出し	953
図 172.	非同期割込み期間のデータ書き込み	953
図 173.	DS、ハイスピード、SDR12、SDR25 向け SDMMC_CK でのクロック停止	956
図 174.	DDR50、SDR50、SDR104 向け SDMMC_CK でのクロック停止	956
図 175.	50 MHz 未満の SDMMC_CK での読出しウェイト	957
図 176.	50 MHz 以上の SDMMC_CK での読出しウェイト	958
図 177.	CMD12 ストリームタイミング	960
図 178.	CMD5 SLEEP AWAKE 手順	962
図 179.	通常ブートモード動作	964
図 180.	代替ブートモード動作	965
図 181.	コマンドレスポンス R1b ビジー信号	966
図 182.	SDMMC の状態制御	967
図 183.	カードサイクルパワー / パワーアップの図	968
図 184.	CMD11 信号電圧スイッチシーケンス	969
図 185.	電圧スイッチトランシーバの標準アプリケーション	971
図 186.	DLYB ブロック図	1000
図 187.	ADC のブロック図	1008
図 188.	ADC クロック構成	1012
図 189.	ADC1 接続性	1013
図 190.	ADC2 接続性	1014
図 191.	ADC 較正	1017
図 192.	ADC 較正係数の更新	1018
図 193.	シングルエンドチャンネルと差動チャンネルの混合	1018
図 194.	ADC の有効化 / 無効化	1020
図 195.	パルスモードのタイミング図	1023
図 196.	アナログ / デジタル変換時間	1026
図 197.	実行中のレギュラ変換の停止	1027
図 198.	実行中のレギュラ変換とインジェクト変換の停止	1027
図 199.	ADC マスタとスレーブ間でのトリガの共有	1029
図 200.	インジェクト変換の遅延時間	1030

図 201.	コンテキストの JSQR キューの例 (シーケンスの変更)	1034
図 202.	コンテキストの JSQR キューの例 (トリガの変更)	1034
図 203.	変換前にオーバーフローが発生した場合のコンテキストの JSQR キューの例	1035
図 204.	変換中にオーバーフローが発生した場合のコンテキストの JSQR キューの例	1035
図 205.	空のキューがあるコンテキストの JSQR キューの例 (JQM = 0 の場合)	1036
図 206.	空のキューがあるコンテキストの JSQR キューの例 (JQM = 1)	1037
図 207.	JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に JADSTP が発生	1037
図 208.	JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に JADSTP が発生し、新しいトリガが発生	1038
図 209.	JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に外部で JADSTP が発生	1038
図 210.	JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 1)	1039
図 211.	ADDIS = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0)	1039
図 212.	ADDIS = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 1)	1040
図 213.	シーケンスのシングル変換、ソフトウェアトリガ	1042
図 214.	シーケンスの連続変換、ソフトウェアトリガ	1042
図 215.	シーケンスのシングル変換、ハードウェアトリガ	1043
図 216.	シーケンスの連続変換、ハードウェアトリガ	1043
図 217.	右詰め (オフセット無効、符号なしの値)	1045
図 218.	右詰め (オフセット有効、符号付きの値)	1045
図 219.	左詰め (オフセット無効、符号なしの値)	1046
図 220.	左詰め (オフセット有効、符号付きの値)	1046
図 221.	オーバーラン (OVRMOD = 0) の例	1047
図 222.	オーバーラン (OVRMOD = 1) の例	1048
図 223.	AUTDLY = 1、連続モードのレギュラ変換、ソフトウェアトリガ	1051
図 224.	AUTDLY = 1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN = 0 ; JDISCEN = 0)	1051
図 225.	AUTDLY = 1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN = 1, JDISCEN = 1)	1052
図 226.	AUTDLY = 1、インジェクト変換によって中断されたレギュラ連続変換	1053
図 227.	自動インジェクトモードの AUTDLY = 1 (JAUTO = 1)	1053
図 228.	アナログウォッチドッグによって保護される領域	1054
図 229.	ADCy_AWDx_OUT 信号生成 (すべてのレギュラチャネル)	1056
図 230.	ADCy_AWDx_OUT 信号生成 (ソフトウェアによって AWDx フラグがクリアされない場合)	1057
図 231.	ADCy_AWDx_OUT 信号生成 (1 つのレギュラチャネル)	1057
図 232.	ADCy_AWDx_OUT 信号生成 (すべてのインジェクトチャネル)	1057
図 233.	20 ビットから 16 ビットへの結果の切詰め	1058
図 234.	5 ビットシフトと丸めの数値例	1059
図 235.	オーバーサンプリングモードのトリガ (TROVS ビット = 1)	1060
図 236.	レギュラオーバーサンプリングモード (4x の比率)	1061
図 237.	レギュラおよびインジェクトオーバーサンプリングモードの同時使用	1062
図 238.	インジェクションでのトリガレギュラオーバーサンプリング	1062
図 239.	自動インジェクトモードでのオーバーサンプリング	1063
図 240.	デュアル ADC ブロック図 ⁽¹⁾	1065
図 241.	4 チャネルのインジェクト同時モード: デュアル ADC モード	1066
図 242.	16 チャネルのレギュラ同時モード: デュアル ADC モード	1068
図 243.	連続変換モードにおける 1 チャネルのインタリーブモード: デュアル ADC モード	1069
図 244.	シングル変換モードにおける 1 チャネルのインタリーブモード: デュアル ADC モード	1070
図 245.	インジェクションによるインタリーブ変換	1070
図 246.	オルタネートトリガ: 各 ADC のインジェクトグループ	1071
図 247.	オルタネートトリガ: 不連続モードにおける 4 つのインジェクトチャネル (各 ADC)	1072

図 248.	オルタネート + レギュラ同時	1073
図 249.	インジェクト変換中にトリガが発生した場合	1073
図 250.	インジェクトシーケンス CH11、CH12 とインタリーブシングルチャネル CH 0	1074
図 251.	インジェクトシーケンス CH11、CH12 と 2 つのインタリーブチャネル (CH1、CH2) - ケース 1: マスタ割込み優先	1074
図 252.	インジェクトシーケンス CH11、CH12 と 2 つのインタリーブチャネル (CH1、CH2) - ケース 2: スレーブ割込み優先	1074
図 253.	レギュラ同時モードにおける DMA リクエスト (MDMA = 0b00)	1075
図 254.	レギュラ同時モードにおける DMA リクエスト (MDMA = 0b10)	1076
図 255.	インタリーブモードにおける DMA リクエスト (MDMA = 0b10)	1076
図 256.	温度センサチャネルのブロック図	1078
図 257.	VBAT チャネルのブロック図	1079
図 258.	VREFINT チャネルのブロック図	1080
図 259.	温度センサの機能ブロック図	1122
図 260.	REF_CLK の周波数が低い場合の方法	1124
図 261.	REF_CLK の周波数が高い場合の方法	1124
図 262.	温度センサシーケンス	1127
図 263.	デュアルチャネル DAC ブロック図	1140
図 264.	シングル DAC チャネルモードのデータレジスタ	1143
図 265.	デュアル DAC チャネルモードのデータレジスタ	1143
図 266.	トリガ無効 (TEN = 0) 時の変換タイミング図	1145
図 267.	DAC LFSR レジスタ計算アルゴリズム	1147
図 268.	LFSR 波形生成による DAC 変換 (SW トリガ有効)	1147
図 269.	DAC 三角波生成	1148
図 270.	三角波生成による DAC 変換 (SW トリガ有効)	1148
図 271.	DAC サンプルおよびホールドモードフェーズの図	1151
図 272.	VREFBUF のブロック図	1175
図 273.	DCMI のブロック図	1180
図 274.	DCMI 信号の波形	1181
図 275.	タイミング図	1183
図 276.	スナップショットモードにおけるフレームキャプチャ波形	1185
図 277.	連続グラブモードにおけるフレームキャプチャ波形	1186
図 278.	クロップ後のウィンドウの座標とサイズ	1186
図 279.	データキャプチャ波形	1187
図 280.	ピクセルラスタースキャンの順序	1188
図 281.	PSSI のブロック図	1203
図 282.	最上位ブロック図	1203
図 283.	受信モードでのデータイネーブルの波形図 (CKPOL=0)	1207
図 284.	送信モードでのデータイネーブルの波形図 (CKPOL=0)	1207
図 285.	受信モードでのレディの波形図 (CKPOL=0)	1208
図 286.	双方向 PSSI_DE/PSSI_RDY の波形	1209
図 287.	双方向 PSSI_DE/PSSI_RDY の接続図	1209
図 288.	RNG のブロック図	1218
図 289.	NIST SP800-90B エントロピーソースモデル	1219
図 290.	RNG 初期化の概要	1222
図 291.	AES ブロック図	1234
図 292.	暗号化／復号の一般的な使用法	1236
図 293.	認証時の一般的な動作	1238
図 294.	サスペンドモードの管理例	1239
図 295.	ECB 暗号化	1240
図 296.	ECB 復号	1240
図 297.	CBC 暗号化	1240

図 298.	CBC 復号	1241
図 299.	CTR モードでのメッセージ構造	1243
図 300.	CTR 暗号化	1244
図 301.	GCM でのメッセージ構造	1246
図 302.	GCM 認証暗号化	1247
図 303.	GMAC モードでのメッセージ構造	1250
図 304.	GMAC 認証モード	1251
図 305.	CCM モードでのメッセージ構造	1252
図 306.	CCM モード認証暗号化	1253
図 307.	データ型に応じた 128 ビットブロックの構成	1258
図 308.	SAES ブロック図	1279
図 309.	暗号化／復号の一般的な使用法	1281
図 310.	認証時の一般的な動作	1283
図 311.	サスペンドモードの管理例	1284
図 312.	ECB 暗号化	1285
図 313.	ECB 復号	1285
図 314.	CBC 暗号化	1286
図 315.	CBC 復号	1286
図 316.	CTR モードでのメッセージ構造	1289
図 317.	CTR 暗号化	1290
図 318.	GCM でのメッセージ構造	1291
図 319.	GCM 認証暗号化	1292
図 320.	GMAC モードでのメッセージ構造	1295
図 321.	GMAC 認証モード	1296
図 322.	CCM モードでのメッセージ構造	1297
図 323.	CCM モード認証暗号化	1298
図 324.	ECB および CBC モードでの SAES 用のラップキーを使った動作	1302
図 325.	CTR モードでの SAES 用のラップキーを使った動作	1304
図 326.	共有キーモードの使用	1305
図 327.	データ型に応じた 128 ビットブロックの構成	1308
図 328.	キー保護メカニズム	1310
図 329.	HASH ブロック図	1333
図 330.	メッセージデータスワッピング機能	1335
図 331.	HASH サスペンド／レジュームメカニズム	1341
図 332.	PKA のブロック図	1356
図 333.	OTFDEC ブロック図	1386
図 334.	SoC での OTFDEC の標準的な使用方法	1387
図 335.	AES CTR 復号フロー	1388
図 336.	OTFDEC フロー制御の概要（デュアルバースト読み出しリクエスト）	1389
図 337.	高機能制御タイマのブロック図	1409
図 338.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	1415
図 339.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	1415
図 340.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1417
図 341.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1417
図 342.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1418
図 343.	内部クロック分周比が N の場合のカウンタのタイミング図	1418
図 344.	ARPE=0（TIMx_ARR はプリロードされない）の場合の更新イベント時のカウンタの タイミング図	1419
図 345.	ARPE=1（TIMx_ARR がプリロードされる）の場合の更新イベント時のカウンタの タイミング図	1420
図 346.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1421
図 347.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1422

図 348.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1422
図 349.	内部クロック分周比が N の場合のカウンタのタイミング図	1423
図 350.	繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図	1423
図 351.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	1425
図 352.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1425
図 353.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	1426
図 354.	内部クロック分周比が N の場合のカウンタのタイミング図	1426
図 355.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図	1427
図 356.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	1428
図 357.	モードと TIMx_RCR レジスタの設定に応じた更新レート	1429
図 358.	外部トリガ入力ブロック	1430
図 359.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	1431
図 360.	tim_ti2 外部クロックの接続例	1431
図 361.	外部クロックモード 1 の制御回路	1432
図 362.	外部トリガ入力ブロック	1433
図 363.	外部クロックモード 2 の制御回路	1434
図 364.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	1434
図 365.	キャプチャ/比較チャンネル 1 メイン回路	1435
図 366.	キャプチャ/比較チャンネル (チャンネル 1、同じくチャンネル 2、3 および 4) の出力ステージ	1436
図 367.	キャプチャ/比較チャンネル (チャンネル 5、同じくチャンネル 6) の出力ステージ	1436
図 368.	PWM 入力モードタイミング	1438
図 369.	出力比較モード、tim_oc1 のトグル	1440
図 370.	エッジアライン PWM 波形 (ARR=8)	1442
図 371.	センターアライン PWM 波形 (ARR=8)	1443
図 372.	ディザリング原理	1444
図 373.	ディザリングモードでのデータフォーマットとレジスタのコーディング	1445
図 374.	PWM 分解能 対 周波数	1446
図 375.	PWM ディザリングのパターン	1447
図 376.	センターアライン PWM モードでのデューティサイクルへのディザリングの影響	1448
図 377.	2 位相シフトされた 50% デューティサイクルの PWM 信号の生成	1450
図 378.	チャンネル 1 および 3 における組み合わせ PWM モード	1451
図 379.	周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号	1452
図 380.	対称のデッドタイム挿入のある相補出力	1453
図 381.	非対称のデッドタイム	1454
図 382.	負のパルスより長い遅延があるときのデッドタイムの波形	1454
図 383.	正のパルスより長い遅延があるときのデッドタイムの波形	1454
図 384.	ブレークおよびブレーク 2 回路の概要	1457
図 385.	tim_brk (OSS1=1) でのブレークイベントに対するさまざまな出力の動作	1459
図 386.	tim_brk および tim_brk2 のアサート後の PWM 出力状態 (OSS1=1)	1460
図 387.	tim_brk アサート後の PWM 出力状態 (OSS1=0)	1460
図 388.	出力先変更 (tim_brk2 リクエストは表示なし)	1462
図 389.	tim_ocref_clr 入力選択マルチプレクサ	1463
図 390.	TIMx tim_ocxref のクリア	1463
図 391.	6 ステップ生成 COM の例 (OSSR=1)	1464
図 392.	ワンパルスモードの例	1465
図 393.	再トリガ可能なワンパルスモード	1467
図 394.	パルス発生回路	1467
図 395.	エッジアラインおよびエンコーダモードでの比較イベントによるパルスの生成	1468
図 396.	同時トリガの場合の拡張されたパルス幅	1469
図 397.	エンコーダインタフェースモードにおけるカウンタの動作例	1471
図 398.	tim_ti1fp1 の極性を反転したエンコーダインタフェースモードの例	1471
図 399.	直交エンコーダのカウントモード	1472

図 400.	方向プラスクロックエンコーダモード	1473
図 401.	方向クロックエンコーダモード (CC1P = CC2P = 0)	1473
図 402.	方向クロックエンコーダモード (CC1P = CC2P = 1)	1474
図 403.	インデックスゲーティングオプション	1475
図 404.	ジッタのあるインデックス信号	1475
図 405.	IPOS[1:0] = 11 の場合のインデックス生成	1476
図 406.	インデックスがチャネル A でゲート処理された (IPOS[1:0] = 11) 場合のカウンタの読出し	1476
図 407.	インデックスのゲート処理なしの (IPOS[1:0] = 00) 場合のカウンタの読出し	1477
図 408.	インデックスがチャネル A および B でゲート処理された場合のカウンタの読出し	1477
図 409.	狭いインデックスパルスの場合のエンコーダモードの動作 (IPOS[1:0] = 11)	1478
図 410.	狭いインデックスパルスによるカウンタリセット (拡大図、ARR = 0x07)	1479
図 411.	x1 および x2 モードでのインデックスの動作 (IPOS[1:0] = 01)	1480
図 412.	インデックスの方向検知	1480
図 413.	FIDX ビット設定によって変わるカウンタリセット	1481
図 414.	インデックスのブランキング	1482
図 415.	クロックプラス方向モードでのインデックスの動作 (IPOS[0] = 1)	1482
図 416.	方向クロックモードでのインデックスの動作 (IPOS[0] = 1)	1483
図 417.	直交エンコードされた信号の状態図	1483
図 418.	アップカウントでのエンコーダエラー検出	1484
図 419.	ダウンカウントでのエンコーダエラー検出	1485
図 420.	更新によって転送されるプリロードを使ったエンコーダのモード変更 (SMSPS = 0)	1486
図 421.	3 つの信号上のエッジ間の時間間隔の測定	1487
図 422.	ホールセンサインタフェースの例	1489
図 423.	リセットモードの制御回路	1490
図 424.	ゲートモードの制御回路	1491
図 425.	トリガモードの制御回路	1492
図 426.	外部クロックモード 2 + トリガモードの制御回路	1493
図 427.	汎用タイマのブロック図	1552
図 428.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	1557
図 429.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	1557
図 430.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1558
図 431.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1559
図 432.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1559
図 433.	内部クロック分周比が N の場合のカウンタのタイミング図	1560
図 434.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	1560
図 435.	ARPE=1 (TIMx_ARR はプリロードされる) のときの更新イベント時のカウンタの タイミング図	1561
図 436.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1562
図 437.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1562
図 438.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1563
図 439.	内部クロック分周比が N の場合のカウンタのタイミング図	1563
図 440.	更新イベント時のカウンタのタイミング図	1564
図 441.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	1565
図 442.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1566
図 443.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	1566
図 444.	内部クロック分周比が N の場合のカウンタのタイミング図	1567
図 445.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図	1567
図 446.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	1568
図 447.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	1569
図 448.	tim_t12 外部クロックの接続例	1569
図 449.	外部クロックモード 1 の制御回路	1570

図 450.	外部トリガ入力ブロック	1571
図 451.	外部クロックモード 2 の制御回路	1572
図 452.	キャプチャ／比較チャンネル (例: チャンネル 1 入力ステージ)	1572
図 453.	キャプチャ／比較チャンネル 1 メイン回路	1573
図 454.	キャプチャ／比較チャンネル (チャンネル 1、同じくチャンネル 2、3、および 4) の出力ステージ	1573
図 455.	PWM 入力モードタイミング	1576
図 456.	出力比較モード、tim_oc1 のトグル	1577
図 457.	エッジアライン PWM 波形 (ARR=8)	1579
図 458.	センターアライン PWM 波形 (ARR=8)	1580
図 459.	ディザリング原理	1581
図 460.	ディザリングモードでのデータフォーマットとレジスタのコーディング	1582
図 461.	PWM 分解能 対 周波数 (16 ビットモード)	1583
図 462.	PWM 分解能 対 周波数 (32 ビットモード)	1583
図 463.	PWM ディザリングのパターン	1584
図 464.	センターアライン PWM モードでのデューティサイクルへのディザリングの影響	1585
図 465.	50% デューティサイクルの 2 位相シフトされた PWM 信号の生成	1587
図 466.	チャンネル 1 および 3 における組み合わせ PWM モード	1588
図 467.	OCREF_CLR 入力選択マルチプレクサ	1589
図 468.	TIMx tim_ocxref のクリア	1589
図 469.	ワンパルスモードの例	1590
図 470.	再トリガ可能なワンパルスモード	1592
図 471.	パルス発生回路	1592
図 472.	エッジアラインおよびエンコーダモードでの比較イベントによるパルスの生成	1593
図 473.	同時トリガの場合の拡張されたパルス幅	1594
図 474.	エンコーダインタフェースモードにおけるカウンタの動作例	1596
図 475.	tim_ti1fp1 の極性を反転したエンコーダインタフェースモードの例	1596
図 476.	直交エンコーダのカウントモード	1597
図 477.	方向プラスクロックエンコーダモード	1598
図 478.	方向クロックエンコーダモード (CC1P = CC2P = 0)	1598
図 479.	方向クロックエンコーダモード (CC1P = CC2P = 1)	1599
図 480.	インデックスゲーティングオプション	1600
図 481.	ジッタのあるインデックス信号	1600
図 482.	IPOS[1:0] = 11 の場合のインデックス生成	1601
図 483.	チャンネル A でゲート処理された (IPOS[1:0] = 11) インデックスによるカウンタの読み値	1601
図 484.	ゲート処理なしの (IPOS[1:0] = 00) インデックスによるカウンタの読み値	1602
図 485.	チャンネル A および B でゲート処理されたインデックスによるカウンタの読み値	1602
図 486.	狭いインデックスパルスの場合のエンコーダモードの動作 (IPOS[1:0] = 11)	1603
図 487.	狭いインデックスパルスによるカウンタリセット (拡大図、ARR = 0x07)	1604
図 488.	x1 および x2 モードでのインデックスの動作 (IPOS[1:0] = 01)	1605
図 489.	インデックスの方向感受性	1605
図 490.	FIDX ビット設定によって変わるカウンタリセット	1606
図 491.	インデックスのブランキング	1607
図 492.	クロックプラス方向モードでのインデックスの動作 (IPOS[0] = 1)	1607
図 493.	方向クロックモードでのインデックスの動作 (IPOS[0] = 1)	1608
図 494.	直交エンコードされた信号の状態図	1608
図 495.	アップカウントでのエンコーダエラー検出	1609
図 496.	ダウンカウントでのエンコーダエラー検出	1610
図 497.	更新によって転送されるプリロードを使ったエンコーダのモード変更 (SMSPS = 0)	1611
図 498.	リセットモードの制御回路	1613
図 499.	ゲートモードの制御回路	1614
図 500.	トリガモードの制御回路	1615
図 501.	外部クロックモード 2 + トリガモードの制御回路	1616

図 502.	マスタ／スレーブタイマの例	1617
図 503.	チャンネルが 1 つしかないタイマーによるマスタ／スレーブ接続の例	1617
図 504.	TIM_mstr の tim_oc1ref による TIM_slv のゲート操作	1618
図 505.	TIM_mstr の有効化による TIM_slv のゲート操作	1619
図 506.	TIM_mstr の更新による TIM_slv のトリガ	1620
図 507.	TIM_mstr の有効化による TIM_slv のトリガ	1620
図 508.	TIM_mstr の tim_ti1 入力による TIM_mstr と TIM_slv のトリガ	1621
図 509.	基本タイマブロック図	1663
図 510.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	1664
図 511.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	1665
図 512.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	1666
図 513.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1667
図 514.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1667
図 515.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1668
図 516.	内部クロック分周比が N の場合のカウンタのタイミング図	1668
図 517.	ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	1669
図 518.	ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタの タイミング図	1670
図 519.	ディザリング原理	1671
図 520.	ディザリングモードでのデータフォーマットとレジスタのコーディング	1671
図 521.	FCnt 分解能 対 周波数	1672
図 522.	PWM ディザリングのパターン	1672
図 523.	汎用タイマのブロック図 (TIM12)	1684
図 524.	汎用タイマのブロック図 (TIM13/TIM14)	1685
図 525.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	1688
図 526.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	1688
図 527.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1689
図 528.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1690
図 529.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1690
図 530.	内部クロック分周比が N の場合のカウンタのタイミング図	1691
図 531.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	1691
図 532.	ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタの タイミング図	1692
図 533.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	1693
図 534.	tim_ti2 外部クロックの接続例	1693
図 535.	外部クロックモード 1 の制御回路	1694
図 536.	キャプチャ／比較チャンネル 1 入力ステージ (TIM13/TIM14)	1695
図 537.	キャプチャ／比較チャンネル 1 入力ステージ (TIM12)	1695
図 538.	キャプチャ／比較チャンネル 1 メイン回路	1696
図 539.	キャプチャ／比較チャンネル 1 出力ステージ	1696
図 540.	PWM 入力モードタイミング	1699
図 541.	出力比較モード、tim_oc1 のトグル	1700
図 542.	エッジアライン PWM 波形 (ARR=8)	1702
図 543.	ディザリング原理	1703
図 544.	ディザリングモードでのデータフォーマットとレジスタのコーディング	1703
図 545.	PWM 分解能 対 周波数	1704
図 546.	PWM ディザリングのパターン	1705
図 547.	チャンネル 1 および 2 における組み合わせ PWM モード	1707
図 548.	ワンパルスモードの例	1708
図 549.	再トリガ可能なワンパルスモード	1709

図 550.	2 つの信号上のエッジ間の時間間隔の測定	1710
図 551.	リセットモードの制御回路	1711
図 552.	ゲートモードの制御回路	1712
図 553.	トリガモードの制御回路	1712
図 554.	TIM15 ブロック図	1747
図 555.	TIM16/TIM17 ブロック図	1748
図 556.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	1752
図 557.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	1753
図 558.	内部クロック分周比が 1 の場合のカウンタのタイミング図	1754
図 559.	内部クロック分周比が 2 の場合のカウンタのタイミング図	1755
図 560.	内部クロック分周比が 4 の場合のカウンタのタイミング図	1755
図 561.	内部クロック分周比が N の場合のカウンタのタイミング図	1756
図 562.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	1756
図 563.	ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタの タイミング図	1757
図 564.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	1758
図 565.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	1759
図 566.	tim_ti2 外部クロックの接続例	1759
図 567.	外部クロックモード 1 の制御回路	1760
図 568.	キャプチャ/比較チャネル (例: チャネル 1 入力ステージ)	1761
図 569.	キャプチャ/比較チャネル 1 メイン回路	1761
図 570.	キャプチャ/比較チャネル (チャネル 1) の出力ステージ	1762
図 571.	キャプチャ/比較チャネル (TIM15 の場合、チャネル 2) の出力ステージ	1762
図 572.	PWM 入力モードタイミング	1765
図 573.	出力比較モード、tim_oc1 のトグル	1767
図 574.	エッジアライン PWM 波形 (ARR=8)	1768
図 575.	ディザリング原理	1769
図 576.	ディザリングモードでのデータフォーマットとレジスタのコーディング	1769
図 577.	PWM 分解能 対 周波数	1770
図 578.	PWM ディザリングのパターン	1771
図 579.	チャネル 1 および 2 における組み合わせ PWM モード	1773
図 580.	対称のデッドタイム挿入のある相補出力。	1774
図 581.	非対称のデッドタイム	1775
図 582.	負のパルスより長い遅延があるときのデッドタイムの波形	1775
図 583.	正のパルスより長い遅延があるときのデッドタイムの波形	1775
図 584.	ブレーク回路の概要	1777
図 585.	tim_brk でのブレークイベントに対する出力の動作	1779
図 586.	出力先変更	1781
図 587.	tim_ocref_clr 入力選択マルチプレクサ	1782
図 588.	6 ステップ生成 COM の例 (OSSR=1)	1783
図 589.	ワンパルスモードの例	1784
図 590.	再トリガ可能なワンパルスモード	1786
図 591.	2 つの信号上のエッジ間の時間間隔の測定	1786
図 592.	リセットモードの制御回路	1787
図 593.	ゲートモードの制御回路	1788
図 594.	トリガモードの制御回路	1789
図 595.	LPTIM1/2/3/5/6 タイマのブロック図 (1)	1851
図 596.	LPTIM4 タイマのブロック図 (1)	1852
図 597.	グリッチフィルタのタイミング図	1856
図 598.	LPTIM 出力波形、シングルカウントモードの設定 (繰り返しレジスタの内容が 0 ではない場合 (かつ PRELOAD = 1))	1858

図 599.	LPTIM 出力波形、シングルカウントモードの設定、セットワンスモードのアクティブ化 (WAVE ビットがセットされている場合)	1858
図 600.	LPTIM 出力波形、連続カウントモードの設定	1859
図 601.	波形生成	1860
図 602.	エンコーダモードのカウントシーケンス	1863
図 603.	繰り返しレジスタ LPTIM_RCR が 0 ではない場合の連続カウントモード (PRELOAD = 1)	1864
図 604.	キャプチャ/比較入力ステージ (チャネル 1)	1865
図 605.	キャプチャ/比較出力ステージ (チャネル 1)	1865
図 606.	エッジアライン PWM モード (PRELOAD = 1)	1868
図 607.	エッジアライン PWM 波形 (ARR = 8、CCxP = 0)	1868
図 608.	即時更新とプリロード更新の PWM モード	1869
図 609.	独立型ウォッチドッグのブロック図	1900
図 610.	タイムアウトによるリセットのタイミング	1902
図 611.	許可されていない領域でのリフレッシュによるリセットのタイミング	1903
図 612.	ウィンドウコンパレータの更新例	1905
図 613.	独立型ウォッチドッグ割込みのタイミング図	1907
図 614.	早期ウェイクアップ・コンパレータの更新例	1908
図 615.	ウォッチドッグのブロック図	1917
図 616.	ウィンドウ型ウォッチドッグのタイミング図	1919
図 617.	RTC ブロック図	1925
図 618.	TAMP のブロック図	1979
図 619.	バックアップレジスタ保護ゾーン	1983
図 620.	アクティブタンパフィルタ	1990
図 621.	I ² C ブロック図	2027
図 622.	I ² C バスプロトコル	2029
図 623.	セットアップおよびホールドタイミング	2031
図 624.	I ² C 初期化フロー	2033
図 625.	データ受信	2034
図 626.	データ送信	2035
図 627.	スレーブ初期化フロー	2038
図 628.	I ² C スレーブトランスミッタの転送シーケンスフロー (NOSTRETCH = 0)	2040
図 629.	I ² C スレーブトランスミッタの転送シーケンスフロー (NOSTRETCH = 1)	2041
図 630.	I ² C スレーブトランスミッタの転送バス図 (必須イベントのみ)	2042
図 631.	スレーブレシーバの転送シーケンスフロー (NOSTRETCH = 0)	2043
図 632.	スレーブレシーバの転送シーケンスフロー (NOSTRETCH = 1)	2044
図 633.	I ² C スレーブレシーバの転送バス図 (必須イベントのみ)	2044
図 634.	マスタクロック生成	2046
図 635.	マスタ初期化フロー	2048
図 636.	HEAD10R = 0 のときの 10 ビットアドレス読出しアクセス	2048
図 637.	HEAD10R = 1 のときの 10 ビットアドレス読出しアクセス	2049
図 638.	N ≤ 255 バイトの場合の I ² C マスタトランスミッタの転送シーケンスフロー	2050
図 639.	N > 255 バイトの場合の I ² C マスタトランスミッタの転送シーケンスフロー	2051
図 640.	I ² C マスタトランスミッタの転送バス図 (必須イベントのみ)	2052
図 641.	N ≤ 255 バイトの場合の I ² C マスタレシーバの転送シーケンスフロー	2054
図 642.	N > 255 バイトの場合の I ² C マスタレシーバの転送シーケンスフロー	2055
図 643.	I ² C マスタレシーバの転送バス図 (必須イベントのみ)	2056
図 644.	t _{LOW:SEXT} 、t _{LOW:MEXT} のタイムアウト間隔	2060
図 645.	N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフロー	2063
図 646.	SMBus スレーブトランスミッタの転送バス図 (SBC = 1)	2064
図 647.	N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフロー	2065
図 648.	SMBus スレーブレシーバのバス転送図 (SBC = 1)	2066
図 649.	SMBus マスタトランスミッタのバス転送図	2067

図 650.	SMBus マスタレシーバのバス転送図	2069
図 651.	I3C ブロック図	2094
図 652.	I3C (1 次) コントローラの状態とプログラミングシーケンス図	2098
図 653.	I3C ターゲットの状態とプログラミングシーケンス図	2103
図 654.	I3C CCC メッセージ (コントローラの場合)	2116
図 655.	I3C ブロードキャスト ENTDAACCC (コントローラの場合)	2117
図 656.	I3C ブロードキャスト、直接読出しおよび直接書込み RSTACT CCC (コントローラの場合)	2118
図 657.	I3C CCC メッセージ (ターゲットの場合)	2120
図 658.	I3C ブロードキャスト ENTDAACCC (ターゲットの場合)	2121
図 659.	I3C ブロードキャスト DEFTGTS CCC (ターゲットの場合)	2122
図 660.	I3C ブロードキャスト DEFGRPA CCC (ターゲットの場合)	2123
図 661.	I3C プライベート読出し / 書込みメッセージ (コントローラの場合)	2125
図 662.	I3C プライベート読出し / 書込みメッセージ (ターゲットの場合)	2126
図 663.	従来の I2C 読出し / 書込みメッセージ (コントローラの場合)	2127
図 664.	IBI 転送 (コントローラ / ターゲットの場合)	2129
図 665.	ホットジョイン・リクエスト転送 (コントローラ / ターゲットの場合)	2130
図 666.	コントローラ機能リクエスト転送 (コントローラ / ターゲットの場合)	2131
図 667.	C-FIFO 管理 (コントローラの場合)	2132
図 668.	TX-FIFO 管理 (コントローラの場合)	2134
図 669.	RX-FIFO 管理 (コントローラの場合)	2136
図 670.	S-FIFO 管理 (コントローラの場合)	2139
図 671.	RX-FIFO 管理 (ターゲットの場合)	2140
図 672.	I3C_TGTTDR による TX-FIFO 管理 (ターゲットの場合)	2142
図 673.	TX-FIFO サイズよりバイト数が少ない読出しの I3C_TGTTDR を使用しない ソフトウェアによる TX-FIFO 管理 (ターゲットの場合)	2144
図 674.	USART のブロック図	2202
図 675.	ワード長のプログラミング	2206
図 676.	設定可能なストップビット	2208
図 677.	送信時の TC/TXE の動作	2210
図 678.	16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出	2211
図 679.	usart_ker_ck クロック分周回路のブロック図	2214
図 680.	データサンプリング (16 倍のオーバーサンプリング)	2216
図 681.	データサンプリング (8 倍のオーバーサンプリング)	2216
図 682.	アイドルライン検出を使用したミュートモード	2222
図 683.	アドレスマーク検出を使用したミュートモード	2223
図 684.	LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1)	2226
図 685.	LIN モードでのブレーク検出とフレーミングエラー検出	2227
図 686.	USART の同期マスタ送信の例	2228
図 687.	同期マスタモードでの USART データクロックタイミング図 (M ビット = 00)	2228
図 688.	同期マスタモードでの USART データクロックタイミング図 (M ビット = 01)	2229
図 689.	同期スレーブモードでの USART データクロックタイミング図 (M ビット = 00)	2230
図 690.	ISO 7816-3 非同期プロトコル	2232
図 691.	ストップビット 1.5 個を使用したパリティエラー検出	2234
図 692.	IrDA SIR ENDEC ブロック図	2237
図 693.	IrDA データ変調 (3/16) - 通常モード	2238
図 694.	DMA を使用した送信	2239
図 695.	DMA を使用した受信	2240
図 696.	2 つの USART 間のハードウェアフロー制御	2241
図 697.	RS232 RTS フロー制御	2241
図 698.	RS232 CTS フロー制御	2242
図 699.	確認されたウェイクアップイベント (ウェイクアップイベント = アドレス一致、FIFO 無効)	2245

図 700.	確認されていないウェイクアップイベント（ウェイクアップイベント = アドレス一致、FIFO 無効）	2245
図 701.	LPUART ブロック図	2288
図 702.	LPUART ワード長のプログラミング	2291
図 703.	設定可能なストップビット	2293
図 704.	送信時の TC/TXE の動作	2295
図 705.	lpuart_ker_ck クロック分周回路のブロック図	2298
図 706.	アイドルライン検出を使用したミュートモード	2303
図 707.	アドレスマーク検出を使用したミュートモード	2304
図 708.	DMA を使用した送信	2306
図 709.	DMA を使用した受信	2307
図 710.	2 つの LPUART 間のハードウェアフロー制御	2308
図 711.	RS232 RTS フロー制御	2308
図 712.	RS232 CTS フロー制御	2309
図 713.	確認されたウェイクアップイベント（ウェイクアップイベント = アドレス一致、FIFO 無効）	2312
図 714.	確認されていないウェイクアップイベント（ウェイクアップイベント = アドレス一致、FIFO 無効）	2312
図 715.	SPI/I2S ブロック図	2342
図 716.	全二重シングルマスタ / シングルスレーブアプリケーション	2345
図 717.	半二重シングルマスタ / シングルスレーブアプリケーション	2346
図 718.	単方向シングルマスタ / シングルスレーブアプリケーション（送信専用モードのマスタ / 受信専用モードのスレーブ）	2347
図 719.	スター接続形態で接続されたマスタと 3 つの独立したスレーブ	2348
図 720.	サーキュラ（デジチチェーン）接続形態で接続されたマスタと 3 つのスレーブ	2349
図 721.	マルチマスタアプリケーション	2350
図 722.	SS 制御ロジックの仕組み	2352
図 723.	データフロータイミング制御（SSOE = 1、SSOM = 0、SSM = 0）	2352
図 724.	SS によるデータ間パルスのインターリーブ（SSOE = 1、SSOM = 1、SSM = 0）	2353
図 725.	データクロックのタイミング図	2355
図 726.	データサイズが 8、16、または 32 ビットと等しくない場合のデータ配置	2356
図 727.	全機能がセットされたインスタンスで、FIFO での送受信のデータのパッキング	2365
図 728.	TI モードでの転送	2367
図 729.	スレーブ検出中のアンダーラン状態に関するオプションの設定	2369
図 730.	波形の例	2377
図 731.	マスタ I ² S フィリップスプロトコルの波形（16/32-bit フル精度）	2378
図 732.	フィリップス I ² S 規格の波形	2378
図 733.	マスタ MSB 詰め 16-bit または 32-bit 長フル精度	2379
図 734.	マスタ MSB 詰め 16-bit または 24-bit データ長	2379
図 735.	スレーブ MSB 詰め	2380
図 736.	LSB 詰め 16-bit または 24-bit データ長	2380
図 737.	フレーム長がデータ長と等しいときのマスタ PCM	2381
図 738.	マスタ PCM 規格の波形（16-bit または 24-bit データ長）	2382
図 739.	スレーブ PCM 波形	2383
図 740.	フィリップス I2S 規格、マスタの起動シーケンス	2386
図 741.	フィリップス I ² S 規格、スレーブの起動シーケンス	2387
図 742.	フィリップス I ² S 規格、マスタの停止シーケンス	2387
図 743.	I ² S クロックジェネレータのアーキテクチャ	2388
図 744.	データフォーマット	2390
図 745.	アンダーラン状況の取り扱い	2392
図 746.	オーバーラン状況の取り扱い	2393
図 747.	FIXCH=0 でのフレームエラー検出	2394
図 748.	FIXCH=1 でのフレームエラー検出	2394

図 749.	SAI 機能ブロック図	2419
図 750.	オーディオフレーム	2423
図 751.	FS の機能は、フレーム開始 + チャネルサイド識別 (FSDEF = TRIS = 1)	2425
図 752.	FS の機能はフレーム開始 (FSDEF = 0)	2426
図 753.	スロットサイズ設定 (SAI_xSLOTR で FBOFF = 0 の場合)	2427
図 754.	先頭ビットオフセット	2427
図 755.	オーディオブロッククロックジェネレータの概要	2429
図 756.	PDM 標準接続とタイミング	2433
図 757.	PDM インタフェースの詳細ブロック図	2434
図 758.	起動シーケンス	2435
図 759.	TDM の SAI_ADR フォーマット (32 ビットスロット幅)	2436
図 760.	TDM の SAI_ADR フォーマット (16 ビットスロット幅)	2437
図 761.	TDM の SAI_ADR フォーマット (8 ビットスロット幅)	2438
図 762.	AC'97 オーディオフレーム	2441
図 763.	2 つ以上の内蔵 SAI を搭載したデバイス (3 個の外部 AC'97 デコーダ) の AC'97 設定の代表例	2442
図 764.	SPDIF フォーマット	2443
図 765.	SAI_xDR レジスタの順序	2444
図 766.	SAI オーディオブロックのデータコンパニングハードウェア	2447
図 767.	非アクティブスロットにおける SD 出力ラインのトライステート化	2448
図 768.	I2S のようなプロトコルにおける出力データラインのトライステート化	2449
図 769.	オーバーラン検出エラー	2450
図 770.	FIFO アンダーランイベント	2450
図 771.	CAN サブシステム	2486
図 772.	FDCAN ブロック図	2488
図 773.	ビットタイミング	2489
図 774.	トランシーバ遅延測定	2494
図 775.	バスモニタモードでのピン制御	2495
図 776.	ループバックモードでのピン制御	2497
図 777.	メッセージ RAM 設定	2498
図 778.	標準メッセージ ID フィルタパス	2501
図 779.	拡張メッセージ ID フィルタパス	2502
図 780.	USB ペリフェラルのブロック図	2549
図 781.	パケットバッファ領域とバッファ記述テーブルの位置の例	2555
図 782.	UCPD ブロック図	2596
図 783.	クロック分周とタイミング要素	2598
図 784.	K コード送信	2600
図 785.	さまざまなデータサイズの送信順	2602
図 786.	パケットフォーマット	2602
図 787.	Hard Reset のラインフォーマット	2603
図 788.	Cable Reset のラインフォーマット	2603
図 789.	BIST テストデータフレーム	2604
図 790.	BIST キャリアモード 2 フレーム	2605
図 791.	UCPD BMC トランスミッターアーキテクチャ	2605
図 792.	UCPD BMC レシーバのアーキテクチャ	2607
図 793.	Ethernet ハイレベルブロック図	2639
図 794.	DMA 送信フロー (標準モード)	2642
図 795.	DMA 送信フロー (OSP モード)	2644
図 796.	受信 DMA フロー	2646
図 797.	MAC 送信フローの概要	2650
図 798.	MAC 受信フロー	2652
図 799.	パケットフィルタシーケンス	2656

図 800.	ネットワーク時間の同期	2665
図 801.	ピアツーピアパス補正をサポートするクロック単位での伝搬遅延の計算	2666
図 802.	密補正法を使用したシステム時間の更新	2675
図 803.	TCP セグメンテーションオフロードの概要	2691
図 804.	TCP セグメンテーションオフロードのフロー	2692
図 805.	分割されたパケットのヘッダおよびペイロードフィールド	2695
図 806.	サポートされる PHY インタフェース	2705
図 807.	SMA インタフェースブロック	2705
図 808.	MDIO のパケット構造 (Clause 45)	2706
図 809.	MDIO パケットの構造 (Clause 22)	2707
図 810.	SMA 書き込み動作フロー	2708
図 811.	書き込みデータパケット	2709
図 812.	読み出しデータパケット	2709
図 813.	メディア独立インタフェース (MII) 信号	2711
図 814.	RMII ブロック図	2713
図 815.	送信ビットの順序	2714
図 816.	受信ビットの順序	2715
図 817.	LPI 遷移 (送信, 100 Mbps)	2723
図 818.	LPI Tx クロックのゲーティング (LPITCSE = 1 の場合)	2724
図 819.	LPI 遷移 (受信, 100 Mbps)	2725
図 820.	ディスクリプタのリング構造	2746
図 821.	DMA ディスクリプタリング	2747
図 822.	送信ディスクリプタ (読み出しフォーマット)	2748
図 823.	送信ディスクリプタの書き戻しフォーマット	2753
図 824.	送信コンテキストディスクリプタのフォーマット	2757
図 825.	受信通常ディスクリプタ (読み出しフォーマット)	2760
図 826.	受信通常ディスクリプタ (書き戻しフォーマット)	2763
図 827.	受信コンテキストディスクリプタ	2771
図 828.	ETH_DMAISR フラグの生成	2788
図 829.	デバッグサポートインフラストラクチャのブロック図	2905
図 830.	製品のライフサイクル状態とデバッグ認証	2909
図 831.	JTAG TAP ステートマシン	2912
図 832.	CoreSight トポロジ	2934
図 833.	トレースポートインタフェースユニット (TPIU)	3020
図 834.	組み込みクロストリガ	3033

1 このマニュアルにおける表記の規則

1.1 一般情報

STM32H563/H573および STM32H562 デバイスには、Arm^{®(a)} Cortex-M33 コアが搭載されています。



1.2 レジスタに関する略記

レジスタの説明では、次の略記^(b) が使用されます。

読出し／書込み (rw)	このビットは、ソフトウェアによる読出しと書込みが可能です。
読出し専用 (r)	このビットは、ソフトウェアによる読出しのみが可能です。
書込み専用 (w)	このビットは、ソフトウェアによる書込みのみが可能です。このビットを読み出すと、リセット値が返されます。
読出し／クリア書込み 0 (rc_w0)	このビットは、ソフトウェアによって読み出すことができ、“0”を書き込むことによってクリアできます。“1”を書き込んでも、ビットの値は変化しません。
読出し／クリア書込み 1 (rc_w1)	このビットは、ソフトウェアによって読み出すことができ、“1”を書き込むことによってクリアできます。“0”を書き込んでも、ビットの値は変化しません。
読出し／クリア書込み (rc_w)	このビットは、ソフトウェアによって読み出すことができ、レジスタへの書込みによってクリアできます。このビットに書き込まれた値は重要ではありません。
読出し／読出しによるクリア (rc_r)	このビットは、ソフトウェアによって読み出すことができます。このビットを読み出すと、自動的に“0”にクリアされます。このビットを書き込んでも、ビットの値は変化しません。
読出し／読出しによるセット (rs_r)	このビットは、ソフトウェアによって読み出すことができます。このビットを読み出すと、自動的に“1”にセットされます。このビットを書き込んでも、ビットの値は変化しません。
読出し／セット (rs)	このビットは、ソフトウェアによって読出しとセットが可能です。“0”を書き込んでも、ビットの値は変化しません。
読出し／1 回だけ書込み (rwo)	このビットは、ソフトウェアによって 1 回だけ書き込むことができ、いつでも読み出すことができます。1 回リセットするだけで、このビットをそのリセット値に戻すことができます。
反転 (t)	このビットは、ソフトウェアによって“1”を書き込むことで反転できます。0 を書き込んでも、ビットの値は変化しません。
読出し専用書込みトリガ (rt_w1)	このビットは、ソフトウェアによって読み出すことができます。“1”を書き込むと、イベントがトリガされますが、ビットの値は変化しません。
予約済み (Res.)	予約済みビットであり、リセット値に保持する必要があります。

a. Arm は、米国内およびその他の地域にある Arm Limited（またはその子会社）の登録商標です。

b. これは STMicroelectronics マイクロコントローラに適用されるすべての略語の包括的リストですが、本書では使用されていないものもあります。

1.3 用語

このセクションでは、本書で用いられる略語についての定義の概要を掲載しています。

- **ワード** : 32 ビット長のデータ。
- **ハーフワード** : 16 ビット長のデータ。
- **バイト** : 8 ビット長のデータ。
- **AHB** : アドバンストハイパフォーマン斯巴ス。

2 メモリとバスのアーキテクチャ

2.1 システムアーキテクチャ

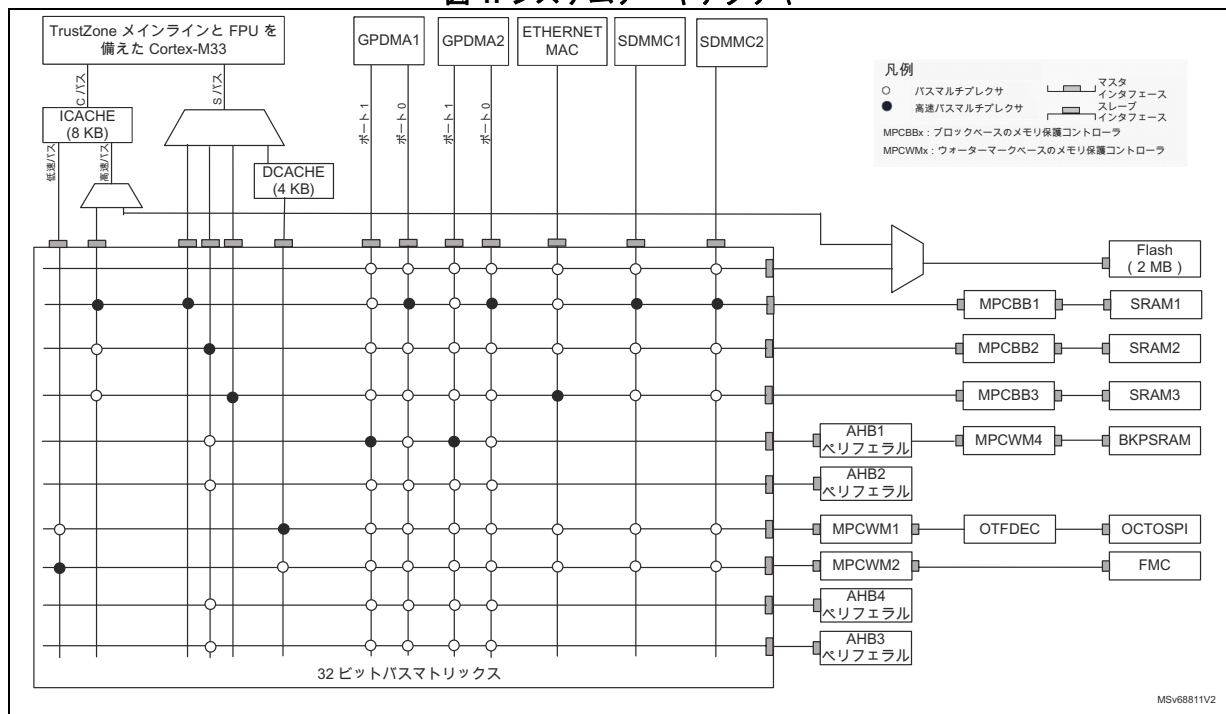
STM32H563/H573およびSTM32H562アーキテクチャは、内蔵Flashメモリに直接アクセスできる命令キャッシュによって実行に最適化されたArm Cortex-M33コアに依存しています。

このアーキテクチャは、以下を相互接続する32ビットマルチレイヤAHBバスマトリクスも備えています。

- 最大13のマスタ：
 - TrustZone[®]メインラインとFPUコアCバスを持つCortex-M33を命令キャッシュ経由で内部SRAMに接続する高速Cバス
 - TrustZoneメインラインとFPUコアCバスを持つCortex-M33を命令キャッシュ経由で外部メモリに接続する低速Cバス
 - TrustZoneメインラインとFPUコアSバスを持つCortex-M33（3つのマスタを3つの内部SRAMに遅延なく接続）
 - TrustZoneメインラインとFPUコアSバスを持つCortex-M33をデータキャッシュ経由で外部メモリに接続
 - GPDMA1（2つのマスタポートを備えた汎用DMA）
 - GPDMA2（2つのマスタポートを備えた汎用DMA）
 - SDMMC1
 - SDMMC2
 - Ethernet MAC
- 最大10のスレーブ：
 - 内部Flashメモリ（2MB）
 - 内部SRAM1（256KB）
 - 内部SRAM2（64KB）
 - 内部SRAM3（320KB）
 - AHB-APBブリッジとAPBペリフェラル（APB1とAPB2に接続）を含むAHB1ペリフェラルおよびバックアップRAM（4KBのBKPSRAM）
 - AHB2ペリフェラル
 - FMC（Flexible memory controller：フレキシブルメモリコントローラ）
 - OCTOSPI
 - AHB-APBブリッジとAPBペリフェラル（APB3に接続）を含むAHB3ペリフェラル
 - AHB4ペリフェラル

複数のハイスピードペリフェラルが同時動作中であっても、バスマトリクスによって、マスタからスレーブへのアクセスが可能になるとともに、同時アクセスや効率的な動作を行うことが可能になっています。このアーキテクチャを次の図に示します。

図 1. システムアーキテクチャ



2.1.1 高速 C バス

このバスは、Cortex-M33 の C バスを命令キャッシュ経由で内部 Flash メモリと BusMatrix に接続します。このバスは、命令フェッチおよびコード領域にマップされた内部メモリへのデータアクセスに使用されます。このバスは、内部 Flash メモリと内部 SRAM (SRAM1、SRAM2、および SRAM3) をターゲットとします。

SRAM1、SRAM2、および SRAM3 は、連続マッピングにより、このバスでアクセス可能です。

2.1.2 低速 C バス

このバスは、Cortex-M33 コアの C バスを命令キャッシュ経由で BusMatrix に接続します。このバスは、命令フェッチおよびコード領域にマップされた外部メモリへのデータアクセスに使用されます。このバスは、外部メモリ (FMC および OCTOSPI) をターゲットとします。

2.1.3 S-バス

このバスは、Cortex-M33 コアのシステムバスを BusMatrix に接続します。コアがこのバスを使用してペリフェラルまたは SRAM 領域内のデータにアクセスします。このバスは、内部 SRAM (SRAM1、SRAM2、SRAM3、および BKPSRAM)、AHB1 ペリフェラル (APB1、APB2、AHB2、AHB3、および AHB4 ペリフェラルを含む) をターゲットとします。

SRAM1、SRAM2、および SRAM3 は、連続マッピングにより、このバスでアクセス可能です。

注: バスマトリクスが SRAM1、SRAM2、および SRAM3 にアクセスするときにはゼロ遅延です。

2.1.4 DCache S バス

このバスは、Cortex-M33 コアの システムバスをデータキャッシュ経由で BusMatrix に接続します。このバスは、命令フェッチおよびデータ領域にマップされた外部メモリへのデータアクセスに使用されます。このバスは、外部メモリ（FMC および OCTOSPI）をターゲットとします。

注： このバス経由で命令をフェッチするのは、低速 C バス経由で命令をフェッチするより非効率的です。

2.1.5 GPDMA1 および GPDMA2 バス

これらのバスは、GPDMA1 および GPDMA2 の 4 つの AHB マスタインタフェースを BusMatrix に接続します。これらのバスは、内部 Flash メモリ、内部 SRAM（SRAM1、SRAM2、SRAM3、および BKPSRAM）、AHB1 ペリフェラル（APB1 および APB2 ペリフェラルを含む）、AHB2 ペリフェラル、AHB3 ペリフェラル、AHB4 ペリフェラル、および外部メモリを FMC または OCTOSPI 経由でターゲットとします。

2.1.6 SDMMC1 および SDMMC2 コントローラ DMA バス

これらのバスは、SDMMC1 および SDMMC2 DMA マスタインタフェースを BusMatrix に接続します。これらのバスは、SDMMC1 および SDMMC2 DMA によってデータをメモリからロードするか、データをメモリに保存するためにのみ使用されます。これらのバスは、データメモリ、つまり、内部 Flash メモリ、内部 SRAM（SRAM1、SRAM2、および SRAM3）、および外部メモリを FMC または OCTOSPI 経由でターゲットとします。

2.1.7 バスマトリックス

このバスマトリックスによって、マスタ間のアクセス調停を管理します。調停には、ラウンド-ロビン方式を使用します。この BusMatrix は、各マスタを特定のスレーブに遅延なく接続するための高速バスマルチプレクサを備えています（[図 1](#) を参照）。同じマスタの場合、他のスレーブでは、新しいアクセスごとに少なくとも 1 サイクルの遅延が発生します。

2.1.8 AHB/APB ブリッジ

3 つの AHB/APB ブリッジが AHB バスと APB バスの間の完全同期接続を提供するため、ペリフェラル周波数の柔軟な選択が可能です。

これらのブリッジに接続されたペリフェラルのアドレスマッピングについては、[セクション 2.3.2: メモリマップとレジスタ境界アドレス](#) を参照してください。

各デバイスのリセット後、すべてのペリフェラルクロックは無効になります（内部 SRAM および Flash メモリインタフェースは除きます）。ペリフェラルを使用する前に、RCC_AHBxENR および RCC_APBxENR レジスタでクロックを有効にする必要があります。

注： APB レジスタに 8 または 16 ビットアクセスが行われるときには、アクセスは 32 ビットアクセスに変換されます。すなわち、ブリッジが 8 または 16 ビットのデータを複製して、32 ビットのベクタを供給します。

2.1.9 Ethernet MAC

Ethernet MAC は、AHB バスマトリックスに接続された 32 ビットバスを使用します。システムバスマトリックスを通じて、内部 Flash メモリ、内部メモリ、および外部メモリに OCTOSPI および FMC 経由でアクセスできます。

2.2 TrustZone セキュリティアーキテクチャ

セキュリティアーキテクチャは、Armv8-M メインライン拡張を備えた Arm TrustZone に基づきます。

TrustZone セキュリティは Flash_OTPR レジスタの TZEN オプションビットで有効化されます。

TrustZone が有効なとき、SAU (security-attribution unit : セキュリティ属性ユニット) と IDAU (implementation-defined-attribution unit : 実装定義属性ユニット) は、セキュア状態と非セキュア状態に基づいてアクセス許可を定義します。

- SAU : セキュリティ属性には、最大 8 つの SAU 設定可能領域が利用できます。
- IDAU : 非セキュアまたは非セキュア呼出し可能属性として、最初のメモリパーティションを提供します。IDAU メモリマップパーティションは設定することができず、ハードウェア実装により固定されます (図 2 : IDAU マッピングに基づくメモリマップを参照)。そして、SAU セキュリティ属性からの結果と組み合わせられ、より高度なセキュリティ状態が選択されます。

IDAU セキュリティ属性に基づいて、Flash メモリ、システム SRAM、およびペリフェラルのメモリ空間には、セキュア状態と非セキュア状態についてエイリアスが 2 回作成されます。ただし、外部のメモリ空間にはエイリアスは作成されません。

2.2.1 デフォルトの TrustZone セキュリティ状態

TrustZone セキュリティが FLASH_OTPR の TZEN オプションビットによって有効になると、デフォルトのシステムセキュリティ状態は以下のとおりです。

- CPU :
 - Cortex-M33 は、リセット後はセキュアな状態です。ブートアドレスは、セキュアなアドレスである必要があります。
- メモリマップ :
 - SAU は、リセット後は完全にセキュアです。そのため、すべてのメモリマップは完全にセキュアです。セキュリティ属性には、最大 8 つの SAU 設定可能領域が利用できます。
- Flash メモリ :
 - Flash メモリのセキュリティ領域は、ウォーターマークユーザオプションによって定義されます。
 - Flash ブロックベースのセキュリティ属性は、リセット後は非セキュアです。
- SRAM :
 - すべての SRAM はリセット後はセキュアです。MPCBBx (block-based memory protection controller : ブロックベースのメモリ保護コントローラ) はセキュアです。
- 外部メモリ :
 - FMC および OCTOSPI バンクは、リセット後はセキュアです。MPCWMx (watermark-based memory protection controller : ウォーターマークベースのメモリ保護コントローラ) はセキュアです。
- ペリフェラル (セキュリティ保護可能なペリフェラルおよび TrustZone 対応ペリフェラルのリストについては、表 1 および 表 2 を参照)
 - セキュリティ保護可能なペリフェラルは、リセット後は非セキュアです。
 - TrustZone を認識するペリフェラルは、リセット後は非セキュアです。それらのセキュア設定レジスタはセキュアです。
- すべての GPIO はリセット後はセキュアです。

- 割込み：
 - NVIC：すべての割込みはリセット後はセキュアです。NVIC はセキュアな状態と非セキュアな状態でバンク切替えを行います。
 - TZIC：すべての不正アクセス割込みはリセット後に無効になります（[GTZC の TrustZone システムアーキテクチャ](#) を参照）。

2.2.2 TrustZone ペリフェラルの分類

TrustZone セキュリティが有効な場合、ペリフェラルは、次のように、セキュリティ保護可能タイプまたは TrustZone 対応タイプのいずれかです。

- セキュリティ保護可能：セキュリティ特性を定義するために、TZSC コントローラから制御される AHB/APB ファイアウォールゲートによって保護されるペリフェラル
- TrustZone 対応：AHB または APB バスに直接接続され、セキュアなレジスタのサブセットなど、特定の TrustZone 動作を実装するペリフェラル。

詳細については、[GTZC の TrustZone システムアーキテクチャ](#)を参照してください。

次の表は、システム内のセキュリティ保護可能ペリフェラルと TrustZone 対応ペリフェラルのリストです。

表 1. TZSC によってセキュリティ保護可能なペリフェラル

バス	ペリフェラル
AHB4	OCTOSPI
	FMC
	SDMMC2
	SDMMC1
AHB2	PKA
	SAES
	RNG
	HASH
	AES
	DCMI
	ADC1 / ADC2
	DAC 1
AHB1	DCACHE レジスタ
	ICACHE レジスタ
	ETHERNET
	RAMCFG
	FMAC
	CORDIC
	CRC

表 1. TZSC によってセキュリティ保護可能なペリフェラル (続き)

バス	ペリフェラル
APB3	VREFBUF
	LPTIM1
	LPTIM3
	LPTIM4
	LPTIM5
	LPTIM6
	I2C4
	I2C3
	LPUART1
	SPI5
APB2	USB FS
	SAI2
	SAI1
	SPI6
	SPI4
	TIM17
	TIM16
	TIM15
	USART1
	TIM8
	SPI1
	TIM1

表 1. TZSC によってセキュリティ保護可能なペリフェラル (続き)

バス	ペリフェラル
APB1	UCPD
	FDCAN2
	FDCAN1
	LPTIM2
	DTS
	UART12
	UART9
	UART8
	UART7
	HDMI-CEC
	USART11
	USART10
	USART6
	CRS
	I3C1
	I2C2
	I2C1
	UART5
	UART4
	USART3
	USART2
	SPI3/I2S3
	SPI2/I2S2
	IWDG
	WWDG
	TIM14
	TIM13
	TIM12
	TIM7
	TIM6
	TIM5
	TIM4
	TIM3
	TIM2

表 2. TrustZone 対応ペリフェラル

バス	ペリフェラル
AHB3	EXTI
	RCC
	PWR
AHB4	OTFDEC1
AHB2	GPIOI
	GPIOH
	GPIOG
	GPIOF
	GPIOE
	GPIOD
	GPIOC
	GPIOB
	GPIOA
AHB1	GTZC
	DCACHE
	ICACHE
	FLASH
	GPDMA-2
	GPDMA-1
APB3	TAMP
	RTC
	SBS

2.3 メモリ構成

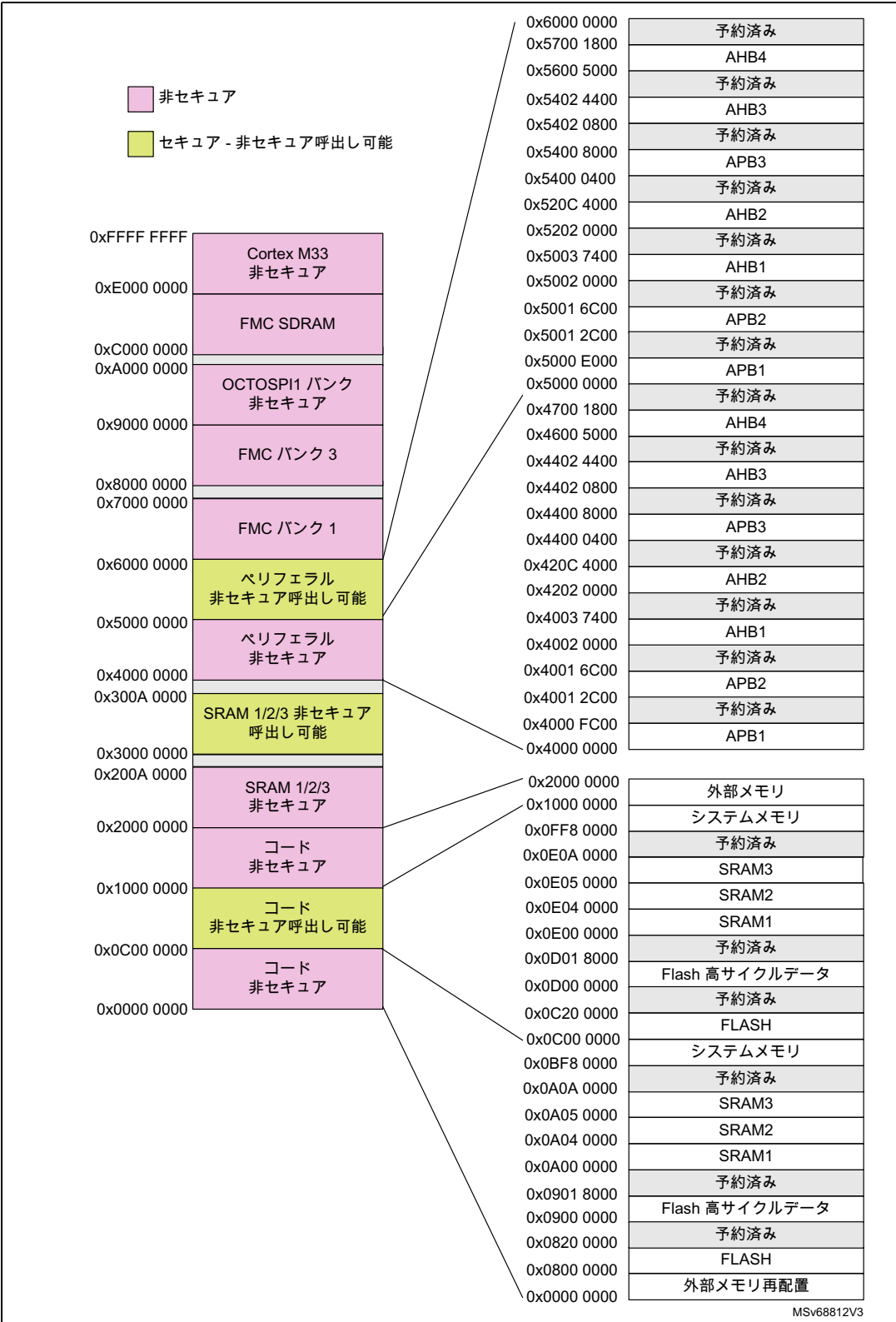
2.3.1 概要

プログラムメモリ、データメモリ、レジスタ、および I/O ポートは、同じ 4 GB のリニアなアドレス空間に配置されています。

バイトは、メモリ内でリトルエンディアン形式でコード化されます。ワード内で最も小さな番号のバイトがワードの最下位バイトとみなされ、最も大きな番号のバイトが最上位バイトとみなされます。

2.3.2 メモリマップとレジスタ境界アドレス

図 2. IDAU マッピングに基づくメモリマップ



MSv68812V3

オンチップメモリとペリフェラルに割り当てられていないメモリマップ領域はすべて、「予約済み」とみなされます。次の表に、デバイスで使用可能なペリフェラルの境界アドレスを示します。

表 3. メモリマップとペリフェラルレジスタ境界アドレス

バス	セキュアな境界アドレス	非セキュアな境界アドレス	ペリフェラル	ペリフェラルレジスタマップ
AHB4	0x5700 1800 - 0x57FF FFFF	0x4700 1800 - 0x47FF FFFF	予約済み	-
	0x5700 1400 - 0x5700 17FF	0x4700 1400 - 0x4700 17FF	OCTOSPI1	OCTOSPI レジスタマップ
	0x5700 0800 - 0x5700 13FF	0x4700 0800 - 0x4700 13FF	予約済み	-
	0x5700 0400 - 0x5700 07FF	0x4700 0400 - 0x4700 07FF	FMC	FMC レジスタマップ
	0x5600 F800 - 0x5700 03FF	0x4600 F800 - 0x4700 03FF	予約済み	-
	0x5600 F000 - 0x5600 F3FF	0x4600 F000 - 0x4600 F3FF	DLYBOS1	DLYB レジスタマップ
	0x5600 9000 - 0x5600 EFFF	0x4600 9000 - 0x4600 EFFF	予約済み	-
	0x5600 8C00 - 0x5600 8FFF	0x4600 8C00 - 0x4600 8FFF	SDMMC2	SDMMC レジスタマップ
	0x5600 8800 - 0x5600 8BFF	0x4600 8800 - 0x4600 8BFF	DLYBSD2	DLYB レジスタマップ
	0x5600 8400 - 0x5600 87FF	0x4600 8400 - 0x4600 87FF	DLYBSD1	
	0x5600 8000 - 0x5600 83FF	0x4600 8000 - 0x4600 83FF	SDMMC1	SDMMC レジスタマップ
	0x5600 5400 - 0x5600 7FFF	0x4600 5400 - 0x4600 7FFF	予約済み	-
	0x5600 5000 - 0x5600 53FF	0x4600 5000 - 0x4600 53FF	OTFDEC1	OTFDEC レジスタマップ
	0x5600 0000 - 0x5600 4FFF	0x4600 0000 - 0x4600 4FFF	予約済み	-
AHB3	0x5402 4400 - 0x54FF FFFF	0x4402 4400 - 0x44FF FFFF	予約済み	-
	0x5402 4000 - 0x5402 43FF	0x4402 4000 - 0x4402 43FF	デバッグ	DEBUG レジスタマップ
	0x5402 3000 - 0x5402 3FFF	0x4402 3000 - 0x4402 3FFF	予約済み	-
	0x5402 2000 - 0x5402 23FF	0x4402 2000 - 0x4402 23FF	EXTI	EXTI レジスタマップ
	0x5402 1000 - 0x5402 1FFF	0x4402 1000 - 0x4402 1FFF	予約済み	-
	0x5402 0C00 - 0x5402 0FFF	0x4402 0C00 - 0x4402 0FFF	RCC	RCC レジスタマップ
	0x5402 0800 - 0x5402 0BFF	0x4402 0800 - 0x4402 0BFF	PWR	PWR レジスタマップ
	0x5402 0000 - 0x5402 07FF	0x4402 0000 - 0x4402 07FF	予約済み	-
APB3	0x5400 8000 - 0x5400 FFFF	0x4400 8000 - 0x4400 FFFF	予約済み	-
	0x5400 7C00 - 0x5400 7FFF	0x4400 7C00 - 0x4400 7FFF	TAMP	TAMP レジスタマップ
	0x5400 7800 - 0x5400 7BFF	0x4400 7800 - 0x4400 7BFF	RTC	RTC レジスタマップ
	0x5400 7400 - 0x5400 77FF	0x4400 7400 - 0x4400 77FF	VREFBUF	VREFBUF レジスタマップ
	0x5400 5800 - 0x5400 73FF	0x4400 5800 - 0x4400 73FF	予約済み	-

表 3. メモリマップとペリフェラルレジスタ境界アドレス (続き)

バス	セキュアな境界アドレス	非セキュアな境界アドレス	ペリフェラル	ペリフェラルレジスタマップ
APB3 (続き)	0x5400 5400 - 0x5400 57FF	0x4400 5400 - 0x4400 57FF	LPTIM6	LPTIM レジスタマップ
	0x5400 5000 - 0x5400 53FF	0x4400 5000 - 0x4400 53FF	LPTIM5	
	0x5400 4C00 - 0x5400 4FFF	0x4400 4C00 - 0x4400 4FFF	LPTIM4	
	0x5400 4800 - 0x5400 4BFF	0x4400 4800 - 0x4400 4BFF	LPTIM3	
	0x5400 4400 - 0x5400 47FF	0x4400 4400 - 0x4400 47FF	LPTIM1	
	0x5400 3000 - 0x5400 43FF	0x4400 3000 - 0x4400 43FF	予約済み	-
	0x5400 2C00 - 0x5400 2FFF	0x4400 2C00 - 0x4400 2FFF	I2C4	I2C レジスタマップ
	0x5400 2800 - 0x5400 2BFF	0x4400 2800 - 0x4400 2BFF	I2C3	
	0x5400 2400 - 0x5400 27FF	0x4400 2400 - 0x4400 27FF	LPUART1	LPUART レジスタマップ
	0x5400 2000 - 0x5400 23FF	0x4400 2000 - 0x4400 23FF	SPI5	SPI レジスタマップ
	0x5400 0800 - 0x5400 1FFF	0x4400 0800 - 0x4400 1FFF	予約済み	-
	0x5400 0400 - 0x5400 07FF	0x4400 0400 - 0x4400 07FF	SBS	SBS レジスタマップ
	0x5400 0000 - 0x5400 03FF	0x4400 0000 - 0x4400 03FF	予約済み	-
AHB2	0x520C 4000 - 0x53FF FFFF	0x420C 4000 - 0x43FF FFFF	予約済み	-
	0x520C 2000 - 0x520C 3FFF	0x420C 2000 - 0x420C 3FFF	PKA +RAM	PKA レジスタマップと リセット値
	0x520C 1000 - 0x520C 1FFF	0x420C 1000 - 0x420C 1FFF	予約済み	-
	0x520C 0C00 - 0x520C 0FFF	0x420C 0C00 - 0x420C 0FFF	SAES	SAES レジスタマップ
	0x520C 0800 - 0x520C 0BFF	0x420C 0800 - 0x420C 0BFF	RNG	RNG レジスタマップ
	0x520C 0400 - 0x520C 07FF	0x420C 0400 - 0x420C 07FF	HASH	HASH レジスタマップ
	0x520C 0000 - 0x520C 03FF	0x420C 0000 - 0x420C 03FF	AES	AES レジスタマップ
	0x5202 C800 - 0x520B FFFF	0x4202 C800 - 0x420B FFFF	予約済み	-
	0x5202 C400 - 0x5202 C7FF	0x4202 C400 - 0x4202 C7FF	PSSI	PSSI レジスタマップ
	0x5202 C000 - 0x5202 C3FF	0x4202 C000 - 0x4202 C3FF	DCMI	DCMI レジスタマップ
	0x5202 8800 - 0x5202 BFFF	0x4202 8800 - 0x4202 BFFF	予約済み	-
	0x5202 8400 - 0x5202 87FF	0x4202 8400 - 0x4202 87FF	DAC1	DAC レジスタマップ
	0x5202 8000 - 0x5202 83FF	0x4202 8000 - 0x4202 83FF	ADC1 / ADC2	ADC レジスタマップ
	0x5202 2400 - 0x5202 7FFF	0x4202 2400 - 0x4202 7FFF	予約済み	-
	0x5202 2000 - 0x5202 23FF	0x4202 2000 - 0x4202 23FF	GPIOI	GPIO レジスタマップ
	0x5202 1C00 - 0x5202 1FFF	0x4202 1C00 - 0x4202 1FFF	GPIOH	
	0x5202 1800 - 0x5202 1BFF	0x4202 1800 - 0x4202 1BFF	GPIOG	
	0x5202 1400 - 0x5202 17FF	0x4202 1400 - 0x4202 17FF	GPIOF	
	0x5202 1000 - 0x5202 13FF	0x4202 1000 - 0x4202 13FF	GPIOE	
	0x5202 0C00 - 0x5202 0FFF	0x4202 0C00 - 0x4202 0FFF	GIPOD	
	0x5202 0800 - 0x5202 0BFF	0x4202 0800 - 0x4202 0BFF	GPIOC	
	0x5202 0400 - 0x5202 07FF	0x4202 0400 - 0x4202 07FF	GPIOB	
	0x5202 0000 - 0x5202 03FF	0x4202 0000 - 0x4202 03FF	GPIOA	

表 3. メモリマップとペリフェラルレジスタ境界アドレス (続き)

バス	セキュアな境界アドレス	非セキュアな境界アドレス	ペリフェラル	ペリフェラルレジスタマップ
AHB1	0x5003 7400 - 0x51FF FFFF	0x4003 7400 - 0x41FF FFFF	予約済み	-
	0x5003 6400 - 0x5003 73FF	0x4003 6400 - 0x4003 73FF	MPC_WM_BKP RAM	GTZC1 TZSC レジスタマップ
	0x5003 3800 - 0x5003 63FF	0x4003 3C00 - 0x4003 63FF	予約済み	-
	0x5003 2400 - 0x5003 37FF	0x4003 2400 - 0x4003 37FF	GTZC1	GTZC1 レジスタマップ
	0x5003 1800 - 0x5003 23FF	0x4003 1800 - 0x4003 23FF	予約済み	-
	0x5003 1400 - 0x5003 17FF	0x4003 1400 - 0x4003 17FF	DCACHE	DCACHE レジスタマップ
	0x5003 0800 - 0x5003 13FF	0x4003 0800 - 0x4003 13FF	予約済み	-
	0x5003 0400 - 0x5003 07FF	0x4003 0400 - 0x4003 07FF	ICACHE	ICACHE レジスタマップ
	0x5002 9400 - 0x5003 03FF	0x4002 9400 - 0x4003 03FF	予約済み	-
	0x5002 8000 - 0x5002 93FF	0x4002 8000 - 0x4002 93FF	ETHERNET MAC	ETHERNET レジスタマップ
	0x5002 7000 - 0x5002 7FFF	0x4002 7000 - 0x4002 7FFF	予約済み	-
	0x5002 6000 - 0x5002 6FFF	0x4002 6000 - 0x4002 6FFF	RAMCFG	RAMCFG レジスタマップ
	0x5002 4000 - 0x5002 5FFF	0x4002 4000 - 0x4002 5FFF	予約済み	-
	0x5002 3C00 - 0x5002 3FFF	0x4002 3C00 - 0x4002 3FFF	FMAC	FMAC レジスタマップ
	0x5002 3800 - 0x5002 3BFF	0x4002 3800 - 0x4002 3BFF	CORDIC	CORDIC レジスタマップ
	0x5002 3400 - 0x5002 37FF	0x4002 3400 - 0x4002 37FF	予約済み	-
	0x5002 3000 - 0x5002 33FF	0x4002 3000 - 0x4002 33FF	CRC	CRC レジスタマップ
	0x5002 2400 - 0x5002 2FFF	0x4002 2400 - 0x4002 2FFF	予約済み	-
	0x5002 2000 - 0x5002 23FF	0x4002 2000 - 0x4002 23FF	FLASH	Flash レジスタマップ
	0x5002 1000 - 0x5002 1FFF	0x4002 1000 - 0x4002 1FFF	DMA2	DMA レジスタマップ
	0x5002 0000 - 0x5002 0FFF	0x4002 0000 - 0x4002 0FFF	DMA1	

表 3. メモリマップとペリフェラルレジスタ境界アドレス (続き)

バス	セキュアな境界アドレス	非セキュアな境界アドレス	ペリフェラル	ペリフェラルレジスタマップ
APB2	0x5001 6C00 - 0x5001 FFFF	0x4001 6C00 - 0x4001 FFFF	予約済み	-
	0x5001 6400 - 0x5001 6BFF	0x4001 6400 - 0x4001 6BFF	USB_FS RAM	USB レジスタマップ
	0x5001 6000 - 0x5001 63FF	0x4001 6000 - 0x4001 63FF	USB_FS	
	0x5001 5C00 - 0x5001 5FFF	0x4001 5C00 - 0x4001 5FFF	予約済み	-
	0x5001 5800 - 0x5001 5BFF	0x4001 5800 - 0x4001 5BFF	SAI2	SAI レジスタマップ
	0x5001 5400 - 0x5001 57FF	0x4001 5400 - 0x4001 57FF	SAI1	
	0x5001 5000 - 0x5001 53FF	0x4001 5000 - 0x4001 53FF	SPI6	SPI レジスタマップ
	0x5001 4C00 - 0x5001 4FFF	0x4001 4C00 - 0x4001 4FFF	SPI4	
	0x5001 4800 - 0x5001 4BFF	0x4001 4800 - 0x4001 4BFF	TIM17	TIM16/TIM 17 レジスタマップ
	0x5001 4400 - 0x5001 47FF	0x4001 4400 - 0x4001 47FF	TIM16	
	0x5001 4000 - 0x5001 43FF	0x4001 4000 - 0x4001 43FF	TIM15	TIM15 レジスタマップ
	0x5001 3C00 - 0x5001 3FFF	0x4001 3C00 - 0x4001 3FFF	予約済み	-
	0x5001 3800 - 0x5001 3BFF	0x4001 3800 - 0x4001 3BFF	USART1	USART レジスタマップ
	0x5001 3400 - 0x5001 37FF	0x4001 3400 - 0x4001 37FF	TIM8	TIMx レジスタマップ
	0x5001 3000 - 0x5001 33FF	0x4001 3000 - 0x4001 33FF	SPI1 / I2S1	SPI レジスタマップ
	0x5001 2C00 - 0x5001 2FFF	0x4001 2C00 - 0x4001 2FFF	TIM1	TIMx レジスタマップ
	0x5001 0000 - 0x5001 2BFF	0x4001 0000 - 0x4001 2BFF	予約済み	-
APB1	0x5000 E000 - 0x5000 FFFF	0x4000 E000 - 0x4000 FFFF	予約済み	-
	0x5000 DC00 - 0x5000 DFFF	0x4000 DC00 - 0x4000 DFFF	UCPD1	UCPD レジスタマップ
	0x5000 B400 - 0x5000 DBFF	0x4000 B400 - 0x4000 DBFF	予約済み	-
	0x5000 AC00 - 0x5000 B3FF	0x4000 AC00 - 0x4000 B3FF	FDCAN SRAM	FDCAN レジスタマップ
	0x5000 A800 - 0x5000 ABFF	0x4000 A800 - 0x4000 ABFF	FDCAN2	
	0x5000 A400 - 0x5000 A7FF	0x4000 A400 - 0x4000 A7FF	FDCAN1	
	0x5000 9800 - 0x5000 A3FF	0x4000 9800 - 0x4000 A3FF	予約済み	-
	0x5000 9400 - 0x5000 97FF	0x4000 9400 - 0x4000 97FF	LPTIM2	LPTIM レジスタマップ
	0x5000 9000 - 0x5000 93FF	0x4000 9000 - 0x4000 93FF	予約済み	-
	0x5000 8C00 - 0x5000 8FFF	0x4000 8C00 - 0x4000 8FFF	DTS	DTS レジスタマップ
	0x5000 8800 - 0x5000 8BFF	0x4000 8800 - 0x4000 8BFF	予約済み	-
	0x5000 8400 - 0x5000 87FF	0x4000 8400 - 0x4000 87FF	UART12	USART レジスタマップ
	0x5000 8000 - 0x5000 83FF	0x4000 8000 - 0x4000 83FF	UART9	
	0x5000 7C00 - 0x5000 7FFF	0x4000 7C00 - 0x4000 7FFF	UART8	
	0x5000 7800 - 0x5000 7BFF	0x4000 7800 - 0x4000 7BFF	UART7	
	0x5000 7400 - 0x5000 77FF	0x4000 7400 - 0x4000 77FF	予約済み	-

表 3. メモリマップとペリフェラルレジスタ境界アドレス (続き)

バス	セキュアな境界アドレス	非セキュアな境界アドレス	ペリフェラル	ペリフェラルレジスタマップ
APB1	0x5000 7000 - 0x5000 73FF	0x4000 7000 - 0x4000 73FF	HDMI-CEC	HDMI レジスタマップ
	0x5000 6C00 - 0x5000 6FFF	0x4000 6C00 - 0x4000 6FFF	USART11	USART レジスタマップ
	0x5000 6800 - 0x5000 6BFF	0x4000 6800 - 0x4000 6BFF	USART10	
	0x5000 6400 - 0x5000 67FF	0x4000 6400 - 0x4000 67FF	USART6	
	0x5000 6000 - 0x5000 63FF	0x4000 6000 - 0x4000 63FF	CRS	CRS レジスタマップ
	0x5000 5C00 - 0x5000 5FFF	0x4000 5C00 - 0x4000 5FFF	I3C1	I3C レジスタマップ
	0x5000 5800 - 0x5000 5BFF	0x4000 5800 - 0x4000 5BFF	I2C2	I2C レジスタマップ
	0x5000 5400 - 0x5000 57FF	0x4000 5400 - 0x4000 57FF	I2C1	
	0x5000 5000 - 0x5000 53FF	0x4000 5000 - 0x4000 53FF	UART5	USART レジスタマップ
	0x5000 4C00 - 0x5000 4FFF	0x4000 4C00 - 0x4000 4FFF	UART4	
	0x5000 4800 - 0x5000 4BFF	0x4000 4800 - 0x4000 4BFF	USART3	
	0x5000 4400 - 0x5000 47FF	0x4000 4400 - 0x4000 47FF	USART2	
	0x5000 4000 - 0x5000 43FF	0x4000 4000 - 0x4000 43FF	予約済み	-
	0x5000 3C00 - 0x5000 3FFF	0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3	SPI レジスタマップ
	0x5000 3800 - 0x5000 3BFF	0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2	
	0x5000 3400 - 0x5000 37FF	0x4000 3400 - 0x4000 37FF	予約済み	-
	0x5000 3000 - 0x5000 33FF	0x4000 3000 - 0x4000 33FF	IWDG	IWDG ハードウェア構成 レジスタ (IWDG_HWCFGR)
	0x5000 2C00 - 0x5000 2FFF	0x4000 2C00 - 0x4000 2FFF	WWDG	WWDG レジスタマップ
	0x5000 2400 - 0x5000 2BFF	0x4000 2400 - 0x4000 2BFF	予約済み	-
	0x5000 2000 - 0x5000 23FF	0x4000 2000 - 0x4000 23FF	TIM14	TIMx レジスタマップ
	0x5000 1C00 - 0x5000 1FFF	0x4000 1C00 - 0x4000 1FFF	TIM13	
	0x5000 1800 - 0x5000 1BFF	0x4000 1800 - 0x4000 1BFF	TIM12	
	0x5000 1400 - 0x5000 17FF	0x4000 1400 - 0x4000 17FF	TIM7	
	0x5000 1000 - 0x5000 13FF	0x4000 1000 - 0x4000 13FF	TIM6	
	0x5000 0C00 - 0x5000 0FFF	0x4000 0C00 - 0x4000 0FFF	TIM5	
	0x5000 0800 - 0x5000 0BFF	0x4000 0800 - 0x4000 0BFF	TIM4	
	0x5000 0400 - 0x5000 07FF	0x4000 0400 - 0x4000 07FF	TIM3	
	0x5000 0000 - 0x5000 03FF	0x4000 0000 - 0x4000 03FF	TIM2	

2.3.3 内蔵 SRAM

デバイスは、最大 644 KB の SRAM を搭載しています。

- 256 KB の SRAM1
- 64 KB の SRAM2
- 320 KB の SRAM3
- 4 KB の BKPSRAM

これらの SRAM には、バイト、ハーフワード（16 ビット）、またはフルワード（32 ビット）によるアクセスが可能です。これらのメモリは、CPU と DMA の両方によるアドレス指定が可能です。

CPU は、選択したアドレスに応じて、システムバスまたは C バス経由で SRAM1、SRAM2、および SRAM3 にアクセスできます。CPU は、システムバス経由でのみ BKPSRAM にアクセスできます。

TrustZone セキュリティが有効な場合、すべての SRAM はリセット後はセキュアです。SRAM は、ブロック単位で非セキュアとしてプログラムできます。詳細については、[セクション 5：グローバル TrustZone® コントローラ \(GTZC\)](#) を参照してください。

SRAM 機能の詳細については、[セクション 6.3.1：内部 SRAM 機能](#) を参照してください。

2.3.4 Flashメモリの概要

Flashメモリは2つの別個の物理領域で構成されます。

- アプリケーションプログラムとユーザデータを含むメイン Flash メモリブロック。
- 以下の部分で構成される情報ブロック。
 - ハードウェアとメモリ保護のユーザ設定のオプションバイト。
 - ST 独自仕様コードを含むシステムメモリ
 - OTP（一度だけプログラム可能な）領域。

Flashインタフェースは、AHB プロトコルに基づいて命令アクセスとデータアクセスを実装します。また、Flash レジスタとセキュリティアクセス制御機能によって制御される Flash メモリ操作（プログラム/消去）の実行に必要なロジックを実装します。詳細については、[セクション 7：内蔵 Flash メモリ \(FLASH\)](#) を参照してください。

2.3.5 ブートモード

起動時、コアは、BOOT0 ピン、BOOT_ADDR オプションバイト、TZEN オプションビット、デバッグリクエスト、および製品状態によって設定されたブートアドレスにジャンプします。

BOOT_ADDR がまだ設定されていない場合、専用ライブラリ（ST 生産時にプログラム）をセキュアブートに使用できます。それらはシステム Flash メモリにあります。

- システム Flash メモリの ST ライブラリは、セキュアブートとセキュアファームウェアインストール（SFI-RSS）などの特殊機能でアプリケーションソフトウェアのブートを支援します。
- ユーザ Flash メモリの ST iROT（immutable root of trust：不変の信頼の起点）セキュアソフトウェアは、セキュアファームウェア更新とプロビジョニング（SFU）に使用されます。

デバッグが製品にアタッチされている場合、エントリポイントはデバッグ認証ポリシーであり、デバッグにアタッチされるデバイスをアンロックするために使用されます。デバッグが許される製品状態へのリグレーションを実行するには、デジタル署名が必要です。

内蔵ブートローダ

内蔵ブートローダは、システムメモリに配置され、生産時に ST によってプログラムされています。アプリケーションノート「STM32 マイクロコントローラシステムメモリブートモード」(AN2606)を参照してください。

3 システムセキュリティ

STM32H563/H573および STM32H562 は、包括的なセキュリティ機能のセットで設計され、その一部は標準の Arm TrustZone® テクノロジーに基づきます。

これらのセキュリティ機能は、IoT デバイスをセキュリティ標準に照らして評価するプロセスを簡素化します。また、再利用を容易にし、相互運用性を高め、API の断片化を最小限に抑えるため、OEM およびサードパーティ開発者にとって、ソフトウェア開発のコストと複雑性が大幅に軽減します。

このセクションでは、STM32H563/H573および STM32H562 デバイスで使用可能なさまざまなセキュリティ機能について説明します。

3.1 主なセキュリティ機能

- Cortex-M33 の特権モードと Armv8-M メインラインセキュリティ拡張をセキュリティ保護可能な I/O、メモリ、およびペリフェラルに拡張して使用することによるリソースの分離
- デバイス固有の暗号鍵ペアによるセキュアなファームウェアインストール (SFI)
 - USART、USB、I²C、I³C、SPI、FDCAN、および JTAG 経由のイメージダウンロードをサポートするオンチップの不変ブートローダの利用
- ブートエントリ：このプラットフォームでは、ネイティブの不変な信頼の起点または独自仕様のブートエントリ（ユーザ Flash メモリ内）の選択が可能です。
- セキュリティサービス（システム Flash 内）：このプラットフォームにはネイティブセキュリティサービスが付属し、システムメモリに内蔵されて、信頼の起点サービスを管理します。ネイティブの信頼の起点サービスが処理するもの：セキュアブート、次のブートレベルのセキュアな更新（uROT：updatable root of trust）、セキュアなデバッグ制御（デバッグの再開、リグレーション制御）などのプラットフォームセキュリティ。サービスは、プロビジョニングツールを使用して、OEM ごとにパーソナライズできます。
- 時間的分離：HDPL（hide protect level：保護レベルの非表示）単調カウンタにより、ブートレベルが分離されます。
- 以下の特徴を持つセキュアストレージ：
 - HDPL と TrustZone によって保護されたセキュアストレージ専用の 5 つの不揮発性領域。
 - タンパ発生時に自動的に消去される、バッテリー駆動の揮発性セキュアストレージ
 - AES エンジンの書込み専用キーレジスタ
 - デバイス 96 ビットユニーク ID と JTAG 32 ビットデバイス固有 ID
 - セキュアストレージは、SAES エンジンに依存して、保存されたデータを暗号化し、DHUK プロパティのメリットを活かすことができます。DHUK で暗号化されたすべてのデータは、時間的およびランタイム分離（HDPL および TrustZone）、RHUK、EPOCH（バージョンカウンタ）プロパティのメリットを活かすことができます。
- a) データの分離：SAES+DHUK に依存して暗号化/復号されるすべてのデータは、さまざまな分離領域（HDPL0、HDPL1、HDPL2、HDPL3S、HDPL3NS）について、DHUK のバリエーションのメリットを活かすことができます。
- b) RHUK：ハードウェアシークレット不揮発性、デバイスキーごとにユニーク（コピー保護）
- c) EPOCH：リグレーションが行われるたびにインクリメントされるカウンタ（アンチロールバック）。

- 汎用暗号化アクセラレーション
 - ECB、CBC、CTR、GCM、および CCM 連鎖モードをサポートする AES 256 ビットエンジン
 - ECB、CBC、CTR、GCM、および CCM 連鎖モードをサポートする、サイドチャネル対策と緩和策を備えたセキュア AES 256 ビットセキュリティコプロセッサ
 - SHA-1 チェックサムと SHA-2 セキュアハッシュ（SHA2、SHA2-384、SHA2-512）をサポートする HASH プロセッサ
 - シークレットを操作するときのサイドチャネル対策と緩和策を実装する、RSA/DH（最大 4096 ビット）および ECC（最大 640 ビット）向け公開鍵アクセラレータ（PKA）
 - 真の乱数発生器（RNG）、NIST SP800-90B 事前認定済み
- OCTOSPI 経由で接続された外部 Flash メモリに保存された暗号化されたイメージのオンザフライ復号
 - 標準 NOR Flash メモリで遅延がほぼゼロ
 - デバイスユニーク秘密キーを使用してイメージを暗号化するのに使用できます。
 - タンパ発生時の自動的な鍵消去
- 新しい柔軟なライフサイクルスキーム
 - 現場での製品メンテナンスが可能（デバッグの再開またはリグレーション）
 - 不変の信頼の起点、セキュアアプリケーション、および非セキュアアプリケーションの最大 3 ステップで保護されたファームウェア配布が可能
- 温度、電圧、および周波数攻撃に対するアクティブタンパおよび保護
 - さまざまな電力モードで使用可能な最大 8 つのアクティブ入力、8 つのアクティブ出力タンパピン（TAMP 低電力モードを参照）

3.2 セキュアインストール

セキュアファームウェアインストール（SFI）は、STMicroelectronics によってデバイスに内蔵された不変のセキュアサービスです。SFI により、信頼できない生産環境（OEM 契約製造業者など）でも OEM ファームウェアのセキュアでカウントされたインストールが可能です。

内部 Flash メモリに、または外部 Flash メモリに暗号化されて書き込まれるインストールイメージの秘匿性も AES を使用して保護されます。

SFI ネイティブサービスは、以下のハードウェアセキュリティ機能を利用します。

- セキュアブート（[セクション 3.3](#) を参照）
- TrustZone を使用したリソース分離（[セクション 3.6](#) を参照）
- 非表示保護を使用した時間的分離（[セクション 3.7.1](#) を参照）
- セキュア実行（[セクション 3.8](#) を参照）
- 関連付けられた暗号化エンジン（[セクション 3.9](#) および [セクション 3.10](#) を参照）と Flash メモリ専用領域によるセキュアストレージ

詳細については、AN4992「セキュアファームウェアインストール（SFI）の概要」を参照してください。

3.3 セキュアブート

セキュアブートは、システムリセット後に常に実行される不変コードです。信頼の起点として、このコードは、デバイスの静的保護をチェックし、使用可能なデバイスランタイム保護を有効にして、プラットフォームでの無効な、または悪意あるコード実行のリスクを軽減します。信頼の起点として、セキュアブートは、実行前に、次のレベルのファームウェアの完全性と認証性もチェックします。

STM32H573 は、リセット後の不変コードの実行について次の 2 つのオプションを提供します (BOOT_UBE を使用して)。

- ST-iROT (システムメモリにネイティブに存在するコード) が選択されたときには、セキュリティサービスが次のブートレベルのセキュアブートを管理します。ST-iROT は、OEM によって行われた設定を使用して、次のレベルの完全性と認証性をチェックします。次のステージは、ST 更新可能サービス (ST-uROT) または独自仕様サービス (uROT) です。
- OEM が連鎖全体を管理するときには (OEM-iROT)、独自仕様のブートエントリ。適切なセキュリティアクティベーションにより、ユーザ Flash メモリにインストールされます。

セキュアブートの実際の機能は、TrustZone 機能の可用性と、デバイスの保存されているファームウェアによって異なります。セキュアブートは、一般に、セキュアストレージを開始し、オンザフライ復号キーを OTFDEC にインストールします。これは、外部 Flash メモリに保存された暗号化されたファームウェアを使用するためです。

TrustZone アプリケーションは、STM32 エコシステムによってサポートされ、セキュアブート機能も含め、信頼の起点ソリューションを提供します。

デバイスでは、セキュアブートは、次のようなハードウェアセキュリティ機能を利用します。

- TrustZone を使用したリソース分離 ([セクション 3.6](#) を参照)
- 非表示保護レベル (HDPL) を使用した時間的分離 ([セクション 3.7.1](#) を参照)
- セキュア実行 ([セクション 3.8](#) を参照)
- セキュアインストールおよび更新 ([セクション 3.2](#) および [セクション 3.4](#) を参照)
- 使用可能な場合、関連付けられた暗号化エンジンによるドメインごとのセキュアストレージ ([セクション 3.9](#) および [セクション 3.10](#) を参照)

このセクションでは、特にセキュアブートのために設計された機能について説明します。

3.3.1 ユニークブートエントリ

ユニークブートエントリ (BOOT_UBE) により、システム Flash (ST-iROT) 内のセキュリティサービスと独自仕様のブートエントリ (OEM-iROT) の間でブートエントリポイントを選択できます。この選択は、TrustZone が有効なとき (TZEN=0xB4)、暗号化アクセラレーション (STM32H573) を内蔵している製品でのみ可能です。

TrustZone が有効で (TZEN = 0xB4)、SECBOOT_LOCK セキュアオプションビットがクリアされているとき、アプリケーションは、システム Flash メモリ (次のセクションを参照) またはセキュアユーザ Flash メモリのいずれかの、SECBOOTADD オプションバイトによって定義されたアドレスにあるブートエントリポイントを選択します。

TrustZone が有効で (TZEN = 0xB4)、SECBOOT_LOCK セキュアオプションビットがセットされているとき、デバイスユニークブートエントリは、SECBOOTADD オプションバイトによって定義された変更不能セキュアアドレスです。SECBOOT_LOCK がセットされているとき、アプリケーションはこれらのオプションバイトを変更できません。

TrustZone が無効で (TZEN = 0xC3)、NSBOOT_LOCK オプションビットがクリアされているとき、アプリケーションは、非セキュアユーザ Flash メモリの、NSBOOTADD オプションバイトによって定義されたアドレスにあるブートエントリポイントを選択します。

TrustZone が無効で (TZEN = 0xC3)、NSBOOT_LOCK オプションビットがセットされているとき、デバイスユニークブートエントリは、NSBOOTADD オプションバイトによって定義された変更不能な非セキュアアドレスです。NSBOOT_LOCK がセットされているとき、アプリケーションはこれらのオプションバイトを変更できません。

注： SECBOOT_LOCK および NSBOOT_LOCK は、PRODUCT_STATE = オープン、プロビジョニング、またはリグレーションのときのみ変更できます。

ブートメカニズムの詳細については、[セクション 4：ブートモード](#)を参照してください。

3.3.2 システム Flash メモリの不変の信頼の起点

システム Flash メモリに保存される不変の信頼の起点コードは、まず、SFI の実行に使用されて、信頼できない生産環境 (OEM 契約製造業者など) での OEM ファームウェアのセキュアでカウントされるインストールを可能にします。

STMicroelectronics の不変コードは、セキュアアプリケーションが SBS_RSSCMDR レジスタを非ヌル値に設定しているとき、システムリセットをトリガする前に、ランタイムに呼び出すことができるセキュアランタイムサービスも含んでいます。このランタイム機能は、BOOT_LOCK セキュアオプションビットがセットされているとき、または PRODUCT_STATE が Open でないときには無効です。

STM32H563/H573 製品は、セキュリティサービスを備えています。

- ST-DA (デバッグ認証) は、デバッグ認証制御機能を管理し、製品のアフターサービス (現場返品) として、製品のデバッグの再開とリグレーションの制御を可能にします。

STM32H573xx 製品は、セキュリティサービスを備えています。

- ST-iROT (不変の信頼の起点) は、プラットフォームの最初の更新可能レベルのセキュアプロビジョニング、セキュアブート、セキュア更新を処理します。

3.4 セキュア更新

セキュアファームウェア更新は、セキュアブート後に実行するセキュアサービスです。実際の機能は、TrustZone 機能の可用性と、デバイスに保存されているファームウェアによって異なります。

Cortex-M ベースのアプリケーションプロセッサは、一般に、信頼されるブートと信頼される OS を実行して、信頼される実行環境 (Trusted Execution Environment : TEE) を作成するために使用されます。信頼されるブートと TEE アプリケーションは、STM32 エコシステムによってサポートされ、マイクロコントローラビルトインプログラムの新しいファームウェアバージョンへの更新を可能にして、新しい機能を追加し、潜在的な問題を修正します。更新プロセスはセキュアな方法で行われ、無許可の更新やデバイス上の機密データへの無許可アクセスを防止します。

ファームウェアの更新は、セキュア部分と非セキュア部分の両方を含んだ単一のファームウェアイメージで、またはファームウェアイメージのセキュア (それぞれ非セキュア) 部分を個別に行うことができます。

デバイスでは、セキュア更新アプリケーションは、[セクション 3.2](#) で説明されているファームウェアインストールと同じハードウェアセキュリティを利用します。

3.5 非表示保護レベルを使用したリソース分離

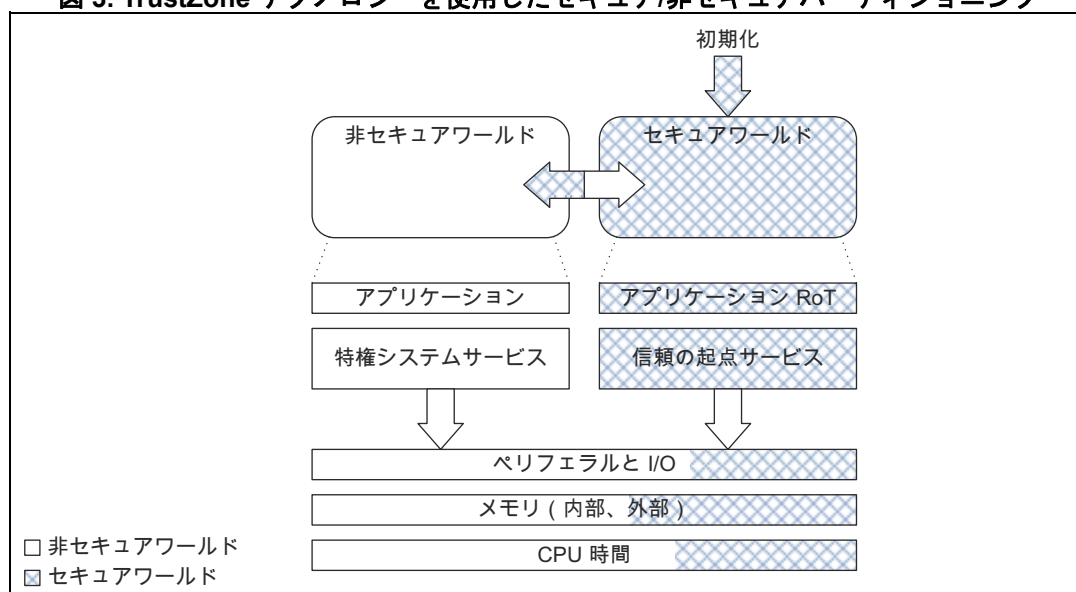
STM32H563/H573および STM32H562 デバイスでは、ブートに使用されるハードウェアおよびソフトウェアリソースを分離できます。これを時間的分離といいます。

これは、レベルのインクリメントのみを処理する単調カウンタに基づきます。カウンタがインクリメントされると、前のレベルのリソースは非表示になります（コードとデータ）。

3.6 TrustZone を使用したリソースの分離

STM32H563/H573および STM32H562 デバイスでは、ハードウェアおよびソフトウェアリソースをパーティショニングして、図 3 に示されているように、セキュアワールドまたは非セキュアワールドのいずれかに配置できます。

図 3. TrustZone テクノロジーを使用したセキュア/非セキュアパーティショニング



注：プラットフォームの初期のパーティショニングは、デバイスのリセット後に実行されるセキュアファームウェアの責任で行われます。

このリソース分離テクノロジーにより、セキュアワールドを使用して、非セキュアワールドで実行する、より公開度の高いコードからの意図的または偶発的なタンパに対して重要なコードを保護できます。

注：セキュアコードは、一般に小さく、ほとんど変更されませんが、非セキュアコードは、より公開度が高く、ファームウェアの更新が多い傾向があります。

3.6.1 TrustZone セキュリティアーキテクチャ

Armv8-M TrustZone テクノロジーは、包括的なハードウェアアーキテクチャであり、プロセッサおよびシステム全体に及ぶ包括的でホリステックな保護を開発者に提示します。デバイスの TrustZone ハードウェア機能は、次のとおりです。

- Cortex-M33 の Armv8-M メインラインセキュリティ拡張。関連するセキュア割込みにより、新しいプロセッサセキュア状態を有効にします。
- Cortex-M33 の 8 つのセキュリティ属性ユニット (SAU) 領域を使用した、メモリおよびペリフェラルの TrustZone への動的割り当て
- グローバル TrustZone フレームワーク (GTZC)。TrustZone 保護を Cortex-M33 以外のシステムのマスタから着信するトランザクションに対して拡張します。
- TrustZone 対応の内蔵 Flash メモリおよびペリフェラル

注： **FLASH_OPTSR2_PRG** レジスタの **TZEN** オプションビットは、TrustZone セキュリティを有効にします。

3.6.2 Cortex-M33 の Armv8-M セキュリティ拡張

Cortex-M33 の Arm セキュリティ拡張は、変革ではなく進化です。Cortex-M4 など、以前の Cortex-M サブファミリのプログラマモデルを使用します。実際、Armv8-M は、アーキテクチャとしては Armv7-M と同様であり、同じ 32 ビットアーキテクチャ、MPU で保護された同じメモリマップリソースを使用します。Armv8-M は、ネスト化されたベクタ割込みコントローラ (NVIC) も使用します。

STM32H563/H573 および STM32H562 デバイスでの Armv8-M TrustZone 実装は、以下の機能で構成されています。

- 新しいプロセッサ状態。追加のコード / サイクルのオーバーヘッドがほとんどなく (Armv8-A TrustZone と違って)、専用の例外ルーチンを使用してセキュア/非セキュアワールドの変更をトリガします。
- 共有 4 GB のアドレス空間の 2 つのメモリマップビュー
- セキュアと非セキュアの両方のドメインの低割込み遅延と、セキュリティグルーピングとプライオリティ設定用の新しい割込み設定
- セキュアおよび非セキュア例外用の個別の例外ベクタテーブル
- マイクロコード化されたコンテキスト保持
- スタック限界チェッカー付きのスタックポインタを含め、セキュア/非セキュア状態の特定のレジスタのバンキング
- 次の Cortex-M33 プログラム可能コンポーネントのバンキング (セキュアと非セキュアの 2 つの個別のユニット)。
 - SysTick タイマ
 - MPU 設定レジスタ (セキュアに 12 の MPU 領域、非セキュアに 8 つ)
 - システム制御ブロック (SCB) レジスタの一部
- セキュリティ違反を処理するための新しいシステム例外 (SecureFault)
- [セクション 3.12](#) で定義されている設定可能なデバッグサポート

詳細については、Cortex-M33 プログラミングマニュアル (PM0264) を参照してください。

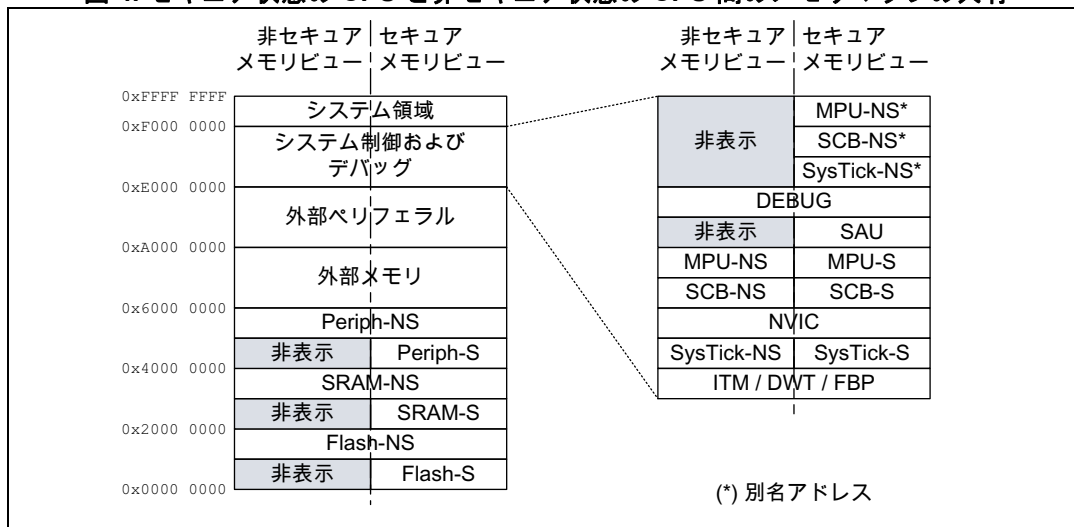
3.6.3 IDAU/SAU を使用したメモリおよびペリフェラルの割り当て

セキュリティ属性

図 4 に示されているように、Armv8-M 非セキュアメモリビューは Armv7-M (Cortex-M4 で見られる) と同様ですが、セキュアメモリが非表示である点が異なります。セキュアメモリビューは、Cortex プロセッサがセキュア状態で実行しているときにのみアクセス可能な Flash メモリ、SRAM、およびペリフェラルを示します。

注： 図 4 は、SAU 設定後にセキュアコードによって表示される 32 ビットアドレス空間を示しています。

図 4. セキュア状態の CPU と非セキュア状態の CPU 間のメモリマップの共有

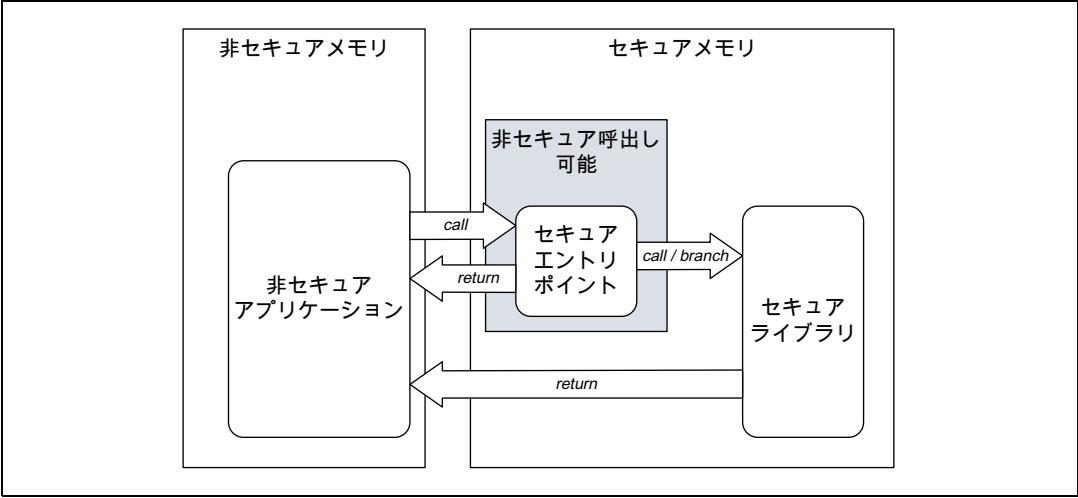


Cortex プロセッサの状態（および関連する権利）は、実行されるメモリ領域に割り当てられたセキュリティ属性に依存します。

- 非セキュア/セキュア状態のプロセッサは、非セキュア（NS）/セキュア（S）プログラムメモリからのみ実行します。
- セキュア状態で実行するとき、プロセッサは S と NS の両方のメモリのデータにアクセスできます。非セキュア状態で実行するとき、CPU は非セキュアメモリに制限されます。

セキュアワールドへの遷移を管理するには、開発者はセキュアライブラリへの有効なエントリポイントを含んだ非セキュア呼出し可能（NSC）領域を作成する必要があります。これらのエントリポイントの最初の命令は、非セキュアコードがセキュア関数を呼び出すために使用する新しいセキュアゲート（SG）命令でなければなりません（図 5 を参照）。

図 5. セキュアワールド遷移とメモリパーティショニング



セキュリティ属性のプログラミング

Cortex-M33 では、静的実装によって定義された属性ユニット（IDAU）は、プログラム可能なセキュリティ属性ユニット（SAU）と連携して機能して、表 4 に示されているように、特定のセキュリティ属性（S、NS、または NSC）を特定のアドレスに割り当てます。

表 4. IDAU および SAU によるセキュリティ属性の設定

IDAU セキュリティ属性	SAU セキュリティ属性 ⁽¹⁾	最終的なセキュリティ属性
非セキュア	セキュア	セキュア
	セキュア-NSC	セキュア-NSC
	非セキュア	非セキュア
セキュア-NSC	セキュア	セキュア
	非セキュア	セキュア-NSC

1. 定義された領域は 32 バイト境界に揃えられます。

Cortex-M33 は、セキュア特権状態でのみ SAU を設定できます。TrustZone が有効なとき、SAU は、デフォルトですべてのアドレスをセキュア（S）として定義します。セキュアブートアプリケーションは、その後、表 4 に示されているように、SAU をプログラムして、NSC または NS 領域を作成できます。

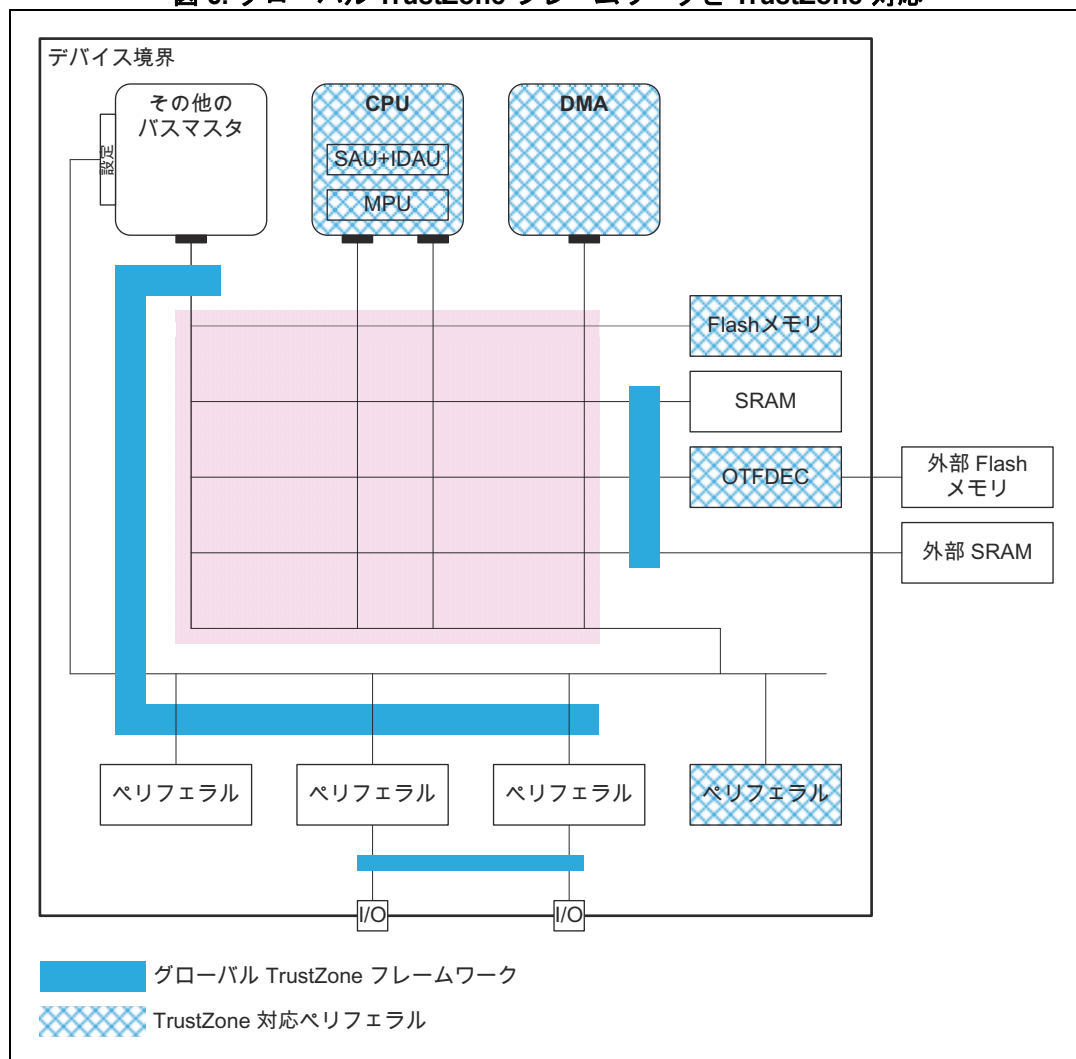
注： SAU/IDAU 設定は Cortex-M33 にのみ適用されます。DMA など、他のマスタは、これらのポリシーの影響を受けません。

3.6.4 GTZC を使用したメモリおよびペリフェラルの割り当て

グローバル TrustZone フレームワークアーキテクチャ

Cortex-M33 の Armv8-M TrustZone セキュリティ拡張の上に、デバイスは、セキュアワールドと非セキュアワールドの分離を柔軟に補強する補足的なセキュリティ機能を内蔵します。SAU/IDAU と違って、GTZC は、レガシーメモリおよびペリフェラルを Cortex-M33 以外のマスタから着信する非セキュアトランザクションに対して保護できます。

図 6. グローバル TrustZone フレームワークと TrustZone 対応



TZSC によるペリフェラルのセキュリティ保護

TrustZone セキュリティがアクティブなとき、ペリフェラルは、前の図に示されているように、GTZC を通じてセキュリティ保護可能か、ネイティブで TrustZone 対応のいずれかです。

- セキュリティ保護可能なペリフェラルまたはメモリは、TrustZone セキュリティコントローラ (TZSC) によって制御される AHB/APB ファイアウォールゲートによって保護されます。
- TrustZone 対応ペリフェラルまたはメモリは、AHB または APB 相互接続に直接接続されて、セキュアレジスタのサブセットやセキュアメモリ領域など、特定の TrustZone 動作を実装します。

セキュリティ保護可能なペリフェラルが GTZC によりセキュア専用になっているとき、このペリフェラルが相互接続のマスタの場合、セキュアトランザクションを自動的に発行します。SDMMC は、セキュリティ保護可能なマスタの例です。Cortex-M33 や DMA など、TrustZone 対応 AHB マスタは、GTZC とは無関係に、セキュリティモードに従って、AHB 相互接続でセキュア信号を駆動します。

注： TrustZone と同様、ペリフェラルを TZSC で特権専用にできます ([セクション 3.7.2](#) を参照)。この場合、このペリフェラルが相互接続のマスタである場合、特権トランザクションを自動的に発行します。

TZSC および MPCBB によるメモリのセキュリティ保護

GTZC の TZSC ブロックは、セキュリティ保護可能な外部メモリのすべてのセキュリティと特権を管理し、表 5 で定義されているように、MPCWM リソースをプログラムする機能を備えています。

表 5. MPCWMx リソース

メモリ	MPC リソース	フィルタリングのタイプ	領域数	デフォルトセキュリティ	オンザフライ復号 ⁽¹⁾
OCTOSPI1	MPCWM1	非セキュア特権 または非特権領域 (ウォーターマーク)	2	セキュア特権 ⁽²⁾	はい
FMC_NOR バンク	MPCWM2		2		いいえ
FMC_NAND / FMC_SDRAM バンク	MPCWM3		1		
バックアップ SRAM (BKPSRAM) / FMC_SDRAM	MPCWM4		1		

1. OTFDEC の使用。
2. デバイスで TrustZone が有効であることを前提とし、そうでない場合は、非セキュア非特権。

GTZC 内の MPCBB リソースは、表 6 で定義されているように、内蔵 SRAM ブロックのセキュリティと特権を設定する機能を備えています。

表 6. MPCBBx リソース

メモリ	MPC リソース	フィルタリングのタイプ	メモリサイズ	ブロックサイズ	スーパーブロックの数	デフォルトセキュリティ
SRAM1	GTZC1_MPCBB1	セキュリティと特権の ブロックベースの管理	256 KB	512 ⁽¹⁾ バイト	16	セキュア特権 ⁽²⁾
SRAM2	GTZC1_MPCBB2		64 KB		4	
SRAM3	GTZC1_MPCBB3		320 KB		20	

1. ブロックは、32 の連続したブロックから成るスーパーブロックにグループ化され、設定ロックを管理します。
2. デバイスで TrustZone が有効であることを前提とし、そうでない場合は、非セキュア非特権。

GTZC 設定の適用

TZSC および MPCBB ブロックは、次のいずれかの方法で使用できます。

- セキュアブート時に静的にプログラムされ、ロックされて、その後は変更されない
- 特定のアプリケーションコードまたはリアルタイムカーネルを使用して動的に再プログラムされる

動的オプションが選択され、設定がロックされていないとき：

- MPCBB セキュアブロックまたは MPCWM 非セキュア領域サイズをセキュアソフトウェアによって変更できます。このソフトウェアは MPCWM に対する特権を持つ必要があり、特定のブロックが特権専用でない場合は、非特権になります。
- 各ペリフェラルのセキュア（それぞれ特権）状態は、GTZC_TZSC_SECCFRGx（それぞれ GTZC_TZSC_PRIVCFGRx）レジスタに書き込むことで変更できます。

TZSC によるペリフェラルのセキュリティ保護

GTZC 内の TZSC ブロックは、セキュリティ保護可能なすべてのペリフェラルのセキュリティと特権を管理する機能を備えています。これらのペリフェラルのリストは、[セクション 5：グローバル TrustZone® コントローラ \(GTZC\)](#) にあります。

注： TrustZone が無効なときでも、リソース分離ハードウェア GTZC を使用して、ペリフェラルを特権コード専用に分離できます ([セクション 3.7.2](#) を参照)。

TrustZone が有効なとき、ペリフェラルは、リセット後は非セキュアおよび非特権として設定されます。

TrustZone 対応ペリフェラル

デバイスには、以下の TrustZone 対応ペリフェラルが含まれます。

- GPIOA ~ GPIOI
- GTZCx_MPCBB、GTZCx_TZIC、および GTZCx_TZSC (GTZC ブロック)
- OTFDEC1、TZEN = 1 の場合、セキュアでのみ書込み可能
- EXTI
- Flashメモリ
- RCC および PWR
- GPDMA1 および GPDMA2
- SBS レジスタ
- RTC および TAMP
- MCU デバッグユニット DBGMCU

それらのペリフェラルへの不正アクセスを TZIC レジスタを介して監視する方法は、[セクション 5：グローバル TrustZone® コントローラ \(GTZC\)](#) で説明されています。

詳細については、[セクション 3.6.5](#) を参照してください。

TrustZone 不正アクセスコントローラ (TZIC)

GTZC 内の TZIC は、GTZC または TrustZone 対応によって保護されているソースから発生したすべての不正アクセスイベントを収集して、NVIC に対して 1 つのグローバルセキュア割込みを生成します。

TZIC は、システムの TrustZone 機能が有効 (TZEN = 0xB4) なときのみ使用可能です。TZIC レジスタへのアクセスはすべて、セキュアである必要があります。

各不正イベントソースについて、ステータスフラグとクリアビットがあります。各不正イベントをマスクして、NVIC に対する割込みを生成しないようにできます。

注： デフォルトでは、すべてのイベントがマスクされています。

3.6.5 TrustZone 対応ペリフェラルのセキュリティの管理

このセクションでは、前のセクションでリストされた TrustZone 対応ペリフェラルでのセキュリティの実装方法について詳しく説明します。

内蔵 Flash メモリ

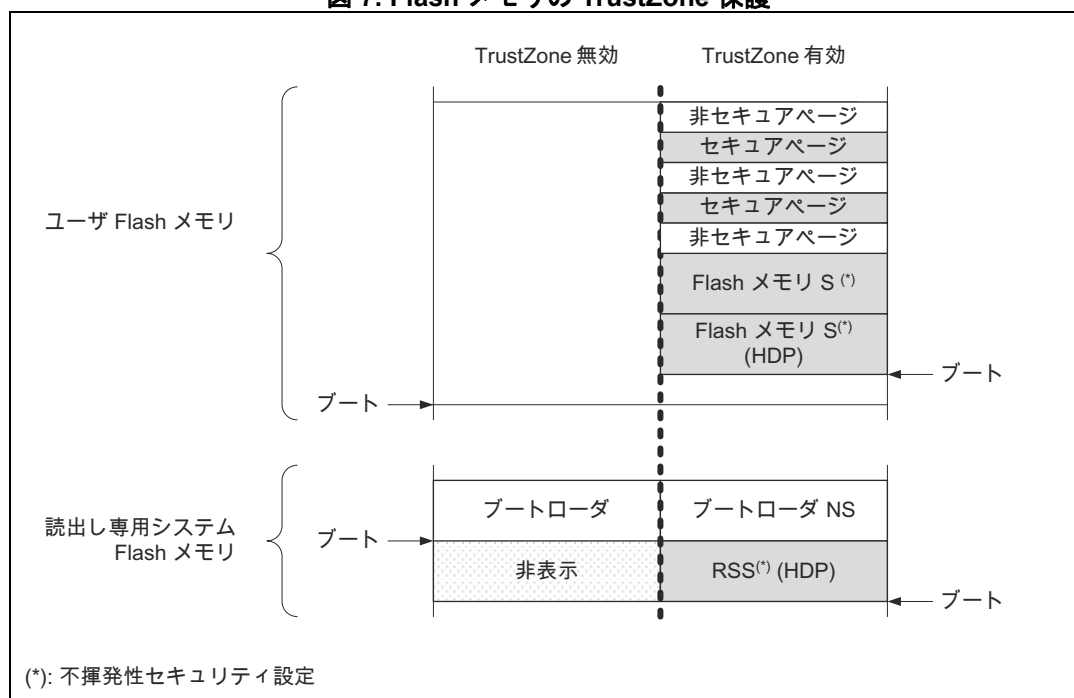
TrustZone セキュリティがオプションバイトを通じて有効なとき (TZEN = 0xB4)、リセット後はメモリ全体がセキュアです。アプリケーションは、図 7 に示されている以下の保護を使用できます。

- 不揮発性ユーザセキュア領域、不揮発性セキュアユーザオプションバイトで定義
 - ウォーターマークベースのセキュア専用領域 (x2)
 - セキュア非表示保護 (HDP) 領域、ブート後は厳格に非表示 (x2)
- 揮発性ユーザセキュアページ、揮発性セキュアレジスタで定義 (リセット後は失われる)
 - 非セキュアとして設定されたページ (外部のウォーターマークベースのセキュア専用領域など) は、ブロックベースの設定レジスタを使用してオンザフライでセキュアとして設定できます。

注： すべての領域は Flash メモリのページ単位に揃えられ、Cortex-M33 IDAU/SAU で非セキュアとしてタグが付けられているときにセキュアとして設定できます。この場合、CPU から Flash メモリへの非セキュアアクセスは拒否されます。

セキュア/非セキュアコードに対する消去またはプログラム操作は、セキュア/非セキュアページまたはメモリについてのみ使用可能です。少なくとも 1 つのページがセキュアの場合、Flash メモリはセキュアとみなされます。

図 7. Flash メモリの TrustZone 保護



上に示されているように、TrustZone が有効なとき (TZEN = 0xB4)、アプリケーションコードは、Flash メモリのウォーターマークベースのセキュア領域の一部である HDP 領域を使用できます。実際、アプリケーションが HDPx_ACCDIS ビットをセットすると、この HDP 領域でのデータの読み取り、書き込み、および命令フェッチは、次のシステムリセットまで拒否されます。

たとえば、セキュア Flash メモリの HDP 領域内のソフトウェアコードは一度だけ実行でき、この領域へのその後のアクセスは、次のシステムリセットまで拒否されます。また、アクティブな HDP 領域に属する Flash メモリのページは、それ以上消去できなくなります。

TrustZone が無効なとき (TZEN = 0xC3)、揮発性/不揮発性セキュア領域機能は無効化され、すべてのセキュアレジスタは RAZ/WI です。

詳細は、[セクション 7 : 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。

オンザフライ暗号化／復号 (OTFDEC)

TrustZone セキュリティが有効なとき (TZEN = 0xB4)、OTFDEC はセキュアアプリケーションによってのみ初期化できます。4 つの暗号化領域のそれぞれは、設定が確定されると、次のパワーオンリセットまで、書き込みロックされます。

注： アプリケーション (セキュアまたは非セキュア) は、ペリフェラルレジスタを読み取ることにより、各 OTFDEC 領域の初期化コンテキスト (キーの CRC を含む) を確認できます。

各 OTFDEC 内のキーレジスタは書き込み専用です。

この暗号化エンジンの詳細については、[セクション 3.10.3](#) を参照してください。

ダイレクトメモリアクセスコントローラ (GPDMAx)

DMA チャンネル x がセキュアとして定義されているとき (GPDMA_SECCFGR の SECx = 1)、セキュアアプリケーションは、GPDMA_CxTR1 の SSEC および DSEC ビットを使用して、ソースおよびデスティネーション転送を個別にセキュアまたは非セキュアとして設定できます。[表 7](#) に、各 DMA チャンネルで使用可能なこれらのセキュリティオプションの概要を示します。

表 7. DMA チャンネルの使用 (セキュリティ)⁽¹⁾

デスティネーションタイプ	セキュア DMA チャンネル x (SECx = 1)		非セキュア DMA チャンネル y (SECy = 0)	
	セキュアソース	非セキュアソース	セキュアソース	非セキュアソース
セキュアデスティネーション	OK	OK ⁽²⁾	転送はブロックされる	
非セキュアデスティネーション	OK ⁽³⁾	OK ⁽⁴⁾	転送はブロックされる	OK

1. 転送がブロックされると、転送は完了しますが、対応する書き込みは無視され、読み取りは 0 を返します。TZIC への不正アクセスイベントは、ソースまたはデスティネーションとして使用されるメモリ/ペリフェラルによって自動的にトリガされます。
2. ソースがメモリの場合、SSEC = 0 の場合のみ転送が可能であり、そうでない場合、転送はブロックされます。
3. デスティネーションがメモリの場合、DSEC = 0 の場合のみ転送が可能であり、そうでない場合、転送はブロックされます。
4. 転送がメモリ間の場合、SSEC = 0 かつ DSEC = 0 の場合のみ可能であり、そうでない場合はブロックされます。

チャンネルがセキュアとして設定されているとき：

- このチャンネルに割り当てられたレジスタ (GPDMA_SECCFGR、GPDMA_PRIVCFGR、および GPDMA_RCFGLOCKR を除く) は、0 として読み取られます。非セキュアアクセスの場合、書き込みは無視されます。セキュア不正アクセスイベントは、TZIC ペリフェラルに対してトリガされることもあります。
 - GPDMA_SECCFGR および GPDMA_RCFGLOCKR への書き込みはセキュアである必要があります。GPDMA_PRIVCFGR 内の各ビットについて、GPDMA_SECCFGR 内の対応するビットがセットされている場合、書き込みはセキュアでなければなりません。
- リンクリストモードでは、メモリからの次のリンクリストデータ構造のロードは、セキュア転送で行われます。
- 非セキュア状態に切り替えるとき、セキュアアプリケーションは、切り替えを行う前に、チャンネルをアボートするか、セキュアチャンネルが完了するまで待つ必要があります。

注： DMA セキュアチャネルは、TrustZone が無効なときには使用できません。チャネルが非セキュアとして設定されているとき、リンクリストモードでは、メモリからの次のリンクリストデータ構造のロードは、非セキュア転送で行われます。

詳細は、[セクション 16：汎用ダイレクト・メモリ・アクセス・コントローラ（GPDMA）](#)を参照してください。

電源制御（PWR）

TrustZone セキュリティが有効なとき（TZEN = 0xB4）、選択した PWR レジスタを PWR_SECCFGR を通じてセキュリティ保護して、以下の PWR 機能を保護できます。

- 低電力モードのセットアップ
- ウェイクアップ（WKUP）ピンの定義
- 電圧の検出と監視
- バックアップドメイン制御

他の PWR 設定ビットは、次のときにセキュアになります。

- システムクロック選択が RCC でセキュアなとき：電圧スケーリング（VOS）はセキュアになります。
- GPIO がセキュアとして設定されているとき：STANDBY モードのブルアップ/ブルダウン設定の対応するビットはセキュアになります。
- USB Type-C/USB Power Delivery インタフェース（UCPD）は、TZSC でセキュアとして設定されます。PWR_UCPDR レジスタはセキュアになります。

詳細については、[セクション 10：電源制御（PWR）](#)を参照してください。

セキュアクロックおよびリセット（RCC）

TrustZone セキュリティが有効で（TZEN = 0xB4）、RCC でセキュリティが有効なとき、ペリフェラルクロックおよびリセットの制御するビットは TrustZone 対応になります。

- ペリフェラルがセキュリティ保護可能で、TZSC でセキュアとしてプログラムされている場合、ペリフェラルクロックおよびリセットビットはセキュアになります。
- ペリフェラルが TrustZone 対応な場合、ペリフェラル内で少なくとも 1 つの機能がセキュアとして設定されると、ペリフェラルクロックおよびリセットビットはセキュアになります。

注： セキュリティ保護可能および TrustZone 対応ペリフェラルのリストについては、[セクション 3.6.4](#)を参照してください。

また、以下の設定は、RCC_SECCFGR を使用してセキュア専用に行えます。

- 外部クロック（HSE または LSE など）、内部オシレータ（HSI、CSI、または LSI など）
- メイン PLL および AHB プリスケータ
- システムクロックソース選択
- MCO クロック出力
- リセットフラグ
- 自動内部オシレータウェイクアップ設定

詳細については、[セクション 11：リセットおよびクロック制御（RCC）](#)を参照してください。

リアルタイム・クロック（RTC）

すべての TrustZone 対応ペリフェラルと同様、セキュア RTC レジスタに対する非セキュアな読み取り/書き込みアクセスは RAZ/WI です。また、RTC 不正アクセスイベントが TZIC で有効な場合、不正アクセスイベントを生成して、セキュア不正アクセス割込みをトリガします。

バックアップドメインのパワーオンリセット後、すべての RTC レジスタは、セキュアモードと非セキュアモードの両方で読み書きできます。セキュアブートコードは、RTC_SECCFGR を使用してセキュリティセットアップを変更し、レジスタアラーム A、アラーム B、ウェイクアップタイマ、およびタイムスタンプをセキュアまたは非セキュアにできます。

SEC ビットがセキュア専用 RTC_SECCFGR でセットされているとき：

- RTC レジスタへの書込みはセキュアモードでのみ可能です。
- RTC_SECCFGR、RTC_PRIVCFGR、RTC_MISR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER、および RTC_CALR の読出しは、セキュアモードと非セキュアモードで常に可能です。その他のすべての RTC レジスタは、セキュアモードでのみ読出しできます。

SEC がセキュア専用 RTC_SECCFGR でクリアされているときでも、以下の専用制御ビットをセットすることにより、セキュアモードでのアクセスを一部の RTC レジスタに制限することが可能です。INITSEC、CALSEC、TSSEC、WUTSEC、ALRASEC、および ALRBSEC。

注： RTC セキュリティ設定は、システムリセットの影響を受けません。

詳細は、[セクション 46：リアルタイムクロック（RTC）](#)を参照してください。

タンパおよびバックアップレジスタ（TAMP）

すべての TrustZone 対応ペリフェラルと同様、セキュア TAMP レジスタに対する非セキュアな読取り/書込みアクセスは RAZ/WI です。また、TAMP 不正アクセスイベントが TZIC で有効な場合、不正アクセスイベントを生成して、セキュア不正アクセス割込みをトリガします。

バックアップドメインのパワーオンリセット後、すべての TAMP レジスタは、セキュアモードと非セキュアモードの両方で読み書きできます。セキュアブートコードは、TAMP_SECCFGR レジスタを使用してこのセキュリティセットアップを変更し、一部のレジスタを必要に応じてセキュアまたは非セキュアにできます。

TAMP_SECCFGR で TAMPSEC がセットされているとき：

- TAMP レジスタへの書込みはセキュアモードでのみ可能です。バックアップレジスタには専用の書込み保護があります（下記を参照）。
- TAMP レジスタ（TAMP_SECCFGR、TAMP_PRIVCFGR、および TAMP_MISR を除く）の読出しは、アクセスが非セキュアの場合、0 を返します。バックアップレジスタには専用の読出し保護があります（下記を参照）。

アプリケーションは、以下のこともできます。

- TAMP_SECCFGR セキュアレジスタの CNT1SEC ビットをセットすることにより、TAMP_COUNTNTR レジスタを読出しおよび書込みセキュア専用に行えます。
- TAMP_SECCFGR の BKPRWSEC[7:0] および BKPWSEC[7:0] ビットフィールドを使用して設定された 3 つの保存ゾーンのうちの 2 つについて、バックアップレジスタのセキュリティを高めます。
 - 保護ゾーン 1 は、読出し非セキュア、書込み非セキュアです。
 - 保護ゾーン 2 は、読出し非セキュア、書込みセキュアです。
 - 保護ゾーン 3 は、読出しセキュア、書込みセキュアです。

注： TAMP セキュリティ設定は、システムリセットの影響を受けません。

詳細は、[セクション 47：タンパおよびバックアップレジスタ（TAMP）](#)を参照してください。

汎用 I/O (GPIO)

TrustZone セキュリティが有効なとき (TZEN = 0xB4)、GPIO ポートの各 I/O ピンは、GPIOx_SECCFGR レジスタによって個別にセキュアとして設定できます。セキュアアプリケーションのみが、GPIOx_SECCFGR レジスタに書き込むことができます。ブート後、各 I/O ピンはセキュアとして設定されます。

I/O ピンがセキュアとして設定されているとき、オルタネート機能 (AF)、モード選択 (MODE)、および I/O データの対応する設定ビットは、非セキュアアクセスの場合 RAZ/WI です。

セキュアペリフェラルによって管理される I/O との間で転送されるデータを保護するためにデータデジタルオルタネート機能を使用されるとき (入力/出力モード)、デバイスはペリフェラルと割り当てられた I/O の間のパスにセキュアオルタネート機能ゲートを追加します。

- ペリフェラルがセキュアな場合、データの入出力を可能にするには、I/O ピンもセキュアである必要があります。
- ペリフェラルがセキュアでない場合、I/O ピンの状態に関係なく、接続が許されます。

GPIO ポート周辺の TrustZone 対応ロジックは、オルタネート機能として使用され、表 8 に概要が示されています。

表 8. ペリフェラルと割り当てられた I/O 間のセキュアオルタネート機能

セキュリティ設定		オルタネート機能ロジック		コメント
ペリフェラル	割り当てられた I/O ピン	入力	出力	
セキュア	セキュア	I/O データ	ペリフェラルデータ	-
非セキュア				リセット解除設定
セキュア	非セキュア	0	0	-
非セキュア		I/O データ	ペリフェラルデータ	

アナログスイッチ付きのアナログ機能を使用されるとき、ペリフェラルが非セキュアであり、I/O がセキュアな場合、表 9 にリストされているペリフェラルへの接続はハードウェアによってブロックされます。

表 9. セキュア I/O に接続できない非セキュアペリフェラル機能

ペリフェラル	アナログ機能 ⁽¹⁾	入力	出力
ADC12	ADC12_INy (y = 1 ~ 17)	X	-

1. パッケージ上の信号/機能に対応する I/O を見つけるために使用されます (製品データシートを参照)。

最後に、GPIO とセキュリティに関して、表 10 に、TrustZone にリンクされたハードウェア保護を持たない I/O のリストを示します。リストされている信号 (入力または出力) は、I/O がセキュアとして設定され、関連するペリフェラルが非セキュアのときにはブロックされません。

たとえば、セキュアアプリケーションが PA4 を LPTIM2_CH1 として使用されるセキュアとして設定したとき、DAC が非セキュアの場合、データを PA4 に出力するようにプログラムできますが、セキュアアプリケーションが誤動作することがあります。

同様に、セキュアアプリケーションが PA0 を UART4_TX として使用されるセキュアとして設定したとき、TAMP が非セキュアの場合、USART 入力トラフィックを TAMP_IN 信号によってキャプチャするようにプログラムできます。

表 10 で説明されている各ケースについて、セキュアアプリケーションは、データの整合性や秘匿性への潜在的な影響が重大かどうかを判断することが重要です。たとえば、上記で説明されている USART 状況が許容不能な場合（セキュア USART でのデータ遷移は機密である）、セキュアアプリケーションは、セキュアアプリケーションによって使用されない場合でも、TAMP をセキュアとして設定する必要があります。

表 10. セキュア I/O に接続できる非セキュアペリフェラル機能

ペリフェラル	信号 ⁽¹⁾	入力	出力	ペリフェラルまたは機能をセキュアとして設定する方法
DAC	DAC1_OUTx (x = 1, 2)	-	X	GTZC1_TZSC_SECCFGR1 の DAC1SEC をセットします。
UCPD	UCPD1_CCx (x = 1, 2)	X	X	GTZC1_TZSC_SECCFGR1 の UCPD1SEC をセットします。
	UCPD1_DBx (x = 1, 2)	X	-	
TAMP	TAMP_INx (x = 1 ~ 8)	X	-	TAMP_SECCFGR の TAMPSEC をセットします。
	TAMP_OUTx (x = 1 ~ 8)	-	X	
RTC	RTC_OUTx (x = 1, 2)	-	X	RTC_SECCFGR の SEC をセットします。
	RTC_TS	X	-	RTC_SECCFGR の TSSEC をセットします。
PWR	WKUPx (x = 1 ~ 8)	X	-	PWR_SECCFGR の WUPxSEC をセットします。
RCC	LSCO	-	X	RCC_SECCFGR の LSESEC をセットします。
EXTI	EXTIx (x = 0 ~ 58)	X	-	EXTI_SECCFGR の SECx ビットをセットします。

1. パッケージ上の信号/機能に対応する I/O を見つけるには、製品データシートを参照してください。

詳細については、[セクション 13 : 汎用 I/O \(GPIO\)](#) を参照してください。

拡張割込みとイベントコントローラ (EXTI)

TrustZone セキュリティが有効なとき (TZEN = 0xB4)、EXTI はイベントレジスタのビットを非セキュアアクセスによる変更から保護できます。EXTI_SECCFGRx のレジスタビットにより、入力イベントごとに個別に保護を有効化できます。入力イベントがセキュアとして設定されているとき、セキュアアプリケーションのみが設定（セキュリティを含む）を変更でき、マスキングを変更でき、またはこの入力イベントのステータスをクリアできます。

EXTI_SECCFGR1 および EXTI_SECCFGR2 のセキュリティ設定は、EXTI_LOCKR リセット後にグローバルにロックすることができます。

詳細は、[セクション 23 : Octo-SPI インタフェース \(OCTOSPI\)](#) を参照してください。

システム設定、ブート、およびセキュリティ (SBS)

すべての TrustZone 対応ペリフェラルと同様、TrustZone セキュリティが有効なとき (TZEN = 0xB4)、セキュア SBS レジスタに対する非セキュアな読取り/書込みアクセスは RAZ/WI です。SBS 不正アクセスイベントが TZIC でマスクされてない場合、そのようなアクセスは不正アクセスイベントを生成して、セキュア不正アクセス割込みをトリガします。

詳細は、[セクション 14 : システム設定、ブート、およびセキュリティ \(SBS\)](#) を参照してください。

マイクロコントローラデバッグユニット (DBGMCU)

MCU デバッグコンポーネント (DBGMCU) は、デバッグによる以下のサポート機能を支援します。

- デバッグ時の低電力モードの動作
- デバッグ時のペリフェラルのフリーズ。I2C、IWDG、WWDG、タイマ、低電力タイマ、および GPDMA チャンネルに適用可能。

DBGCMU は TrustZone 対応ペリフェラルであり、表 11 に示されているように、制御レジスタへのアクセスを管理します。

表 11. TrustZone 対応 DBGMCU の非セキュアアクセス管理

デバッグプロファイル	ペリフェラルのステータス ⁽¹⁾	DBG_xx_STOP 制御ビット	
		書き込みアクセス	読出しアクセス
非セキュア浸入型 (SPIDEN = 0)	NS	はい (S ⁽²⁾ または NS)	はい (S または NS)
	S	なし (S または NS)	
セキュア浸入型 (SPIDEN = 1)	NS	はい (S または NS)	
	S	はい (S のみ)	

1. GTZC によって報告される TrustZone 対応ペリフェラルまたは DMA チャンネル。

2. デバッグからのセキュアアクセスは、デバイスで非セキュアアクセスに変換されます。

詳細については、[セクション 58.12: マイクロコントローラデバッグユニット \(DBGMCU\)](#) を参照してください。

3.6.6 TrustZone セキュリティの有効化

TrustZone は、すべての STM32H563/H573 および STM32H562 デバイスでデフォルトで無効です。PRODUCT_STATE が Open のとき、FLASH_OPTSR2_PRG の TZEN オプションビットをセットすることによって有効化できます。TZEN が無効から有効に (0xC3 から 0xB4 に) 変更されると、リセット後のデフォルトのセキュリティ状態は、常に次のようになります。

- CPU サブシステム
 - Cortex-M33 はセキュア状態でリセットを抜けるため、ブートアドレスはセキュアメモリ領域を指すはずです。
 - すべての割込みソースはセキュアです (NVIC で)。
 - IDAU/SAU 経由で CPU によって表示されるメモリマップは完全にセキュアです。
- 内蔵 Flash メモリ
 - Flash メモリの不揮発性セキュア領域 (HDP ゾーン付き) は、不揮発性レジスタ FLASH_SECWMxR (x = 1, 2) で定義されます。デフォルトのセキュアオプションバイトの設定は、すべてユーザ Flash セキュアであり、HDP 領域は定義されていません。
 - Flash メモリのブロックベースの揮発性セキュリティ属性は非セキュアです。
- 内蔵 SRAM
 - すべての SRAM は、GTZC/MPCBB で定義されるようにセキュアです ([セクション 3.6.4](#) を参照)。セキュアブートコードは、このセキュリティ設定を変更して、ブロックをセキュアまたは非セキュアにできます。
- 外部メモリ
 - FMC および OCTOSPI に接続されたすべてのメモリデバイスは、GTZC/MPCWM で定義されるようにセキュアです ([セクション 3.6.4](#) を参照)。セキュアブートコードは、このセキュリティ設定を変更して、コンポーネントをセキュアまたは非セキュアにできます。
- すべての GPIO はセキュアです。
- すべての GPDMA チャンネルは非セキュアです。
- バックアップレジスタは非セキュアです。
- ペリフェラルと GTZC :
 - セキュリティ保護可能なペリフェラルは非セキュアであり、非特権です。

- TrustZone 対応ペリフェラルは非セキュアであり、セキュア設定レジスタはセキュアです。
- GTZC/TZIC におけるすべての不正アクセス割込みは無効です。

注： セキュリティ保護可能および TrustZone 対応ペリフェラルのリストについては、[セクション 3.6.4](#) を参照してください。

3.6.7 TrustZone セキュリティの無効化

TrustZone が有効になると、オープンへの PRODUCT_STATE リグレーションのときにのみ無効化できます。

注： そのような PRODUCT_STATE リグレーションは内蔵メモリ（SRAM2、Flash）の消去と、OTFDEC およびすべての暗号化エンジンを含め、すべてのペリフェラルのリセットをトリガします。

TrustZone が無効化されると、[セクション 3.6](#) で言及されたほとんどの機能は使用できなくなります。

- 内蔵 Flash メモリの不揮発性セキュア領域は、HDP 領域も含めて無効になります。
- NVIC のみが非セキュア割込みを管理します。
- TrustZone 対応ペリフェラルのすべてのセキュアレジスタは RAZ/WI です。

注： TrustZone が無効化されても、特権を使用するリソース分離は使用可能なままです（詳細については、[セクション 3.7.2](#) を参照してください）。

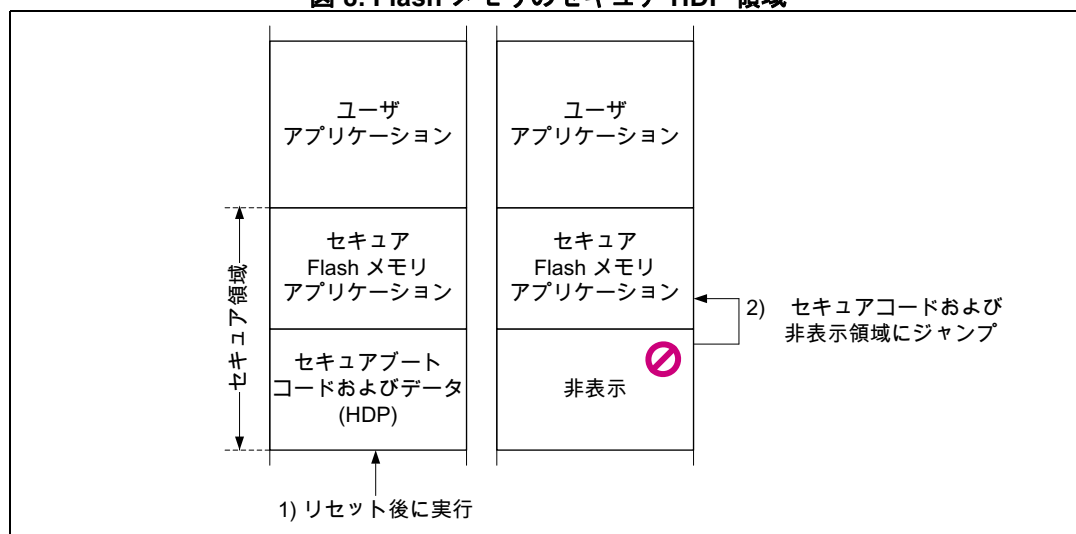
3.7 その他のリソース分離

これらは、TrustZone テクノロジーの上にもう 1 つの分離レベルを提供するハードウェアメカニズムです。

3.7.1 セキュア非表示保護（HDP）を使用した時間的分離

TrustZone セキュリティが有効なとき（TZEN = 0xB4）、内蔵 Flash メモリで各バンク（8 KB のページ単位）のウォーターマークセキュア領域ごとに HDP 領域を定義できます。この HDP 領域で実行されるコードは、関連データおよびキーとともに、ブート後は次のシステムリセットまで非表示にできます。非表示保護の原理を [図 8](#) に示します。

図 8. Flash メモリのセキュア HDP 領域



ユーザ Flash メモリの HDP 領域の有効化は、HDPL1 に関連します。HDPL \geq 2 になるとすぐに、データの読出し、書込み、および命令フェッチ (FLASH_HDPxR_PRG オプションバイトの HDPx_PSTRT および HDPx_PEND によって定義された領域に対するもの) は、次のデバイスリセットまで拒否されます。

FLASH_HDPEXTR Flash メモリレジスタにより、HDPx 領域の終わりを拡張できます (アプリケーションによって動的に)。

注： バンク消去に書き込み保護領域 (WRP または HDP 領域) が含まれていると、バンク消去はアボートされます。

3.7.2 Cortex 特権モードを使用したリソース分離

[セクション 3.6](#) で説明されている TrustZone 分離と並行して、ハードウェアおよびソフトウェアリソースをパーティショニングして、Cortex 特権モードで実行しているソフトウェアに制限できます。

このハードウェア分離テクノロジーは、TrustZone が無効なときでも (TZEN = 0XC3) 使用できるため、重要なコードやデータを、より公開されることが多い非特権コードからの意図的または偶発的のタンパに対して保護できます。

MPU を使用したメモリおよびペリフェラルの特権割当て

Cortex-M33 MPU は統合メモリを非セキュアの 8 つの領域と TrustZone の 12 の領域に分割し、これらは、それぞれ 32 バイトの倍数に揃えられています。各メモリ領域は、非特権ソフトウェアによって不適切にアクセスされたときにエラーを生成するようにプログラムできます。

GTZC を使用したメモリおよびペリフェラルの特権割当て

Cortex-M33 マスタの場合、MPU によって提供される簡易分離を補足するために、GTZC は、ペリフェラルと選択されたメモリについて、特権タスクと非特権タスク間の分離を柔軟に補強します。

Cortex-M33 以外のマスタについては、GTZC はそれらを非特権イニシエータとして割り当てて、特権として定義されたリソースをこのマスタから自動的に保護します。

- TZSC によるペリフェラルのセキュリティ保護 (特権のみ)

デバイスでは、ペリフェラルは、GTZC を通じてセキュリティ保護可能な特権専用か、ネイティブに特権対応かのいずれかです。

- セキュリティ保護可能な特権専用ペリフェラルまたはメモリは、TZSC によって制御される AHB/APB ファイアウォールゲートによって保護されます。
- 特権対応ペリフェラルまたはメモリは、AHB または APB 相互接続に直接接続されて、特定の動作を実装します (レジスタのサブセットまたはメモリ領域が特権専用など)。

そのようなペリフェラルが GTZC により特権専用になれ、相互接続のマスタである場合 (SDMMC)、特権トランザクションを自動的に発行します。GPDMA1 や GPDMA2 などの特権対応マスタは、GTZC とは無関係に、内部特権モードに従って、AHB 相互接続で特権信号を駆動します。

セキュリティ保護可能なペリフェラルのリストは、[セクション 5: グローバル TrustZone@コントローラ \(GTZC\)](#) にあります。

- TZSC および MPCBB によるメモリのセキュリティ保護 (特権専用)

GTZC の TZSC ロジックは、セキュリティ保護可能な外部メモリのすべての特権レベルを管理し、[セクション 3.6.4](#) で定義されているように、MPCWM リソースをプログラムする機能を備えています。

同様に、GTZC の TZSC ロジックは、内蔵 SRAM ブロックの特権レベルを設定し、[セクション 3.6.4](#) で定義されている MPCBB リソースをプログラムする機能を備えています。

- エラー管理（特権専用）
 - 特権リソースへのアクセスを試みる非特権トランザクションはすべて不正アクセスとみなされます。不正非特権読出しおよび書込みアクセスについては、不正アクセスイベントは生成されません。
 - アドレス指定されたリソースはサイレントフェイル動作に従い、読出しにはすべてゼロのデータを返し、書込みはすべて無視します。
 - 不正非特権アクセスが発生すると、バスエラーは生成されませんが、特権メモリまたはペリフェラルレジスタにアクセスする命令フェッチの場合は例外です。

特権対応ペリフェラルのセキュリティの管理

TrustZone 対応ペリフェラルは、特権専用アクセスモードも実装します。TrustZone が無効 (TZEN = 0xC3) の場合でも、特権保護は有効です。

- 内蔵 Flash メモリ

デフォルトでは、すべてのレジスタは、特権モードと非特権モードの両方で読出しまたはプログラムが可能です。

FLASH_PRIVCFGR でセキュア特権ビット SPRIV がセットされているとき、セキュアレジスタの読出しおよび書込みは特権モードでのみ可能です。このビットへの書込みアクセスは、TrustZone が無効なときには (TZEN = 0xC3)、無視されます。

FLASH_PRIVCFGR の非セキュア特権ビット NSPRIV がセットされているときには、Flash メモリの非セキュアレジスタの読出しおよび書込みは、特権モードでのみ可能です。

特権保護に関して、デバイスは以下の機能を備えています。

 - システム Flash メモリは、特権モードと非特権モードの両方でアクセスできます。
 - 各ウォーターマークベースのセキュア領域は、セキュア HDP 領域も含め、セキュア特権モードとセキュア非特権モードでアクセス可能です（該当する場合）。
 - 内蔵 Flash メモリの各8 KB ページは、ブロックベースの特権設定レジスタ FLASH_PRIV1BBRx および FLASH_PRIV2BBRx を使用して、オンザフライで特権専用としてプログラムできます。非特権ページには、特権アクセスまたは非特権アクセスによってアクセスできます。

注： ページを特権から非特権に切り替えても、内容は消去されません。

該当する場合、消去またはプログラム操作は、特権コードに対しては常に使用可能であり、非特権コードに対しては、非特権ページまたは非特権メモリについてのみ使用可能です。

- オンザフライ暗号化／復号 (OTFDEC)

OTFDEC_PRIVCFGR の特権ビット PRIV がセットされているとき、特権アプリケーションのみが OTFDEC を初期化できます。

注： OTFDEC_PRIVCFGR は、特権コードと非特権コードの両方によって読み出すことができます。

- ダイレクトメモリアccessコントローラ (GPDMAx)

DMA チャンネル x が特権として定義されているとき (GPDMA_PRIVCFGR の PRIVx = 1)、特権/非特権ソースまたはデスティネーションにアクセスするときには、特別な規則が適用されます（表 12 を参照）。

表 12. DMA チャンネルの使用（特権）

転送先	特権 DMA チャンネル x (PRIVx = 1)		非特権 DMA チャンネル y (PRIVy = 0)	
	特権ソース	非特権ソース	特権ソース	非特権ソース
特権	OK		転送はブロックされる ⁽¹⁾	
非特権			転送はブロックされる	OK

1. 転送がブロックされると、転送は完了しますが、対応する書込みは無視され、読取りは 0 を返します。

- 電源制御 (PWR)

デフォルトでは、パワーオンまたはシステムリセット後、PWR_PRIVCFGR 以外のすべての PWR レジスタは、特権モードと非特権モードの両方で読み書きできます。

PWR_PRIVCFGR のセキュア特権ビット SPRIV がセットされているとき、PWR セキュリティ保護可能レジスタの読み書きは、特権モードでのみ可能です。このビットへの書き込みアクセスは、TrustZone が無効なときには (TZEN = 0xC3)、無視されます。

PWR_PRIVCFGR の非セキュア特権ビット NSPRIV がセットされているときには、PWR 非セキュアレジスタの読出しおよび書き込みは、特権モードでのみ可能です。

詳細については、[セクション 10：電源制御 \(PWR\)](#) を参照してください。
- セキュアクロックおよびリセット (RCC)

デフォルトでは、パワーオンまたはシステムリセット後、RCC_PRIVCFGR 以外のすべての RCC レジスタは、特権モードと非特権モードの両方で読み書きできます。

RCC_PRIVCFGR のセキュア特権ビット SPRIV がセットされているとき、RCC セキュリティ保護可能ビットの読み書きは、特権モードでのみ可能です。このビットへの書き込みアクセスは、TrustZone が無効なときには (TZEN = 0xC3)、無視されます。

RCC_PRIVCFGR の非セキュア特権ビット NSPRIV がセットされているとき、RCC 非セキュアビットの読出しおよび書き込みは、特権モードでのみ可能です。

詳細については、[セクション 11：リセットおよびクロック制御 \(RCC\)](#) を参照してください。
- リアルタイム・クロック (RTC)

デフォルトでは、バックアップドメインリセット後、RTC_PRIVCFGR 以外のすべての RTC レジスタは、特権モードと非特権モードの両方で読み書きできます。

PRIV ビットが特権専用 RTC_PRIVCFGR でセットされているとき：

 - RTC レジスタへの書き込みは特権モードでのみ可能です。
 - RTC_SECCFGR、RTC_PRIVCFGR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER、および RTC_CALR の読出しは、特権モードと非特権モードで常に可能です。

その他すべての RTC レジスタは、特権モードでのみ読み出せます。

PRIV ビットが特権専用 RTC_PRIVCFGR レジスタ でクリアされているときでも、以下の専用制御ビットをセットすることにより、アクセスを一部の RTC レジスタへの特権モードに制限することが可能です。INITPRIV、CALPRIV、TSPRIV、WUTPRIV、ALRAPRV、または ALRBPRIV。

詳細については、[セクション 46：リアルタイムクロック \(RTC\)](#) 章 を参照してください。
- タンパおよびバックアップレジスタ (TAMP)

デフォルトでは、バックアップドメインリセット後、TAMP_PRIVCFGR 以外のすべての TAMP レジスタは、特権モードと非特権モードの両方で読み書きできます。

PRIV ビットが特権専用 TAMP_PRIVCFGR でセットされているとき：

 - TAMP レジスタへの書き込みは、独自の保護設定を持つバックアップレジスタと単調カウンタを除き、特権モードでのみ可能です。
 - TAMP_SECCFGR または TAMP_PRIVCFGR の読出しは、特権モードと非特権モードで常に可能です。他の TAMP レジスタはすべて、独自の保護設定を持つバックアップレジスタと単調カウンタを除き、特権モードでのみ読み出すことができます。

アプリケーションは、以下のこともできます。

 - TAMP_PRIVCFGR の CNTPRIV ビットをセットすることにより、TAMP_COUNT1R レジスタの読み書きを特権専用に行えます。

- TAMP_PRIVCFGR の BKPRWPRIV および BKPWPRIV ビットを使用して、バックアップレジスタの 3 つの保護ゾーンのうちの 2 つについてセキュリティを高めます。
 - 保護ゾーン 1 を読出し特権、書込み特権にします。
 - 保護ゾーン 2 を読出し特権または非特権、書込み特権にします。
 - 保護ゾーン 3 は、常に読み書き特権または非特権です。
- 汎用 I/O (GPIO)

すべての GPIO レジスタは、セキュリティ状態 (セキュアまたは非セキュア) に関係なく、特権および非特権アクセスで読み書きできます。
- 拡張割込みとイベントコントローラ (EXTI)

EXTI ペリフェラルは、イベントレジスタビットを非特権アクセスによる変更から保護できます。保護は、特権専用 EXTI_PRIVCFGRx レジスタのレジスタビットを介して、入力イベントごとに個別に有効化されます。入力イベントが特権として設定されているとき、特権アプリケーションのみが設定 (該当する場合はセキュリティを含む) を変更でき、この入力イベントのマスクを変更したり、ステータスをクリアしたりできます。

EXTI_PRIVCFGR1 および EXTI_PRIVCFGR2 のセキュリティ設定は、EXTI_LOCKR リセット後にグローバルにロックすることができます。

詳細は、[セクション 18 : 拡張割込み/イベントコントローラ \(EXTI\)](#) を参照してください。
- システム設定ブートおよびセキュリティ (SBS)

すべての SBS レジスタは、特権モードと非特権モードの両方で読み書きできますが、以下は例外です。

 - SBS_SECCFGR レジスタの FPUSEC ビット (特権専用)
 - CPU 設定の SBS レジスタ: SBS_CSLOCKR、SBS_FPUIMR、および SBS_CNSLOCKR

詳細は、[セクション 14: システム設定、ブート、およびセキュリティ \(SBS\)](#) を参照してください。

3.8 セキュアな実行

ソフトウェアとハードウェアの特別な機能の組み合わせにより、デバイスは、プログラムのエラー、ネットワークアクセスを通じたソフトウェア攻撃、またはコード実行をタンパするローカルな試みに対して、機能の正常な動作を確保します。

このセクションでは、セキュアな実行のために特に設計されたハードウェアについて説明します。

3.8.1 メモリ保護ユニット (MPU)

Cortex-M33 にはメモリ保護ユニット (MPU) が含まれ、これは、以下のパラメータの 1 つ以上に基づいて、各メモリ領域 (ペリフェラルにマップされたものも含む) への読出しおよび書込みアクセスを制限できます。

- Cortex-M33 動作モード (特権、非特権)
- データ/命令フェッチ

非セキュアおよびセキュア MPU のメモリマップとプログラミングにより、メモリは複数の領域に分割されます (非セキュアの場合は最大 8、TrustZone の場合は最大 12)。セキュア MPU は、TrustZone が有効なときにのみ使用できます。

3.8.2 内蔵 Flash メモリ書き込み保護

内蔵 Flash メモリ書き込み保護 (WRP) は、内蔵 Flash メモリユーザ領域の特別なセクションへの不正または望ましくない書き込み/消去を防止します (システム領域は永続的に書き込み保護されています)。

書き込み保護領域はオプションバイトによって開始アドレスと終了アドレスを書き込むことで定義されます。各バンクに 2 つの書き込み保護領域を 32 KB のページ単位で定義できます。

WRP 領域は、オプションバイトの変更によって変更できますが、対応する FLASH_WRPSTN_xR の UNLOCK オプションビットがクリアされている (ROM エミュレーションを意味します) 場合を除きません。UNLOCK は、PRODUCT_STATE=Open にリグレーションするときのみセットできます。

注： バンク消去に書き込み保護領域 (WRP または HDP 領域) が含まれていると、バンク消去はアボートされます。

3.8.3 タンパの検出と対応

原則

デバイスは、温度、電圧、および周波数攻撃に対する重要なセキュリティアセットのアクティブ保護を含んでいて、次のような機能を備えています。

- タンパ検出時のデバイスシークレットの消去
- 次のような、CPU と関連のセキュリティペリフェラルの安全な実行の保証の向上。
 - 範囲外の電圧 (例: V_{BAT} 、 V_{DDA})、温度、およびクロッキング (LSE) の検出
 - 内部オシレータ LSI によってクロックされるセキュリティウォッチドッグ IWDG
 - 内部オシレータ HSI をシステムクロックとして選択可能
- 電源装置の保護
 - V_{DD} または V_{BAT} によって自動的に電力供給される RTC/TAMP ドメイン

詳細は、[セクション 47：タンパおよびバックアップレジスタ \(TAMP\)](#) を参照してください。

タンパ検出ソース

デバイスは 8 つのアクティブ入出力ピンをサポートしているため、4 つの独立したアクティブタンパメッシュ、または同じ出力ピンが複数の入力ピンで共有される場合は最大 7 つのメッシュが可能です (合計 8 つのアクティブタンパ I/O)。アクティブタンパボールは、パッケージの中心にマップされ、POS 市場で使用できます (WLCSP80 など)。

アクティブピンは LSE によってクロックされ、さまざまなシステム動作モード (Run、Sleep、Stop、または Standby) と VBAT モードで機能します。タンパピンのリストとさまざまな電力モードでの可用性については、「モードによる TAMP ピンの機能」を参照してください。

検出時間はプログラム可能であり、デジタルフィルタリングが使用可能です (4 つの連続した比較サンプルのうち、2 つが誤った比較になると、タンパがトリガされます)。

注： タンパイベントが発生すると、タイムスタンプが自動的に生成されます。

内部タンパソースを [表 13](#) に示します。

表 13. TAMP の内部タンパ

タンパ入力	TAMP_CR3 の NOER ビット番号	タンパソース
itamp1	0	バックアップドメイン電圧連続監視、VBAT モードで機能
itamp2	1	温度監視、VBAT モードで機能
itamp3	2	LSE 監視 ⁽¹⁾ 、VBAT モードで機能
itamp4	3	HSE 監視
itamp5	4	RTC カレンダーオーバーフロー (rtc_calovf)
itamp6	5	JTAG/SWD アクセス
itamp7、12、 13	6、11、12	電圧監視 (V _{CORE} 、V _{REF+})、ADC アナログウォッチドッグを介して
itamp8	7	単調カウンタオーバーフロー (内部生成)
itamp9	8	暗号化ペリフェラルのフォールト生成 (SAES、PKA、AES、RNG)
itamp11	10	IWDG タイムアウトおよび潜在的タンパ (少なくとも 1 つの有効タンパフラグがセットされると、IWDG リセット)
itamp15	14	システムフォールト検出

1. LSE の周波数過不足検出 (> 2 MHz)、グリッチフィルタ (> 2 MHz)。

タンパへの対応

デバイス内のタンパの各ソースは、以下のイベントをトリガするように設定できます。

- デバイスを Stop および Standby モードからウェイクアップできる割込みを生成 (TAMP_CR2 レジスタの TAMPxMSK ビットを参照)。
- 低電力タイマのハードウェアトリガを生成。
- 対応する TAMPxNOER ビットが TAMP_CR2 (タンパピンの場合) または TAMP_CR3 (内部タンパの場合) でクリアされている場合、デバイスシークレットを消去。消去可能なシークレットは、次のとおりです。
 - バックアップレジスタ (x32)、AES、HASH、および OTFDEC に保存された対称鍵 (暗号化 Flash メモリ領域は 0 として読み出されます)。
 - PKA SRAM に保存された非対称鍵、V_{DD}が存在するときに消去。
 - SRAM2 および CPU 命令キャッシュメモリに保存されたその他のシークレット (SRAM2 は V_{DD} が存在するときに消去される)。
 - SAES の DHUK を導出するために使用される不揮発性情報は、SRAM2 の完全な消去の場合を除き、0 にされます。
 - 4 KB のバックアップ SRAM (設定ビットに依存)、V_{DD}が存在するときに消去

これらすべてのシークレットに対するソフトウェアからの読み出し/書き込みアクセスは、TAMP_CR2 の BKBLOCK ビットをセットすることによってブロックできます。デバイスシークレットへのアクセスは、BKBLOCK がクリアされ、有効なタンパソースのいずれについてもタンパフラグがセットされていないときのみ可能です。

V_{DD}が存在しない場合、V_{DD}が存在するときに消去されるシークレットは、次のV_{DD}パワーオン時のみ消去されます。

注： デバイスシークレットの消去は、TAMP_CR2 の BKERASE ビットをセットすることによって、または [セクション 3.11.2](#) で定義されている PRODUCT_STATE リグレーションを実行することによってトリガされます。

デバイスシークレットは、システムリセットや、デバイスが Standby モードからウェイクアップしたときにもリセットされません。

ソフトウェアフィルタリングメカニズム

各タンパソースは、即時消去を開始しないように設定できます。そのためには、TAMP_CR2（外部タンパピンの場合）または TAMP_CR3（内部タンパの場合）の対応する TAMPxNOER ビットをセットします。

そのような状況ではタンパフラグが立てられると、すべてのタンパフラグがクリアされるまで、以下のシークレットへのアクセスはブロックされます。

- SAES の DHUK：ダミー値に固定
- バックアップレジスタ、バックアップ SRAM、SRAM2：0 として読出し、書込みは無視
- AES、SAES、および HASH ペリフェラル：RCC によって自動的にリセット
- PKA ペリフェラル：リセットされ、メモリ使用はブロック（PKA は使用不可）

タンパイブントの通知を受けたアプリケーションが状況を分析した場合、次の 2 つのケースがあります。

- アプリケーションはソフトウェアコマンドによってシークレットの消去を開始します（確認されたタンパ）。
- アプリケーションはフラグをクリアして、シークレットのブロックを解除します（偽タンパ）。

注： タンパソフトウェアがそのようなタンパフラグに反応できなかった場合、IWDG リセットによりシークレットの自動消去がトリガされます。

タンパ検出と低電力モード

低電力モードによるタンパ検出への影響を [表 14](#) に示します。

表 14. 低消費電力モードが TAMP に与える影響

モード	説明
SLEEP	タンパ検出機能への影響はありません。 TAMP 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	タンパ検出機能への影響はありませんが、フィルタリングによるレベル検出とアクティブタンパモードは、クロックソースが LSE または LSI のときのみアクティブです。 タンパイブントによって、デバイスは STOP モードを終了します。
STANDBY	タンパ検出機能への影響はありませんが、フィルタリングによるレベル検出とアクティブタンパモードは、クロックソースが LSE または LSI のときのみアクティブです。 タンパイブントによって、デバイスは STANDBY モードを終了します。

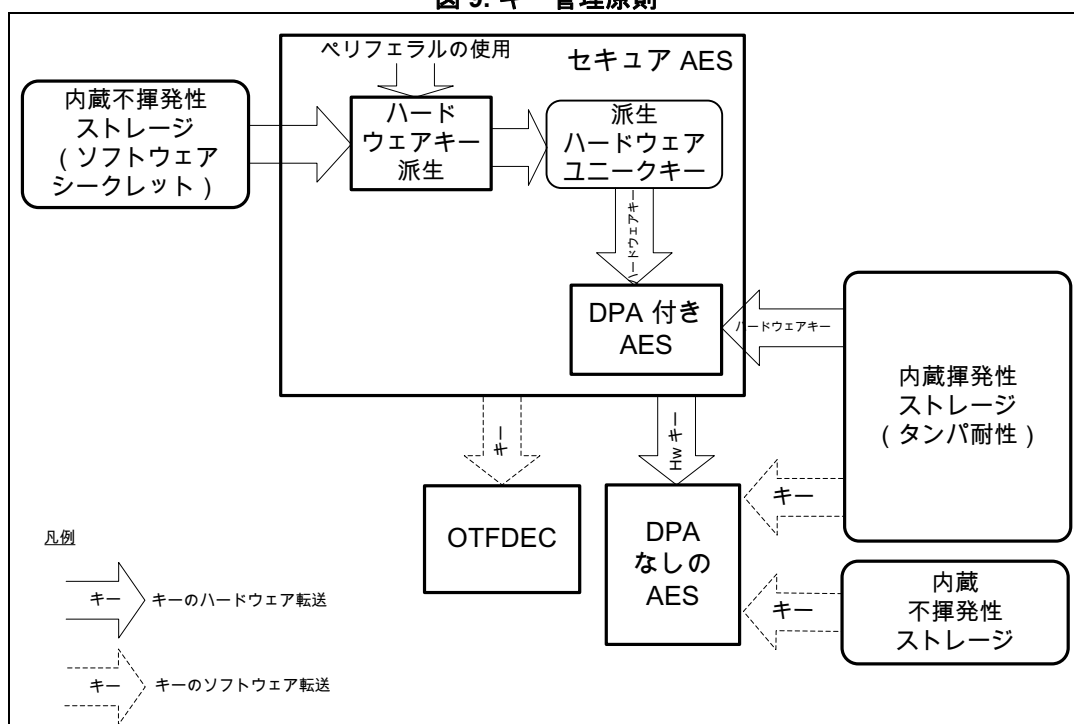
3.9 セキュアストレージ

セキュリティシステムの重要な機能は、ロングタームキーの保存、保護、およびプロビジョニングの方法です。そのようなキーは、一般に、ブートイメージのロードや重要なユーザデータの処理に使用されます。

図 9 は、キー管理サービスアプリケーションが AES エンジンを使用して、たとえば、外部イメージ復号キーを計算する方法を示しています。不揮発性キーは、Flash メモリのアクセス制御を含む専用領域に保存されますが（[セクション 3.7.1](#) を参照）、揮発性キーストレージは、バッテリー駆動のタンパ保護された SRAM または TrustZone 対応 TAMP のレジスタで構成されます。

図 9 は、ソフトウェアによって操作されるキー（OTFDEC キーなど）や、ハードウェアによってのみ管理されるキー（DHUK など）も示しています。これらのハードウェアキーの詳細については、[セクション 3.9.1](#) を参照してください。

図 9. キー管理原則



タンパ保護の詳細については [セクション 3.8.3](#) を、TAMP TrustZone 機能については [セクション 3.6.5](#) を参照してください。

3.9.1 ハードウェア秘密キー管理

前の図に示されているように、デバイスはハードウェア秘密キーを使用した、アプリケーションキーの保護の向上を提案します。これらの AES キーは、クリアテキスト（暗号化されていない）で公開しなくても、アプリケーションにとって使用可能にできます。タンパが発生した場合は、そのようなキーを直ちに使用不可にすることもできます。

ハードウェア秘密キーには 3 つの異なるソースがあります。

- DHUK : Flash メモリ内の 256 ビットの非揮発性、デバイス一意シークレットに基づいて導出されたキー。Flash メモリは、製品製造時にプロビジョニングされた値を提供します (RHUK と呼ばれます)。DHUK キーの生成では、TrustZone の状態、OBK-HDPLx (時間的分離カウンタ)、

EPOCH (アンチリプレイ保護が可能なリグレーションカウンタ)、およびキー使用状態 (KMOD) が考慮されます。

- BHK : TAMP のタンパ耐性の揮発性ストレージに保存された 256 ビットのアプリケーションキー。
このキーは、ブート時に書き込まれ、その後は次のリセットまで、アプリケーションに対して読み書きがロックされます。
- XORK : BHK と DHUK の XOR の結果。

これらのキーは、次のモードで使用できます。

- 通常のキーとして、書き込み専用キーレジスタにロードして (ソフトウェアキーモード)
- 別のキーの暗号化/復号キーとして、DPA 耐性 SAES で使用 (ラップキーモード)
- 別のキーの暗号化/復号キーとして、高速な AES エンジンで使用 (共有キーモード)

3.9.2 ユニーク ID

デバイスは、各デバイスに一意な 96 ビットの ID を保存します ([セクション 59.1 : ユニークデバイス ID レジスタ \(96 ビット\)](#) を参照)。

アプリケーションサービスは、この一意 ID キーを使用して、クラウドネットワーク製品を識別したり、偽造デバイスまたはクローンが、信頼できないデータをネットワークに注入するのを困難にしたりできます。

または、256 ビットのデバイス一意キー (DHUK) を使用することもできます ([セクション 3.9.1](#) を参照)。

3.10 暗号化エンジン

デバイスは最新の暗号アルゴリズムを実装しています。これらは、米国の NIST、ドイツの BSI、フランスの ANSSI など、各国の安全保障機関によって推奨されているキー長とコンピューティング保護が特徴です。これらのアルゴリズムは、プライバシー、認証、整合性、エントロピー、および身元確認のサポートに使用されます。

内蔵暗号化エンジンは、重要な暗号機能の実装における弱点を減らし、たとえば、弱い暗号アルゴリズムやキー長の使用を防止します。また、暗号操作を実行する際の処理時間の短縮と電力消費の低減も可能にし、これらの計算負荷を Cortex-M33 から取り除きます。特に、非対称暗号の場合はそうです。

製品認定の目的から、ST は、これらのセキュリティ機能の実装方法と検証方法に関する認定済みデバイス情報を提供できます。

暗号化エンジンの処理時間の詳細については、リファレンスマニュアルの該当セクションを参照してください。

3.10.1 暗号化エンジンの機能

[表 15](#) に、デバイスで使用可能な高速化暗号操作を示します。2 つの AES アクセラレータが使用可能です (どちらもセキュアアプリケーション専用に予約できます)。

注 : ファームウェアを使用して、追加の操作を追加できます。PKA は、非対称暗号操作を加速できます (キーペアの生成、ECC スカラー乗算、曲線上の点の確認など)。詳細は、[セクション 36 : 公開鍵アクセラレータ \(PKA\)](#) を参照してください。

表 15. 加速された暗号操作

操作	アルゴリズム	仕様	キーの長さ (ビット)	モード
エントロピーの取得	RNG	NIST SP800-90B ⁽¹⁾	N/A	並列実行するソフトウェアおよびハードウェア ⁽²⁾ モード
暗号化、復号	AES	FIPS PUB 197 NIST SP800-38A	128, 256	ECB、CBC、CTR ⁽³⁾
認証済みの暗号化または復号		NIST SP800-38C NIST SP800-38D	128, 256	GCM、CCM
暗号ベースのメッセージ認証コード		NIST SP800-38D	128, 256	GMAC
チェックサム	SHA-1	FIPS PUB 180-4	N/A	ダイジェスト 160 ビット
暗号化ハッシュ	SHA-2		-	SHA-224, SHA-256 SHA2-384, SHA2-512
メッセージ認証のための鍵付きハッシング	HMAC	FIPS PUB 198-1 IETF RFC 2104	ショート、 ロング (> 64 バイト)	-
暗号化/復号キーペア生成 ⁽⁴⁾	RSA	IETF RFC 8017 NIST SP800-56B	最大 4160	RSAES-OAEP
ハッシング付き署名 ⁽⁴⁾	RSA	IETF RFC 8017 FIPS PUB 186-4	最大 4160	PKCS1-v1_5、PSS
署名の検証	ECDSA	ANSI X9.62 IETF RFC 7027 FIPS PUB 186-4	最大 640	詳細については、PKA セクションの「ECC 操作でサポートされる曲線群」の表を参照。
鍵の合意	ECDH	ANSI X9.42		

1. STMicroelectronics レビュードキュメントを使用して認定可能

2. 専用ハードウェアバスを使用して、SAES および PKA に乱数を分散

3. ECB および CBC 連鎖モードをサイドチャネルや SAES のタイミング攻撃から保護（[セクション 3.10.2](#) を参照）。

4. 私有鍵暗号をサイドチャネルおよびタイミング攻撃から保護。

注： 2 値曲線、エドワーズ曲線、および Curve25519 は、PKA ではサポートされていません。

3.10.2 セキュア AES コプロセッサ (SAES)

デバイスは、追加のオンチップハードウェア AES 暗号化および復号エンジンを備え、電力および電磁的サイドチャネル攻撃に対する対策と緩和を実装します。

システムクロックによってクロックされるため、SAES は dpa 耐性ハードウェアアクセラレータのパフォーマンスを非常に良好に保ちます。SAES エンジンには、電子コードブック (ECB)、暗号ブロック連鎖 (CBC)、(CTR)、(GCM)、(CCM)、(GMAC) モードの 128 ビットまたは 256 ビットキーをサポートします。

[セクション 3.9](#) に示されているように、SAES は機密情報の超セキュアなオンチップストレージに使用できます。セキュア専用にすることもできます。

詳細については、[セクション 34 : セキュアな AES コプロセッサ \(SAES\)](#) を参照してください。

3.10.3 オンザフライ復号エンジン (OTFDEC)

OTFDEC TrustZone 対応ペリフェラルは、外部 Flash メモリに保存され、OCTOSPI を通じて接続される暗号化されたイメージのオンザフライ復号を可能にします。標準の NOR Flash メモリが使用される場合、復号プロセスによる追加のサイクルオーバーヘッドはほとんどありません。OTFDEC は、デバイス上の Flash メモリイメージを暗号化するためにも使用できます（デバイス一意秘密キーによる暗号化など）。

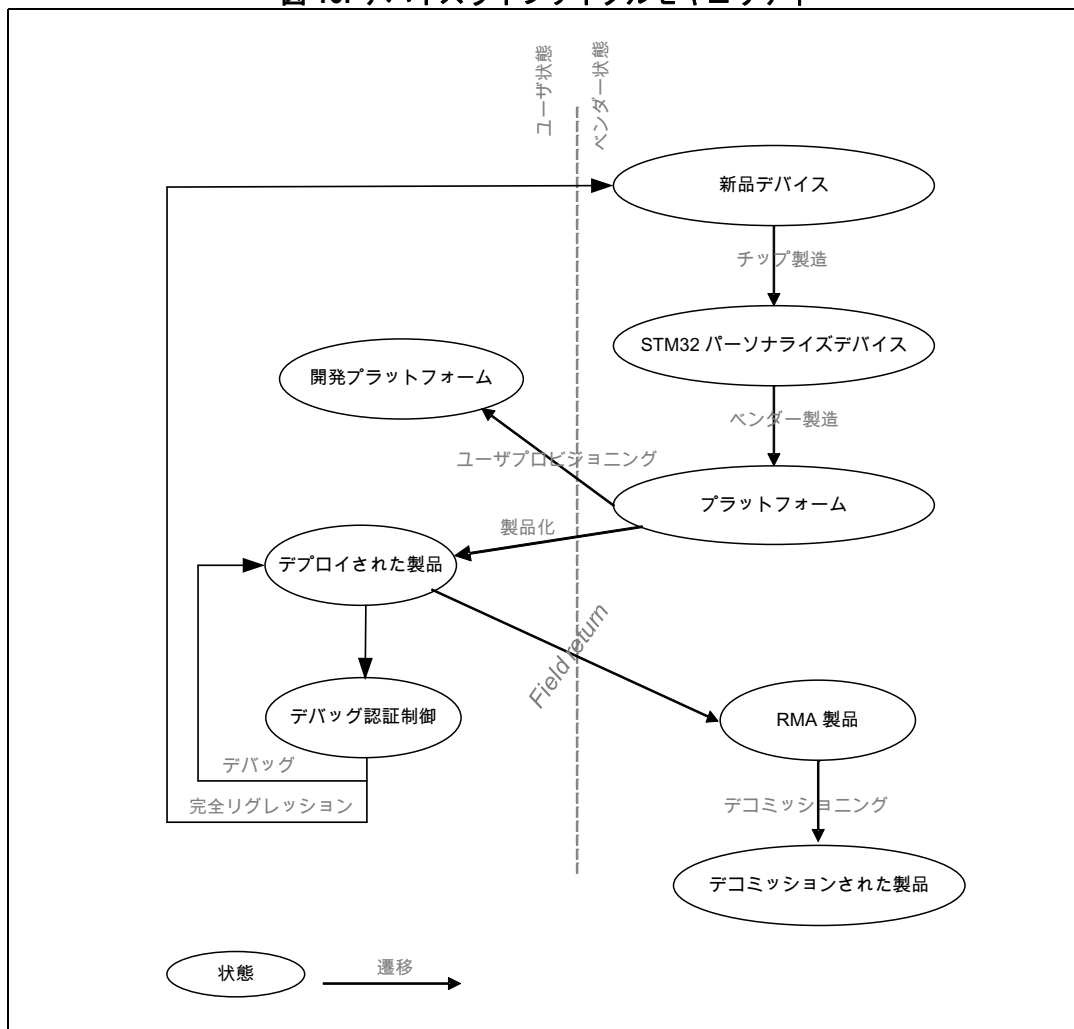
TAMP でタンパイイベントが確定されると、すべての OTFDEC キーは消去され、OTFDEC が再び正しく初期化されるまで、暗号化された領域は 0 として読み出されます。

OTFDEC の一般的な用途については、[セクション 3.12.2](#) を参照してください。ペリフェラルプログラミングの詳細については、[セクション 37: オンザフライ復号エンジン \(OTFDEC\)](#) を参照してください。

3.11 製品ライフサイクル

一般的な IoT デバイスのライフサイクルを図 10 に示します。各ステップについて、デバイスはハードウェアに内蔵されたセキュアなライフサイクル管理メカニズムを提示します。

図 10. デバイスライフサイクルセキュリティ



ベンダーまたはエンドユーザ構内で見られるさまざまなフェーズと関連する遷移の詳細を [表 16](#) に示します。

表 16. 主な製品ライフサイクル遷移

遷移	説明
デバイス製造	ST は常に製造上の欠陥をチェックしつつ、新しい STM32 デバイスを作成します。このプロセスでは、デバイスに ROM ファームウェア、セキュアファームウェアインストール (SFI) 一意キーペア、およびパブリック ID がプロビジョニングされます。
ベンダー製造	1 つ（または複数）のベンダーが、エンドユーザに提供される前のプラットフォームのアセンブリ、初期化、およびプロビジョニングを担当します。このエンドユーザは、最終製品を使用するか（「製品化」遷移）、ソフトウェア開発にプラットフォームを使用できます（「ユーザプロビジョニング」遷移）。
製品化	エンドユーザは、すぐに使える製品を入手します。プラットフォームのすべてのセキュリティ機能は有効化され、デバッグ/テスト機能は制限/無効化され、不変コードへの一意なブートエントリが行われます。
ユーザプロビジョニング	プラットフォームベンダーは、生産クラウドネットワークに接続されない、開発用の個別プラットフォームを準備します。
現場返品	製品は分析のために現場返品センターに返送されます。分析は、部分的に開くことによって、または部分的あるいは完全なリグレッションを開始することによって可能です。そのようなアクセスには、ベンダーによって提供された証明書などのプロビジョニングが必要です。

以下に説明する機能は、デバイスライフサイクルのセキュリティ保護に貢献します。

3.11.1 製品の設定とセキュリティサービス

製品は、システム Flash に ST セキュリティサービスがプロビジョニングされますが、ブート連鎖の完全な制御を OEM に委ねるように設定することもできます。これは、一意なブートエントリ (FLASH_OPTSR : BOOT_UBE) オプションバイトによって行われ、セキュリティサービス (ST-iROT : ST immutable root of trust (ST 不変の信頼の起点)) と OEM ブート実装 (OEM-iROT : OEM immutable root of trust (OEM 不変の信頼の起点)) のいずれを Flash メモリのユーザ部分にインストールするか選択できます。

セキュリティサービスは、ST によってシステム Flash (不変) にプロビジョニングされます。これらは、最初の更新可能コード (uROT) の確認と更新を管理するプラットフォームの信頼の起点となります。

3.11.2 ライフサイクル管理

製品のライフサイクルでは、開発時、製造時、アフターサービスなども含めて、製品のさまざまなアセット（コードとデータ）へのアクセスを制御できます。

プロビジョニングされたコードとデータを処理するさまざまな配布モデルで製品をプロビジョニングできます。

表 17. 一般的な製品ライフサイクルの フェーズ

PRODUCT_STATE + DebugState		デバッグ (デフォルト設定)	コメント
オープン	デバイスオープン	セキュア ⁽¹⁾ と 非セキュア	コードのデバッグを提供することにより、製品の開発を可能にします。 TrustZone が有効なとき、ブートアドレスはセキュア領域をターゲットにする必要があります。 ブートピンを使用して、ブートローダを起動できます。ブートピンと BOOT_UBE を使用することで、ST-iROT を通じてユーザ Flash コードを起動できます。
プロビジョニング	デバッグの部分的な オープン（非セキュアのみ）	HDPL3 + 非セキュアのみ	製品（部分的または完全）のプロビジョニングの管理が可能です。セキュアファームウェアインストールまたはブートローダを起動して、製品をプロビジョニングできます。 TrustZone が有効なとき、ブートアドレスはセキュア領域をターゲットにする必要があります。SRAM でのブートは許されません。
iROT プロビジョン	デバッグの部分的な オープン（非セキュアのみ）	HDPL3 + 非セキュアのみ	構成（コード、オプションバイト、セキュアストレージ）も含め、不変の信頼の起点がインストールされていることを前提とします。 TrustZone が有効なとき、ブートアドレスはセキュア領域をターゲットにする必要があります。SRAM でのブートは許されません。
TZ クローズド	デバッグの部分的な オープン（非セキュアのみ）	HDPL3 + 非セキュアのみ	セキュア OS が TrustZone にインストールされていることを前提とします（この状態は、TZEN = 0xB4 の場合のみ存在します）。 デバッグは非セキュアアプリケーションについてオープンされます
クローズド	デバッグクローズド	NoDebug + デバッグ認証制御	製品構成が最終決定していることを前提とします。現場修理（専用アプリケーションノートの読出し）のデバッグ認証をサポートできます。
ロック	デバッグロック	なし	製品構成が最終決定していることを前提とします。デバッグ認証は許されません。製品はこの状態で決定的です。
NS- リグレーション	デバッグクローズド	NoDebug	これは、TZ クローズドへの部分的なリグレーションを管理し、セキュアストレージに含まれるすべての非セキュアコードおよびデータを削除する一時的な（ただし不揮発性の）状態です。
リグレーション	デバッグクローズド	NoDebug	これは、Open 状態への完全なリグレーションを管理するための一時的な（ただし不揮発性の）状態であり、セキュアストレージ（HDPL1 から HDPL2）に含まれているすべてのユーザ Flash コードおよびデータを削除します。
制限付きデバッグ	デバッグ認証アクセス許可に従ってオープンされたデバッグ	アクセス許可に依存	これは、デバッグ認証プロトコルによって取得されたアクセス許可に基づいてデバッグを管理するための一時的な（パワーオンリセットまでの）状態です。

1. RSS コードを実行しているときには、デバッグはできません。

サポートされる遷移は、図 11 に示されていますが、デバッグインタフェースまたはシステムブートローダを介して要求できます（使用可能な場合）。

図 11. PRODUCT_STATE（簡略化した TrustZone 有効ビュー）

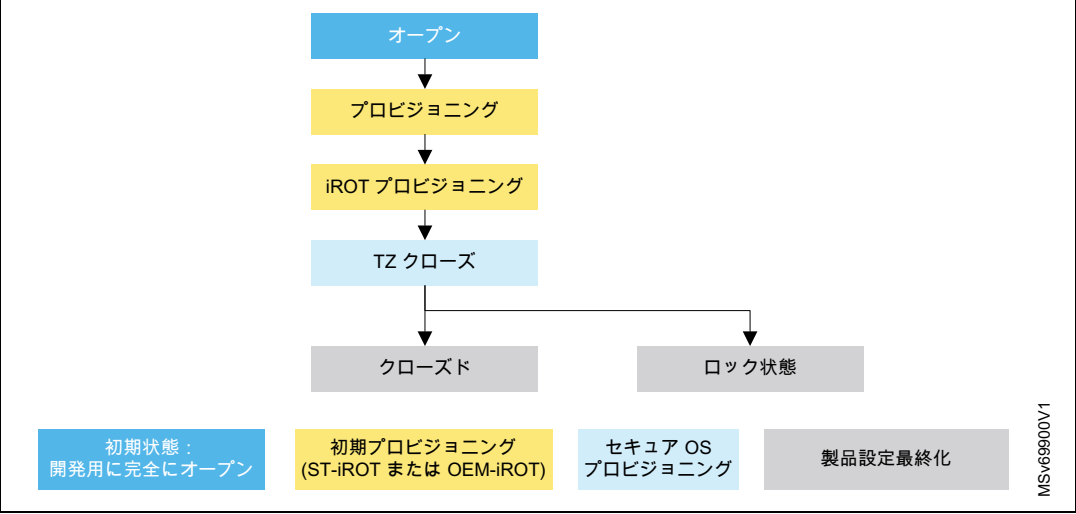
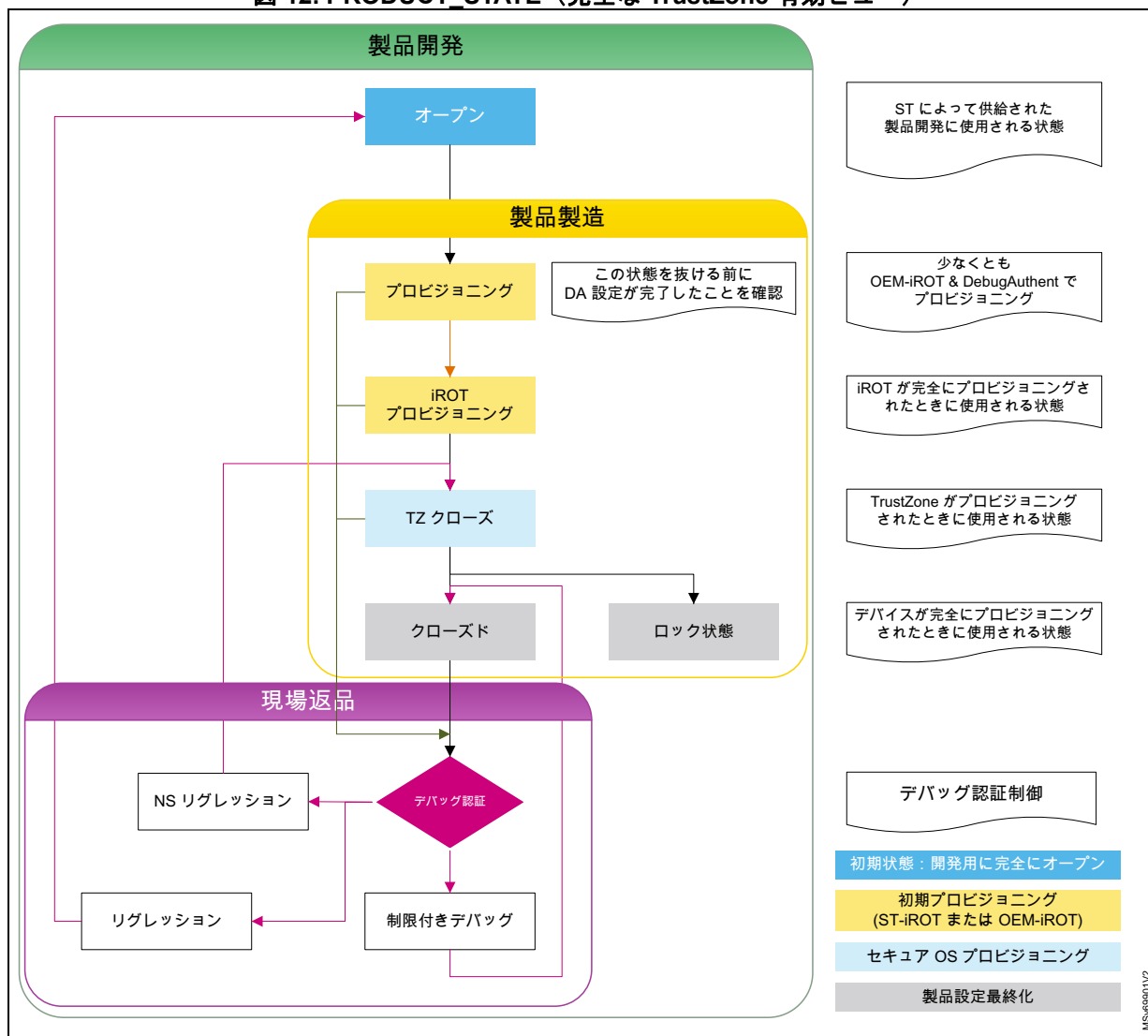


図 12. PRODUCT_STATE (完全な TrustZone 有効ビュー)



ライフサイクルの完全なビューには、デバッグ認証部分が含まれ、製品保守担当者による現場返品（現場センター）の管理を容易にします。デバッグ認証制御は、Arm PSA ADAC 仕様に基づくプロトコルによって確保されます。これにより、ホストがアクセス許可付きの信頼できる証明書を持つかどうかを制御できます。アクセス許可の定義により、デフォルトモデルでの柔軟性が高まります。完全または部分的なリグレーション、または非セキュアについてのデバッグをオープンするため。

TrustZone が無効なとき (TZEN = 0xC3)、状態 TZ クローズドは存在せず、デバッグ認証はパスワード認証方式に基づきます。デバッグ認証機能の詳細については、対応するアプリケーションノートを参照してください。

プロトコルには、ターゲットの基本情報を取得できる「DISCOVER」コマンドが含まれています。つまり、製造業者、品名、デバイス識別子、サポートされるプロトコルバージョン、認証の暗号方式（証明書またはパスワードの選択に役立ちます）です。

3.11.3 推奨されるプロトコル設定

製品のメンテナンス（現場での）を容易にするために、デバッグ認証制御と呼ばれる機能を利用することをお勧めします。これにより、製品のメンテナンス担当者はデバッグを有効にして、機密情報のセキュリティを考慮しながら、リグレッションの管理を可能にできます。

これは、以下のアクションを意味します。

- 製品が設定されたときに、PRODUCT_STATE をクローズド状態に設定します。
- デバッグ認証制御が証明書に基づくとき（TrustZone が有効（TZEN = 0xB4））、**OEM-PublicKey** および **SOC-Mask** をプロビジョニングします。非セキュアデバッグと部分的小および完全なリグレッションを提示するように SOC-Mask を設定すると、Arm PSA セキュリティモデルの準拠が保証されます。
- デバッグ認証制御がパスワードに基づくときには（TrustZone が無効（TZEN = 0xC3））、パスワードの HASH のみをプロビジョニングする必要があります。

プラットフォームの設定、ホストツールの設定、および証明書の生成に役立つ専用のアプリケーションノートがあります。

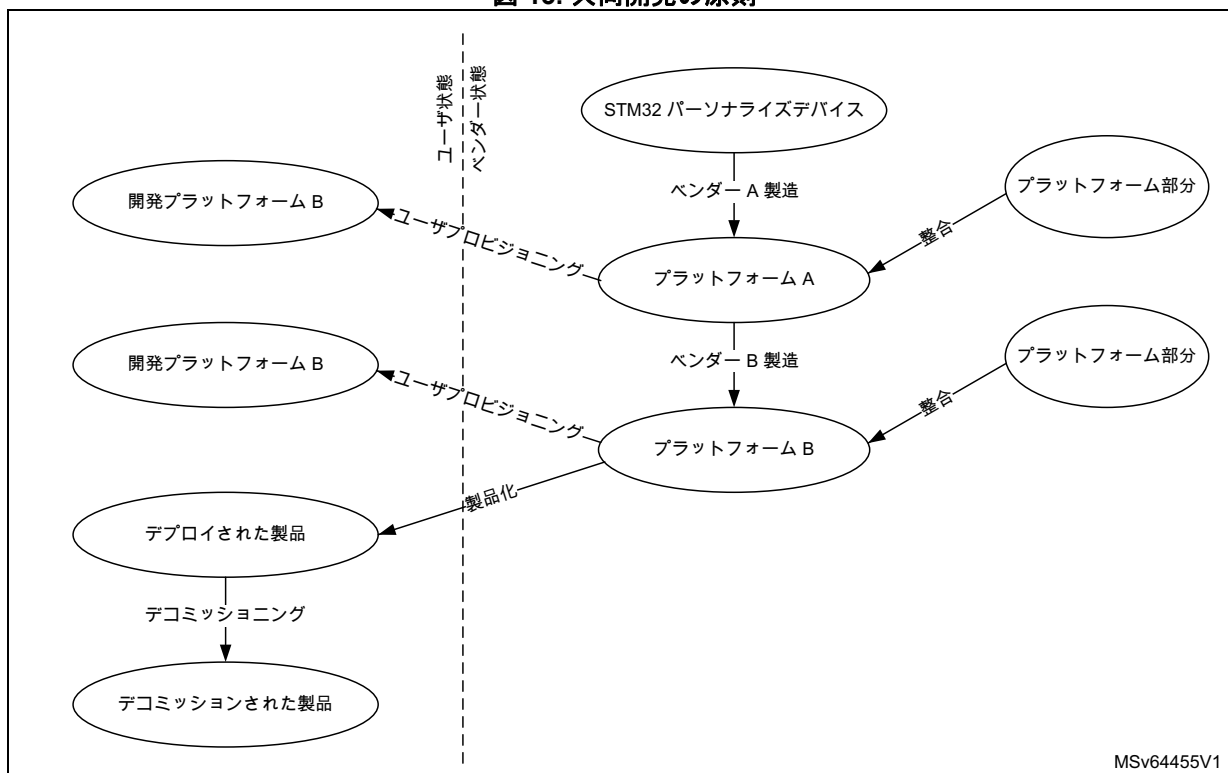
3.12 ソフトウェア知的財産保護と共同開発

ソフトウェア知的財産保護と共同開発により、デバイスは画期的なサードパーティライブラリを統合するソリューションの設計を可能にします。

共同開発の概要を [図 13](#) に示します。STMicroelectronics によって販売されたパーソナライズされたデバイスから始めて、ベンダー A はハードウェアとソフトウェアの一部をプラットフォーム A で統合できます。それをベンダー B が使用して、同じことを行ってから、最終製品をエンドユーザーにデプロイします。

注： 各プラットフォームベンダーは、生産クラウドネットワークに接続されない開発用の個別プラットフォームをプロビジョニングできます（「開発プラットフォーム X」）。

図 13. 共同開発の原則



以下で説明されている機能は、そのような共同開発におけるソフトウェア知的財産のセキュリティ保護に貢献します。

3.12.1 ソフトウェア知的財産保護

セクション 3.11.2 で説明されているように、ハードウェア PRODUCT_STATE メカニズムは、デバイスにプロビジョニングされたシークレットへのアクセスを自動的に制御します。これらのシークレットの保護は 表 18 で定義されています。

表 18. PRODUCT_STAT によるソフトウェア知的財産保護

PRODUCT_STATE 保護レベル		シークレット保護
オープン	デバイスオープン	特別な保護なし。
プロビジョニング	デバッグの部分的なオープン (非セキュアのみ)	HDPL1 で保護されているすべての領域は、ダンプ、デバッグ、またはトレースができません。iROT は、より高いレベルの保護をセットアップできます。
iROT プロビジョン	デバッグの部分的なオープン (非セキュアのみ)	HDPL1 で保護されているすべての領域は、ダンプ、デバッグ、またはトレースができません。iROT は、より高いレベルの保護をセットアップできます。
TZ クローズド	デバッグの部分的なオープン (非セキュアのみ)	セキュアブート時にセキュアとしてマップされたすべてのペリフェラルとメモリは、ダンプ、デバッグ、またはトレースができません。
クローズド	デバッグなし	デバッグ認証以外のデバッグはできません。
ロック	デバッグなし	デバイスに保存され、または外部 Flash メモリに暗号化されたすべてのデータとコードは、クリアテキストのダンプ、デバッグ、またはトレースができません。

3.12.2 OTFDEC によるソフトウェア知的財産保護

セクション 3.10.3 で説明されているように、OCTOSPI に関連付けられた OTFDEC は、外部 SPI Flash メモリに保存された暗号化されたイメージをオンザフライで復号できます。

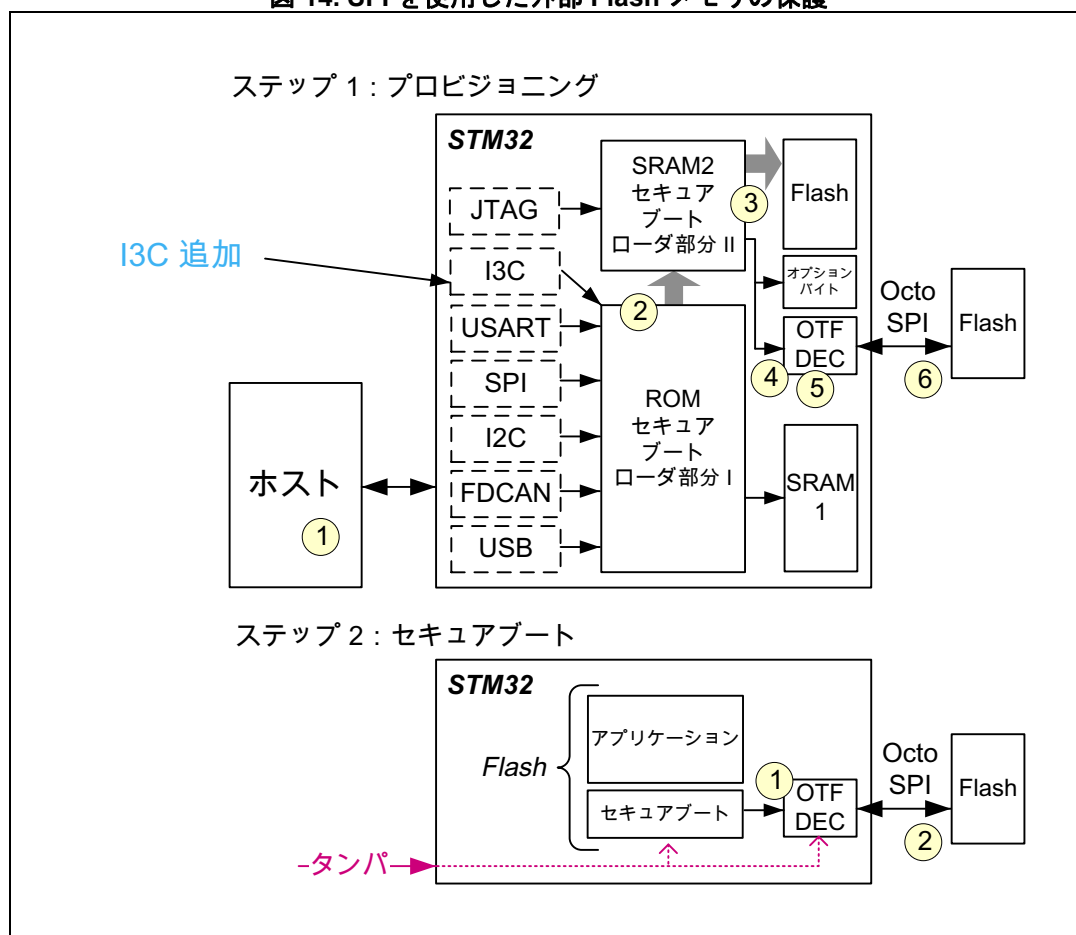
この機能により、デバイスは次のいずれかの方法で知的財産のインストールができます。

- 無線で。デバイスにプロビジョニングされたキーですでに暗号化されたイメージで。
- 信頼できる、または信頼できない環境/設備にあるプロビジョニングホストを通じて。

図 14 は、この最後のケースを示しています。信頼できない環境での、内蔵 Flash メモリと外部 SPI Flash メモリ（暗号化されている）の両方のソフトウェア知的財産のプロビジョニング。

注： OTFDEC はカウンタモード（CTR）の AES を使用して、可能な限り低い遅延を達成するため、1 つの暗号化領域の内容が変更されるたびに、対応する暗号コンテキスト（キーまたは初期化ベクトル）を変更する必要があります。この制約により、OTFDEC は、外部 NOR Flash メモリに保存された読み出し専用データまたはコードの復号に適しています。

図 14. SFI を使用した外部 Flash メモリの保護



プロビジョニング

デバイスが新品であると仮定して、最初のステップは、次のように両方のメモリをプロビジョニングすることです。

- ユーザは、以下で構成される SFI イメージを作成します。
 - 暗号化された内部ファームウェアおよびデータ（外部 Flash メモリのドライバを含む）
 - 暗号化された外部ファームウェアおよびデータ AES キー（最大 4）
 - 暗号化された外部ファームウェアおよびデータイメージ
- システムメモリに保存されたセキュアブートローダは、サポートされる通信ポート（USART、SPI、I2C、FDCAN、USB、I3C、および JTAG）を通じて、セキュアブートローダの 2 番目の部分を SRAM2 にロードします。この 2 番目の部分は、セキュア SRAM2 で実行して、上記のサポートされる通信ポートを通じて受信したコマンドにより、SFI プロセスの実行と SFI プロトコルの適用を担当します。
- 内部 Flash メモリは、復号されたオプションバイト、内部ファームウェアおよびデータ、外部ファームウェアおよびデータ AES キーでプログラムされます。または、そのようなグローバルキーの代わりに、デバイス-意外部ファームウェア AES キーを使用することもできます。
- OTFDEC は、対応する外部ファームウェアおよびデータ AES キーを含め、暗号化された領域情報で適切に初期化されます。
- Sfi プロセスの実行により、暗号化された外部ファームウェアおよびデータイメージのチャンクは、デバイスで復号されてから、OTFDEC で再び暗号化されます。

6. OTFDEC によるチャンクの再暗号化後、ユーザ外部 Flash メモリのプログラマは、OCTOSPI を通じて最後の暗号化チャンクを外部 SPI Flash メモリにプログラムします。

セキュアブート

プロビジョニング後、デバイスが信頼できるファームウェアで初期化するたびに、以下のアクションが必要です。

1. セキュアブートファームウェアが実行して、外部ファームウェアおよびデータ AES キーを、その他の必要な情報とともに、OTFDEC 書込み専用キーレジスタにプログラムします。
2. アプリケーションは、暗号化された外部ファームウェアおよびデータを OCTOSPI を通じてメモリマップモードで読み出し、または実行しますが、タンパイイベントが検出された場合を除きます。この場合は、すべての OTFDEC キーが消去され、OTFDEC が再び正しく初期化されるまで、暗号化領域は 0 として読み出されます。

デバイスの SFI ソリューションの詳細については、AN4993『セキュアファームウェアインストール (SFI) の概要』を参照してください。

3.12.3 その他のソフトウェア知的財産保護

ソフトウェア知的財産に対するデバイスのその他の保護としては、以下のものがあります。

- 物理的タンパまたは摂動などの侵入攻撃は、検出後のデバイスのデコミッションングによって、検出された攻撃が成功する前に対応されます。
- サイドチャネル攻撃などの非侵入型攻撃は、サイドチャネル経由でシークレット情報を漏出させない（タイミング、電力、EM エミッションなど）ことによって対応されます。

4 ブートモード

起動時、BOOT0 ピンと NSBOOTADD[31:8]/SECBOOTADD[31:8] オプションバイトを使用して、以下を含むブートメモリアドレスを選択します。

- ユーザ Flash メモリ内の任意のアドレスからのブート
- システム・メモリからのブート
 - ブートローダ
 - ST の不変の信頼の基点(ルート・オブ・トラスト) (ST-iROT)
 - ルート・セキュア・サービス (RSS)
 - オーセンティケーション・ライブラリのデバッグ (RSS-DA)

内蔵ブートローダ

内蔵ブートローダは、システムメモリに配置され、生産時に ST によってプログラムされています。これは、デバイスモードで USART、I2C、I3C、SPI、FDCAN、または USB_FS を使用し、DFU（デバイス・ファームウェア・アップグレード）により Flash メモリを再プログラムするために使用します。

AN2606「STM32 マイクロコントローラのシステム・メモリ・ブート・モード」を参照してください。

内蔵のルート・セキュリティ・サービス (RSS)

内蔵 RSS は、セキュア情報ブロックに配置されるよう生産時に ST によってプログラムされています。

AN4992「セキュア・ファームウェア・インストール (SFI) の概要」を参照してください。

内蔵の不変の信頼の基点 (ルート・オブ・トラスト) (ST-iROT)

システム・メモリの内蔵 ST-iROT は、生産時に ST によってプログラムされています。ST-iROT は、ブート・シーケンスで実行する最初のアップデート可能なレベルのセキュア・ブートとセキュア・インストールを管理している、不変の信頼の基点(ルート・オブ・トラスト)です。

内蔵のデバッグ・オーセンティケーション (ST-DA)

システム・メモリの内蔵 ST-DA は、生産時に ST によってプログラムされています。ST-DA は、デバッグをセキュアに再度オープンできるようにするか、あるいはセキュアなプロダクトのリグレーションを現場で立ち上げられるようにすることで、デバッグ・オーセンティケーション・プロトコルを管理するライブラリです。

4.1 STM32H562/H563 ブートモード

表 19に、TrustZone が無効な場合（TZEN = 0xC3）の、STM32H562/H563 デバイスのブートモードの詳細を示します。

表 19. TrustZone が無効な場合（TZEN = 0xC3）の STM32H562/H563 のブートモード

PRODUCT_STATE	BOOT0 ピン	ブート・アドレスの オプション・バイトの選択	ブート領域	ST によってプログラム されたデフォルト値
オープン	0	NSBOOTADD[31:8]	ユーザ・オプション・ バイト NSBOOTADD[31:8] で 定義されたブート・アドレス	Flash : 0x0800 0000
-	1	該当なし	ブートローダ	ブートローダ
プロビジョニング	x	該当なし	RSS	RSS
プロビジョンド、 クローズド、ロックド	x	NSBOOTADD[31:8]	ユーザ・オプション・バイト NSBOOTADD[31:8] で 定義されたブート・アドレス	Flash : 0x0800 0000

注： BOOT_UBE は STM32H573x デバイスでのみ使用可能です。セクション 4.2 を参照してください。

表 20 に、TrustZone が有効な場合（TZEN = 0xB4）の、STM32H562/H563 デバイスのブートモードの詳細を示します。

表 20. TrustZone が有効な場合（TZEN = 0xB4）の STM32H562/H563 のブートモード

PRODUCT_STATE	BOOT0 ピン	ブート・アドレスの オプション・バイトの選択	ブート領域	ST によってプログラム されたデフォルト値
オープン	0	SECBOOTADD[31:8]	ユーザ・オプション・バイト SECBOOTADD[31:8] で 定義されたブート・アドレス	Flash : 0x0C00 0000
-	1	該当なし	ブートローダ	ブートローダ
プロビジョニング	x	該当なし	RSS	RSS
プロビジョンド、 TZ_クローズド、 クローズド、ロックド	x	SECBOOTADD[31:8]	ユーザ・オプション・バイト SECBOOTADD[31:8] で 定義されたブート・アドレス	Flash : 0x0C00 0000

注： BOOT_UBE は STM32H573x デバイスでのみ使用可能です。セクション 4.2 を参照してください。

TrustZone が有効な場合（TZEN = 0xB4）、ブート領域はセキュア領域になければなりません。SECBOOTADD0 [24:0] オプション・バイトは、セキュア・ブート・メモリ・アドレスを選択するために使用します。一意なブート・エントリ・オプションは、SECBOOT_LOCK オプション・ビットを設定することによって選択できます。

4.2 STM32H573x ブートモード

表 21に、TrustZone が無効な場合（TZEN = 0xC3）の、STM32H573x 製品のブート・モードの詳細を示します。

表 21. TrustZone が無効な場合（TZEN = 0xC3）の STM32H573x のブートモード

PRODUCT_STATE	BOOT0 ピン	BOOT_UBE FLASH_OPTSR [29:22]	ブート・アドレスの オプション・バイトの 選択	ブート領域	ST によって プログラムされた デフォルト値
オープン	0	該当なし	NSBOOTADD[31:8]	ユーザ・オプション・ バイト NSBOOTADD[31:8] で定義されたブート・ アドレス	Flash : 0x0800 0000
	1	該当なし	該当なし	ブートローダ	ブートローダ
プロビジョニング	x	該当なし	該当なし	RSS	RSS
プロビジョンド、 クローズド、ロックド	x	該当なし	NSBOOTADD[31:8]	ユーザ・オプション・ バイト NSBOOTADD[31:8] で定義されたブート・ アドレス	Flash : 0x0800 0000

表 22 に、TrustZone が有効な場合（TZEN = 0xB4）の、STM32H573x 製品のブートモードの詳細を示します。

表 22. TrustZone が有効な場合（TZEN = 0xB4）の STM32H573x のブートモード

PRODUCT_STATE	BOOT0 ピン	BOOT_UBE FLASH_OPTSR [29:22]	ブート・アドレスの オプション・ バイトの選択	ブート領域	ST によってプログラム されたデフォルト値
オープン	0	x	SECBOOTADD [31:8]	ユーザ・オプション・バ イト SECBOOTADD[31:8] で 定義されたブート・アド レス	Flash : 0x0C00 0000
-	1	0xB4	該当なし	ブートローダ	ブートローダ
-	1	0xC3	該当なし	ST-iROT	ST-iROT
プロビジョニング	x	該当なし	該当なし	RSS	RSS
プロビジョンド、 TZ_クローズド、 クローズド、 ロックド	x	0xC3	ST-iROT	ST-iROT	ST-iROT
		0xB4	SECBOOTADD [31:8]	ユーザ・オプション・ バイト SECBOOTADD[31:8] で 定義された ブート・アドレス	Flash : 0x0C00 0000

TrustZone が有効な場合（TZEN = 0xB4）、ブート領域はセキュア領域になければなりません。SECBOOTADD0 [24:0] オプション・バイトは、セキュア・ブート・メモリ・アドレスを選択するために使用します。一意なブート・エントリ・オプションは、SECBOOT_LOCK オプション・ビットを設定することによって選択できます。

5 グローバル TrustZone® コントローラ (GTZC)

5.1 概要

このセクションでは、次のようなサブブロックを含むグローバル TrustZone コントローラ (GTZC) ブロックについて説明します。

- **TZSC** : TrustZone セキュリティ・コントローラ
このサブブロックは、スレーブ・ペリフェラルのセキュア／特権状態を定義します。また、ウォーターマーク・メモリ・ペリフェラル・コントローラ (MPCWM) のサブリージョン領域サイズおよびプロパティの制御も行います。TZSC は、セキュリティ保護可能な各ペリフェラルのセキュア状態を RCC および I/O ロジックと共有することで、いくつかのペリフェラル (RCC、GPIO など) に通知します。
- **MPCBB** : ブロックベースのメモリ保護コントローラ
このサブブロックでは、512 バイトのページでセグメント化された SRAM を持ち、TrustZone システム製品にある内蔵 RAM の設定を、プログラム可能なセキュリティと特権属性により行います。
- **TZIC** : TrustZone 不正アクセスコントローラ
このサブブロックは、システム内のすべての不正アクセスイベントを集め、NVIC へのセキュア割込みを生成します。

これらのサブブロックは、以下に挙げるようなプログラム可能なセキュリティおよび特権属性を備えたバス・エージェントを持つ製品において、TrustZone システム・セキュリティを設定するために使用します。

- プログラム可能なセキュア／特権ブロック (ページ) を持つオンチップ RAM (内蔵 RAM)
- プログラム可能なセキュア／特権アクセスモードを持つ AHB および APB ペリフェラル
- セキュア／特権領域を持つオフチップメモリ (外付けメモリ)

5.2 GTZC の主な機能

GTZC の主な機能を以下に列記します。

- TZSC、TZIC、および MPCBB 用の 3 つの独立した 32 bit AHB インタフェース
- セキュアなトランザクションでのみ TZIC へのアクセスが可能
- TZSC および MPCBB の特権および非特権領域への、セキュアおよび非セキュア・アクセスのサポート
- 一連のレジスタを使用して製品のセキュリティ設定を定義
 - 内蔵 SRAM 用のセキュア／特権ブロック
 - 外部メモリおよび内部バックアップ SRAM 用のセキュア／特権領域
 - セキュリティ保護可能なペリフェラルのセキュア／特権アクセスモード
 - セキュリティ保護可能なマスタのセキュア／特権アクセスモード

GTZC の TrustZone システムアーキテクチャ

Armv8-M は、次の 2 つの世界を分離することで、TrustZone-M モデル単位のセキュリティをサポートします。

- 通常、セキュリティに厳しいアプリケーションが実行されたり、重要なリソースが配置されたりするセキュアなワールド
- (普通の非セキュア OS やユーザスペースなどの) 非セキュアあるいはパブリックなワールド

TrustZone アーキテクチャは、以下の手段により AHB および Armv8-M を超えて拡張されます。

- APB エージェントに対するセキュア/非セキュアおよび特権/非特権トランザクションをブロックまたは伝搬するためのセキュア・ゲートとして AHB/APB ブリッジを用いる。
- AHB エージェントに対するセキュア/非セキュアおよび特権/非特権トランザクションをブロックまたは伝搬するためのセキュア・ゲートとして PPC (ペリフェラル・プロテクション・コントローラ) を用いる。
- 内蔵 SRAM へのセキュア/非セキュア、特権/非特権アクセスをフィルタするためのセキュア・ゲートとして、ブロックベースの TrustZone MPC ファイアウォールを用いる。
- 外部メモリへのセキュア/非セキュア、特権/非特権アクセスをフィルタするためのセキュア・ゲートとして、ウォーターマークベース TrustZone MPC ファイアウォールを用いる。

AHB および APB ペリフェラルは以下のように分類することができます。

- **特権ペリフェラル**：特権プロパティを定義するために、TZSC コントローラから制御される AHB/APB ファイアウォールスタブによって保護されているペリフェラル。
- **セキュアペリフェラル**：AHB/APB ファイアウォール・スタブによって常時保護されているペリフェラル。これらのペリフェラルは常にセキュアです (TZIC など)。
- **セキュリティ保護可能なペリフェラル**：セキュリティ・プロパティを定義するために、TZSC から制御される AHB/APB ファイアウォール・スタブによって保護されているペリフェラル (オプション)。
- **非セキュアおよび非特権ペリフェラル**：一切のセキュアゲートなしで直接 AHB/APB に相互接続されているペリフェラル。
- **TrustZone を認識するペリフェラル**：AHB または APB バスに直接接続され、TrustZone 固有の動作 (セキュアレジスタのサブセットなど) を実装しているペリフェラル。TrustZone を認識する AHB マスタは、常にそのセキュリティモードに合った HNONSEC 信号をドライブします (Armv8-M コアまたは DMA など)。

AHB によるセキュリティ保護可能なマスタは、TZSC でセキュア/非セキュアや特権/非特権に設定することができます。

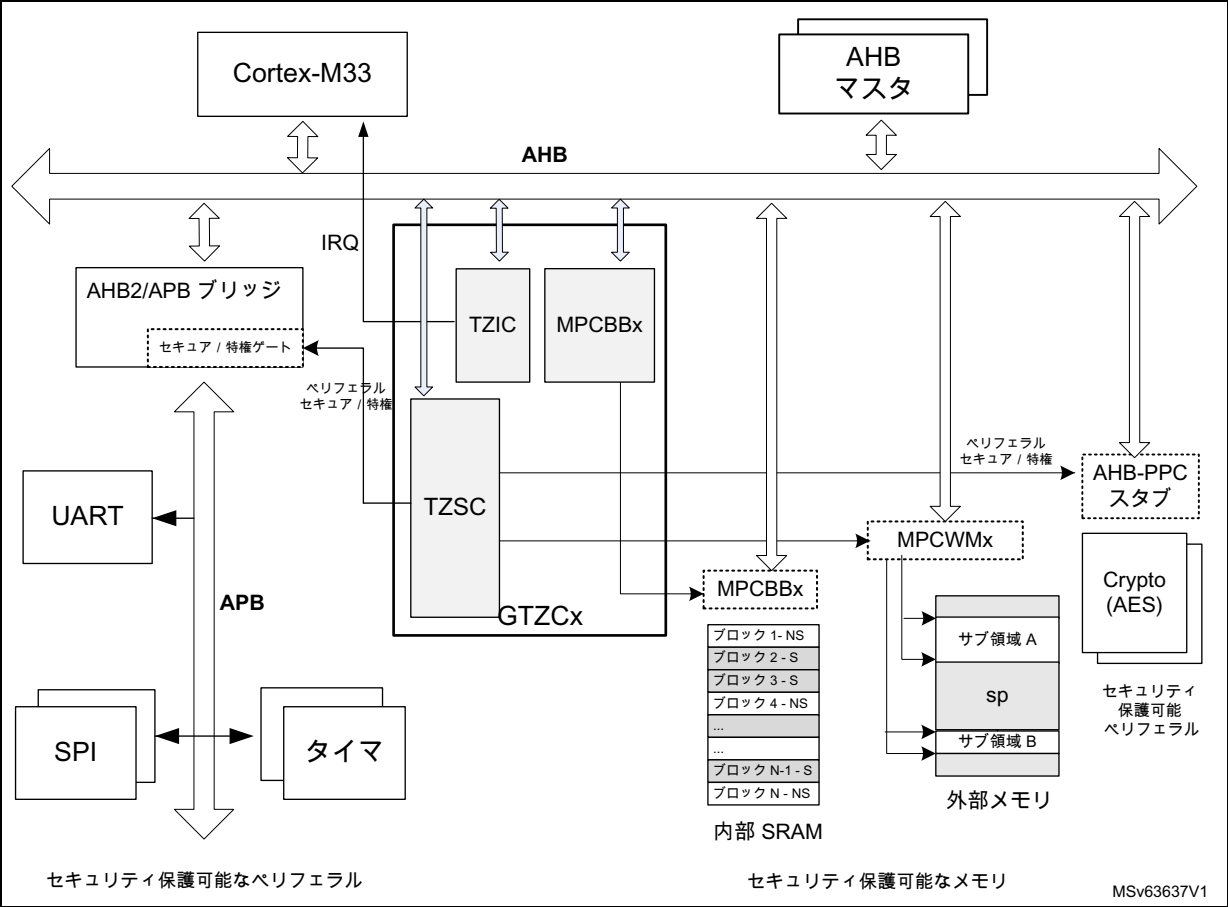
アプリケーション情報

TZSC、MPCBB、および TZIC は次のいずれかの方法で使用できます。

- 必ずセキュアブート中にプログラムし、その後はロックした状態で変更しません。
- 特定のアプリケーションコードまたはセキュアカーネル (マイクロバイザ) の使用時に動的に再プログラムします。ロックされていない場合、MPC セキュア・ブロックまたは領域のサイズは、セキュア Flash 領域またはセキュア SRAM から実行するセキュアなソフトウェアによって変更することができます。各ペリフェラルのセキュア/特権状態を定義する GTZC1_TZSC_SECCFGRx/PRIVCFGRx レジスタにも同じことが言えます。

セキュア、セキュリティ保護可能、および TrustZone を認識するペリフェラルを備えた Armv8-M セキュリティアーキテクチャを 図 15 に示します。

図 15. Armv8-M サブシステム内の GTZC のブロック図



5.3 GTZC の実装

STM32H563/H573および STM32H562 デバイスには 1 個の GTZC インスタンスが埋め込まれています。

表 23. GTZC の機能

GTZC サブブロック	GTZC1
TZSC	X
TZIC	X
MPCBB サブブロック (MPCBB の数)	X (3)

表 24 に GTZC ベース・アドレスに対する GTZC サブブロックのアドレスオフセットを示します (GTZC1 のベース・アドレスは [セクション 2.3](#) を参照してください)。

表 24. GTZC1 サブブロック・アドレスオフセット

GTZC1 サブブロック	アドレスオフセット
GTZC1_TZSC	0x0
GTZC1_TZIC	0x400
GTZC1_MPCBB1	0x800
GTZC1_MPCBB2	0xC00
GTZC1_MPCBB3	0x1000

表 25 に、使用可能な MPCWM の特性を示します。

表 25. MPCWM リソースの割当

GTZC	MPC	対象メモリアインタフェース	セキュア／非セキュアおよび 特権／非特権領域の数	ウォーターマークの 粒度 (バイト)
GTZC1	MPCWM1	OCTOSPI1	2	128 K
	MPCWM2	FMC_NOR バンク	2	128 K
	MPCWM3	FMC_NAND バンク	1	128 K
		FMC_SDRAM バンク 1	1	128 K
	MPCWM4	BKPSRAM	1	32
		FMC_SDRAM バンク 2	1	128 K

表 26 に、使用可能な MPCBB の特性を示します。

表 26. MPCBB リソースの割当

GTZC	MPC	リソース	メモリサイズ (KB)	ブロック・ サイズ (バイト)	ブロック数	スーパー ブロックの数
GTZC1	MPCBB1	SRAM1	256	512	512	16
	MPCBB2	SRAM2	64		128	4
	MPCBB3	SRAM3	320		640	20

5.4.2 不正アクセスの定義

次の 3 種類の不正アクセスが存在します。

- 不正な非セキュアアクセス
セキュア・リソースの書込みを試みる非セキュア・トランザクションは、すべて不正アクセスとみなされます。そのため、そのアドレス指定されたリソースが、不正書込みアクセスへの不正アクセス割り込みと、不正フェッチ・アクセスへのバス・エラーを生成します。セキュアおよび特権設定レジスタには例外もあります。特権設定レジスタは、セキュア・レジスタへの許可された非セキュア読出しアクセスを行います (GTZC1_TZSC_SECCFGRx および GTZC1_TZSC_PRIVCFGRx を参照)。
- 不正なセキュアアクセス
ブロックベースの内蔵 SRAM またはウォーターマーク・メモリにおいて非セキュア・ブロックへのアクセスを試みるセキュア・トランザクションはすべて不正とみなされます。
TZIC の正しい設定は、関連イベントのキャプチャを可能にし、さらに NVIC への GTZC_IRQn 割り込みを生成します。これには、書込み、読出し、および実行アクセスが該当します。
MPCBB コントローラについては、GTZC1_MPCBBz_CR レジスタの SRWILADIS ビットを設定することにより、非セキュア SRAM ブロックへのセキュアなデータ読出し／書込みアクセスを無視するという選択肢があります。その場合、非セキュア SRAM ブロックに対するセキュアなデータ読出しおよび書込みトランザクションは許可されますが、セキュアな実行アクセスは許可されないままです。
非セキュア・ペリフェラルのレジスタへのアクセスを試みるセキュア実行トランザクションはすべて不正アクセスとみなされ、バスエラーが生成されます。
- 不正な非特権アクセス
特権リソースへのアクセスを試みる非特権トランザクションはすべて不正アクセスとみなされます。不正読出しおよび書込みアクセスに対して生成される不正アクセスイベントはありません。アドレス指定されたリソースはサイレントフェイル動作に従い、読出しにはすべてゼロデータを返し、書込みはすべて無視します。バスエラーは生成されません。非特権実行トランザクションが特権メモリへのアクセスを試みた場合、バスエラーが生成されます。

5.4.3 TrustZone セキュリティ・コントローラ (TZSC)

TZSCは、設定可能な一連のレジスタで構成され、以下の機能を提供します。

- 次のレジスタによる、すべてのペリフェラルのセキュアおよび特権状態の制御。
 - セキュリティ保護可能なペリフェラルの AHB/APB ファイアウォール・スタブを制御するための GTZC1_TZSC_SECCFGRx レジスタ
 - 特権ペリフェラルの AHB/APB ファイアウォール・スタブを制御するための GTZC1_TZSC_PRIVCFGRx レジスタ
- ウォーターマークベースのメモリ保護コントローラ (外部メモリおよびバックアップ SRAM) には 2 つの独立した領域を定義することができ、次のフィールドを使ってプログラムをします。
 - 外部メモリ／バックアップ SRAM の最初の保護サブ領域の始め: SUBA_START[10:0]
 - 外部メモリ／バックアップ SRAM の最初の保護サブ領域の長さ: SUBA_LENGTH[11:0]
 - 外部メモリ／バックアップ SRAM の 2 番目の保護サブ領域の始め: SUBB_START[10:0]
 - 外部メモリ／バックアップ SRAM の 2 番目の保護サブ領域の長さ: SUBB_LENGTH[11:0]

各サブ領域の制御レジスタは、ウォーターマークベースのメモリの保護コントローラを有効化／無効化するほか、各サブ領域の適切な属性の定義に使用することができます。

図 17. ウォーターマークベースのメモリ保護コントローラ (領域 x/サブ領域 A および B)

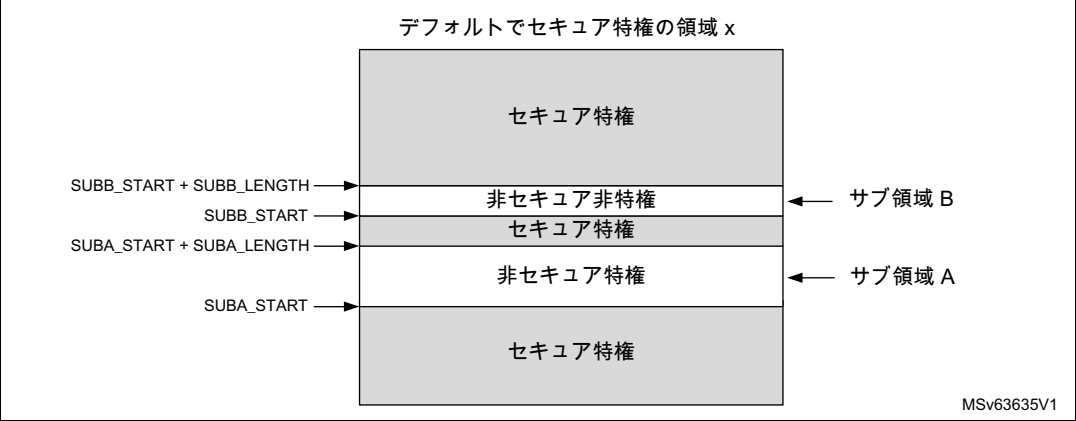


図 17 において、領域 x は外部メモリまたはバックアップ SRAM 領域 (FMC バンク、OCTOSPI1、あるいは BKPSRAM など) を表します。サブ領域 A と B のセキュアおよび特権属性は、独立して設定可能です。領域 x においてサブ領域が定義または有効化されていない場合、領域 x のデフォルト属性はセキュア/特権として設定されます。

以下の表は、オーバーラップがない場合のサブ領域 A と B の共通領域のセキュア/特権プロパティの説明です。

表 27. サブ領域 A と B のセキュア・プロパティ

サブ領域 A	サブ領域 B	領域 A と B のオーバーラップ領域のプロパティ
非セキュア	非セキュア	非セキュア
非セキュア	セキュア	非セキュア
セキュア	非セキュア	非セキュア
セキュア	セキュア	セキュア

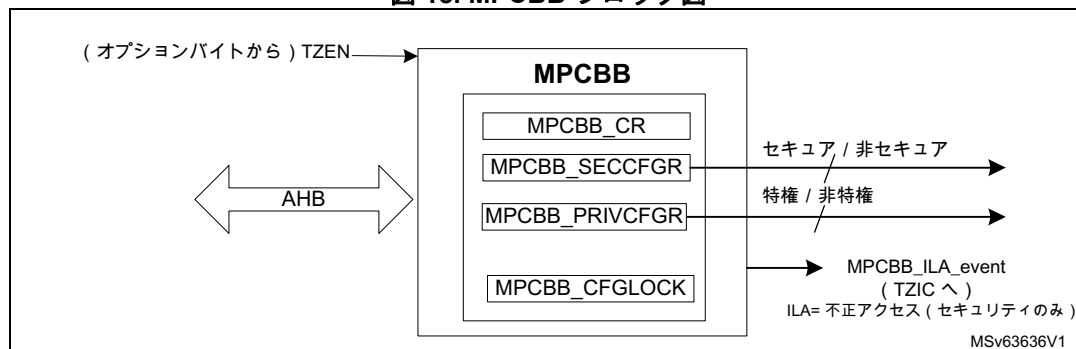
表 28. サブ領域 A と B の特権プロパティ

サブ領域 A	サブ領域 B	領域 A と B のオーバーラップ領域のプロパティ
非特権	非特権	非特権
非特権	特権	非特権
特権	非特権	非特権
特権	特権	特権

5.4.4 ブロックベースのメモリ保護コントローラ (MPCBB)

MPCBB は一連の設定可能なレジスタで構成され、内蔵 SRAM メモリのセキュリティと特権ポリシーの定義を可能にします。セキュリティおよび特権ポリシーは、SRAM の各 512 バイト・ブロックごとに個別に設定が可能です。

図 18. MPCBB ブロック図



MPCBB を設定するには、以下の操作が必要です (例えばブート時)。

- セキュアファームウェアは、GTZC1_MPCBBz_SECCFGRx に正しいビットを設定することにより、どのメモリ・ブロックがセキュア・ブロックかを定義しなくてはなりません。
- 特権ファームウェアは、GTZC1_MPCBBz_PRIVCFGRx に正しいビットを設定することにより、どのメモリ・ブロックが特権ブロックかを定義しなくてはなりません。

MPCBB スーパーブロックは 32 の連続したブロックからなります。セキュア・アプリケーションは、それぞれのスーパーブロックに対し、GTZC1_MPCBBz_CFGLOCK の正しいビットを使用して、すべての関連するセキュリティ／特権ビットをロックすることができます。このロックは次のシステム・リセットまで有効です。

注： ブロック・サイズは 512 バイトです。スーパーブロックのサイズは $512 \times 32 = 16 \text{ KB}$ です。

5.4.5 TrustZone 不正アクセスコントローラ (TZIC)

TZIC は、すべての不正アクセスのソースとなるイベントに的を絞っています。これは、システムの TrustZone 機能 (TZEN = 0xB4) が有効な場合にのみ使用します。

TZIC は、セキュアな不正アクセス割り込みをトリガしたイベントのトレース (フラグ) を可能にします。不要なイベントのフィルタリングには、レジスタ・マスク (GTZC1_TZIC_IERx) が用意されています。マスクされていない不正イベントでは、TZIC が NVIC に対して GTZC_IRQn 割り込みを生成します。

不正イベントのソースごとに、ステータス・フラグおよびクリア・ビットが (それぞれ GTZC1_TZIC_SRx および GTZC1_TZIC_FCRx 内に) 存在します。マスクレジスタ (GTZC1_TZIC_IERx) のリセット値では、すべてのイベントがマスクされていなければなりません。

5.4.6 パワーオン／リセット状態

TZSC のパワーオンおよびリセット状態は、GTZC1_TZSC_SECCFGRx と GTZC1_TZSC_PRIVCFGRx のすべてのビットをクリアして 0 にします。これはすべてのセキュリティ保護可能なペリフェラルが、それぞれ非セキュアおよび非特権に設定されることを意味します。

内蔵 SRAMx (x = 1 から 3) では、すべての GTZC1_MPCBBz_SECCFGRx および GTZC1_MPCBBz_PRIVCFGRx は次の値に設定されます。

- 0xFFFF FFFF : TrustZone セキュリティがシステム・レベルで有効になっている場合 (TZEN = 0xB4)、これらの内蔵メモリブロックをデフォルトでセキュアおよび特権にします。
- 0x0000 0000 : TrustZone セキュリティがシステム・レベルで無効になっている場合 (TZEN = 0xC3)、これらの内蔵メモリブロックをデフォルトで非セキュアおよび非特権にします。

外部メモリとバックアップ SRAM では、すべての GTZC1_TZSC_MPCWMxR レジスタが次の値に設定されます。

- 0x0000 0000 : TrustZone セキュリティがシステム・レベルで有効になっている場合 (TZEN = 0xB4)、これらのメモリをデフォルトでセキュアおよび特権にします。
- 0x0800 0000 : TrustZone セキュリティがシステム・レベルで無効になっている場合 (TZEN = 0xC3)、これらのメモリをデフォルトで非セキュアおよび非特権にします。

次に、セキュア・ブート・コードによりセキュリティ設定をプログラムし、必要に応じてコンポーネントをセキュアまたは非セキュアにすることができます。

5.5 GTZC 割込み

TZIC はセキュアペリフェラルなので、非セキュア・アクセスが行われると、自動的に不正アクセスイベントを生成します。MPCBB と TZSC は TrustZone を認識するペリフェラルであり、ペリフェラル内にセキュアレジスタと非セキュアレジスタが混在しています。

表 29. GTZC 割込みリクエスト

項目 (割込みの 略称)	割込み イベント	イベントフラグ	有効制御ビット	割込みのクリア方法	SLEEP モード の終了	STOP モード の終了	STANDBY モード の終了
GTZC	不正 アクセス	GTZC1_TZIC_SRx の すべてのフラグ	GTZC1_TZIC_IERx の すべてのビット	ビット GTZC1_TZIC_FCRx に 1 を書き込む	はい	はい	不可

5.6 GTZC1 TZSC レジスタ

すべてのレジスタは、必ずワード（32 ビット）単位でアクセスされます。

5.6.1 GTZC1 TZSC 制御レジスタ (GTZC1_TZSC_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCK
															rs

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **LCK** : GTZC1_TZSC_SECCFGRx および GTZC1_TZSC_PRIVCFGRx の設定を次のリセットまでロックします。

このビットはデフォルトではクリアされており、一旦、設定すると、システムリセットまでリセットできません。

0 : GTZC1_TZSC_SECCFGRx および GTZC1_TZSC_PRIVCFGRx がすべてロックされない設定。

1 : GTZC1_TZSC_SECCFGRx および GTZC1_TZSC_PRIVCFGRx がすべてロックされる設定。

5.6.2 GTZC1 TZSC セキュア設定レジスタ 1 (GTZC1_TZSC_SECCFGR1)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

書込みセキュアアクセスのみ。

このレジスタへの書込みは、対応する GTZC1_TZSC_PRIVCFGR の信号が 1 に設定されている場合、セキュア特権トランザクションによってのみ可能です。所定の PRIV ビットが設定されていない場合、相当する SEC ビットはセキュア非特権トランザクションによって書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM2 SEC	DTSSE C	UART1 2SEC	UART9 SEC	UART8 SEC	UART7 SEC	DAC1S EC	HDMIC ECSEC	USART 11SEC	USART 10SEC	USART 6SEC	CRSSE C	I3C1SE C	I2C2SE C	I2C1SE C	UART5 SEC
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART4 SEC	USART 3SEC	USART 2SEC	SPI3SE C	SPI2SE C	IWDGS EC	WWDG SEC	TIM14S EC	TIM13S EC	TIM12S EC	TIM7S EC	TIM6S EC	TIM5S EC	TIM4S EC	TIM3S EC	TIM2S EC
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **LPTIM2SEC** : LPTIM2 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

- ビット 30 **DTSSEC** : DTS のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 29 **UART12SEC** : UART12 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 28 **UART9SEC** : UART9 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 27 **UART8SEC** : UART8 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 26 **UART7SEC** : UART7 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 25 **DAC1SEC** : DAC1 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 24 **HDMICECSEC** : HDMICEC のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 23 **USART11SEC** : USART11 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 22 **USART10SEC** : USART10 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 21 **USART6SEC** : USART6 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 20 **CRSSEC** : CRS のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 19 **I3C1SEC** : I3C1 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 18 **I2C2SEC** : I2C2 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 17 **I2C1SEC** : I2C1 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 16 **UART5SEC** : UART5 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア

- ビット 15 **UART4SEC** : UART4 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 14 **USART3SEC** : USART3 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 13 **USART2SEC** : USART2 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 12 **SPI3SEC** : SPI3 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 11 **SPI2SEC** : SPI2 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 10 **IWDGSEC** : IWDG のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 9 **WWDGSEC** : WWDG のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 8 **TIM14SEC** : TIM14 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 7 **TIM13SEC** : TIM13 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 6 **TIM12SEC** : TIM12 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 5 **TIM7SEC** : TIM7 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 4 **TIM6SEC** : TIM6 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 3 **TIM5SEC** : TIM5 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 2 **TIM4SEC** : TIM4 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 1 **TIM3SEC** : TIM3 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア

ビット 0 **TIM2SEC** : TIM2 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

5.6.3 GTZC1 TZSC セキュア設定レジスタ 2 (GTZC1_TZSC_SECCFGR2)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

書込みセキュアアクセスのみ。

このレジスタへの書込みは、対応する GTZC1_TZSC_PRIVCFGR の信号が 1 に設定されている場合、セキュア特権トランザクションによってのみ可能です。所定の PRIV が設定されていない場合、相当する SEC ビットはセキュア非特権トランザクションによって書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM5 SEC	LPTIM4 SEC	LPTIM3 SEC	LPTIM1 SEC	I2C4SE C	I2C3SE C	LPUAR T1SEC	SPI5SE C	Res.	Res.	Res.	Res.	USBSE C	SAI2SE C	SAI1SE C	SPI6SE C
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4SE C	TIM17S EC	TIM16S EC	TIM15S EC	USART 1SEC	TIM8S EC	SPI1SE C	TIM1S EC	Res.	Res.	Res.	Res.	Res.	UCPDS EC	FDCAN 2SEC	FDCAN 1SEC
rw	rw	rw	rw	rw	rw	rw	rw						rw	rw	rw

ビット 31 **LPTIM5SEC** : LPTIM5 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 30 **LPTIM4SEC** : LPTIM4 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 29 **LPTIM3SEC** : LPTIM3 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 28 **LPTIM1SEC** : LPTIM1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 27 **I2C4SEC** : I2C4 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 26 **I2C3SEC** : I2C3 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 25 **LPUART1SEC** : LPUART のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 24 **SPI5SEC** : SPI5 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **USBSEC** : USB のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 18 **SAI2SEC** : SAI2 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 17 **SAI1SEC** : SAI1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 16 **SPI6SEC** : SPI6 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 15 **SPI4SEC** : SPI4 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 14 **TIM17SEC** : TIM17 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 13 **TIM16SEC** : TIM16 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 12 **TIM15SEC** : TIM15 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 11 **USART1SEC** : USART1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 10 **TIM8SEC** : TIM8 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 9 **SPI1SEC** : SPI1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 8 **TIM1SEC** : TIM1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **UCPDSEC** : UCPD のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 1 **FDCAN2SEC** : FDCAN2 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 0 **FDCAN1SEC** : FDCAN1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

5.6.4 GTZC1 TZSC セキュア設定レジスタ 3 (GTZC1_TZSC_SECCFGR3)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

書込みセキュアアクセスのみ。

このレジスタへの書込みは、対応する GTZC1_TZSC_PRIVCFGR が 1 に設定されている場合、セキュア特権トランザクションによってのみ可能です。所定の PRIV が設定されていない場合、相当する SEC ビットはセキュア非特権トランザクションによって書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RAMC FGSEC	Res.	OCTOS PI1SEC C	FMCSE C	SDMM C2SEC	SDMM C1SEC	PKASE C	SAESS EC	RNGS EC	HASHS EC	AESSE C
					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCMIS EC	ADC12 SEC	DCAC HESEC	ICACH ESEC	ETHSE C	FMACS EC	CORDI CSEC	CRCSE C	Res.	Res.	Res.	Res.	Res.	Res.	VREFB UFSEC	LPTIM6 SEC
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **RAMCFGSEC** : RAMSCFG のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OCTOSPI1SEC** : OCTOSPI1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 23 **FMCSEC** : FMC のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 22 **SDMMC2SEC** : SDMMC2 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

ビット 21 **SDMMC1SEC** : SDMMC1 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

- ビット 20 **PKASEC** : PKA のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 19 **SAESSEC** : SAES のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 18 **RNGSEC** : RNG のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 17 **HASHSEC** : HASH のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 16 **AESSEC** : AES のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 15 **DCMISEC** : DCMI のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 14 **ADC12SEC** : ADC1 および ADC2 のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 13 **DCACHESEC** : DCACHE のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 12 **ICACHESEC** : ICACHE のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 11 **ETHSEC** : ETH のレジスタのセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 10 **FMACSEC** : FMAC のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 9 **CORDICSEC** : CORDIC のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 8 **CRCSEC** : CRC のセキュアアクセスモード
0 : 非セキュア
1 : セキュア
- ビット 7:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **VREFBUFSEC** : VREFBUF のセキュアアクセスモード
0 : 非セキュア
1 : セキュア

ビット 0 **LPTIM6SEC** : LPTIM6 のセキュアアクセスモード

0 : 非セキュア

1 : セキュア

5.6.5 GTZC1 TZSC 特権設定レジスタ 1 (GTZC1_TZSC_PRIVCFGR1)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

書込み特権アクセスのみ。

このレジスタは、対応する GTZC1_TZSC_SECCFGRの信号が 1 に設定されている場合、セキュア特権トランザクションによってのみ読出しや書込みができます。所定の SEC ビットが設定されていない場合、相当する PRIV ビットは非セキュア特権トランザクションによって読出しや書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM2 PRIV	DTSPR IV	UART1 2PRIV	UART9 PRIV	UART8 PRIV	UART7 PRIV	DAC1P RIV	HDMIC ECPRI V	USART 11PRIV	USART 10PRIV	USART 6PRIV	CRSPR IV	I3C1PR IV	I2C2PR IV	I2C1PR IV	UART5 PRIV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART4 PRIV	USART 3PRIV	USART 2PRIV	SPI3P RIV	SPI2P RIV	IWDGP RIV	WWDG PRIV	TIM14P RIV	TIM13P RIV	TIM12P RIV	TIM7P RIV	TIM6P RIV	TIM5P RIV	TIM4P RIV	TIM3P RIV	TIM2P RIV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **LPTIM2PRIV** : LPTIM2 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 30 **DTSPRIV** : DTS の特権アクセスモード

0 : 非特権

1 : 特権

ビット 29 **UART12PRIV** : UART12 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 28 **UART9PRIV** : UART9 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 27 **UART8PRIV** : UART8 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 26 **UART7PRIV** : UART7 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 25 **DAC1PRIV** : DAC1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 24 **HDMICECPRIV** : HDMICEC の特権アクセスモード

0 : 非特権

1 : 特権

ビット 23 **USART11PRIV** : USART11 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 22 **USART10PRIV** : USART10 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 21 **USART6PRIV** : USART6 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 20 **CRSPRIV** : CRS の特権アクセスモード

0 : 非特権

1 : 特権

ビット 19 **I3C1PRIV** : I3C1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 18 **I2C2PRIV** : I2C2 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 17 **I2C1PRIV** : I2C1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 16 **UART5PRIV** : UART5 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 15 **UART4PRIV** : UART4 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 14 **USART3PRIV** : USART3 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 13 **USART2PRIV** : USART2 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 12 **SPI3PRIV** : SPI3 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 11 **SPI2PRIV** : SPI2 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 10 **IWDGPRIV** : IWDG の特権アクセスモード

0 : 非特権

1 : 特権

ビット 9 **WWDGPRIV** : WWDG の特権アクセスモード

0 : 非特権

1 : 特権

ビット 8 **TIM14PRIV** : TIM14 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 7 **TIM13PRIV** : TIM13 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 6 **TIM12PRIV** : TIM12 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 5 **TIM7PRIV** : TIM7 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 4 **TIM6PRIV** : TIM6 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 3 **TIM5PRIV** : TIM5 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 2 **TIM4PRIV** : TIM4 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 1 **TIM3PRIV** : TIM3 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 0 **TIM2PRIV** : TIM2 の特権アクセスモード

0 : 非特権

1 : 特権

5.6.6 GTZC1 TZSC 特権設定レジスタ 2 (GTZC1_TZSC_PRIVCFGR2)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

書込み特権アクセスのみ。

このレジスタは、対応する GTZC1_TZSC_SECCFGR の信号が 1 に設定されている場合、セキュア特権トランザクションによってのみ読出しや書込みができます。所定の SEC ビットが設定されていない場合、相当する PRIV ビットは非セキュア特権トランザクションによって読出しや書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM5 PRIV	LPTIM4 PRIV	LPTIM3 PRIV	LPTIM1 PRIV	I2C4PR IV	I2C3PR IV	LPUAR T1PRIV	SPI5P RIV	Res.	Res.	Res.	Res.	USBPR IV	SAI2P RIV	SAI1P RIV	SPI6P RIV
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4P RIV	TIM17P RIV	TIM16P RIV	TIM15P RIV	USART 1PRIV	TIM8P RIV	SPI1P RIV	TIM1P RIV	Res.	Res.	Res.	Res.	Res.	UCPD RIV	FDCAN 2PRIV	FDCAN 1PRIV
rw	rw	rw	rw	rw	rw	rw	rw						rw	rw	rw

ビット 31 **LPTIM5PRIV** : LPTIM5 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 30 **LPTIM4PRIV** : LPTIM4 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 29 **LPTIM3PRIV** : LPTIM3 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 28 **LPTIM1PRIV** : LPTIM1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 27 **I2C4PRIV** : I2C4 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 26 **I2C3PRIV** : I2C3 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 25 **LPUART1PRIV** : LPUART の特権アクセスモード

0 : 非特権

1 : 特権

ビット 24 **SPI5PRIV** : SPI5 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

- ビット 19 **USBPRIV** : USB の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 18 **SAI2PRIV** : SAI2 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 17 **SAI1PRIV** : SAI1 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 16 **SPI6PRIV** : SPI6 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 15 **SPI4PRIV** : SPI4 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 14 **TIM17PRIV** : TIM17 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 13 **TIM16PRIV** : TIM16 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 12 **TIM15PRIV** : TIM15 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 11 **USART1PRIV** : USART1 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 10 **TIM8PRIV** : TIM8 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 9 **SPI1PRIV** : SPI1 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 8 **TIM1PRIV** : TIM1 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 7:3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **UCPDPRIV** : UCPD の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 1 **FDCAN2PRIV** : FDCAN2 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 0 **FDCAN1PRIV** : FDCAN1 の特権アクセスモード
0 : 非特権
1 : 特権

5.6.7 GTZC1 TZSC 特権設定レジスタ 3 (GTZC1_TZSC_PRIVCFGR3)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

書込み特権アクセスのみ。

このレジスタは、対応する GTZC1_TZSC_SECCFGR の信号が 1 に設定されている場合、セキュア特権トランザクションによってのみ読出しや書込みができます。所定の SEC ビットが設定されていない場合、相当する PRIV ビットは非セキュア特権トランザクションによって読出しや書込みができます。

読出しアクセスは、セキュア／非セキュア、特権／非特権のどのタイプのトランザクションにも許可されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RAMC FG PRIV	Res.	OCTOS PI1 PRIV	FMC PRIV	SDMM C2 PRIV	SDMM C1 PRIV	PKA PRIV	SAES PRIV	RNG PRIV	HASH PRIV	AES PRIV
					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCMI PRIV	ADC12 PRIV	DCAC HE PRIV	ICACH E PRIV	ETH PRIV	FMAC PRIV	CORDI C PRIV	CRC PRIV	Res.	Res.	Res.	Res.	Res.	Res.	VREF BUF PRIV	LPTIM6 PRIV
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **RAMCFGPRIV** : RAMSCFG の特権アクセスモード

0 : 非特権

1 : 特権

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OCTOSPI1PRIV** : OCTOSPI1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 23 **FMCPRIV** : FMC の特権アクセスモード

0 : 非特権

1 : 特権

ビット 22 **SDMMC2PRIV** : SDMMC2 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 21 **SDMMC1PRIV** : SDMMC1 の特権アクセスモード

0 : 非特権

1 : 特権

ビット 20 **PKAPRIV** : PKA の特権アクセスモード

0 : 非特権

1 : 特権

ビット 19 **SAESPRIV** : SAES の特権アクセスモード

0 : 非特権

1 : 特権

- ビット 18 **RNGPRIV** : RNG の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 17 **HASHPRIV** : HASH の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 16 **AESPRIV** : AES の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 15 **DCMIPRIV** : DCMI の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 14 **ADC12PRIV** : ADC1 および ADC2 の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 13 **DCACHEPRIV** : DCACHE の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 12 **ICACHEPRIV** : ICACHE の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 11 **ETHPRIV** : ETH のレジスタの特権アクセスモード
0 : 非特権
1 : 特権
- ビット 10 **FMACPRIV** : FMAC の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 9 **CORDICPRIV** : CORDIC の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 8 **CRCPRIV** : CRC の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 7:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **VREFBUFPRIV** : VREFBUF の特権アクセスモード
0 : 非特権
1 : 特権
- ビット 0 **LPTIM6PRIV** : LPTIM6 の特権アクセスモード
0 : 非特権
1 : 特権

5.6.8 GTZC1 TZSC メモリ x サブ領域 z ウォーターマーク設定レジスタ (GTZC1_TZSC_MPCWMxzCFGR) (z = A から B)

アドレスオフセット：ブロック A : $0x40 + 0x10 \times (x - 1)$ (x = 1 から 4)

アドレスオフセット：ブロック B : $0x48 + 0x10 \times (x - 1)$ (x = 1 から 4)

リセット値：0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	PRIV	SEC	Res.	Res.	Res.	Res.	Res.	Res.	SRLOCK	SREN
						rw	rw							rs	rw

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **PRIV**：ベース領域 x の特権サブ領域 z

このビットは SREN が設定されている場合のみ考慮されます。

0：サブ領域 z には特権および非特権アクセスが認められています。

1：領域 x のサブ領域 z には特権アクセスのみが認められています。

ビット 8 **SEC**：ベース領域 x のセキュアサブ領域 z

このビットは SREN が設定されている場合のみ考慮されます。

0：領域 x のサブ領域 z には非セキュア・データ・アクセスのみが認められています。

1：領域 x のサブ領域 z にはセキュア・データ・アクセスのみが認められています。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SRLOCK**：サブ領域 z のロック

このビットが一旦設定されると、システムリセットでしかクリアできません。

0：GTZC1_TZSC_MPCWMxCFGR、GTZC1_TZSC_MPCWMxAR、GTZC1_TZSC_MPCWMxBR への書き込みが可能。

1：GTZC1_TZSC_MPCWMxCFGR、GTZC1_TZSC_MPCWMxAR、GTZC1_TZSC_MPCWMxBR への書き込みを無視。

ビット 0 **SREN**：サブ領域 z の有効化

0：サブ領域 z を無効化。ベース領域 x のアクセス制御が、このサブ領域の開始アドレスから終了アドレスの間のあらゆるアクセスに適用されます。

1：領域 x のサブ領域 z を有効化。GTZC1_TZSC_MPCWMxCFGR で定義されたアクセス制御が、GTZC1_TZSC_MPCWMxAR と GTZC1_TZSC_MPCWMxBR の両方で定義された、このサブ領域の開始アドレスと終了アドレスの間のあらゆるアクセスに適用されます。

注： ウォーターマーク制御される外部メモリは TZEN = 0xC3 の場合、リセット時点で完全に非セキュア／非特権な状態を開始します。TZEN = 0xB4 のとき、外部メモリは完全にセキュア／特権的な状態を開始します（リセット値の反転）。

5.6.9 GTZC1 TZSC メモリ x サブ領域 A ウォーターマーク・レジスタ (GTZC1_TZSC_MPCWMxAR)

アドレスオフセット : $0x44 + 0x10 \times (x - 1)$ ($x = 1$ から 4)

リセット値 : 0x0000 0000

指定のリセット値は TZEN = 0xB4 のときに有効です。TZEN = 0xC3 のときのリセット値は 0x0800 0000 です。

セキュア特権アクセスのみ。

SUBA_START + SUBA_LENGTH がメモリの最大許容サイズより大きい場合、自動的に SUBA_LENGTH の飽和が適用されます。

サブ領域 A と B にオーバーラップが存在する場合、両方のサブ領域のセキュア／特権属性が共通部分に適用されます (セクション 5.4.3 を参照してください)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SUBA_LENGTH[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SUBA_START[10:0]										
					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **SUBA_LENGTH[11:0]** : 領域 x のサブ領域 A の長さ

このフィールドはサブ領域 A の長さを定義し、表 25 で定義された粒度を掛け算します。

SUBA_START + SUBA_LENGTH がメモリに許されている最大許容サイズより大きい場合、

SUBA_LENGTH の飽和が自動的に適用されます。

SUBA_LENGTH = 0 の場合、サブ領域 A が無効になります (GTZC1_TZSC_MPCMWxACFGR の SREN ビットがクリアされます)。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **SUBA_START[10:0]** : 領域 x のサブ領域 A の開始

領域 x の開始に対し、このフィールドでサブ領域 A のアドレス・オフセットを定義して、表 25 で定義される粒度を掛け算します。

ウォーターマーク制御される外部メモリは、TZEN = 0xC3 の場合、リセット時点で完全に非セキュアな状態を開始します。TZEN = 0xB4 のとき、外部メモリは完全にセキュアな状態を開始します (リセット値の反転)。

5.6.10 GTZC1 TZSC メモリ x サブ領域 B ウォーターマーク・レジスタ (GTZC1_TZSC_MPCWMxBR)

アドレスオフセット : $0x4C + 0x10 \times (x - 1)$ ($x = 1$ から 4)

リセット値 : $0x0000\ 0000$

指定のリセット値は $TZEN = 0xB4$ のときに有効です。 $TZEN = 0xC3$ のときのリセット値は $0x0800\ 0000$ です。

セキュア特権アクセスのみ。

SUBB_START + SUBB_LENGTH がメモリの最大許容サイズより大きい場合、自動的に SUBB_LENGTH の飽和が適用されます。

サブ領域 A と B にオーバーラップが存在する場合、両方のサブ領域のセキュア／特権属性が共通部分に適用されます (セクション 5.4.3 を参照してください)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SUBB_LENGTH[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SUBB_START[10:0]										
					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **SUBB_LENGTH[11:0]** : 領域 x のサブ領域 B の長さ

このフィールドはサブ領域 B の長さを定義し、表 25 で定義された粒度を掛け算します。

SUBB_START + SUBB_LENGTH がメモリに許されている最大許容サイズより大きい場合、

SUBB_LENGTH の飽和が自動的に適用されます。

SUBB_LENGTH = 0 の場合、サブ領域 B が無効になります (GTZC1_TZSC_MPCMWxBCFGR の SREN ビットがクリアされます)。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **SUBB_START[10:0]** : 領域 x のサブ領域 B の開始

領域 x の開始に対し、このフィールドでサブ領域 B のアドレスオフセットを定義して、表 25 で定義される粒度を掛け算します。

ウォーターマーク制御される外部メモリは、 $TZEN = 0xC3$ の場合、リセット時点で完全に非セキュアな状態を開始します。 $TZEN = 0xB4$ のとき、外部メモリは完全にセキュアな状態を開始します (リセット値の反転)。

5.6.11 GTZC1 TZSC レジスタのマップ

表 30. GTZC1 TZSC レジスタのマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	GTZC1_TZSC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	0	LOCK
	リセット値																																	0
0x004 - 0x00C	予約済み	予約済み																																
0x010	GTZC1_TZSC_SECCFGR1	LPTIM2SEC	DT5SEC	UART12SEC	UART9SEC	UART8SEC	UART7SEC	DAC1SEC	HDMICECSEC	USART11SEC	USART10SEC	USART6SEC	CRSSEC	I3C1SEC	I2C2SEC	I2C1SEC	UART5SEC	UART4SEC	USART3SEC	USART2SEC	SP13SEC	SP12SEC	IWDGSEC	WWDGSEC	TIM14SEC	TIM13SEC	TIM112SEC	TIM7SEC	TIM6SEC	TIM5SEC	TIM4SEC	TIM3SEC	TIM2SEC	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	GTZC1_TZSC_SECCFGR2	LPTIM5SEC	LPTIM4SEC	LPTIM3SEC	LPTIM1SEC	I2C4SEC	I2C3SEC	LPUART1SEC	SP15SEC	Res	Res	Res	Res	USBSEC	SAI2SEC	SAI1SEC	SP16SEC	SP14SEC	TIM17SEC	TIM16SEC	TIM15SEC	USART11SEC	TIM8SEC	SP11SEC	TIM1SEC	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	GTZC1_TZSC_SECCFGR3	Res	Res	Res	Res	Res	RAMCFGSEC	Res	OCTOSP11SEC	FMCSEC	SDMMC2SEC	SDMMC1SEC	PKASEC	SAESSEC	RNGSEC	HASHSEC	AESSEC	DCM1SEC	ADC12SEC	DCACHESEC	ICACHESEC	ETHSEC	FMACSEC	CORDICSEC	CRCSEC	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	予約済み	予約済み																																
0x020	GTZC1_TZSC_PRIVCFGR1	LPTIM2PRIV	DTSPRIV	UART12PRIV	UART9PRIV	UART8PRIV	UART7PRIV	DAC1PRIV	HDMICECPRIV	USART11PRIV	USART10PRIV	USART6PRIV	CRSPRIV	I3C1PRIV	I2C2PRIV	I2C1PRIV	UART5PRIV	UART4PRIV	USART3PRIV	USART2PRIV	SP13PRIV	SP12PRIV	IWDGPRIV	WWDGPRIV	TIM14PRIV	TIM13PRIV	TIM112PRIV	TIM7PRIV	TIM6PRIV	TIM5PRIV	TIM4PRIV	TIM3PRIV	TIM2PRIV	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	GTZC1_TZSC_PRIVCFGR2	LPTIM5PRIV	LPTIM4PRIV	LPTIM3PRIV	LPTIM1PRIV	I2C4PRIV	I2C3PRIV	LPUART1PRIV	SP15PRIV	Res	Res	Res	Res	USBPRIV	SAI2PRIV	SAI1PRIV	SP16PRIV	SP14PRIV	TIM17PRIV	TIM16PRIV	TIM15PRIV	USART11PRIV	TIM8PRIV	SP11PRIV	TIM1PRIV	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	GTZC1_TZSC_PRIVFGR3	Res	Res	Res	Res	Res	RAMCFGPRIV	Res	OCTOSP11PRIV	FMCPRIV	SDMMC2PRIV	SDMMC1PRIV	PKAPRIV	SAESPRIV	RNGPRIV	HASHPRIV	AESPRIV	DCM1PRIV	ADC12PRIV	DCACHEPRIV	ICACHEPRIV	ETHPRIV	FMACPRIV	CORDICPRIV	CRCPRIV	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C - 0x03C	予約済み	予約済み																																
0x040	GTZC1_TZSC_MPCWM1ACFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN	
	リセット値																						0	0								0	0	0
0x44	GTZC1_TZSC_MPCWM1AR	Res	Res	Res	Res	SUBA_LENGTH[11:0]														Res	Res	Res	Res	Res	SUBA_START[10:0]									
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	GTZC1_TZSC_MPCWM1BCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN	
	リセット値																						0	0								0	0	0
0x04C	GTZC1_TZSC_MPCWM1BR	Res	Res	Res	Res	SUBB_LENGTH[11:0]														Res	Res	Res	Res	Res	SUBB_START[10:0]									
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	GTZC1_TZSC_MPCWM2ACFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN	
	リセット値																						0	0								0	0	0

表 30. GTZC1 TZSC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x54	GTZC1_TZSC_MPCWM2AR	Res	Res	Res	Res	SUBA_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBA_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0
0x058	GTZC1_TZSC_MPCWM2BCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN
	リセット値																							0	0								0	0
0x05C	GTZC1_TZSC_MPCWM2BR	Res	Res	Res	Res	SUBB_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBB_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x060	GTZC1_TZSC_MPCWM3ACFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN
	リセット値																							0	0								0	0
0x64	GTZC1_TZSC_MPCWM3AR	Res	Res	Res	Res	SUBA_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBA_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x068	GTZC1_TZSC_MPCWM3BCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN
	リセット値																							0	0								0	0
0x06C	GTZC1_TZSC_MPCWM3BR	Res	Res	Res	Res	SUBB_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBB_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x070	GTZC1_TZSC_MPCWM4ACFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN
	リセット値																							0	0								0	0
0x74	GTZC1_TZSC_MPCWM4AR	Res	Res	Res	Res	SUBA_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBA_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x078	GTZC1_TZSC_MPCWM4BCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV	SEC	Res	Res	Res	Res	Res	Res	Res	SRLOCK	SREN
	リセット値																							0	0								0	0
0x07C	GTZC1_TZSC_MPCWM4BR	Res	Res	Res	Res	SUBB_LENGTH[11:0]										Res	Res	Res	Res	Res	SUBB_START[10:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0

表 24 : GTZC1 サブブロック・アドレスオフセットを参照してください。

5.7 GTZC1 TZIC レジスタ

すべてのレジスタは、必ずワード（32 ビット）単位でアクセスされます。

5.7.1 GTZC1 TZIC 割込み有効レジスタ 1 (GTZC1_TZIC_IER1)

アドレスオフセット：0x000

リセット値：0x0000 0000

セキュア特権アクセスのみ。

このレジスタは、不正アクセスの割込みを有効にするために使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM2 IE	DTSIE	UART1 2IE	UART9 IE	UART8 IE	UART7 IE	DAC1 IE	HDMIC ECIE	USART 11IE	USART 10IE	USART 6IE	CRSIE	I3C1IE	I2C2IE	I2C1IE	UART5 IE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART4 IE	USART 3IE	USART 2IE	SPI3IE	SPI2IE	IWDG IE	WWDG IE	TIM14 IE	TIM13 IE	TIM12 IE	TIM7IE	TIM6IE	TIM5IE	TIM4IE	TIM3IE	TIM2IE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **LPTIM2IE** : LPTIM2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 30 **DTSIE** : DTS の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 29 **UART12IE** : UART12 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 28 **UART9IE** : UART9 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 27 **UART8IE** : UART8 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 26 **UART7IE** : UART7 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 25 **DAC1IE** : DAC1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 24 **HDMICECIE** : HDMICEC の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 23 **USART11IE** : USART11 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 22 **USART10IE** : USART10 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 21 **USART6IE** : USART6 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 20 **CRSIE** : CRS の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 19 **I3C1IE** : I3C1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 18 **I2C2IE** : I2C2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 17 **I2C1IE** : I2C1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 16 **UART5IE** : UART5 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 15 **UART4IE** : UART4 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 14 **USART3IE** : USART3 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 13 **USART2IE** : USART2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 12 **SPI3IE** : SPI3 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 11 **SPI2IE** : SPI2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 10 **IWDGIE** : IWDG の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 9 **WWDGIE** : WWDG の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 8 **TIM14IE** : TIM14 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 7 **TIM13IE** : TIM13 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 6 **TIM12IE** : TIM12 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 5 **TIM7IE** : TIM7 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 4 **TIM6IE** : TIM6 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 3 **TIM5IE** : TIM5 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 2 **TIM4IE** : TIM4 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 1 **TIM3IE** : TIM3 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 0 **TIM2IE** : TIM2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

5.7.2 GTZC1 TZIC 割込み有効レジスタ 2 (GTZC1_TZIC_IER2)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

このレジスタは、不正アクセスの割込みを有効にするために使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM5IE	LPTIM4IE	LPTIM3IE	LPTIM1IE	I2C4IE	I2C3IE	LPUART1IE	SPI5IE	Res.	Res.	Res.	Res.	USBIE	SAI2IE	SAI1IE	SPI6IE
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4IE	TIM17IE	TIM16IE	TIM15IE	USART1IE	TIM8IE	SPI1IE	TIM1IE	Res.	Res.	Res.	Res.	Res.	UCPDIE	FDCAN2IE	FDCAN1IE
rw	rw	rw	rw	rw	rw	rw	rw						rw	rw	rw

ビット 31 **LPTIM5IE** : LPTIM5 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

- ビット 30 **LPTIM4IE** : LPTIM4 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 29 **LPTIM3IE** : LPTIM3 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 28 **LPTIM1IE** : LPTIM1 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 27 **I2C4IE** : I2C4 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 26 **I2C3IE** : I2C3 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 25 **LPUART1IE** : LPUART の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 24 **SPI5IE** : SPI5 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 23:20 予約済みであり、リセット値に保持する必要があります。
- ビット 19 **USBIE** : USB の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 18 **SAI2IE** : SAI2 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 17 **SAI1IE** : SAI1 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 16 **SPI6IE** : SPI6 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 15 **SPI4IE** : SPI4 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 14 **TIM17IE** : TIM17 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 13 **TIM16IE** : TIM16 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。

ビット 12 **TIM15IE** : TIM15 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 11 **USART1IE** : USART1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 10 **TIM8IE** : TIM8 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 9 **SPI1IE** : SPI1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 8 **TIM1IE** : TIM1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **UCPDIE** : UCPD の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 1 **FDCAN2IE** : FDCAN2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 0 **FDCAN1IE** : FDCAN1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

5.7.3 GTZC1 TZIC 割込み有効レジスタ 3 (GTZC1_TZIC_IER3)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

このレジスタは、不正アクセスの割込みを有効にするために使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RAMC FGIE	Res.	OCTOS PI1IE	FMCIE	SDMM C2IE	SDMM C1IE	PKAIE	SAESI E	RNGIE	HASHI E	AESIE
					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCMIIE	ADC12I E	DCAC HEIE	ICACH EIE	ETHIE	FMACI E	CORDI CIE	CRCIE	Res.	Res.	Res.	Res.	Res.	Res.	VREFB UFIE	LPTIM6 IE
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **RAMCFGIE** : RAMSCFG の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OCTOSPI1IE** : OCTOSPI1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 23 **FMCIE** : FMC の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 22 **SDMMC2IE** : SDMMC2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 21 **SDMMC1IE** : SDMMC1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 20 **PKAIE** : PKA の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 19 **SAESIE** : SAES の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 18 **RNGIE** : RNG の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 17 **HASHIE** : HASH の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 16 **AESIE** : AES の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 15 **DCMIIE** : DCMI の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 14 **ADC12IE** : ADC1 および ADC2 の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 13 **DCACHEIE** : DCACHE の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 12 **ICACHEIE** : ICACHE の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 11 **ETHIE** : ETH のレジスタの不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 10 **FMACIE** : FMAC の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 9 **CORDICIE** : CORDIC の不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 8 **CRCIE** : CRC の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **VREFBUFIE** : VREFBUF の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 0 **LPTIM6IE** : LPTIM6 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

5.7.4 GTZC1 TZIC 割込み有効レジスタ 4 (GTZC1_TZIC_IER4)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

このレジスタは、不正アクセスの割込みを有効にするために使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	MPCB B3_RE GIE	SRAM3 IE	MPCB B2_RE GIE	SRAM2 IE	MPCB B1_RE GIE	SRAM1 IE	Res.	Res.	Res.	BKPSR AMIE	FMC_ MEMIE	OCTOS PI1_M EMIE	TZIC1I E	TZSC1I E
		rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXTIIE	RCCIE	PWRIE	TAMPI E	RTCIE	SBSIE	Res.	OTFDE C1IE	FLASHI E	Flash_ REGIE	GPDM A2IE	GPDM A1IE
				rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **MPCBB3_REGIE** : MPCBB3 レジスタの不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 28 **SRAM3IE** : SRAM3 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 27 **MPCBB2_REGIE** : MPCBB2 レジスタの不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

- ビット 26 **SRAM2IE** : SRAM2 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 25 **MPCBB1_REGIE** : MPCBB1 レジスタの不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 24 **SRAM1IE** : SRAM1 の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 23:21 予約済みであり、リセット値に保持する必要があります。
- ビット 20 **BKPSRAMIE** : MPCWM4 (BKPSRAM) メモリ・バンクへの不正アクセス割込みの有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 19 **FMC_MEMIE** : MPCWM2 (FMC_NOR バンク)、MPCWM3 (FMC_NAND バンク、および FMC_SDRAM バンク 1)、MPCWM4 (FMC_SDRAM バンク 2) の不正アクセス割込みの有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 18 **OCTOSPI1_MEMIE** : MPCWM1 (OCTOSPI1) メモリ・バンクの不正アクセス割込みの有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 17 **TZIC1IE** : GTZC1 TZIC レジスタの不正アクセス割込みの有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 16 **TZSC1IE** : GTZC1 TZSC レジスタの不正アクセス割込みの有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 15:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **EXTIIE** : EXTI の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 10 **RCCIE** : RCC の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 9 **PWRIE** : PWR の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 8 **TAMPIE** : TAMP の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 7 **RTCIE** : RTC の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 6 **SBSIE** : SBS の不正アクセス割込み有効化
0 : 割込みは無効です。
1 : 割込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OTFDEC1IE** : OTFDEC1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 3 **FLASHIE** : FLASH メモリの不正アクセス割込みの有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 2 **Flash_REGIE** : Flash レジスタの不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 1 **GPDMA2IE** : GPDMA2 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 0 **GPDMA1IE** : GPDMA1 の不正アクセス割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

5.7.5 GTZC1 TZIC ステータス・レジスタ 1 (GTZC1_TZIC_SR1)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM2 F	DTSF	UART1 2F	UART9 F	UART8 F	UART7 F	DAC1F	HDMIC ECF	USART 11F	USART 10F	USART 6F	CRSF	I3C1F	I2C2F	I2C1F	UART5 F
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART4 F	USART 3F	USART 2F	SPI3F	SPI2F	IWDGF	WWDG F	TIM14F	TIM13F	TIM12F	TIM7F	TIM6F	TIM5F	TIM4F	TIM3F	TIM2F
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 **LPTIM2F** : LPTIM2 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 30 **DTSF** : DTS の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 29 **UART12F** : UART12 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 28 **UART9F** : UART9 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 27 **UART8F** : UART8 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

- ビット 26 **UART7F** : UART7 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 25 **DAC1F** : DAC1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 24 **HDMICECF** : HDMICEC の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 23 **USART11F** : USART11 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 22 **USART10F** : USART10 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 21 **USART6F** : USART6 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 20 **CRSF** : CRS の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 19 **I3C1F** : I3C1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 18 **I2C2F** : I2C2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 17 **I2C1F** : I2C1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 16 **UART5F** : UART5 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 15 **UART4F** : UART4 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 14 **USART3F** : USART3 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 13 **USART2F** : USART2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 12 **SPI3F** : SPI3 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント

- ビット 11 **SPI2F** : SPI2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 10 **IWDGF** : IWDG の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 9 **WWDGF** : WWDG の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 8 **TIM14F** : TIM14 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 7 **TIM13F** : TIM13 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 6 **TIM12F** : TIM12 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 5 **TIM7F** : TIM7 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 4 **TIM6F** : TIM6 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 3 **TIM5F** : TIM5 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 2 **TIM4F** : TIM4 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 1 **TIM3F** : TIM3 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 0 **TIM2F** : TIM2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント

5.7.6 GTZC1 TZIC ステータス・レジスタ 2 (GTZC1_TZIC_SR2)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM5 F	LPTIM4 F	LPTIM3 F	LPTIM1 F	I2C4F	I2C3F	LPUAR T1F	SPI5F	Res.	Res.	Res.	Res.	USBF	SAI2F	SAI1F	SPI6F
r	r	r	r	r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4F	TIM17F	TIM16F	TIM15F	USART 1F	TIM8F	SPI1F	TIM1F	Res.	Res.	Res.	Res.	Res.	UCPDF	FDCAN 2F	FDCAN 1F
r	r	r	r	r	r	r	r						r	r	r

ビット 31 **LPTIM5F** : LPTIM5 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 30 **LPTIM4F** : LPTIM4 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 29 **LPTIM3F** : LPTIM3 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 28 **LPTIM1F** : LPTIM1 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 27 **I2C4F** : I2C4 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 26 **I2C3F** : I2C3 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 25 **LPUART1F** : LPUART の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 24 **SPI5F** : SPI5 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **USBF** : USB の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 18 **SAI2F** : SAI2 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

- ビット 17 **SAI1F** : SAI1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 16 **SPI6F** : SPI6 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 15 **SPI4F** : SPI4 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 14 **TIM17F** : TIM17 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 13 **TIM16F** : TIM16 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 12 **TIM15F** : TIM15 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 11 **USART1F** : USART1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 10 **TIM8F** : TIM8 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 9 **SPI1F** : SPI1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 8 **TIM1F** : TIM1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 7:3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **UCPDF** : UCPD の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 1 **FDCAN2F** : FDCAN2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 0 **FDCAN1F** : FDCAN1 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント

5.7.7 GTZC1 TZIC ステータス・レジスタ 3 (GTZC1_TZIC_SR3)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RAMC FGF	Res.	OCTOS PI1F	FMCF	SDMM C2F	SDMM C1F	PKAF	SAESF	RNGF	HASHF	AESF
					r		r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCMIF	ADC12 F	DCAC HEF	ICACH EF	ETHF	FMACF	CORDI CF	CRCF	Res.	Res.	Res.	Res.	Res.	Res.	VREFB UFF	LPTIM6 F
r	r	r	r	r	r	r	r							r	r

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **RAMCFGF** : RAMSCFG の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OCTOSPI1F** : OCTOSPI1 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 23 **FMCF** : FMC の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 22 **SDMMC2F** : SDMMC2 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 21 **SDMMC1F** : SDMMC1 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 20 **PKAF** : PKA の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 19 **SAESF** : SAES の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 18 **RNGF** : RNG の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 17 **HASHF** : HASH の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

- ビット 16 **AESF** : AES の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 15 **DCMIF** : DCMI の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 14 **ADC12F** : ADC1 および ADC2 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 13 **DCACHEF** : DCACHE の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 12 **ICACHEF** : ICACHE の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 11 **ETHF** : ETH のレジスタの不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 10 **FMACF** : FMAC の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 9 **CORDICF** : CORDIC の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 8 **CRCF** : CRC の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 7:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **VREFBUFF** : VREFBUF の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント
- ビット 0 **LPTIM6F** : LPTIM6 の不正アクセスフラグ
0 : 不正アクセスイベントなし
1 : 不正アクセスイベント

5.7.8 GTZC1 TZIC ステータス・レジスタ 4 (GTZC1_TZIC_SR4)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	MPCBB3_REGF	SRAM3F	MPCBB2_REGF	SRAM2F	MPCBB1_REGF	SRAM1F	Res.	Res.	Res.	BKPSRAMF	FMC_MEMF	OCTOSPI1_MEMF	TZIC1F	TZSC1F
		r	r	r	r	r	r				r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXTIF	RCCF	PWRF	TAMPF	RTCF	SBSF	Res.	OTFDEC1F	FLASHF	Flash_REGF	GPDM_A2F	GPDM_A1F
				r	r	r	r	r	r		r	r	r	r	r

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **MPCBB3_REGF** : MPCBB3 レジスタの不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 28 **SRAM3F** : SRAM3 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 27 **MPCBB2_REGF** : MPCBB2 レジスタの不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 26 **SRAM2F** : SRAM2 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 25 **MPCBB1_REGF** : MPCBB1 レジスタの不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 24 **SRAM1F** : SRAM1 の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **BKPSRAMF** : MPCWM4 (BKPSRAM) メモリ・バンクの不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 19 **FMC_MEMF** : MPCWM2 (FMC_NOR バンク)、MPCWM3 (FMC_NAND バンク、および FMC_SDRAM バンク 1)、MPCWM4 (FMC_SDRAM バンク 2) の不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 18 **OCTOSPI1_MEMF** : MPCWM1 (OCTOSPI1) メモリ・バンクの不正アクセスフラグ

- 0 : 不正アクセスイベントなし
- 1 : 不正アクセスイベント

ビット 17 **TZIC1F** : GTZC1 TZIC レジスタの不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 16 **TZSC1F** : GTZC1 TZSC レジスタの不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **EXTIF** : EXTI の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 10 **RCCF** : RCC の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 9 **PWRF** : PWR の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 8 **TAMPF** : TAMP の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 7 **RTCF** : RTC の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 6 **SBSF** : SBS の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OTFDEC1F** : OTFDEC1 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 3 **FLASHF** : FLASH メモリの不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 2 **Flash_REGF** : Flash レジスタの不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 1 **GPDMA2F** : GPDMA2 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

ビット 0 **GPDMA1F** : GPDMA1 の不正アクセスフラグ

0 : 不正アクセスイベントなし

1 : 不正アクセスイベント

5.7.9 GTZC1 TZIC フラグ・クリア・レジスタ 1 (GTZC1_TZIC_FCR1)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLPTI M2F	CDTSF	CUART 12F	CUART 9F	CUART 8F	CUART 7F	CDAC1 F	CHDMI CECF	CUSAR T11F	CUSAR T10F	CUSAR T6F	CCRSF	CI3C1F	CI2C2F	CI2C1F	CUART 5F
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUART 4F	CUSAR T3F	CUSAR T2F	CSPI3F	CSPI2F	CIWDG F	CWWD GF	CTIM1 4F	CTIM1 3F	CTIM1 2F	CTIM7 F	CTIM6 F	CTIM5 F	CTIM4 F	CTIM3 F	CTIM2 F
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **CLPTIM2F** : LPTIM2 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 30 **CDTSF** : DTS の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 29 **CUART12F** : UART12 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 28 **CUART9F** : UART9 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 27 **CUART8F** : UART8 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 26 **CUART7F** : UART7 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 25 **CDAC1F** : DAC1 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 24 **CHDMICECF** : HDMICEC の不正アクセスフラグのクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 23 **CUSART11F** : USART11 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 22 **CUSART10F** : USART10 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

- ビット 21 **CUSART6F** : USART6 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 20 **CCRSF** : CRS の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 19 **CI3C1F** : I3C1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 18 **CI2C2F** : I2C2 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 17 **CI2C1F** : I2C1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 16 **CUART5F** : UART5 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 15 **CUART4F** : UART4 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 14 **CUSART3F** : USART3 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 13 **CUSART2F** : USART2 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 12 **CSPI3F** : SPI3 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 11 **CSPI2F** : SPI2 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 10 **CIWDGF** : IWDG の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 9 **CWWDGF** : WWDG の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 8 **CTIM14F** : TIM14 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 7 **CTIM13F** : TIM13 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア

ビット 6 **CTIM12F** : TIM12 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 5 **CTIM7F** : TIM7 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 4 **CTIM6F** : TIM6 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 3 **CTIM5F** : TIM5 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 2 **CTIM4F** : TIM4 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 1 **CTIM3F** : TIM3 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 0 **CTIM2F** : TIM2 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

5.7.10 GTZC1 TZIC フラグ・クリア・レジスタ 2 (GTZC1_TZIC_FCR2)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLPTI M5F	CLPTI M4F	CLPTI M3F	CLPTI M1F	CI2C4F	CI2C3F	CLPUA RT1F	CSPI5F	Res.	Res.	Res.	Res.	CUSBF	CSAI2F	CSAI1F	CSPI6F
w	w	w	w	w	w	w	w					w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSPI4F	CTIM1 7F	CTIM1 6F	CTIM1 5F	CUSAR T1F	CTIM8 F	CSPI1F	CTIM1 F	Res.	Res.	Res.	Res.	Res.	CUCP DF	CFDCA N2F	CFDCA N1F
w	w	w	w	w	w	w	w						w	w	w

ビット 31 **CLPTIM5F** : LPTIM5 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 30 **CLPTIM4F** : LPTIM4 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 29 **CLPTIM3F** : LPTIM3 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

- ビット 28 **CLPTIM1F** : LPTIM1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 27 **CI2C4F** : I2C4 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 26 **CI2C3F** : I2C3 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 25 **CLPUART1F** : LPUART の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 24 **CSPI5F** : SPI5 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 23:20 予約済みであり、リセット値に保持する必要があります。
- ビット 19 **CUSBF** : USB の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 18 **CSAI2F** : SAI2 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 17 **CSAI1F** : SAI1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 16 **CSPI6F** : SPI6 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 15 **CSPI4F** : SPI4 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 14 **CTIM17F** : TIM17 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 13 **CTIM16F** : TIM16 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 12 **CTIM15F** : TIM15 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 11 **CUSART1F** : USART1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア

ビット 10 **CTIM8F** : TIM8 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 9 **CSPI1F** : SPI1 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 8 **CTIM1F** : TIM1 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CUCPDF** : UCPD の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 1 **CFDCAN2F** : FDCAN2 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 0 **CFDCAN1F** : FDCAN1 の不正アクセスフラグクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

5.7.11 GTZC1 TZIC フラグ・クリア・レジスタ 3 (GTZC1_TZIC_FCR3)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	CRAM CFGF	Res.	COCT OSPI1 F	CFMCF	CSDM MC2F	CSDM MC1F	CPKAF	CSAES F	CRNG F	CHASH F	CAESF
					w		w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDCMI F	CADC1 2F	CDCA CHEF	CICAC HEF	CETHF	CFMA CF	CCOR DICF	CCRCF	Res.	Res.	Res.	Res.	Res.	Res.	CVREF BUFF	CLPTI M6F
w	w	w	w	w	w	w	w							w	w

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **CRAMCFGF** : RAMSCFG の不正アクセスフラグのクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **COCTOSPI1F** : OCTOSPI1 の不正アクセスフラグのクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

ビット 23 **CFMCF** : FMC の不正アクセスフラグのクリア

- 0 : クリアなし
- 1 : ステータスフラグのクリア

- ビット 22 **CSDMMC2F** : SDMMC2 の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 21 **CSDMMC1F** : SDMMC1 の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 20 **CPKAF** : PKA の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 19 **CSAESF** : SAES の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 18 **CRNGF** : RNG の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 17 **CHASHF** : HASH の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 16 **CAESF** : AES の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 15 **CDCMIF** : DCMI の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 14 **CADC12F** : ADC1 および ADC2 の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 13 **CDCACHEF** : DCACHE の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 12 **CICACHEF** : ICACHE の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 11 **CETHF** : ETH のレジスタの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 10 **CFMACF** : FMAC の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 9 **CCORDICF** : CORDIC の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 8 **CCRCF** : CRC の不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CVREFBUFF** : VREFBUF の不正アクセスフラグのクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 0 **CLPTIM6F** : LPTIM6 の不正アクセスフラグのクリア

0 : クリアなし

1 : ステータスフラグのクリア

5.7.12 GTZC1 TZIC フラグ・クリア・レジスタ 4 (GTZC1_TZIC_FCR4)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	CMPC BB3_R EGF	CSRA M3F	CMPC BB2_R EGF	CSRA M2F	CMPC BB1_R EGF	CSRA M1F	Res.	Res.	Res.	CBKPS RAMF	CFMC_ MEMF	COCT OSPI1_ MEMF	CTZIC1 F	CTZSC 1F
		w	w	w	w	w	w				w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CEXTI F	CRCCF	CPWR F	CTAMP F	CRTCF	CSBSF	Res.	COTFD EC1F	CFLAS HF	CFLAS H_REG F	CGPD MA2F	CGPD MA1F
				w	w	w	w	w	w		w	w	w	w	w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **CMPCBB3_REGF** : MPCBB3 レジスタの不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 28 **CSRAM3F** : SRAM3 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 27 **CMPCBB2_REGF** : MPCBB2 レジスタの不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 26 **CSRAM2F** : SRAM2 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 25 **CMPCBB1_REGF** : MPCBB1 レジスタの不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 24 **CSRAM1F** : SRAM1 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

- ビット 20 **CBKPSRAMF** : MPCWM4 (BKPSRAM) メモリ・バンクの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 19 **CFMC_MEMF** : MPCWM2 (FMC_NOR バンク)、MPCWM3 (FMC_NAND バンク、および FMC_SDRAM バンク 1)、MPCWM4 (FMC_SDRAM バンク 2) の不正アクセスフラグのクリア。
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 18 **COCTOSPI1_MEMF** : MPCWM1 (OCTOSPI1) メモリ・バンクの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 17 **CTZIC1F** : GTZC1 TZIC レジスタの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 16 **CTZSC1F** : GTZC1 TZSC レジスタの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 15:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **CEXTIF** : EXTI の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 10 **CRCCF** : RCC の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 9 **CPWRF** : PWR の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 8 **CTAMPF** : TAMP の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 7 **CRTCF** : RTC の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 6 **CSBSF** : SBS の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **COTFDEC1F** : OTFDEC1 の不正アクセスフラグクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 3 **CFLASHF** : FLASH メモリの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア
- ビット 2 **CFLASH_REGF** : Flash レジスタの不正アクセスフラグのクリア
0 : クリアなし
1 : ステータスフラグのクリア

ビット 1 **CGPDMA2F** : GPDMA2 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

ビット 0 **CGPDMA1F** : GPDMA1 の不正アクセスフラグクリア

0 : クリアなし

1 : ステータスフラグのクリア

5.7.13 GTZC1 TZIC レジスタ・マップ

表 31. GTZC1 TZIC レジスタのマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	GTZC1_TZIC_IER1	LPTIM2IE	DTSIE	UART12IE	UART9IE	UART8IE	UART7IE	DAC1IE	HDMICEIE	USART11IE	USART10IE	USART6IE	CRSIE	I3C1IE	I2C2IE	I2C1IE	UART5IE	UART4IE	USART3IE	USART2IE	SPI3IE	SPI2IE	IWDGIE	WWDGIE	TIM14IE	TIM13IE	TIM112IE	TIM7IE	TIM6IE	TIM5IE	TIM4IE	TIM3IE	TIM2IE	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x004	GTZC1_TZIC_IER2	LPTIM5IE	LPTIM4IE	LPTIM3IE	LPTIM1IE	I2C4IE	I2C3IE	LPUART1IE	SPI5IE	Res	Res	Res	Res	USBIE	SAI2IE	SAI1IE	SPI6IE	SPI4IE	TIM17IE	TIM16IE	TIM15IE	USART1IE	TIM8IE	SPI1IE	TIM1IE	Res	Res	Res	Res	Res	Res	UCPDIE	FDCAN2IE	FDCAN1IE
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x008	GTZC1_TZIC_IER3	Res.	Res.	Res.	Res.	Res.	RAMCFGIE	Res	OCTOSP1IE	FMCIE	SDMMC2IE	SDMMC1IE	PKAIE	SAESIE	RNGIE	HASHIE	AESIE	DCMIE	ADC12IE	DCACHEIE	ICACHEIE	ETHIE	FMACIE	CORDICIE	CRCIE	Res	Res	Res	Res	Res	Res	VREFBUFIE	LPTIM6IE	
	リセット値						0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	
0x00C	GTZC1_TZIC_IER4	Res.	Res.	MPCBB3_REGIE	SRAM3IE	MPCBB2_REGIE	SRAM2IE	MPCBB1_REGIE	SRAM1IE	Res.	Res.	Res	BKPSRAMIE	FMC_MEMIE	OCTOSP1_MEMIE	TZIC1IE	TZSC1IE	Res	Res	Res	Res	EXTIE	RCCIE	PWRIE	TAMPIE	RTCIE	SBSIE	Res	Res	Res	Flash_REGIE	Res	Res	
	リセット値						0	0	0				0	0	0	0	0					0	0	0	0	0	0						0	0
0x010	GTZC1_TZIC_SR1	LPTIM2F	DTSF	UART12F	UART9F	UART8F	UART7F	DAC1F	HDMICECF	USART11F	USART10F	USART6F	CRSF	I3C1F	I2C2F	I2C1F	UART5F	UART4F	USART3F	USART2F	SPI3F	SPI2F	IWDGF	WWDGF	TIM14F	TIM13F	TIM112F	TIM7F	TIM6F	TIM5F	TIM4F	TIM3F	TIM2F	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	GTZC1_TZIC_SR2	LPTIM5F	LPTIM4F	LPTIM3F	LPTIM1F	I2C4F	I2C3F	LPUART1F	SPI5F	Res.	Res.	Res.	Res	USBF	SAI2F	SAI1F	SPI6F	SPI4F	TIM17F	TIM16F	TIM15F	USART1F	TIM8F	SPI1F	TIM1F	Res	Res	Res	Res	Res	Res	UCPDF	FDCAN2F	FDCAN1F
	リセット値	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	GTZC1_TZIC_SR3	Res.	Res.	Res.	Res.	Res.	RAMCFGF	Res	OCTOSP1F	FMCF	SDMMC2F	SDMMC1F	PKAF	SAESF	RNGF	HASHF	AESF	DCMIF	ADC12F	DCACHEF	ICACHEF	ETHF	FMACF	CORDICF	CRCF	Res	Res	Res	Res	Res	Res	VREFBUFFF	LPTIM6F	
	リセット値						0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x01C	GTZC1_TZIC_SR4	Res.	Res.	MPCBB3_REGF	SRAM3F	MPCBB2_REGF	SRAM2F	MPCBB1_REGF	SRAM1F	Res.	Res.	Res	BKPSRAMF	FMC_MEMF	OCTOSP1_MEMF	TZIC1F	TZSC1F	Res	Res	Res	Res	EXTIF	RCCF	PWRF	TAMPF	RTCF	SBSF	Res	Res	Res	Flash_REGF	Res	Res	
	リセット値			0	0	0	0	0	0				0	0	0	0	0					0	0	0	0	0	0						0	0
0x020	GTZC1_TZIC_FCR1	CLPTIM2F	CDTSF	CUART12F	CUART9F	CUART8F	CUART7F	CDAC1F	CHDMICECF	CUSART11F	CUSART10F	CUSART6F	CCRSF	CI3C1F	CI2C2F	CI2C1F	CUART5F	CUART4F	CUSART3F	CUSART2F	CSP13F	CSP12F	CIWDGF	CWWDGF	CTIM14F	CTIM13F	CTIM112F	CTIM7F	CTIM6F	CTIM5F	CTIM4F	CTIM3F	CTIM2F	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	GTZC1_TZIC_FCR2	CLPTIM5F	CLPTIM4F	CLPTIM3F	CLPTIM1F	CI2C4F	CI2C3F	CLPUART1F	CSP15F	Res.	Res.	Res.	Res	CUSBF	CSAI2F	CSAI1F	CSP16F	CSP14F	CTIM17F	CTIM16F	CTIM15F	CUSART11F	CTIM8F	CSP11F	CTIM1F	Res	Res	Res	Res	Res	Res	CUCPDF	CFDCAN2F	
	リセット値	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	GTZC1_TZIC_FCR3	Res.	Res.	Res.	Res.	Res.	CRAMCFGF	Res	COCOTOSPI1F	CFMCF	CSDMMC2F	CSDMMC1F	CPKAF	CSAESF	CRNGF	CHASHF	CAESF	CDCMIF	CADC12F	CDACACHEF	CICACHEF	CEHF	CFMACF	CCORDICF	CCRCF	Res	Res	Res	Res	Res	Res	CVREFBUFFF	CLPTIM6F	
	リヤット値						0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0

表 31. GTZC1 TZIC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x02C	GTZC1_TZIC_ FCR4	Res.	Res.	CMPCBB3_REGF	CSRAM3F	CMPCBB2_REGF	CSRAM2F	CMPCBB1_REGF	CSRAM1F	Res.	Res.	Res.	CBKPSRAMF	CFMC_MEMF	COCTOSPI1_MEMF	CTZIC1F	CTZSC1F	Res.	Res.	Res.	Res.	CEXTIF	CRCCF	CPWRF	CTAMPF	CRTCF	CSBSF	Res.	COTFDEC1F	CFLASHF	CFLASH_REGF	CGDMA2F	CGDMA1F
	リセット値			0	0	0	0	0	0				0	0	0	0	0					0	0	0	0	0		0	0	0	0	0	0

表 24 : GTZC1 サブブロック・アドレスオフセットを参照してください。

5.8 GTZC1 MPCBBz レジスタ (z = 1 から 3)

すべてのレジスタは、必ずワード (32 ビット) 単位でアクセスされます。

5.8.1 GTZC1 SRAMz MPCBB 制御レジスタ (GTZC1_MPCBBz_CR) (z = 1 から 3)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRWIL ADIS	INVSE CSTATE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLOCK
															rs

ビット 31 **SRWILADIS** : セキュア読み出し／書き込み不正アクセス無効化

このビットは、セキュア読み出し／書き込みトランザクションがブロックベース SRAM の非セキュアブロックにアクセスするとき、不正アクセスの検出を無効にします (非セキュアブロックにおけるセキュアフェッチは常に不正とみなされます)。

0 : 有効、非セキュア SRAM ブロックへのセキュア読み出し／書き込みアクセスは不可

1 : 無効、非セキュア SRAM ブロックへのセキュア読み出し／書き込みアクセスは可

ビット 30 **INVSECSTATE** : SRAMx クロックのセキュリティ状態

このビットは RCC の内蔵 SRAM のクロック制御がセキュアかどうかを定義するために使用されます。
0 : MPCBB にセキュア領域が存在する場合、SRAM のクロックはセキュアです。セキュア領域が無い場合は非セキュアです。

1 : SRAM のクロックは、MPCBB にセキュア領域が存在しても非セキュアで、MPCBB にセキュアブロックが設定されていない場合でもセキュアです。

ビット 29:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **GLOCK** : MPCBB の制御レジスタを次のリセットまでロックします。

このビットはデフォルトではクリアされており、一旦、設定すると、システムリセットまでリセットできません。

0 : 制御レジスタロック解除

1 : 制御レジスタロック

5.8.2 GTZC1 SRAMz MPCBB 設定ロック・レジスタ 1 (GTZC1_MPCBBz_CFGLOCK1) (z = 1 から 3)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

セキュア特権アクセスのみ。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLCK 31	SPLCK 30	SPLCK 29	SPLCK 28	SPLCK 27	SPLCK 26	SPLCK 25	SPLCK 24	SPLCK 23	SPLCK 22	SPLCK 21	SPLCK 20	SPLCK 19	SPLCK 18	SPLCK 17	SPLCK 16
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPLCK 15	SPLCK 14	SPLCK 13	SPLCK 12	SPLCK 11	SPLCK 10	SPLCK 9	SPLCK 8	SPLCK 7	SPLCK 6	SPLCK 5	SPLCK 4	SPLCK 3	SPLCK 2	SPLCK 1	SPLCK 0
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

ビット 31:0 **SPLCKy** : スーパーブロックのセキュリティ／特権設定のロック (y = 31 から 0)

このビットはソフトウェアでセットされ、システムリセットでのみクリアが可能です。

0 : GTZC1_MPCBBz_SECCFGRy および GTZC1_MPCBBz_PRIVCFGRy への書き込みが可能。

1 : GTZC1_MPCBBz_SECCFGRy および GTZC1_MPCBBz_PRIVCFGRy への書き込みを無視。

5.8.3 スーパーブロック x レジスタの GTZC1 SRAMz MPCBB セキュリティ設定 (GTZC1_MPCBBz_SECCFGRx) (z = 1 から 3)

アドレスオフセット : 0x100 + 0x4 × x, (x = 0 から 31)

リセット値 : 0xFFFF FFFF

指定のリセット値は TZEN = 0xB4 のときに有効です。TZEN = 0xC3 のときのリセット値は 0x0000 0000 です。

このレジスタへの書き込みアクセスはセキュア・アクセスのみです。あらゆる読み込みが許されています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEC31	SEC30	SEC29	SEC28	SEC27	SEC26	SEC25	SEC24	SEC23	SEC22	SEC21	SEC20	SEC19	SEC18	SEC17	SEC16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **SECy** : ブロック y (y = 31 から 0) のセキュリティ設定

0 : スーパーブロック x に属するブロック y へのアクセスは非セキュア・アクセスのみ。
GTZC1_MPCBBz_CR で SRWILADIS ビットが設定されている場合、セキュアアクセスも許可されます。

1 : スーパーブロック x に属するブロック y へのアクセスはセキュア・アクセスのみ。

GTZC1_MPCBBz_PRIVCFGRx で PRIVy ビットが設定されている場合、このビットへの非特権書き込みは無視されます。

GTZC1_MPCBBz_CFGLOCK で SPLCKx ビットが設定されている場合、書き込みは無視されます。

5.8.4 スーパーブロック x レジスタの GTZC1 SRAMz MPCBB 特権設定 (GTZC1_MPCBBz_PRIVCFGRx) (z = 1 から 3)

アドレスオフセット : $0x200 + 0x \cdot x$, ($x = 0$ から 31)

リセット値 : 0xFFFF FFFF

指定のリセット値は TZEN = 0xB4 のときに有効です。TZEN = 0xC3 のときのリセット値は 0x0000 0000 です。このレジスタへの書き込みアクセスは特権アクセスのみです。あらゆる読み込みが許されています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIV31	PRIV30	PRIV29	PRIV28	PRIV27	PRIV26	PRIV25	PRIV24	PRIV23	PRIV22	PRIV21	PRIV20	PRIV19	PRIV18	PRIV17	PRIV16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV15	PRIV14	PRIV13	PRIV12	PRIV11	PRIV10	PRIV9	PRIV8	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **PRIVy** : スーパーブロック x に属するブロック y の特権設定 (y = 31 から 0)。

0 : スーパーブロック x に属するブロック y への特権および非特権アクセス。

1 : スーパーブロック x に属するブロック y へのアクセスは特権アクセスのみ。

GTZC1_MPCBBz_SECCFGRx で SECy ビットが設定されている場合、このビットへの非セキュア書き込みは無視されます。

GTZC1_MPCBBz_CFGLOCK で SPLCKx ビットが設定されている場合、書き込みは無視されます。

5.8.5 GTZC1 MPCBBz レジスタ・マップ (z = 1 から 3)

表 32. GTZC1 MPCBBz レジスタのマップとリセット値 (z = 1 から 3)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	GTZC1_MPCBBz_CR	SRWLADIS	INVSECSTATE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLOCK
	リセット値	0	0																														0
0x004 - 0x00C	予約済み	予約済み																															
0x010	GTZC1_MPCBBz_CFGLOCK1	SPLCK31	SPLCK30	SPLCK29	SPLCK28	SPLCK27	SPLCK26	SPLCK25	SPLCK24	SPLCK23	SPLCK22	SPLCK21	SPLCK20	SPLCK19	SPLCK18	SPLCK17	SPLCK16	SPLCK15	SPLCK14	SPLCK13	SPLCK12	SPLCK11	SPLCK10	SPLCK9	SPLCK8	SPLCK7	SPLCK6	SPLCK5	SPLCK4	SPLCK3	SPLCK2	SPLCK1	SPLCK0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014- 0x0FC	予約済み	予約済み																															
0x100 + 0x04 × x (x = 0 から 31)	GTZC1_MPCBBz_SEC_CFRx	SEC31	SEC30	SEC29	SEC28	SEC27	SEC26	SEC25	SEC24	SEC23	SEC22	SEC21	SEC20	SEC19	SEC18	SEC17	SEC16	SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x180- 0x1FC	予約済み	予約済み																															
0x200 + 0x04 × x (x = 0 から 31)	GTZC1_MPCBBz_PRI_VCFGRx	PRIV31	PRIV30	PRIV29	PRIV28	PRIV27	PRIV26	PRIV25	PRIV24	PRIV23	PRIV22	PRIV21	PRIV20	PRIV19	PRIV18	PRIV17	PRIV16	PRIV15	PRIV14	PRIV13	PRIV12	PRIV11	PRIV10	PRIV9	PRIV8	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 24 : GTZC1 サブブロック・アドレスオフセットを参照してください。

6 RAM 設定コントローラ (RAMCFG)

6.1 概要

RAMCFG は、内部 SRAM (SRAM1、SRAM2、SRAM3、および BKPSRAM) の機能を設定します。

6.2 RAMCFG の主な機能

内部 SRAM は、RAMCFG で設定される下記の機能の一部をサポートします。

- エラーコード訂正 (ECC) :
 - 割込み生成に対応のシングルエラー検出と訂正
 - 割込みまたは NMI 生成に対応のダブルエラー検出
 - 失敗アドレスでのステータス
- 書き込み保護 (1 KB 単位)
- SRAM ソフトウェア消去

6.3 RAMCFG の機能詳細

6.3.1 内部 SRAM 機能

5 つの SRAM がデバイスに内蔵され、それぞれ特定の機能を備えています。

- SRAM1、SRAM2、SRAM3 は、メイン SRAM です。
これらの SRAM は複数のブロックで構成され、STOP モードでパワーダウンして、消費電力を削減できます。
 - SRAM1 : 4 つの 64 KB ブロック (合計 256 KB)
 - SRAM2 : 16 KB + 48 KB ブロック (合計 64 KB)
 - SRAM3 : 5 つの 64 KB ブロック (合計 320 KB)バックアップ SRAM (BKPSRAM) は、すべての低電力モードと VBAT モードで V_{DD} がオフのときに保持できます。
詳細については、[セクション 10 : 電源制御 \(PWR\)](#) を参照してください。
- SRAM2 は、Flash メモリユーザオプションバイトで SRAM2_RST オプションビットが選択された場合、システムリセットが発生したときに消去されます。SRAM1 と SRAM3 は、Flash メモリユーザオプションバイトで SRAM13_RST オプションビットが選択された場合、システムリセットが発生したときに消去されます。詳細については、[セクション 7 : 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。
- SRAM2 とオプションでバックアップ SRAM は、タンパ検出回路によって保護され、タンパ検出の場合にハードウェアによって消去されます。詳細については、[セクション 76 : 「タンパ検出 およびバックアップレジスタ \(TAMP\) \(TBD\)」](#) を参照してください。
- RAMCFG は、内部 SRAM ECC、書き込み保護、およびソフトウェア消去に関するレジスタを内蔵します。

下の表に、各内部 SRAM によってサポートされる機能の概要を示します。

表 33. 内部 SRAM 機能

SRAM の機能	SRAM1 (256 KB)	SRAM2 (64 KB)	SRAM3 (320 KB)	BKPSRAM (4 KB)
STANDBY モードでの内容保持 (オプション)	-	-	-	X
VBAT モードでのオプション保持	-	-	-	X
タンパ検出とバックアップ ドメインリセットによる消去	-	X	-	X ⁽¹⁾
オプションでシステムリセットにより 消去	X	X	X	-
ソフトウェア消去	X	X	X	X
ECC	-	X	X	X
書込み保護機能	-	X	-	-

1. オプション : BKPSRAM は、タンパ検出時に消去するかどうかを設定できます。

6.3.2 エラーコード訂正 (SRAM2、SRAM3、BKPSRAM)

ECC は、SRAM2_ECC、SRAM3_ECC、および BKPSRAM_ECC ユーザオプションビットで有効なとき、SRAM2、SRAM3、および BKPSRAM によってサポートされます。詳細については、[セクション 7 : 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。

7 つの ECC ビットが SRAM の 32 ビットごとに追加され、メモリ読出しアクセス時の 2 ビットのエラー検出と 1 ビットのエラー訂正を可能にします。

ECC は 32 ビットワードについて計算され、チェックされるため、バイトおよびハーフワード書込みアクセスは SRAM インタフェースによって、ワード全体を読み出してから、ワードを新しいバイト/ハーフワード値で再び書き込むことによって管理されます。ECC ダブルエラーも、このようなバイトまたはハーフワード AHB 書込みアクセス中に検出されます (インタフェースによる読出し/変更/書込み)。バイトまたはハーフワード書込みアクセスの遅延は、2 AHB クロックサイクルです。

注意 : ECC 付き SRAM へのバイトまたはハーフワード書込みの場合、読出し/変更/書込み操作はバッファで行われます。SRAM AHB がリリースされてから 2 AHB クロックサイクル後 (SRAM がアクセスされなくなってから)、バッファの内容が SRAM に書き込まれます。

シングルおよびダブル ECC エラー

シングルエラーが検出されると、自動的に訂正され、SEDC/CSEDC ビットがそれぞれ [RAMCFG メモリ割込みステータスレジスタ \(RAMCFG_MxISR\)](#) および [RAMCFG メモリ x 割込みクリアレジスタ x \(RAMCFG_MxICR\)](#) でセットされます。[RAMCFG メモリ x 割込みイネーブルレジスタ \(RAMCFG_MxIER\)](#) の SEIE ビットによって有効化されている場合、割込みが生成されます。[RAMCFG メモリ x 制御レジスタ \(RAMCFG_MxCR\)](#) の ALE ビットがセットされている場合、失敗アドレスは [RAMCFG メモリ x ECC シングルエラーアドレスレジスタ \(RAMCFG_MxSEAR\)](#) に保存されます。

注意 : SEDC ビットがセットされているときには、シングルエラーを検出できません。

ダブルエラーが検出されると、DED および CDED ビットがそれぞれ [RAMCFG メモリ割込みステータスレジスタ \(RAMCFG_MxISR\)](#) と [RAMCFG メモリ x 割込みクリアレジスタ x \(RAMCFG_MxICR\)](#) でセットされます。[RAMCFG メモリ x 割込みイネーブルレジスタ \(RAMCFG_MxIER\)](#) の DEIE または ECCNMI ビットによって有効化されている場合、割込みまたは NMI が生成されます。[RAMCFG メ](#)

メモリ x 制御レジスタ (RAMCFG_MxCR) の ALE ビットがセットされている場合、失敗アドレスは RAMCFG メモリ x ECC ダブルエラーアドレスレジスタ (RAMCFG_MxDEAR) に保存されます。

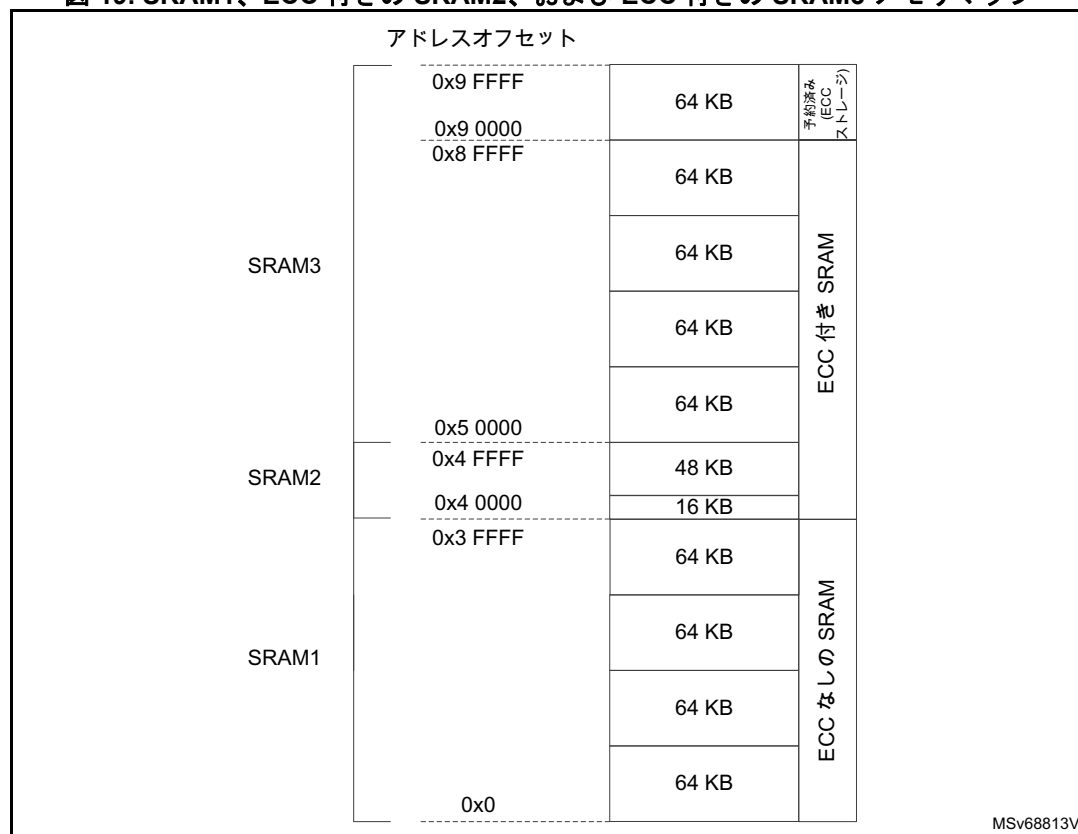
注意： DED ビットがセットされているとき、ダブルエラーを検出できません。

SRAM3 ECC 固有の管理

SRAM3 について ECC が有効なとき、SRAM3 の最初の 256 KB のみについて ECC が行われます。次の 64 KB には ECC がなく、最後のブロックは ECC を保存するために使用されるため、アプリケーションに使用することはできません。

下の図は、SRAM2 と SRAM3 の ECC が有効なときの SRAM 領域を示しています。

図 19. SRAM1、ECC 付きの SRAM2、および ECC 付きの SRAM3 メモリマップ



ユーザオプションビットによって ECC が有効化されると、システムリセット後、関連する RAMCFG メモリ x 制御レジスタ (RAMCFG_MxCR) で ECCE ビットが自動的にセットされます。

ECC は、以下のソフトウェアシーケンスを実行することで無効化できます。

1. RAMCFG メモリ x ECC キーレジスタ (RAMCFG_MxECCKEYR) に 0xAE を書き込みます。
2. RAMCFG メモリ x ECC キーレジスタ (RAMCFG_MxECCKEYR) に 0x75 を書き込みます。
3. RAMCFG メモリ x 制御レジスタ (RAMCFG_MxCR) の ECCE ビットに 0 を書き込みます。

ECC が無効なとき (ECCE = 0)、SRAM3 ECC ストレージ領域は ECC ユーザテスト目的で読出しと書込みが可能です。ECC が有効なとき (ECCE = 1)、この領域は ECC ストレージ目的で予約され、読出しも書込みもできません。

6.3.3 書き込み保護 (SRAM2)

SRAM2 は、64 の 1 KB ページで構成されます。各 1 KB ページは、[RAMCFG メモリ 2 書き込み保護レジスタ 1 \(RAMCFG_M2WPR1\)](#) および [RAMCFG メモリ 2 書き込み保護レジスタ 2 \(RAMCFG_M2WPR2\)](#) の対応する PxWP (x = 0 ~ 63) ビットをセットすることによって書き込み保護できます。

6.3.4 ソフトウェア消去

SRAM 消去は、次のソフトウェアシーケンスを実行することでリクエストできます。

1. [RAMCFG メモリ x 消去キーレジスタ \(RAMCFG_MxERKEYR\)](#) に 0xCA を書き込みます。
2. [RAMCFG メモリ x 消去キーレジスタ \(RAMCFG_MxERKEYR\)](#) に 0x53 を書き込みます。
3. [RAMCFG メモリ x 制御レジスタ \(RAMCFG_MxCR\)](#) の SRAMER ビットに 1 を書き込みます。

SRAMBUSY フラグは、消去が進行中の間だけ、関連する SRAM 割込みステータスレジスタでセットされます。

各 SRAM 消去の合計時間は、N AHB クロックサイクルであり、N は SRAM のサイズ (32 ビットワード数) です。

消去中に SRAM が読出しまたは書き込みが行われた場合、消去操作の終了まで、AHB にウェイトステートが挿入されます。

6.4 RAMCFG 低電力モード

表 34. 低消費電力モードが RAMCFG に与える影響

モード	説明
SLEEP	影響はありません。RAMCFG 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	RAMCFG レジスタの内容が保持されます。
STANDBY	RAMCFG ペリフェラルはパワーダウンされ、STANDBY の終了後に再初期化が必要です。

6.5 RAMCFG 割込み

次の表に、RAMCFG 割込みリクエストの一覧を示します。

表 35. RAMCFG 割込みリクエスト

項目 (割込みの 略称)	割込みイベント	イベント フラグ	有効制御 ビット	割込みの クリア方法	SLEEP モードの 終了	STOP モードの 終了	STANDBY モードの 終了
RAMCFG	ECC シングルエラー検出 および訂正	SEDC	SEIE	CSEDC に 1 を 書き込む	はい	不可	不可
	ECC ダブルエラー検出	DED	DEIE = 1 かつ ECCNMI = 0	CDED に 1 を書 き込む	はい	不可	不可
NMI	ECC ダブルエラー検出	DED	ECCNMI	CDED に 1 を書 き込む	はい	不可	不可

6.6 RAMCFG レジスタ

下記のレジスタでは、x は以下を指します。

- x = 1/2/3 のとき、それぞれ SRAM1/2/3
- x = 5 のとき、BKPSRAM

6.6.1 RAMCFG メモリ x 制御レジスタ (RAMCFG_MxCR)

アドレスオフセット : $0x040 * (x - 1)$ 、(x = 1、2、3、および 5)

リセット値 : 0x0000 : 000X

ECCE リセット値は ECC イネーブルユーザオプションビットに依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAMER	Res.	Res.	Res.	ALE	Res.	Res.	Res.	ECCE
							rs				rw				rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SRAMER** : SRAM消去

このビットは、RAMCFG_MxERKEYR レジスタの ERASEKEY フィールドにアンロックシーケンスを書き込んだ後でのみ、ソフトウェアによってセットできます。このビットをセットすると、SRAM 消去が開始されます。このビットは、消去操作の終了時にハードウェアによって自動的にクリアされます。

0 : 進行中の消去操作はありません

1 : 消去操作が進行中です

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **ALE** : アドレスラッチイネーブル

0 : 失敗アドレスは SRAMx ECC シングル/ダブルエラーアドレスレジスタに保存されません。

1 : 失敗アドレスは SRAMx ECC シングル/ダブルエラーアドレスレジスタに保存されます。

注 : このビットは予約済みであり、SRAM1 制御レジスタでリセット値に保持される必要があります。

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **ECCE** : ECC イネーブル。

このビットのリセット値は、ユーザオプションビット設定によって定義されます。セットされているときには、RAMCFG_MxECCKEYR レジスタにアンロックシーケンスを書き込んだ後でのみ、ソフトウェアによってクリアできます。

0 : ECC は無効です。

1 : ECC は有効です。

注 : このビットは予約済みであり、SRAM1 制御レジスタでリセット値に保持される必要があります。

6.6.2 RAMCFG メモリ x 割込みイネーブルレジスタ (RAMCFG_MxIER)

アドレスオフセット : $0x004 + 0x040 * (x - 1)$ 、($x = 2, 3, 5$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCNMI	Res.	DEIE	SEIE
												rs		rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ECCNMI** : ダブルエラー NMI

このビットは、ソフトウェアによってセットされ、グローバル RAMCFG リセットによってのみクリアされます。

0 : ECC ダブルエラー発生時、NMI は生成されません。

1 : ECC ダブルエラー発生時、NMI が生成されます。

注 : **ECCNMI** がセットされた場合、**DEIE** ビット値に関係なく、**RAMCFG** マスク可能割込みは生成されません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DEIE** : ECC ダブルエラー割込みイネーブル

0 : ダブルエラー割込み無効

1 : ダブルエラー割込み有効

ビット 0 **SEIE** : ECC シングルエラー割込みイネーブル

0 : シングルエラー割込み無効

1 : シングルエラー割込み有効

6.6.3 RAMCFG メモリ割込みステータスレジスタ (RAMCFG_MxISR)

アドレスオフセット : $0x008 + 0x040 * (x - 1)$ 、($x = 1, 2, 3$ 、および 5)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM BUSY	Res.	Res.	Res.	Res.	Res.	Res.	DED	SEDC
							r							r	r

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SRAMBUSY** : SRAM 消去操作によりビジー

0 : 進行中の消去操作はありません

1 : 消去操作が進行中です

注 : **SRAM** に応じて、消去操作は、ソフトウェアリクエストにより、オプションビットが有効な場合はシステムリセットにより、タンパ検出または読み出し保護回帰によって行うことができます。

表 33 : 内部 SRAM 機能を参照してください。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DED** : ECC ダブルエラー検出

0 : ダブルエラーはありません。

1 : ダブルエラーが検出されました。

注 : このビットは予約済みであり、SRAM1 割込みステータスレジスタでリセット値に保持される必要があります。

ビット 0 **SEDC** : ECC シングルエラー検出と訂正

0 : シングルエラーはありません。

1 : シングルエラーが検出され、訂正されました。

注 : このビットは予約済みであり、SRAM1 割込みステータスレジスタでリセット値に保持される必要があります。

6.6.4 RAMCFG メモリ x ECC シングルエラーアドレスレジスタ (RAMCFG_MxSEAR)

アドレスオフセット : $0x00C + 0x040 * (x - 1)$ 、($x = 2, 3, 5$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESEA[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESEA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **ESEA[31:0]** : ECC シングルエラーアドレス

RAMCFG_MxCR レジスタの ALE ビットがセットされたとき、このフィールドは ECC シングルエラーに対応するアドレスで更新されます。

6.6.5 RAMCFG メモリ x ECC ダブルエラーアドレスレジスタ (RAMCFG_MxDEAR)

アドレスオフセット : $0x010 + 0x040 * (x - 1)$ 、($x = 2, 3, 5$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EDEA[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDEA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **EDEA[31:0]** : ECC ダブルエラーアドレス

RAMCFG_MxCR レジスタの ALE ビットがセットされているとき、このフィールドは ECC ダブルエラーに対応するアドレスで更新されます。

6.6.6 RAMCFG メモリ x 割込みクリアレジスタ x (RAMCFG_MxICR)

アドレスオフセット : $0x014 + 0x040 * (x - 1)$ 、(x = 2、3、5)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDED	CSEDC
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CDED** : クリア ECC ダブルエラー検出

このフラグに 1 を書き込むことで、RAMCFG_MxISR レジスタの DED ビットがクリアされます。このフラグを読み出すと、DED 値が返されます。

ビット 0 **CSEDC** : クリア ECC シングルエラー検出および訂正

このフラグに 1 を書き込むことで、RAMCFG_MxISR レジスタの SEDC ビットがクリアされます。このフラグを読み出すと、SEDC 値が返されます。

6.6.7 RAMCFG メモリ 2 書込み保護レジスタ 1 (RAMCFG_M2WPR1)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

ビット 31:0 **PyWP** : SRAM2 1 KB ページ y 書込み保護 (y = 31 ~ 0)

これらのビットは、ソフトウェアによってセットされ、グローバル RAMCFG リセットによってのみクリアされます。

0 : SRAM2 1 KB ページ y の書込み保護は無効です。

1 : SRAM2 1 KB ページ y の書込み保護は有効です。

6.6.8 RAMCFG メモリ 2 書き込み保護レジスタ 2 (RAMCFG_M2WPR2)

アドレスオフセット : 0x05C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

ビット 31:0 **PyWP** : SRAM2 1 KB ページ y 書き込み保護 (y = 63 ~ 32)

これらのビットは、ソフトウェアによってセットされ、グローバル RAMCFG リセットによってのみクリアされます。

0 : SRAM2 1 KB ページ y の書き込み保護は無効です。

1 : SRAM2 1 KB ページ y の書き込み保護は有効です。

6.6.9 RAMCFG メモリ x ECC キーレジスタ (RAMCFG_MxECCKEYR)

アドレスオフセット : $0x024 + 0x040 * (x - 1)$, (x = 2, 3, 5)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	ECCKEY[7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **ECCKEY[7:0]** : ECC書き込み保護キー

RAMCFG_MxCr レジスタの ECCE ビットの書き込み保護をアンロックするには、次の手順が必要です。

1) ECCKEY[7:0] に 0xAE を書き込みます。

2) ECCKEY[7:0] に 0x75 を書き込みます。

注 : 誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

6.6.10 RAMCFG メモリ x 消去キーレジスタ (RAMCFG_MxERKEYR)

アドレスオフセット : $0x028 + 0x040 * (x1)$ ($x = 1$ から 5)

リセット値 : $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERASEKEY[7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **ERASEKEY[7:0]** : 消去書込み保護キー

RAMCFG_MxCr レジスタの SRAMER ビットの書込み保護をアンロックするには、次の手順が必要です。

1) ERASEKEY[7:0] に $0xCA$ を書き込みます。

2) ERASEKEY[7:0] に $0x53$ を書き込みます。

注 : 誤ったキーを書き込むと、書込み保護が再度アクティブになります。

6.6.11 RAMCFG レジスタマップ

表 36. RAMCFG レジスタのマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	RAMCFG_M1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	SPRAMER	Res.	Res.	Res.	Res.	ALE	Res.	Res.	Res.	FCOE		
	リセット値																								0					0				x			
0x04	予約済み	予約済み																																			
0x08	RAMCFG_M1ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	SRAMBUSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																								0												
0x0C から 0x24	予約済み	予約済み																																			
0x28	RAMCFG_M1ERKEYR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERASEKEY[7:0]											
	リセット値																									0	0	0	0	0	0	0	0	0			
0x2C から 0x3C	予約済み	予約済み																																			
0x40	RAMCFG_M2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	SPRAMER	Res.	Res.	Res.	Res.	ALE	Res.	Res.	FCOE		
	リセット値																									0					0			x			
0x44	RAMCFG_M2IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCNMI	Res.	Res.	SEIE				
	リセット値																												0		0	0	0	0			

表 36. RAMCFG レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x048	RAMCFG_M2ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAMBUSY	Res.	Res.	Res.	Res.	Res.	Res.	DED	SEDC
	リセット値																								0							0	0
0x04C	RAMCFG_M2SEAR	ESEA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	RAMCFG_M2DEAR	EDEA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	RAMCFG_M2ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDED	CSEDC	
	リセット値																														0	0	
0x058	RAMCFG_M2WPR1	P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP	P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	RAMCFG_M2WPR2	P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP	P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	予約済み	予約済み																															
0x064	RAMCFG_M2ECKEYR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECKKEY[7:0]							
	リセット値																									0	0	0	0	0	0	0	0
0x068	RAMCFG_M2ERKEYR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERASEKEY[7:0]							
	リセット値																									0	0	0	0	0	0	0	0
0x06C から 0x07C	予約済み	予約済み																															
0x080	RAMCFG_M3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAMER	Res.	Res.	Res.	ALE	Res.	Res.	Res.	ECCE
	リセット値																								0				0				0
0x084	RAMCFG_M3IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCNMI	Res.	DEIE	SEIE
	リセット値																											0			0	0	0
0x088	RAMCFG_M3ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAMBUSY	Res.	Res.	Res.	Res.	Res.	Res.	DED	SEDC
	リセット値																								0							0	0
0x08C	RAMCFG_M3SEAR	ESEA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x090	RAMCFG_M3DEAR	EDEA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094	RAMCFG_M3ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDED	CSEDC	
	リセット値																														0	0	
0x098 から 0x0A0	予約済み	予約済み																															

表 36. RAMCFG レジスタのマップとリセット値 (続き)

[illegible]

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

7 内蔵 Flash メモリ (FLASH)

7.1 概要

内蔵 Flash メモリ (FLASH) は、あらゆるマスタの 2 M バイトの内蔵不揮発性メモリへのアクセスを管理します。これは、読出し、プログラム、および消去操作、エラー訂正の他、各種の整合性と機密性の保護メカニズムを実装しています。

FLASH は、パワーオン・リセット時の不揮発性ユーザ・オプション・バイトの自動ロードを管理し、これらのオプションの動的更新を実装しています。また、STMicroelectronics が製造時に設定した、高サイクル・データ領域やワンタイムプログラマブル (OTP) 領域、セキュア・キー・ストレージ領域 (OBKeys)、読取り専用領域なども備えています。

7.2 FLASH の主な機能

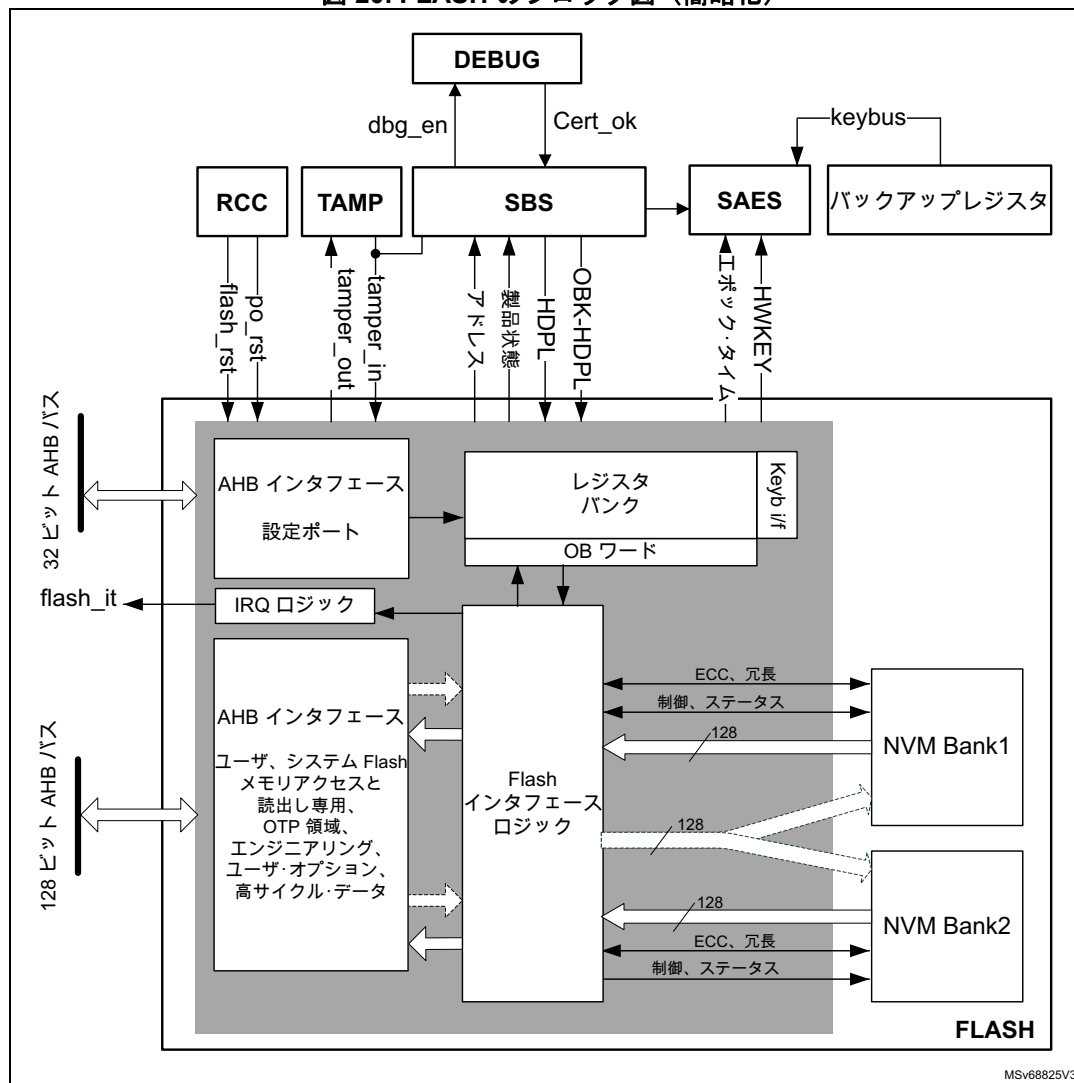
- 2 つの 1 MB のバンクに分割した、最大 2 MB の不揮発性メモリ
- 複数の長さをサポートしている Flash メモリ読出し操作 : 128 bit、64 bit、32 bit、16 bit、1 バイト
- 128 (ユーザ領域、OBKeys) および 16 bit での Flash メモリ・プログラミング (OTP および Flash の高サイクルデータ領域)
- 8 KB セクタ消去、バンク消去、およびデュアルバンク全体消去
- デュアルバンク構成のサポート
 - 同時操作 : Flash の高サイクルデータ領域を含む、書込み中読出し (プログラムおよび消去) をサポートします。2 つのバンクが同一のインタフェースを共有するため、書込みと消去を並行して処理することはできません (「書込み中書込み」はサポートされていません)。
 - バンク・スワッピング : 対応するレジスタとともに、各バンクのユーザ・メモリのアドレス・マッピングのスイッチが可能です。セキュリティ・フラグは物理バンクに対して有効のままであり、より低いセキュリティ設定のバンクにスワップすることでデータが浪費されることはありません。
- エラー・コード訂正 (ECC) : 設定可能な Flash の高サイクル・データ領域内に 6 bit を持つ 16 bit ワードに ECC 9 bit を使用し、128 bit の Flash ワードごとに、1 つのエラー検出/訂正、または 2 つのエラー検出。
- ユーザ設定可能な不揮発性オプション・バイト
- オプション・バイトによって有効化される Flash メモリ拡張保護
 - デバッグ・アクセスからメモリ・コンテンツを保護するための、さまざまな製品状態
 - 最大でバンクあたり 4 セクタ 32 グループ (32 KB) を保護するセクタグループ書込み保護 (Sector Group WRite Protection : WRPSG)
 - 2 つのセキュア・オンリー領域 (ユーザ Flash バンクあたり 1 つ) : 有効になっている場合、これらの領域はマイクロコントローラがセキュア・アクセス・モードで動作してる場合のみアクセス可能。
 - 起動コードを一時的に隔離する HDP 保護
- 2 KB の一度だけプログラム可能な (One Time Programmable : OTP) エリア
- STMicroelectronics によって設定された読出し専用エリア
- プリフェッチが Flash メモリから次の連続命令を読み出します
- バンクごとに最大 48 KB が高サイクル機能をサポート (100K サイクル) し、データに使用 (EEPROM エミュレーション)

7.3 Flash の機能説明

7.3.1 FLASH のブロック図

図 20 に内蔵 Flash メモリのブロック図を示します。

図 20. FLASH のブロック図 (簡略化)



7.3.2 FLASH 信号

Flash メモリには、Flash AHB レジスタ・インタフェースとメイン AHB インタフェースという、2 つの AHB 接続があります。

Flash AHB レジスタ・インタフェース

- データサイズは 32 bit
- いくつかのレジスタ（制御のためにアンロック・シーケンスの挿入に使用される FLASH_NS/SECKEYR、FLASH_NS/SECOBKKEYR、FLASH_OPTKEYR、および 32 bit での書込みができるオプション・レジスタ）を除き、すべてのレジスタが 8 bit、16 bit、32 bit での読出しおよび書込みが可能です。
- 制御のためのアンロック・シーケンスおよびオプション・レジスタが誤っていた場合、バス・エラーが立てられるか、さもなければ読出しまたは書込み失敗エラーがバスに生成されます。

メイン AHB インタフェース

AHB データ・バスサイズは 128 bit です。このインタフェースは次に示す 3 つの異なるターゲットを扱うために使用されます。

- ユーザおよびシステムメモリに配置されたコード。これは 9 bit の ECC により保護されます。
- OBKey セクタに配置されたセキュア・キーで、9 bit の ECC により保護されます。
- 6 bit の ECC で保護される、OTP 読出し専用メモリおよびFlash メモリの高サイクル・データ領域。

メイン AHB インタフェースは次のように実装されます。

- ユーザおよびシステムメモリ、OBKeys ストレージ。
 - 複数のデータ長をサポート：データ幅 128、64、32、16、8 bit。
 - 各バンクには 128 bit の読出しバッファがあり、最後に読み出したデータが格納されます。読出しバッファにデータがある場合、Flash メモリへの読出しアクセスは行われません。書込みアクセス、OTP アクセス、OBK スワップ、OBK 代替セクタ消去、高サイクル・データ領域アクセス、ユーザ・オプション変更リクエスト、消去操作のいずれかが発生した場合には、バッファが一扫されます。
 - 読出しバッファと同じサイズのプリフェッチが行われます。
 - 9 bit ECC は、各 128 bit データの Flash メモリ・ワードに対応しています。
- OTP、読出し専用、および Flash の高サイクル・データ
 - 2 つの 137 bit の専用データ・バッファを用いて 6 bit の ECC により 16 bit データの管理を行います。
 - 同じアドレスを 2 回読みだすことで、2 回の Flash 読出しアクセスをトリガします。
 - 読出しアクセスの間にメモリ・ウェイト・ステートに加え 2 つのウェイト・ステートが追加されます。これらのウェイト・ステートはデータ・バッファを解析するのに必要です。
 - 各書込みアクセスが Flash メモリへの書込みをトリガします。
 - 6 bit ECC は、各 16 bit データに対応しています。

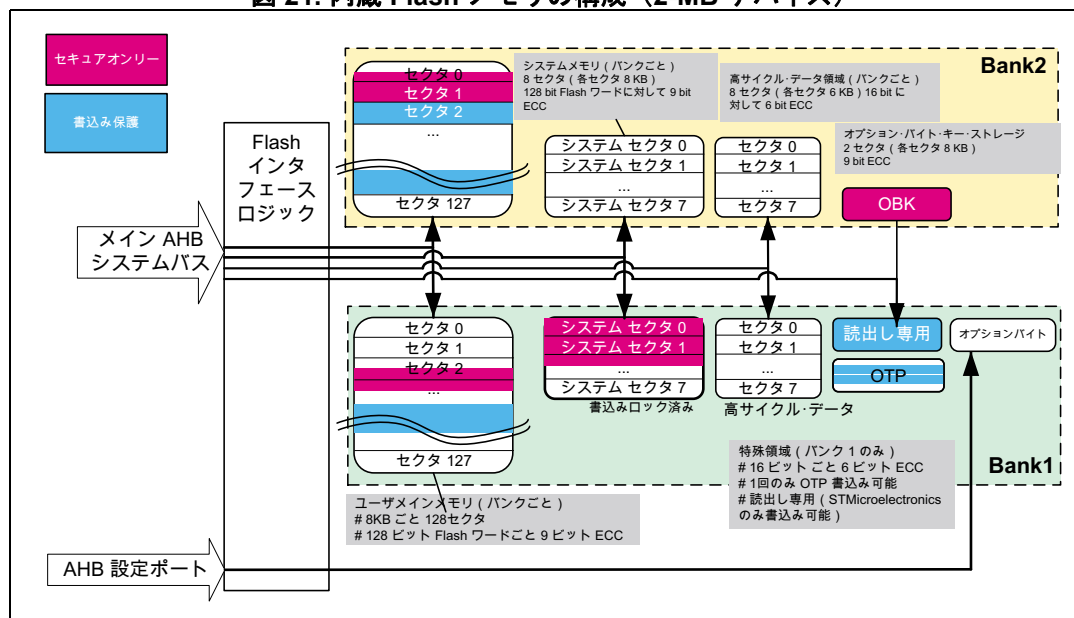
すべての AHB メモリ範囲がデフォルトでキャッシュ可能です。キャッシュが実用的ではない領域（OTP、RO、データ領域）では、ローカル・キャッシュ格納可能性を無効にするために MPU を使用しなくてはなりません。

7.3.3 Flash メモリのアーキテクチャおよび使用

Flashメモリのアーキテクチャ

図 21 に、内蔵 Flash メモリがサポートしている構成を示します。

図 21. 内蔵 Flash メモリの構成 (2-MB デバイス)



内蔵の不揮発性 Flash メモリの構成は次のとおりです。

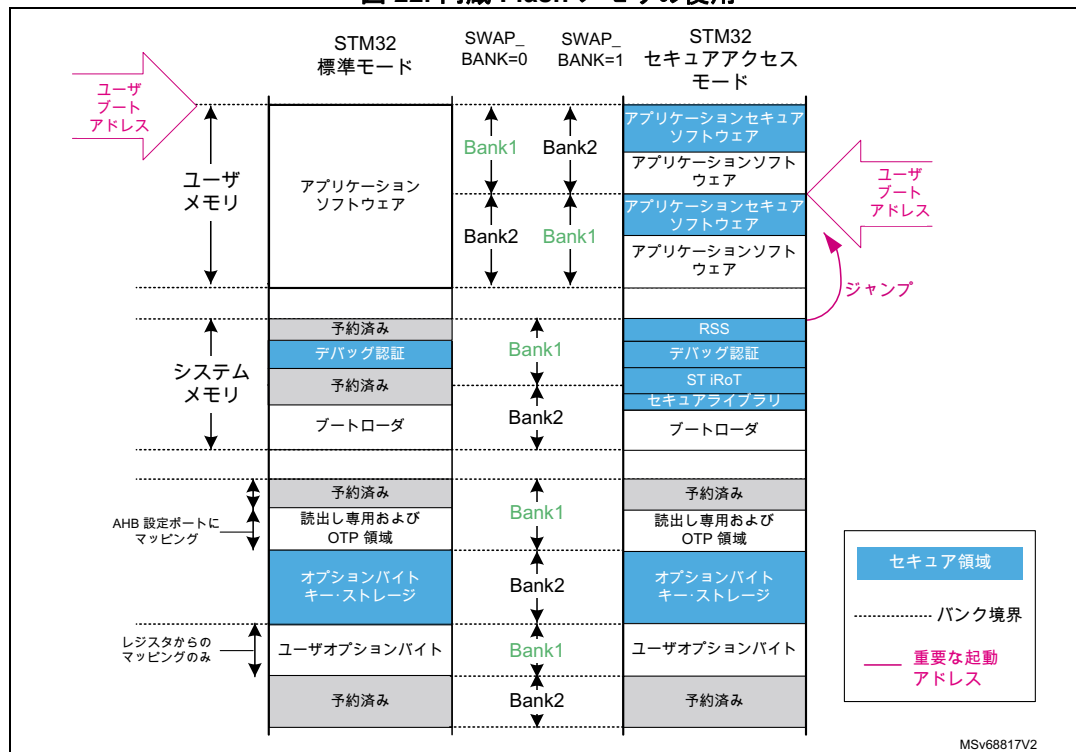
- 各 1 MB の 2 つのバンクで構成されている、2 MB メイン・メモリ・ブロック。各バンクは、8 KB ごとに 128 セクタに分割され、ワードあたり 128 bit + ECC 9 bit の Flash ワード Row を提供します。
- 2 つの 64 KB バンクに分割される 128 KB のシステム・メモリ・ブロック。各バンクは、8 つの 8 KB セクタに分割されます。システム Flash メモリは ECC 保護 (128 bit ワード ごとに 9 bit ECC) されています。
- リセット時に内蔵 Flash メモリによってロードされ、AHB 設定レジスタ・インタフェース経由のみでアプリケーション・ソフトウェアからアクセス可能な、一連の不揮発性オプション・バイト。
- アプリケーション・ソフトウェアによって一度だけ書込み可能な 2 KB の一度だけプログラム可能な (One-Time-Programmable : OTP) 領域。
- 2 KB の読取り専用領域。これには、一意のデバイス ID と製品情報が含まれています。
- セキュア・キー・ストレージの 2 つのメモリ・セクタ (2 × 8 KB)、OBKeys。
- データ用の高サイクリング機能 (100K サイクル) を持ったユーザ Flash メモリの最大 16 セクタ。各バンク 8 セクタ。

Flash メモリのアーキテクチャ全体と、対応するアクセス・インタフェースの概要を表 40 に示します。

パーティションの使用

STMicroelectronics およびアプリケーション・ソフトウェアによる、内蔵 Flash メモリを使用する方法を図 22 に示します。

図 22. 内蔵 Flash メモリの使用



ユーザ・メモリとシステム・メモリの使用方法は、製品状態やその他のオプション・バイトの設定により異なります。

- ユーザメモリはアプリケーション・コードとデータを含んでいますが、システム・メモリはルート・セキュア・サービス (RSS)、デバッグ認証コード、STM32 ブートローダで使用されています。リセットが発生すると、BOOT ピン、(SEC/NS)BOOTADD オプション・バイト、製品状態により設定されるブート・アドレスにコアがジャンプします。
- 一意のブート・エントリ (BOOT_UBE) は、システム Flash メモリに配置された ST iRoT またはユーザ Flash メモリの iRoT のいずれかに設定されます。この機能は暗号化が有効になっているデバイスでのみ使用可能です。
- システム Flash メモリのセキュア・サービス・ライブラリはセキュア・モードで利用可能です。
- デバッグが製品に添付されている場合、エントリ・ポイントは、デバイスをデバッグにアタッチしたときに SBS でデバイスをアンロックするデバッグ認証ポリシーです。デバッグが許可された場合、製品状態の回帰を行うためにデジタル署名を用意しなくてはなりません。
- (SEC/NS)BOOTADD が未だ設定されていない場合は、専用のライブラリをセキュア・ブートに使用することが可能です。これらはシステム Flash メモリにあります。
 - システム Flash メモリの ST ライブラリは、セキュア・ブートやセキュア・ファームウェア・インストール (SFI-RSS) などの特別な機能でアプリケーション・ソフトウェアのブートをサポートします。
 - ユーザ Flash メモリの ST iRoT (不変の信頼の基点 (ルート・オブ・トラスト)) セキュア・ソフトウェアを、セキュア・ファームウェアの更新とプロビジョニング (SFU) に使用します。

注： ブートのためのオプション・バイト設定の詳細については、[セクション 7.4.6](#) を参照してください。

7.3.4 FLASH 読出し操作

読出し操作の概要

メインおよびシステム Flash メモリへの読出しアクセスは次のように動作します。

- 各バンクに関連した、128 bit 読出し・データ・バッファがあり、最後に読み出したデータを格納します。連続するいくつかの読出しアクセスが同じ Flash データワード (128 bit) に属するデータを要求する場合、データは現在のデータ読出しバッファから直接読み出され、さらなる Flash 読出し操作をトリガすることはありません。このメカニズムは、読出しアクセスが許可されるたびに発生します。セキュリティ上の理由により読出しアクセスが拒否されると、対応する読出しエラー応答が内蔵 Flash メモリによって発行され、Flash メモリに対する読出し操作はトリガされません。
- 書込みアクセスまたは OPT アクセス、ユーザ・オプション変更リクエスト、OBK スワップ、OBK 消去、その他の消去操作のいずれかが発生した場合、読出しデータバッファは無効になります。

OTP、RO、Flash の高サイクルデータは次のように動作します。

- 137 bit の Flash データ・ワードが読み出され、一時バッファに格納されます。
- インタフェースが 137 bit データ・ワードを解析し、リクエストされた 16 または 32 ビット・データを選択します。
- 137 bit データ・ワードを解析している間、(2 つの) ウェイト・ステートが追加され、AHB バスが停止します。
- アプリケーションが既に関与していない OTP データ、あるいは Flash の高サイクルデータを読み出すと、**ダブル ECC エラー**が報告され、**設定ビットのみのワードを返します** (詳細は [セクション 7.3.9](#) を参照してください)。ダブル ECC エラーが未使用データによるものか、本当の ECC エラーなのかをユーザが確認できるように、(16 bit の) 読出しデータが FLASH_ECCDR レジスタに格納されます。
- 同一のアドレスを 2 回読み出すと、Flash メモリの 2 回読出しがトリガされます。
- 8 bit アクセスでは AHB バス・エラーが生成されます。

注： 内蔵 Flash メモリは、読出し操作が実行されている間に、シングルエラー訂正とダブルエラー検出を実行可能です ([セクション 7.3.8](#) を参照)。

命令プリフェッチ

Cortex-M33 が有効になっている場合、命令キャッシュを経由し C バスを使って命令とリテラル・プール (定数/データ) をフェッチします。プリフェッチ・ブロックは、命令キャッシュが有効になっているときに、キャッシュのリフィル遅延を減らすことで C バス・アクセスの効率を向上することを目的としています。

プリフェッチは連続コードの場合に効率的で、Flash メモリのプリフェッチは、メモリからの次の連続命令ラインの呼び出しを可能とし、一方、現在の命令ラインは命令キャッシュにフィルされ CPU により実行されます。

プリフェッチは、FLASH アクセス制御 レジスタ (FLASH_ACR) の PRFTEN ビットをセットすると有効になります。Flash メモリのアクセスに 1 つ以上のウェイト・ステートが必要な場合にのみ PRFTEN を設定しなくてはなりません。

読出しタイミング制約の調整

不揮発性メモリからデータを読み出す前に、内蔵クロックを有効にして動作させる必要があります。

メモリから正しくデータを読み出すには、メイン AHB インタフェース・クロック周波数と、デバイスの内部電圧範囲 (V_{CORE}) に合わせて、ウェイト・ステート数 (LATENCY) を、アクセス制御レジスタ (FLASH_ACR) に正しくプログラムしなくてはなりません。

表 37 に、ウェイト・ステート数 (LATENCY)、プログラミング遅延パラメータ (WRHIGHFREQ)、内蔵 Flash メモリ・クロック周波数およびその供給電圧範囲の対応を示します。

表 37. FLASH 推奨ウェイトステート数とプログラミング遅延

ウェイト・ステート数 (LATENCY)	プログラミング 遅延 (WRHIGHFREQ)	インタフェース・クロック周波数対 V_{CORE} レンジ ⁽¹⁾			
		VOS3 範囲 0.95~1.05 V	VOS2 範囲 1.05~1.15 V	VOS1 範囲 1.15~1.26 V	VOS0 範囲 1.30~1.40 V
0 WS (1 FLASH クロックサイクル)	00	0 から 20 MHz	0 から 30 MHz	0 から 34 MHz	0 から 42 MHz
1 WS (2 FLASH クロックサイクル)		20 から 40 MHz	30 から 60 MHz	34 から 68 MHz	42 から 84 MHz
2 WS (3 FLASH クロックサイクル)	01	40 から 60 MHz	60 から 90 MHz	68 から 102 MHz	84 から 126 MHz
3 WS (4 FLASH クロックサイクル)		60 から 80 MHz	90 から 120 MHz	102 から 136 MHz	126 から 168 MHz
4 WS (5 FLASH クロックサイクル)	10	80 から 100 MHz	120 から 150 MHz	136 から 170 MHz	168 から 210 MHz
5 WS (6 FLASH クロックサイクル)		N/A	N/A	170 から 200 MHz	210 から 250 MHz

1. 1.26 から 1.30 V の電圧範囲は使用できません。

システム周波数の調整

電源投入後、VOS3 の電圧スケーリング値に設定された電圧範囲において、内蔵 Flash メモリには 64 MHz ハイスピード内部オシレータ (HSI) からクロックが供給されます。連続した 3 つのウェイト・ステート遅延は FLASH_ACR レジスタで指定されています (表 37 参照)。

バス周波数の変更時、アプリケーション・ソフトウェアはメモリのアクセスに必要なウェイト・ステート数を調整するために、以下に説明するシーケンスに従う必要があります。

CPU 周波数を高くする。

- 必要に応じて、FLASH_ACR レジスタの LATENCY ビットと WRHIGHFREQ ビットの値を表 37 に記載されている正しい値にプログラムします。
- Flash_ACR レジスタを読み直して、新しいウェイトステート数が反映されていることを確認します。
- リセットおよびクロック・コントローラ (RCC) の RCC_CFGR レジスタで、内蔵 Flash メモリ・クロック・ソースやクロック・プリスケラを修正します。
- リセットおよびクロック・コントローラ (RCC) の RCC_CFGR レジスタで、内蔵 Flash メモリ・クロック・ソース・ステータスやプリスケラ値を読み直して、新しい内蔵 Flash メモリのクロック・ソースや新しい AHB クロック・プリスケラ値が反映されていることを確認します。

CPU 周波数を低くする。

1. リセットおよびクロック・コントローラ (RCC) の RCC_CFGR レジスタで、内蔵 Flash メモリ・クロック・ソースやクロック・プリスケアラを修正します。
2. リセットおよびクロック・コントローラ (RCC) の RCC_CFGR レジスタで、内蔵 Flash メモリ・クロック・ソース・ステータスや AHB インタフェース・プリスケアラ値を読み直して、内蔵 Flash メモリの新しいクロックソースや新しいクロック・プリスケアラ値が反映されていることを確認します。
3. 必要に応じて、FLASH_ACR レジスタの LATENCY ビットと WRHIGHFREQ ビットの値を [表 37](#) に記載されている正しい値にプログラムします。
4. Flash_ACR レジスタを読み直して、新しいウェイトステート数が反映されていることを確認します。

エラーコード訂正 (ECC)

このメモリは、エラー訂正メカニズムを内蔵しています。読出し操作のたびにシングルエラー訂正とダブルエラー検出が行われます。詳細については、[セクション 7.3.8](#) を参照してください。

読出しエラー

ECC メカニズムでは読出し操作の修正が行えない場合、[セクション 7.9.10](#) に記載されているように、読出しエラーがメモリから報告されます。

読出し割込み

詳細については、[セクション 7.10](#) を参照してください。

7.3.5 FLASH プログラム操作

プログラム操作の概要

プログラム操作は書き込みコマンドの発行で構成されます。メモリは 1 度に 1 つの書き込みコマンドの実行のみをサポートしています。書き込み中書き込みはサポートしていません。未使用の Flash ワードの上書きは可能ですが、推奨はされていません。結果として不正なデータや一貫しない ECC コードになる可能性があります。

ユーザ Flash、OBK ストレージ、システム Flash メモリ・セクタ

ユーザおよびシステム Flash メモリでは、9 bit ECC が各 128 bit データの Flash ワードに対応しています。この場合、内蔵 Flash メモリは、不揮発性メモリに必ず 128 bit の粒度で書き込み動作を実行しなくてはなりません。書き込みバッファがフル (128 bit) になると、Busy フラグがセットされ、プログラム操作がトリガされます。

バンク 1 と 2 に共通の書き込みバッファがあり、複数の書き込みアクセス・タイプ (128、64、32、16、8 bit) をサポートしています。アプリケーションは 8 bit から 128bit まで書き込みが可能です。この場合、128 bit + ECC への強制書き込み機能が使用されます (FLASH_NS/SECCR レジスタの FW ビット参照)。

書き込みリクエストがメモリに対して発行されると、新しい書き込みリクエストはメイン AHB バスをストールさせ、さらに書き込み動作中の同じバンクへの新しい読込みリクエストも、メイン AHB バスをストールさせます。

OTP、RO、Flash の高サイクルデータ

ターゲット・メモリがOTP、RO、Flash の高サイクル・データ・セクタの場合、各 16 bit データ Flash ワードに対して 6 bit の ECC コードが対応します。内蔵 Flash メモリは 16 bit または 32 bit の書き込み動作をサポートしていますが、8 bit 書き込み動作はサポートしておらず、8 bit アクセスでは書き込みアクセスが無視されます。書き込みデータ・バッファはありません。各書き込みアクセスが Flash メモリの書き込みをトリガします。

注： [セクション 7.3.9](#) に記載されているように、最終製品において OTP エリアは書き込み保護されているのが一般的です。

書き込みリクエストを受け取ると（アドレス・フェーズ中に）書き込み保護確認を行います。書き込みバッファの出力時には書き込み保護確認は行いません。書き込み保護違反が検出された場合、書き込み操作はキャンセルされ、FLASH_NS/SECSR レジスタに書き込み保護エラー（WRPERR）が立ちます。

注： メイン Flash、ICP、OTP の書き込み保護は[セクション 7.6](#)を参照してください。

進行中の書き込み動作の監視

アプリケーション・ソフトウェアは、進行中の書き込み動作を監視するために FLASH_NS/SECSR のステータスフラグを使用できます。1 度に 1 つの操作しかできないので、このフラグは、どのバンクかに関わらず、何らかの操作（書き込み、消去、オプション変更）が進行中であることを示しています。

- **BSY**：不揮発性メモリで、有効な書き込み、消去、オプション・バイトの変更、OBK スワップ、OBK 代替セクタ消去が実行されていることを示します。このフラグは特定のバンク専用ではありません。これは、バンクに関わらずメモリに対して操作が開始しているときに、設定されます。操作のトリガは次のとおりです。
 - 消去（FLASH_NS/SECCR.STRT）
 - 書き込み（FLASH_NS/SECCR.PG + AHB write）
 - オプション変更（FLASH_OPTCR.OPTSTRT）
 - OBKeys セクタ・スワップおよび OBKeys セクタ消去（FLASH_NS/SECOBKCFGR.SWAP_SECT_REQ、FLASH_NS/SECOBKCFGR.ALT_SECT_ERASE）

これらは、現在の操作が終了するか、エラー時にクリアされます。

- **WBNE**：このビットは、新しいデータが 128 ビット書き込みバッファを完了するのを、内蔵 Flash メモリが待っていることを示します。この状態では、書き込みバッファは空ではありません。これは、アプリケーション・ソフトウェアが書き込みバッファを埋めるか、FLASH_NS/SECCR の FW ビットを使用して強制的に書き込みを行うか、あるいはエラーが検出されると直ぐにリセットされます。WBNE がハイの場合、消去、オプション変更、OBK スワップ、あるいは OBK 代替セクタ消去操作を Flash メモリで開始することはできません。
- **DBNE**：このビットは 16 bit データを解析するデータ・バッファが空ではないことを示します。
 - 16 bit データ書き込みアクセスを受け付けてデータ・バッファがフィルになっています。これは、有効な書き込みアクセスを受け付けたときにセットされ、書き込みリクエスト準備の処理が終わると直ぐにリセットされます。

注： AHB 書き込みリクエストの受付時にメモリがビジーだった場合、CPU の実行がストールします。

書き込み操作の有効化

バンク 1 またはバンク 2 のユーザ Flash メモリをプログラムする前に、アプリケーション・ソフトウェアは PG ビットが FLASH_NS/SECCR で 1 に設定されていることを確認しなくてはなりません。そうになっていない場合には、アンロックシーケンスを使用して（[セクション 7.6.7](#)を参照）、PG ビットをセットする必要があります。

オプション・バイトまたはオプション・バイト・キーの修正や全体消去を開始しなくてはならない場合、アプリケーション・ソフトウェアは FLASH_OPTCR がアンロックされていることを確認する必要があります。

があります。そうならない場合には、アンロック・シーケンスを使用する必要があります (セクション 7.6.7 を参照)。

FLASH_NS/SECOBKCFGR には、類似の用途の別のメカニズムが存在します。制御レジスタは、OBKeys ストレージの変更を開始する前に必ずアンロックする必要があります。FLASH_NS/OBKKEYR に正しいシーケンスを書き込むと FLASH_NS/OBKCFGR をアンロックすることができます。セクション 7.6.7 で説明するように、FLASH_SECOBKKEYR は FLASH_SECOBKCFGR にリンクしています。

注： アプリケーション・ソフトウェアは、既にアンロックされているレジスタをアンロックしてはなりません。さもなければ、そのレジスタは次のシステム・リセットまでロックされたままになります。

プログラミングタイミグ制約の調整で説明するように、アプリケーション・ソフトウェアは、必要に応じてプログラミング遅延を更新可能です。

FLASH コントロール・レジスタ、FLASH_NS/SECCR および FLASH_OPTCR への書き込み

FLASH_NS/SECCR、FLASH_OPTCR、FLASH_NS/SECOBKCFGR レジスタは BSY ビットがセットされていると、書き込みモードでのアクセスができません。BSY ビットがセットされているときに、これらのレジスタへの書き込みを行おうとすると、SEC/NSBSY ビットがクリアされるまで AHB バスがストールします。

シングル書き込みシーケンス

推奨されるシングル書き込みシーケンスは、次のとおりです。

1. 保護メカニズムがプログラミングを妨げていないことを確認します。
2. Flash メモリ操作が進行中ではないことを、FLASH_NS/SECSR レジスタの BSY ビットと、FLASH_NS/SECSR レジスタの CDBNE ビットで確認します。書き込みバッファが空であることを FLASH_NS/SECSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング/消去操作によるエラー・フラグをすべて確認し、クリアします。
4. (レジスタがまだアンロックされていない場合に限り) セクション 7.6.7 を参照して FLASH_NS/SECCR レジスタをアンロックします
5. FLASH_NS/SECCR レジスタの PG ビットをセットし書き込み操作を有効にします。
6. 整列アドレスに Flash ワードを 1 つ書き込みます。

注： NS/SECWBNE フラグは 128 bit 書き込みバッファが新しいデータを待っていることを示します。

注： 最初の書き込みと書き込み操作の終了までの間、消去リクエスト、オプション変更リクエスト、OBK 操作は禁止されます。

7. 対応する FLASH_NS/SECSR レジスタの BSY ビットがクリアされるのを待ちます。
8. 他のプログラミング・リクエストが無くなったなら、FLASH_NS/SECCR レジスタの PG ビットをクリアします。

ステップ 6 が順に (例えばバイト単位など) に実行される場合、書き込みバッファが部分的に満たされている可能性があります。この場合、アプリケーション・ソフトウェアは、FLASH_NS/SECCR レジスタの FW ビットを使用して書き込みバッファに格納されているものを強制書き込みすることができます。この特殊な例では、書き込まれないビットは自動的に 1 にセットされます。書き込みバッファのどのビットも 0 にクリアされない場合、FW ビットは影響しません。

注： 強制書き込み操作を使用すると、1 とは異なる値を持つ失われたビットを、後の段階でアプリケーションが更新するのを妨げます。これは予期しない、あるいは一貫しないデータまたは ECC の原因となる場合があります。

プログラミングタイミング制約の調整

プログラム操作のタイミング制約は、パフォーマンスに直接的な影響を及ぼすメモリクロックの周波数に依存します。タイミング制約が厳しすぎると不揮発性メモリが正しく動作せず、緩すぎるとプログラミング速度が最適ではなくなります。

このため、ユーザは Flash_ACR レジスタの WRHIGHFREQ パラメータを通じて最適なプログラミング遅延の調整を行う必要があります。メモリ・クロックの周波数に応じたプログラミング遅延の推奨値については、[表 37](#) を参照してください。

FLASH_ACR 設定レジスタは、両方のバンクに共通です。

アプリケーション・ソフトウェアは、WRHIGHFREQ を変更する前に、プログラム／消去操作が進行中ではないことを確認する必要があります。

注意： プログラミング／メモリの消去中に WRHIGHFREQ を変更すると、その内容を破損する可能性があります。

プログラミングエラー

プログラム操作が失敗すると、[セクション 7.9](#) で説明するようにエラーが報告されます。

プログラミング割込み

詳細については、[セクション 7.10 : FLASH 割込み](#)を参照してください。

7.3.6 FLASH 消去操作

消去操作の概要

メモリは、8 KB ユーザ・セクタ、1 つのユーザ Flash メモリ・バンク、2 つのユーザ Flash メモリ・バンクのいずれかに対して消去操作（例えば全体消去）を実行可能です。ユーザ Flash メモリ、ICP、ユーザ・オプション、OTP 消去保護の詳細については [セクション 7.6](#) を参照してください。

消去コマンドは、AHB 設定インタフェースを通じて発行されます。メモリは 1 回に 1 つの操作のみをサポートしており、書込みと消去のリクエストを同時に受け取ると、エラー・フラグを立てて両方の操作がキャンセルされます。詳細については、[セクション 7.9](#) を参照してください。

消去と WRP

アプリケーション・ソフトウェアが書込み保護されているユーザ・セクタを消去しようとする、[セクション 7.9.2](#) で説明するように、セクタの消去操作をアボートし FLASH_NS/SECCR レジスタの WRPERR フラグが立ちます。

Flash・ビジー

[進行中の書込み動作の監視](#)でビジー信号の説明を行います。

FLASH コントロール・レジスタ、FLASH_NS/SECCR および FLASH_OPTCR への書込み

[FLASH コントロール・レジスタ、FLASH_NS/SECCR および FLASH_OPTCR への書込み](#)を参照してください。

消去操作の有効化

アプリケーション・ソフトウェアは、セクタを消去する前に FLASH_NS/SECCR がアンロックされていることを確認しなくてはなりません。そうっていない場合には、アンロック・シーケンスを使用する必要があります ([セクション 7.6.7](#) を参照)。

注： アプリケーション・ソフトウェアは、既にアンロックされているレジスタをアンロックしてはなりません。さもなければ、そのレジスタは次のシステム・リセットまでロックされたままになります。これを意図的に利用して、レジスタへのさらなるアクセスができないようにロックアウトすることも可能です。

同様な制約がバンク消去リクエストにも適用されます。

標準 Flash セクタ消去シーケンス

8 または 6 KB データのユーザ・セクタをセキュリティ保護なしで消去する場合は、次の手順に従ってください。

1. 保護メカニズムがセクタ消去を妨げていないことを確認します (WRP、セキュアフラグ、HDP)。
2. Flash メモリ操作が何も行われていないことを FLASH_NSSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_NSSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング／消去操作による非セキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#) を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_NSCR レジスタをアンロックします。
5. FLASH_NSCR レジスタの BKSEL ビット、SER ビット、SNB ビット・フィールドをセットします。BKSEL がどの物理バンク・セクタを消去しなくてはならないかを示し、SER はセクタ消去操作を示します。また、SNB にはターゲット・セクタ番号が格納されています。データ・セクタには、対応する通常のセクタの番号を使用します。
6. FLASH_NSCR レジスタの STRT ビットをセットします。
7. FLASH_NSSR レジスタの BSY ビットがクリアされるのを待ちます。
8. STRT ビットはセクタ消去の最後、またはエラーの場合に自動的に消去されます。
9. セクタ消去のリクエストがこれ以上発行されなくなったら、FLASH_NSCR レジスタの SER をクリアします。

注： セクタ消去に別の消去フラグが同時にリクエストされた場合、PGSERR エラーが生成されます。

セキュアな Flash メモリ・セクタの消去シーケンス

8 または 6 KB データのセキュリティが設定されたユーザ・セクタを消去する場合は、次の手順に従ってください。

1. 書込み保護または HDP メカニズムがセクタ消去を妨げていないことを確認します。
2. Flash メモリ操作が何も行われていないことを (FLASH_SECSR レジスタの BSY ビットと DBNE ビットで) 確認し、書込みバッファが空になっていることを (FLASH_SECSR レジスタの WBNE ビットで) 確認します。
3. 以前のプログラミング／消去操作によるセキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#) を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_SECCR レジスタをアンロックします。
5. FLASH_SECCR レジスタの BKSEL ビット、SER ビット、SNB ビット・フィールドをセットします。BKSEL がどの物理バンク・セクタを消去しなくてはならないかを示し、SER はセクタ消

去操作を示します。また、SNB にはターゲット・セキュア・セクタ番号が格納されています。データ・セクタには、対応する通常のセクタの番号を使用します。

6. FLASH_SECCR レジスタの STRT ビットをセットします。
7. FLASH_SECCR レジスタの BSY ビットがクリアされるのを待ちます。
8. STRT ビットはセクタ消去の最後、またはエラーの場合に自動的に消去されます。
9. セキュア・セクタ消去のリクエストがこれ以上発行されなくなったら、FLASH_SECCR レジスタの SER をクリアします。

注： セクタ消去に別の消去フラグが同時にリクエストされた場合、PGSERR エラーが生成されます。

標準 Flash メモリ・バンクの消去シーケンス

どのセクタもセキュアとして設定されていないバンクの消去。

1. 保護メカニズムがセクタ消去を妨げていないことを確認します。
2. Flash メモリ操作が何も行われていないことを FLASH_NSSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_NSSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング／消去操作による非セキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#) を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_NSCR レジスタをアンロックします。
5. FLASH_NSCR レジスタの BKSEL ビットと BER ビットをターゲットの物理バンクにセットします (スワップ設定は無視されます)。
6. 次に、FLASH_NSCR レジスタの STRT ビットをセットして、バンク消去操作を開始します。FLASH_NSSR レジスタの BSY ビットがクリアされるのを待ちます。
7. STRT ビットは消去シーケンスの最後またはエラーの場合に自動的に消去されます。
8. 他のバンク消去のリクエストが発行されなくなったら、FLASH_NSCR レジスタの BER をクリアします。

セキュア Flash メモリ・バンクの消去シーケンス

すべてのセクタがセキュア・フラグで設定されているバンクの消去。

1. 保護メカニズムがセクタ消去を妨げていないことを確認します (全体消去では HDP も考慮され、HDP が定義されている場合は、HDPL = 2, 3 からはこれを完全に実行することができないことに注意してください)。
2. Flash メモリ操作が何も行われていないことを FLASH_SECSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_SECSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング／消去操作によるセキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#) を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_SECCR レジスタをアンロックします。
5. FLASH_SECCR レジスタの BKSEL ビットと BER ビットをターゲットの物理バンクにセットします (スワップ設定は無視されます)。
6. FLASH_SECCR レジスタの STRT ビットをセットして、バンク消去操作を開始します。次に、FLASH_SECSR レジスタの BSY ビットがクリアされるのを待ちます。
7. STRT ビットは消去シーケンスの最後またはエラーの場合に自動的に消去されます。
8. 他のバンク消去のリクエストが発行されなくなったら、FLASH_SECSR レジスタの BER をクリアします。

Flash 全体消去シーケンス

非セキュア・アクセスを使用して両方のバンクのすべてのセクタを消去するには、すべてのセクタが非セキュアに設定されていなくてはなりません。アプリケーション・ソフトウェアは、次に示す手順で FLASH_NSCR レジスタの MER ビットを 1 にセットすることが可能です。

1. 保護メカニズムが全体消去を妨げていないことを確認します([セクション 7.6](#))。
2. Flash メモリ操作が何も行われていないことを FLASH_NSSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_NSSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング/消去操作による非セキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#)を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_NSCR レジスタをアンロックします。
5. FLASH_NSCR レジスタの MER ビットに 1 をセットします。
6. FLASH_NSCR レジスタの STRT ビットをセットします。次に、FLASH_NSSR レジスタの BSY ビットがクリアされるのを待ちます。
7. STRT ビットは消去シーケンスの最後、またはエラーの場合に自動的にクリアされます。
8. FLASH_NSCR レジスタの MER をクリアします。

セキュアな Flash メモリの全体消去シーケンス

セキュア・アクセスを使用して両方のバンクのすべてのセクタを同時に消去するには、すべてのセクタがセキュアに設定されていなくてはなりません。アプリケーション・ソフトウェアは、次に示す手順で FLASH_SECCR レジスタの MER ビットを 1 にセットすることが可能です。

1. 保護メカニズムが全体消去を妨げていないことを確認します([セクション 7.6](#))。
2. Flash メモリ操作が何も行われていないことを FLASH_SECSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_SECSR レジスタの WBNE ビットで確認します。
3. 以前のプログラミング/消去操作によるセキュア・エラー・フラグをすべて確認し、クリアします。詳細については、[セクション 7.9](#)を参照してください。
4. (レジスタがまだアンロックされていない場合に限り) [セクション 7.6.7](#) を参照して FLASH_SECCR レジスタをアンロックします。
5. FLASH_SECCR レジスタの MER ビットに 1 をセットします。
6. FLASH_SECCR レジスタの STRT ビットをセットします。次に、FLASH_SECSR レジスタの BSY ビットがクリアされるのを待ちます。
7. STRT ビットは消去シーケンスの最後、またはエラーの場合に自動的にクリアされます。
8. FLASH_SECCR レジスタの MER をクリアします。

注： 全体消去およびバンク消去は、消去したバンクからエイリアス作成した高サイクル・データ・セクタも消去します。

7.3.7 FLASH 並行操作

不揮発性メモリは同じマクロにカプセル化した 2 つの独立したバンクに分割されるので、内蔵 Flash メモリ・インタフェースは、一方のバンクで書込みをしている間の読出し (RWW 書込み中読出し) または他のバンクで消去を実行している間の読出しをサポートします。ただし書込み中書込みや読出し中読出しはサポートしていません。同じことが高サイクルデータ領域、システム Flash ライブラリ、あるいは OBK (バンク 2 に配置) にも言えます。

すべてのケースで、[セクション 7.3.4](#)、[セクション 7.3.5](#)、[セクション 7.3.6](#)に記載されているシーケンスが適用されます。

7.3.8 FLASH エラー保護

エラー訂正コード (ECC)

内蔵 Flash メモリは、SECDED アルゴリズムに基づくエラー訂正コード (ECC) メカニズムをサポートし、シングル・エラーの訂正、ダブル・エラーの検出を行います。

このメカニズムは 128 bit の Flash ワードごとに ECC 9 bit を使用して、ユーザおよびシステム・メモリに適用します。読出し専用の OTP、Flash の高サイクルデータには、より強い 16 bit ワードごとに ECC 6 bit を使用します。OTP または Flash の高サイクルデータの未使用ワード (例えば 22 bit が 1 のワード) にはダブル ECC エラーが生成されます。この OTP ワードまたは Flash の高サイクルデータワードが未使用でなくなると、ECC エラーは消滅します。

より具体的には、128 bit の Flash ワードからの読出し操作のたびに、内蔵 Flash メモリは 9 bit の ECC 情報を取得して Flash ワードの ECC を計算し、その結果を基準値と比較します。不一致の場合、[セクション 7.9.10](#)に記載されているように、対応する ECC エラーが立てられます。

毎回のプログラム操作の間、9 bit の ECC コードが 128 bit のデータ Flash ワードに関連付けられ、結果として得られる 137 bit の Flash ワード情報が不揮発性メモリに書き込まれます。

類似のメカニズムが読出し専用および OTP 領域にも適用されますが、その場合、16 bit データに対して 6 bit の ECC を使います。

7.3.9 OTP および RO メモリ・アクセス

OTP および RO メモリはメイン AHB インタフェース経由でアクセスします。OTP はアドレス 0x08FF_F000 から 0x08FF_F7FF でアクセスが可能で、読出し専用セクションは 0x08FF_F800 から 0x08FF_FFFF でアクセス可能です。

FLASH 一度だけプログラム可能な領域

内蔵 Flash メモリには、アプリケーションの秘密ではない一度だけプログラム可能なデータ (OTP) 専用の 2048 バイトメモリ領域が備わっています。この領域は、16 bit の 1024ワード (+ ECC の 6 bit) から構成されています。消去不可で、一度だけ書込み可能です。OTP 領域は、アドレス 0x08FF_F000 から 0x08FF_F7FE までのメイン AHB インタフェース経由でアクセス可能です。

OTP データはアプリケーション・ソフトウェアにより 16 bitのチャUNK単位でプログラム可能です。既にプログラムされている 16 bit ハーフワードの上書きは、データおよび ECC エラーにつながる可能性があるため、サポートされていません。

注： OTP エリアは、STMicroelectronics からのデバイス供給時には未使用です。

シングルエラーの修正がダブルエラーの検出が行われた OTP データを読み出すと、[セクション 7.9.10](#)に記載されている読出しエラーが内蔵 Flash メモリから報告されます。

アプリケーション・ソフトウェアにより書き出されていない OTP データ（未使用の OTP など）を読み出すと、ECC 訂正がダブルエラー検出 (ECCD) を報告し、データは FLASH_ECCDR レジスタに格納されることになります。ECCD は NMI が立ったことを暗に示しています。

OTP 書き込み保護

表 38 に示されているように、OTP データは 32 ブロックの 32 OTP ワードとして構成されます。FLASH_OTPBL オプション・バイト・レジスタの各 OTP ブロックに対応する LOCKBi ビット (i = 0 から 31) を設定することにより、OTP ブロック全体を書込みアクセスから保護（ロック）することが可能です。あるブロックが（部分的であっても）プログラム済みであるかどうかに関わらず、そのブロックを書込み保護することができます。

OTP ブロックのロック操作は不可逆的であり、製品のライフ・ステートとは無関係です。

注： OTP 領域は、読み出しモードのみでアクセス可能です。

表 38. Flash メモリ OTP 構成

表 30. Flash メモリ OTP 構成

OTP ブロック	AHB アドレス	AHB ワード		ロックビット
		[31:16]	[15:0]	
ブロック 0	0x08FF : F000	OTP001	OTP000	LOCKBL0
	0x08FF : F004	OTP003	OTP002	
			
	0x08FF : F03C	OTP031	OTP030	
ブロック 1	0x08FF : F040	OTP033	OTP032	LOCKBL1
	0x08FF : F044	OTP035	OTP034	
			
	0x08FF : F07C	OTP063	OTP062	
ブロック 2	0x08FF : F080	OTP065	OTP064	LOCKBL2
	0x08FF : F084	OTP067	OTP066	
			
	0x08FF : F0BC	OTP95	OTP94	
.....				
ブロック 31	0x08FF : F7C0	OTP993	OTP992	LOCKBL31
	0x08FF : F7C4	OTP995	OTP994	
			
	0x08FF : F7FC	OTP1023	OTP1022	

OTP 書き込みシーケンス

OTP ワードを書き込むには、次のシーケンスに従ってください。

- Flash メモリ操作が何も行われていないことを FLASH_NSSR レジスタの BSY ビットで確認し、データバッファが空になっていることを FLASH_NSSR レジスタの DBNE ビットで確認します。
- 以前のプログラミング/消去操作によるエラー・フラグをすべて確認し、クリアします。
- FLASH_NSCR レジスタの PG ビットをセットします。
- 対象 OTP ワードの保護ステータスを確認します (表 38 を参照)。対応する LOCKBi ビットは 1 以外にセットされる必要があります。
- 表 38 に示す 4 バイト整列アドレスに対応した 2 つの OTP ワード (32 ビット) を書き込みます。あるいは、アプリケーションソフトウェアが 16 ビット MSB または 16 ビット LSB を独立してプログラムすることもできます。この場合、最初の 16 ビット書き込み操作は 2 番目の操作を待つことなく直ちに始まります。
- FLASH_NSSR レジスタの BSY ビットがクリアされるのを待ちます。
- そのバンクで、プログラミング・リクエストがこれ以上ない場合、FLASH_NSCR レジスタの PG ビットをクリアします。
- オプションとして、LOCKBLi を使用して OTP ブロックをロックして、データのこれ以上の変更を防止します。

注 : OTP 16 bit ワードに 2 回書き込まないでください。ECC エラーが発生します。

バイト・レベルでの OTP データの書き込みはサポートされておらず、バス・エラーが発生します。

データ破壊を防止するため、オプションの変更を開始する前に、(OTP 値を読み返すなどにより) OTP 書き込みプロセスを完了させることが重要です。

FLASH 読出し専用領域

内蔵 Flash メモリには、読出し専用データを格納するための 2 KB 領域が用意されています。この領域は AHB のメイン・ポートからアクセスすることができ、セクション 7.3.8 に詳しく説明しているように、ロバストな ECC スキームにより保護されています。

アプリケーション・ソフトウェアにより使用可能な、(STMicroelectronics によりプログラムされた) 読出し専用情報は、表 39 で詳しく説明しています。

表 39. 読出し専用公開データ構成

読出し専用データ名	アドレス	コメント
ユニークデバイス ID	0x08FF : F800	U_ID[31:0]
	0x08FF : F804	U_ID[63:32]
	0x08FF : F808	U_ID[96:64]
Flash メモリサイズ/パッケージ	0x08FF : F80C	Flash メモリサイズ[15:0]/パッケージコード[15:0]
予約済みです。	0x08FF F810 から 0x08FF FFFF	予約情報

7.3.10 Flash の高サイクル・データ

内蔵 Flash メモリは、高サイクル能力 (100K サイクル) を持った最大 96 KB のメモリ領域を持っており、データを格納したり EEPROM をエミュレートしたりします。これは、アドレス 0x0900_0000 から 0x0901_7FFF の AHB システム・ポート経由でアクセス可能です (図 23 参照)。これはバンク 1 と 2 の最後の 8 (または 4) セクタにマッピングされます。この領域はロバストな 6 bit ECC で保護されており、セクタ・サイズを 6 KB に縮小する代わりに、16 bit の粒度での読出し、書込みが可能です。

バンクごとのしきい値 (EDATA(1/2)_STRT) はユーザ・オプション・バイトでプログラム可能で、データの Flash メモリ領域の初めを決定します。デフォルトですべての Flash メモリはコードに使用されます。

例えば、48 KB のデータがバンク 1 で必要な場合、EDATA1_EN を 1 に EDATA1_STRT を 7 にセットします。バンク 2 にはデータが必要ない場合、EDATA2_EN を 0 にセットします。この場合、データはバンク 1 のアドレス 0x0900_0000 から 0x0900_BFFF のアドレスでアクセス可能です。

より効率を高めるために、他のバンクの Flash の高サイクル・データのセクタを使用することが常に推奨されます。これによりアプリケーションは、デュアルバンク配置の RWW 機能を利用できます。

SWAP_BANK 機能が有効になっている場合、バンクがスワップされ、バンク 2 の Flash メモリの高サイクル・データが 0x0900_000 から 0x0900_BFFF のアドレスでアクセス可能になり、バンク 1 のデータは 0x0900_C000 から 0x0901_7FFF のアドレスでアクセス可能になります。

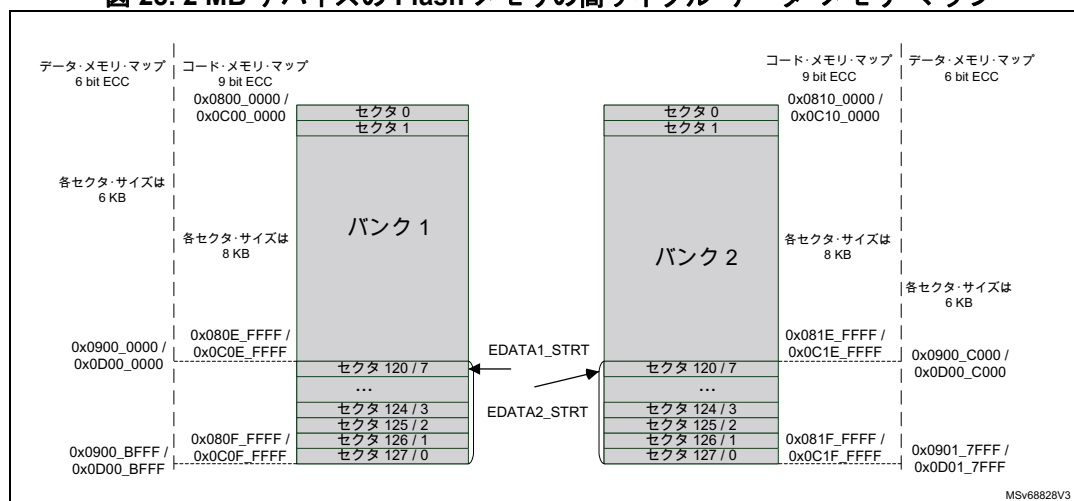
バスエラーは次のように生成されます。

- 0x0900_0000 から 0x0901_7FFF のアドレスへのアクセスを試み、このアドレスが不正である (EDATA(1/2)_EN が有効になっていないか、EDATA(1/2)_STRT が正しくない)。
- Flash メモリの高サイクル・データ領域から命令をフェッチしようとする。

データ領域セクタの消去は、対応するユーザ Flash セクタ (120-127) の通常の消去リクエストで可能です。

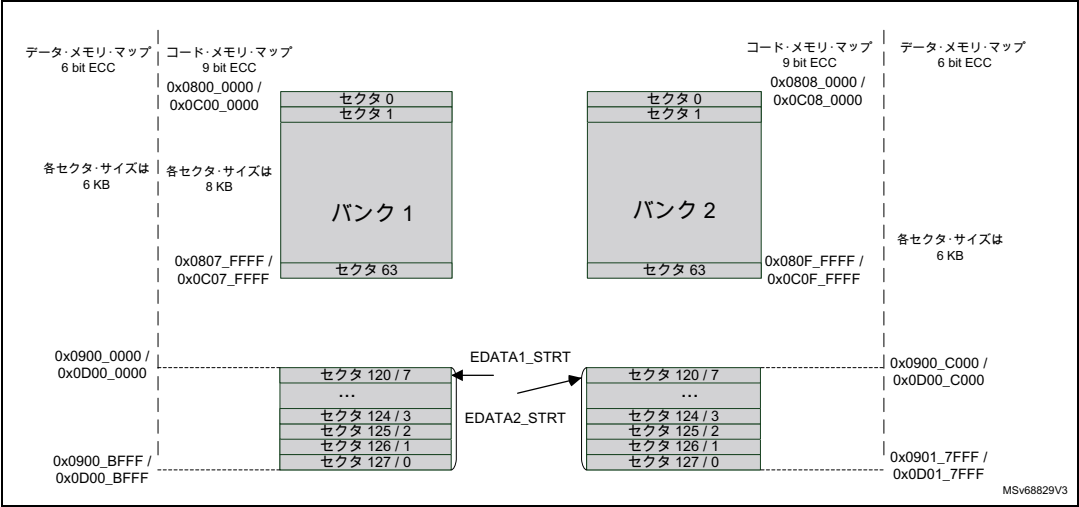
保護と高サイクル・データ領域のセキュリティに関しては、[セクション 7.6.9](#) で詳述しています。

図 23. 2 MB デバイスの Flash メモリの高サイクル・データ・メモリ・マップ



注： バンク 1 の Flash メモリの高サイクル・データ領域が有効になっている場合、2 MB デバイスでは、コード・メモリ・マップがバンク 1 からバンク 2 まで連続ではありません。

図 24. 1 MB デバイスの Flash メモリの高サイクル・データ・メモリ・マップ



7.3.11 Flash バンク・スワッピング

ユーザ Flash メモリのバンク 1 とバンク 2 はスワップ可能です。この機能は、ファームウェアをアップグレードした後に、新しいファームウェアでデバイスを再起動する際に使用できます。バンク・スワッピングは、FLASH_OPTCR レジスタの SWAP_BANK ビットによって制御されるユーザ・オプション・バイト・フラグです。

バンク固有のデータ領域とセキュリティ属性の設定は、元のバンクとその内容に従います。コントロール・ビット BKSEL は常に SWAP_BANK 設定ではなく物理バンクを参照します。

SWAP_BANK ビット値によりアクセス可能なメモリを表 40 に示します。

表 40. メモリ・マップとスワップ・オプション

Flash メモリ 領域	Flash メモリ対応バンク		開始アドレス	終了アドレス	サイズ (バイト)	領域名
	SWAP_BANK = 0	SWAP_BANK = 1				
ユーザ メイン メモリ	Bank1	Bank2	0x0800 0000	0x0800 : 1FFF	8 K	セクタ 0
			0x0800 2000	0x0800 : 3FFF	8 K	セクタ 1
		
			0x080F : E000	0x080F FFFF	8 K	セクタ 127
	Bank2	Bank1	0x0810 0000	0x0810 : 1FFF	8 K	セクタ 0
			0x0810 2000	0x0810 : 3FFF	8 K	セクタ 1
		
			0x081F : E000	0x081F FFFF	8 K	セクタ 127

表 40. メモリ・マップとスワップ・オプション (続き)

Flash メモリ 領域	Flash メモリ対応バンク		開始アドレス	終了アドレス	サイズ (バイト)	領域名
	SWAP_BANK = 0	SWAP_BANK = 1				
システム メモリ	Bank1		0x0BF8 0000	0x0BF8 : 1FFF	8 K	システム 1 セクタ 0
			0x0BF8 2000	0x0BF8 : 3FFF	8 K	システム 1 セクタ 1
		
			0x0BF8 : E000	0x0BF8 FFFF	8 K	システム 1 セクタ 7
	Bank2		0x0BF9 0000	0x0BF9 : 1FFF	8 K	システム 2 セクタ 0
			0x0BF9 0000	0x0BF9 : 3FFF	8 K	システム 2 セクタ 1
		
			0x0BF9 : E000	0x0BF9 FFFF	8 K	システム 2 セクタ 7

FLASH_OPTCR レジスタの SWAP_BANK ビットは、システム・リセットか POR の後に限り、SWAP_BANK オプション・ビットからロードされます。

(たとえば、新しいファームウェア更新の適用のために) SWAP_BANK ビットを変更するには、以下のシーケンスに従ってください。

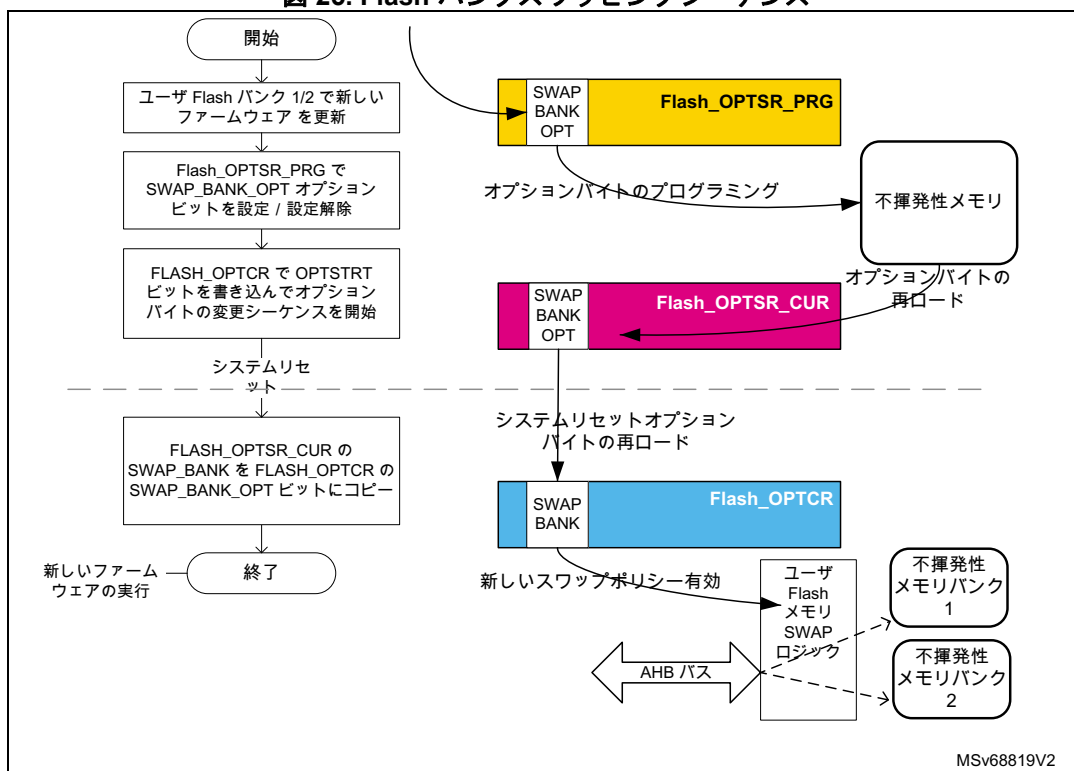
- Flash メモリ操作が何も行われていないことを FLASH_NS/SECSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_NS/SECSR レジスタの WBNE ビットで確認します。
- 以前の操作によるエラー・フラグをすべてクリアします。
- すでにアンロックされていない場合に限り、OPTLOCK ビットをアンロックします。
- 希望する SWAP_BANK の値を FLASH_OPTSR_PRG レジスタにセットします。
- FLASH_OPTCR レジスタの OPTSTRT ビットをセットすることによりオプション・バイトの変更シーケンスを開始します。
- オプション・バイトの変更が完了すると、FLASH_OPTSR_CUR に期待される SWAP_BANK の値が入りますが、FLASH_OPTCR の SWAP_BANK ビットはまだ変更されておらず、バンク・スワッピングはまだ有効ではありません。
- システムリセットまたは POR を強制的に行います。リセットが立ち上がると、バンクスワッピングが有効 (FLASH_OPTCR の SWAP_BANK の値が更新) となり、新しいファームウェアが実行されます。

注 : FLASH_OPTCR の SWAP_BANK ビットは読み出し専用であり、アプリケーション・ソフトウェアは変更できません。

FLASH_OPTSR_PRG の SWAP_BANK オプション・ビットは製品の状態にかかわらず変更可能です。これは PRODUCT_STATE によりロックされる代わりに、(NS/SEC)BOOT_LOCK User OB によってロックされます。

図 25 に、バンクスワッピングシーケンスの概要を示します。

図 25. Flash バンクスワッピングシーケンス



7.3.12 FLASH のリセットとクロック

リセット管理

内蔵 Flash メモリは、リセットおよびクロック制御 (RCC) によって駆動されるコア・ドメイン・リセットによってリセット可能です。このリセットの主な効果は次のとおりです。

- 読出しと書込みの遅延を含めて、オプション・バイトレジスタを除くすべてのレジスタがクリアされます。バンク・スワッピング・オプションが変更されると、それが適用されます。
- 大半の制御レジスタは、書込み操作に対して自動的に保護されています。それらの保護を解除するには、[セクション 7.6.7](#)に記載されている新たなアンロックシーケンスを使用する必要があります。

メモリは、リセットおよびクロック制御 (RCC) によって駆動されるパワーオン・コア・ドメイン・リセットによってリセット可能です。リセットに失敗すると、すべてのオプション・バイトレジスタがリセットされます。リセットが立ち上がると、オプション・バイトがロードされ、場合によっては新しい機能が適用されます。このローディング・シーケンス中、デバイスはリセット状態を維持し、メモリにはアクセスできません。

リセット信号はメモリに重大な影響を与える場合があります。デバイスのリセットが書込みまたは消去動作中に発生した場合、その内容は保証されません。

Flash メモリの操作中に発生するリセット

Flash メモリの操作中（プログラミング、消去、オプション変更）にリセットが発生すると、メモリの内容は保証されません。一貫性を保つために操作を再スタートすることが必須です。ステータスレジスタ FLASH_OPSR はリセットによって中断した操作の情報を提供します。

FLASH_OPSR.CODE_OP は操作のオペコードを与えます。FLASH_OPSR の使い方と、どの操作が必要かを表 41 に示します。

表 41. FLASH_OPSR コンテンツへの推奨対応

CODE_OP	中断した操作	OTP_OP	SYSF_OP	BK_OP	DATA_OP	Flash領域	最小 ADDR_OP	最大 ADDR_OP	推奨対応
0x000	リセットの際に操作は行っていなかった。	0	0	0	0	_(1)	-	-	特別な対応は不要。
0x001	書き込み操作	0	0	0/1	0	ユーザ Flash 領域	0x0000	0xFFFF	セクタを消去して再書き込み
		0	1	0/1	0	システム Flash 領域	0x0000	0x0FFF	
		1	0	0	0	OTP	0x0600	0x07FF	
		1	0	1	0	OBKeys ⁽²⁾	0x0000	0x03FF	
		0	0	0/1	1	データ領域 ⁽³⁾	0xF000	0xFFFF	
0x010	OBK 代替セクタ消去	0	0	1	0	OBKeys	0	0	代替セクタ消去を再開
0x011	セクタ消去 ⁽⁴⁾	0	0	0/1	0	ユーザ Flash 領域	0x0000	0xFFFF	セクタ消去再開
		0	1	0/1	0	システム Flash 領域	0x0000	0x0FFF	
0x100	バンク消去	0	0	0/1	0	ユーザ Flash 領域	-	-	バンク消去再開
0x101	全体消去	0	0	0	0		-	-	全体消去再開
0x110	オプション変更	0	0	0	0	ユーザ設定	-	-	新たなオプション変更の試行
0x111	OBK スワップ・セクタ	0	0	1	0	OBKeys	-	-	代替セクタを消去して新しい OBKeys を再度プログラムし、スワップを再開します。

1. ダッシュは「関係ない」ことを表します。

2. 現在の OBK セクタによる。

3. OB の EDATA 設定による

4. 示したアドレスはセクタ開始に合わせており、データ領域セクタは対応するユーザ Flash メモリ・セクタへの消去リクエストにより消去されます。

クロック管理

メモリは、マイクロコントローラのシステム・クロック (sys_ck)、ここでは AHB インタフェース・クロックを使用します。

7.4 FLASH のオプション・バイト

7.4.1 オプション・バイトについて

メモリには一連の不揮発性オプション・バイトが含まれています。これはパワーオン・リセット時にロードされ、設定レジスタを通じてのみ読み出しと変更が可能です。このセクションでは以下の内容について説明します。

- オプション・バイトがロードされるタイミング
- アプリケーション・ソフトウェアによる変更方法
- 初期値を含む、オプション・バイトの詳細リスト（オプション・バイトが最初に変更される前のユーザデフォルト設定）

7.4.2 オプション・バイトのローディング

内蔵 Flash メモリにオプション・バイトをロードする方法は複数存在します。

- **パワーオンウェイクアップ**
デバイスに電源が初めて供給されると、内蔵 Flash メモリはすべてのオプション・バイトを自動的にロードします。オプション・バイトのローディング・シーケンス中、デバイスはリセット状態を維持し、内蔵 Flash メモリにはアクセスできません。
- **システム STANDBY からのウェイクアップ**
内蔵 Flash メモリを含むコア電力ドメインが STANDBY モードから RUN モードに切り替えられると、内蔵 Flash メモリはパワーオン・シーケンスのように動作します。ローディングの間はパワーオン・シーケンスとは異なり、デバイスはリセットにはなりません。
- **アプリケーションによる専用オプション・バイトの再ロード**
ユーザ・アプリケーションがメモリ・レジスタを経由してオプション・バイトの内容の変更に成功すると、不揮発性のオプション・バイトがプログラムされ、メモリはすべてのオプション・バイトを自動的に再ロードしてオプション・レジスタを更新します。

注： オプション・バイトの読み出しシーケンスは、エラー訂正コードで保護されています。エラー時にはオプション・バイトが初期値（ユーザ・デフォルト設定）とは異なるデフォルト値でロードされ（[セクション 7.4.3](#) 参照）、より制限が多くなります。

7.4.3 オプション・バイトの変更

ユーザ・オプション・バイトの変更

ユーザ・オプション・バイトの変更操作は、不揮発性のオプション・バイト領域に保存された設定や保護設定を変更するために使用できます。

ユーザ・オプション・バイトの変更を行う時には、いくつかの規則に従わなくてはなりません。その概要を[セクション 7.4.7](#)で説明します。これらの規則を守らないと通常はエラーになります（[セクション 7.9.12](#) 参照）。

内蔵 Flash メモリには2セットのオプション・バイト・レジスタがあります。

- 1つ目のレジスタセットには、オプション・バイトの現在値が格納されます。これらの名前には `_CUR` という拡張子がついています。すべての「`_CUR`」レジスタは読み出し専用です。その値は、パワーオンリセット、システム STANDBY からのウェイクアップ、オプション・バイトの変更操作後に不揮発性メモリから自動的にロードされます。
- 2つ目のレジスタセットでは、オプション・バイトを変更することができます。これらの名前には `_PRG` という拡張子が含まれます。すべての「`_PRG`」レジスタは読み出し／書込みモードでアクセスすることができます。

FLASH_OPTCR レジスタの OPTLOCK ビットがセットされると、FLASH_XXX_PRG レジスタの変更は行えません。

OPTSTRT ビットが 1 にセットされると、メモリはプログラミング・シーケンス (PGSERR) と [セクション 7.4.7](#) で説明している条件 (OPTCHANGEERR) を確認します。エラーが検出されると (PGSERR/OPTCHANGEERR) 内蔵 Flash メモリは、その不揮発性メモリのオプション・バイトの変更を開始し、オプション・バイト・レジスタを _CUR 拡張子で更新します。

[セクション 7.4.7](#)、[セクション 7.9.12](#)、[セクション 7.9.5](#)、あるいは[セクション 7.9.4](#) のどれかで説明している条件の 1 つが守られていない場合、メモリはオプション・バイトの変更操作を中止します。この場合、FLASH_XXX_PRG レジスタは現在のオプション値で上書きされません。ユーザアプリケーションは、設定の何が誤っていたのかを確認可能です。

オプション・バイト変更のアンロック

リセット後、OPTLOCK ビットは 1 にセットされ、FLASH_OPTCR はロックされます。その結果として、アプリケーションソフトウェアは、オプション・バイトの変更を試みる前にオプション設定レジスタをアンロックする必要があります。FLASH_OPTCR アンロックシーケンスは、[セクション 7.6.7](#) に記載されています。

オプション・バイトの変更シーケンス

ユーザ・オプション・バイトを変更するには、次のシーケンスに従います。

1. FLASH メモリ操作が何も行われていないことを FLASH_NS/SECSR レジスタの BSY ビットで確認し、書込みバッファが空になっていることを FLASH_NS/SECSR レジスタの WBNE ビットで確認します。
2. FLASH_NS/SECSR レジスタのデータバッファが空 (DBNE = 0) であることを確認します。
3. 以前の操作によるエラー・フラグをすべてクリアします。
4. レジスタがまだアンロックされていない場合、[セクション 7.6.7](#) を参照して FLASH_OPTCR レジスタをアンロックします。
5. 対応するオプションレジスタ (FLASH_XXX_PRG) に希望する新しいオプション・バイトの値を書き込みます。
6. FLASH_OPTCR レジスタでオプション・バイト・スタート変更 OPTSTRT ビットを 1 にセットします。
7. FLASH_NS/SECSR レジスタの BSY ビットがクリアされるまで待ちます。
8. OPTSTRT ビットはシーケンスの最後 (またはエラーの場合) に自動的に消去されます。
9. デバイスをリセットします。このステップは常に推奨されており、次のオプション・バイトのどれか 1 つが影響を受ける場合には必須です。
 - a) SECBOOTADDR
 - b) NSBOOTADDR
 - c) TZEN
 - d) BOOT_UBE

注： オプション・バイトの変更中にリセットまたはパワーダウンが発生した場合は、元のオプション・バイト値が保持されます。これをプログラムするには、新しいオプション・バイトの変更シーケンスが必要です。

オプション・バイトの概要

表 42 は、メモリレジスタを経由して管理されるすべてのユーザ・オプション・バイトと、オプション・バイトが最初に変更される前の初期値（ユーザ・デフォルト設定）のリストです。

表 42. オプション・バイトの構成

レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
FLASH_OPTSR	SWAP_BANK	Res.	BOOT_UBE								IWDG_STDBY	IWDG_STOP	Res.	Res.	IO_VDDIO2_HSLV	IO_VDD_HSLV	PRODUCT_STATE								NRST_STDBY	NRST_STOP	Res.	WWDG_SW	IWDG_SW	BORH_EN	BOR_LEV			
	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	
FLASH_OPTSR2	TZEN								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM2_ECC	SRAM3_ECC	BKPRAM_ECC	SRAM2_RST	SRAM13_RST	Res.	Res.	
	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	
Flash_BOOTR	SECBOOTADD[23:8]															SECBOOTADD[7:0]							SECBOOT_LOCK											
	0x0C00															0x00							0xC3											
Flash_NSBOOTR	NSBOOTADD[23:8]															NSBOOTADD[7:0]							NSBOOT_LOCK											
	0x0800															0x00							0xC3											
Flash_SECWM1R	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_END						Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_STRT							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
Flash_SECWM2R	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_END						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_STRT						
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
Flash_WRPNG1R	WRPSG[127:124]	WRPSG[123:120]													WRPSG[71:68]	WRPSG[67:64]	WRPSG[63:60]	WRPSG[59:56]													WRPSG[7:4]	WRPSG[3:0]
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Flash_WRPNG2R	WRPSG[127:124]	WRPSG[123:120]													WRPSG[71:68]	WRPSG[67:64]	WRPSG[63:60]	WRPSG[59:56]													WRPSG[7:4]	WRPSG[3:0]
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Flash_OTPBLR	LOCKBL																																	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 42. オプション・バイトの構成 (続き)

レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Flash_ EDATA1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA_EN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECTOR_START_1			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Flash_ EDATA2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA_EN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECTOR_START_2		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Flash_ NSEPOCHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NS_EPOCH																								
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Flash_ SECEPOCHR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC_EPOCH																								
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Flash_ HDP1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_END						HDP1_STRT										
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Flash_ HDP2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_END						HDP2_STRT										
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

7.4.4 ユーザ・オプション・バイトとシステムオプション・バイトの説明

アプリケーションが使用可能な汎用オプション・バイトのリストを次に示します。

- ウォッチドッグ管理
 - IWDG_STOP : 1 である場合、独立型ウォッチドッグ IWDG (WDGLS_CD とも言います) カウンタが STOP モードで有効 (0 であればカウンタ停止またはフリーズ)
 - IWDG_STDBY : 1 である場合、独立型ウォッチドッグ IWDG (WDGLS_CD とも言います) カウンタが STANDBY モードで有効 (0 であればカウンタ停止またはフリーズ)
 - IWDG_SW : ハードウェア (0) またはソフトウェア (1) の IWDG (WDGLS_CD とも言います) ウォッチドッグ制御選択

注 : ハードウェアウォッチドッグ「制御選択」機能が有効 (0 にセット) である場合、電源投入時に自動的にウォッチドッグが有効となりますので、カウンタ終了に到達する前にウォッチドッグキーレジスタが書き込まれるか、ダウンカウンタが再ロードされない限り、リセットが発生します。

IWDG_STOP および IWDG_STDBY のオプションの設定に応じて、IWDG は、デバイスがそれぞれ STOP モードまたは STANDBY モードである場合に、カウンタを継続可能であったり (1) そうではなかったり (0) します。STOP または STANDBY モード中に IWDG を動作したままにすると、このモードからデバイスをウェイクアップできます。

- リセット管理
 - BOR_LEV: リセットをアクティブにしたリセットを解除したりする供給レベル閾値を示すブラウンアウト・レベル・オプション
 - BORH_EN : BOR レベル高 (ハイ) 有効化
 - NRST_STDBY : 0 にクリアされている場合 STANDBY モードへの遷移時にリセットを生成

注： NRST_STOP : 0 にクリアされている場合、STOP モードへの遷移時にリセットを生成
NRST_STDBY (NRST_STOP) が 0 にクリアされている場合、STANDBY (STOP) モード遷移シーケンスが正常に実行されるたびに、STANDBY (STOP) モードに遷移する代わりにデバイスがリセットされます。

- デバイス・オプション
 - IO_VDDIO2_HSLV : 1 にセットされている場合、VDDIO2 電源レールに対して 2.7 V 未満のパッドの設定が有効
 - IO_VDD_HSLV : 1 にセットされている場合、VDD 電源レールに対して 2.7V 未満のパッドの設定が有効
 - USBPD_DIS : USB PD を無効にするビット

デバイス納入時、汎用オプション・バイトにプログラムされている値は次のとおりです。

- ウォッチドッグ管理
 - IWDG (WDGLS_CD とも言います) は STANDBY モードと STOP モードで有効 : 0x1
 - IWDG (WDGLS_CD とも言います) は電源投入時に自動的に有効化されません : 0x1
- リセット管理
 - BOR : ブラウンアウト・レベル・オプション (リセット・レベル = 2.1 V) 0x0. デバイスが Standby, Stop 低パワー・モード (値 : 0x1) に入った場合、リセットは生成されません。
- 低電圧での I/O スピードの最適化は無効で、全電圧範囲で動作するデバイス (IO_VDDIO2_HSLV = 0 および IO_VDD_HSLV = 0)

詳細については、[セクション 7.11](#) を参照してください。

7.4.5 データ保護オプション・バイトの説明

以下は、データ保護の強化に使用可能なオプション・バイトのリストです。

- PRODUCT_STATE[7:0] : 製品のライフ・サイクル・ステート (詳細は[セクション 7.6.11](#) を参照してください)。
- WRPSGn1/2 : それぞれ、バンク 1、バンク 2 の 4 つの連続セクタの対応するグループの書き込み保護オプション。アクティブローです。詳細については、[セクション 7.6.8](#) を参照してください。
 - Bit N : セクタ 4 x N から 4 x N + 3 まだが組み込まれているグループ
- SECWMx : TrustZone® セキュアのためのウォーターマーク領域の定義 (詳細は[セクション 7.6.1](#) を参照してください)。
 - SECWM1_STRT (SECWM1_END) はバンク 1 のセキュア・アクセス・オンリー・ゾーンの最初 (最後) のセクタを含んでいます。
 - SECWM2_STRT (SECWM1_END) はバンク 2 のセキュア・アクセス・オンリー・ゾーンの最初 (最後) のセクタを含んでいます。
- TZEN : [セクション 7.6](#) に記載されているように、この不揮発性オプションはセキュア・アクセス・モードを有効にするためにアプリケーションが使用できます。セキュリティのために TZEN は 8 bit の冗長性を持って保存されます。TZEN OB 値はプログラミング後、即座にセットされるものの、実際に TZ 機能を使用し、SBS で TZ_STATE をセットするにはリセットが必要です。TZEN が有効になると、セキュア・ウォーターマーク設定とセキュア・ブート・アドレスがデフォルトにリセットされ、前の値が正しくなかったときにデッドロックになることを回避する場合があります。
- HDPx : セキュアな非表示保護 - ユーザ Flash メモリの HDPL 専用領域制御。

工場でプログラムされるデータ保護オプション・バイトの値は次のとおりです。

- 製品状態は販売の種類により異なります。
- Flash・バンク消去操作はウォーターマークのセキュア・データ領域に影響しません。
- セキュアなウォーターマーク領域保護は無効（開始アドレスが終了アドレスより大きい）。
- 書込み保護は無効（すべてのオプション・バイト・ビットが 1 にセット）
- TrustZone セキュアアクセスモードは無効（TZEN オプション・バイト値 = 0xC3）

詳細については、[セクション 7.11](#) を参照してください。

7.4.6 ブートアドレスオプション・バイトの説明

以下は、アプリケーションに適したブートアドレスの設定に使用可能なオプション・バイトのリストです。

- PRODUCT_STATE
- BOOT_UBE：暗号化が有効になっているデバイスでのみ使用可能です。
ST-iRoT またはユーザ Flash メモリのいずれかをデフォルトのブート・アドレスとして選択。また、次の段階として RSS(SFI) が ブートローダ と ST-iRoT のどちらかを選ぶ際にも使用されます。
- NSBOOTADD：TZ が無効になっている場合、デフォルトのブート・アドレスを選択します。
- SECBOOTADD：TZ が有効になっている場合、デフォルトのブート・アドレスを選択します。
- (NS/SEC)BOOT_LOCK：ブート設定をさらなる変更から保護します。
- SWAP_BANK：バンク・スワップのオプション。1 に設定するとブート後にユーザ Flash メモリのバンクをスワップします（[セクション 7.3.11](#) 参照）。TZEN の状態（SEC/NS）に対応する BOOT_LOCK が有効な場合、SWAP_BANK の値は固定で読取り専用です。

STMicroelectronics がデバイスを納入する際には、PRODUCT_STATE は Open で、BOOT_LOCK はセットされておらず（0xC3）、BOOT_UBE はユーザ Flash メモリに設定されています（0xB4）。アドレスは SECBOOTADD = 0x0C00 0000、NSBOOTADD = 0x0800 0000 です。

詳細については、[セクション 7.11](#) を参照してください。

7.4.7 オプション・バイト変更の特定の規則

OPTLOCK ビットとレジスタ・アクセス規則に加え、セキュリティが重要な特定のオプション・バイト・フィールドには、その他の保護手段も存在します。

他のオプション・バイトも同時に変更可能ですが、保護のために両者が互いに依存している場合、両方の状態を確認します。例えば、PRODUCT_STATE と TZEN を同時に変更しようとする場合、得られる状態は規則に沿ったものでなくてはなりません。このセクションに列記した僅かな例外を除いて、規則に則らない場合には OPTCHANGEERR フラグが立ちます（さらなる詳細は[セクション 7.9.12](#) 参照）。

表 43. OB 変更の特定の規則

オプション・バイト	HDPL	TZ セキュア	値	製品状態
PRODUCT_STATE	0.1 ⁽¹⁾	TZ が有効な場合のみの、いくつかの値。	可能性のある一連の遷移	可能性のある一連の遷移（ 表 37 ）
HDP	0.1 ⁽²⁾	⁽³⁾	-	表 44 を参照してください。
TZEN	-	-	-	回帰、オープンまたはプロビジョニング
EPOCH	0	-	PRG > CUR	回帰
EPOCH_NS	0 ⁽²⁾	-	PRG > CUR	NS-回帰、回帰

表 43. OB 変更の特定の規則 (続き)

オプション・バイト	HDPL	TZ セキュア	値	製品状態
SECWM	-	セキュアモードのみ ⁽²⁾	-	表 44 を参照してください。
BOOT_UBE	-	-	-	オープンまたはプロビジョニング、 SECBOOT_LOCK 無効
SECBOOTADD	-	セキュアモードのみ ⁽²⁾	-	
NSBOOTADD	-	-	-	オープンまたはプロビジョニング、 NSBOOT_LOCK 無効
LOCKBL	-	-	片切スイッチ ⁽²⁾	-
SWAP_BANK	-	TZEN に対応する BOOT_LOCK が設定され ている場合は固定。	-	-
SECBOOT_LOCK	-	セキュアモードのみ ⁽²⁾	-	オープン、回帰、またはアンロックのため にプロビジョニング
NSBOOT_LOCK	-	-	-	

1. 殆どの遷移は HDPL に関係なく可能で、特定の HDPL が必要なのは僅かです。下の説明をご覧ください。
2. これの違反では OPTCHANGEERR フラグは立ちません。
3. ダッシュは、この側には制限がないことを示します。

クローズドの PRODUCT_STATE の進展も、他の制限がすべて満足されている場合には、いくつかの OB は変更が可能です。概要を表 44 に示します。

表 44. 閉じた製品で変更可能な OB

PRODUCT_STATE	変更可能な OB
TZ クローズド	SWAP_BANK, LOCKBL, PRODUCT_STATE, SECWM, HDP
クローズド	SWAP_BANK, LOCKBL, PRODUCT_STATE
ロック	SWAP_BANK, LOCKBL

次の OB を更新するためには、それぞれの規則に従わなくてはなりません。

- **PRODUCT_STATE**

PRODUCT_STATE の遷移は 2 種類の遷移があるステート・マシンに従います。製品のロック・ダウンは制限なく可能です。製品のオープンには必ずデバッグ・インタフェースを用い、デジタル署名の検証に従う必要があります。よりセキュリティが低い状態への回帰は、保護された内容の消去につながります。より詳細については、[セクション 7.6.11](#) を参照してください。

まとめ：選択された変更は HDPL1 と HDPL0 の回帰でのみ可能です。状態の中には TZEN が有効になっている場合のみにアクセスできるものがあります。

- **HDP**

HDPL0 と HDPL1 でのみ変更が可能。Flash メモリの高サイクル・データ領域とオーバーラップするよう設定してはなりません。

- **TrustZone アクセス・モード (TZEN)**

オープン、回帰、プロビジョニングでのみ変更可能。

- **SEC_EPOCH と NS_EPOCH**

プログラムされた値は現在の値より小さくなくてはなりません。インクリメントは次の 2 つの製品状態で行われます：回帰および NS 回帰。

- **セキュアなウォーターマーク領域 (SECWM1/2_STRT および SECWM1/2_END)**
セキュア・モード (TZ_state = 0xB4) でのみ変更可能。TZEN が有効になると自動的にデフォルト値にリセットされます。Flash メモリの高サイクル・データ領域とオーバーラップするよう設定してはなりません。
- **BOOT_UBE**
暗号化が有効になっているデバイスでのみ使用可能です。
オープンやプロビジョニングのようなデバッグに開かれた製品状態でのみ変更可能。変更するには、SECBOOT_LOCK を無効にしなくてはなりません。
- **SECBOOTADD**
オープンおよびプロビジョニングでのみ変更可能です。SECBOOT_LOCK でロックされています。TZEN が有効になると自動的にデフォルト値にリセットされます。
- **NSBOOTADD**
オープンおよびプロビジョニングでのみ変更可能です。NSBOOT_LOCK でロックされています。
- **LOCKBL**
1 方向にのみ自由に変更が可能です。永続的に不可逆なスイッチ。
- **SWAP_BANK**
TZEN と SECBOOT_LOCK の両方が 0xB4 (セット) または TZEN = 0xC3 (無効) で NSBOOT_LOCK が有効(0xB4)な場合には変更不可。
- **SECBOOT_LOCK**
ロック方向にのみ自由に変更が可能です。オープン、プロビジョニング、回帰でアンロックが可能です。
- **NSBOOT_LOCK**
ロック方向にのみ自由に変更が可能です。オープン、プロビジョニング、回帰でアンロックが可能です。

注： 上のすべてのユーザ・オプション・バイト：OBL 中にダブル ECC エラーが発生すると、デフォルト値がロードされ、Tamper が信号を発します。

7.5 オプション・バイトのキー・マネージメント

このセクションでは、SBS により設定された SAES と OBK-HDPL 値に紐づけられたキー・ストレージについて説明します (セクション 14.3.7: SBS ハードウェア・セキュアストレージ制御参照)。OBK は 128 bit で格納され、9 bit ECC (SEC/DED) で保護されます。これらは 2 つのセクタに交互に書き込まれます。

7.5.1 OBK のローディング

OBK はメモリにマッピングされ、0x0FFD 0000 から始まるアドレスでメイン AHB バス (C-Bus) からアクセスが可能です。9 bit ECC は、各 128 bit データの Flash ワードに対応しています。

注： 現在および代替セクタは、最後のアドレスを除き、読取りアクセスが可能です。読取りアクセスの場合、エラーは報告されませんが、読取りデータは常に 0x0 となります (セクション 7.6.12 参照)。

7.5.2 HDPL レベルごとの OBK アクセス

表 45 にオプション・バイト・キー領域を示します。このキー・ストレージは特定のキー専用ではなく、使用方法はアプリケーションにより定義されます。

オプション・バイト・キーは、(SBS に設定されている) OBK-HDPL が (下に示した) ストレージ・オフセットに関連した HDPL に合致する場合にのみアクセス可能です (セクション 14.3.7 : SBS ハードウェア・セキュアストレージ制御参照)。そうでない場合は、OBKERR エラーが立ちます。

表 45. オプション・バイトのキー領域

OBK-HDPL	アドレス・オフセットの 開始	アドレス・オフセットの 終了	コメント
0	0x0000	0x00FF	HDPL0 キー・ストレージに予約済み
1	0x0100	0x08FF	OEM iRoT キー
2	0x0900	0x0BFF	uRoT、OS あるいはセキュア・アプリケーション
3	0x0C00	0x17FF	HDPL 3 セキュア・キー
3	0x1800	0x1FEF	HDPL 3 非セキュア・キー

7.5.3 OBK プログラミング・シーケンス

OBK に書き込むシーケンスは、「代替セクタ」(ALT_SECT) ビットが代替 OBK セクタに書き込むように設定されていなくてはならないことを除いて、メインの Flash メモリへの書込みと同じです。この ALT_SECT ビットは、書込みアクセスを受け取るたびに確認されます。もし、書込みバッファに格納されているものと異なる場合は、エラーが立ちます。

注： 代替 OBK セクタに書き込む前に、ユーザはキー空間全体 (128、256、あるいは 512 bit) が空いていることを、代替 OBK セクタを読み出すことで確認する必要があります。

代替セクタが空いていない場合は、専用の消去命令 ALT_SECT_ERASE を使用してユーザがセクタを消去しなくてはなりません。

セキュア・プログラミング・メソッド

TrustZone が有効な場合 (TZ_STATE = 0xB4)、セキュア・レジスタを使用して以下のステップを実行します。

1. ALT_SECT ビットをプログラムします。
2. 保護メカニズムがプログラミングを妨げていないことを確認します (TZ、PRIV と OBK-HDPL)。
3. FLASH メモリ操作が何も行われていないことを FLASH_SECSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_SECSR レジスタの WBNE ビットで確認します。
4. 以前の操作によるセキュア・エラー・フラグをすべて確認し、クリアします。
5. レジスタがまだアンロックされていない場合に限り、セクション 7.6.7 を参照して FLASH_SECCR レジスタをアンロックします。
6. FLASH_SECCR レジスタの SECPG ビットをセットすることによって書込み操作を有効にします。
7. 16 バイトの整列アドレスで始まる 16 バイトデータに対応する Flash ワードを 1 つ書き込みます。

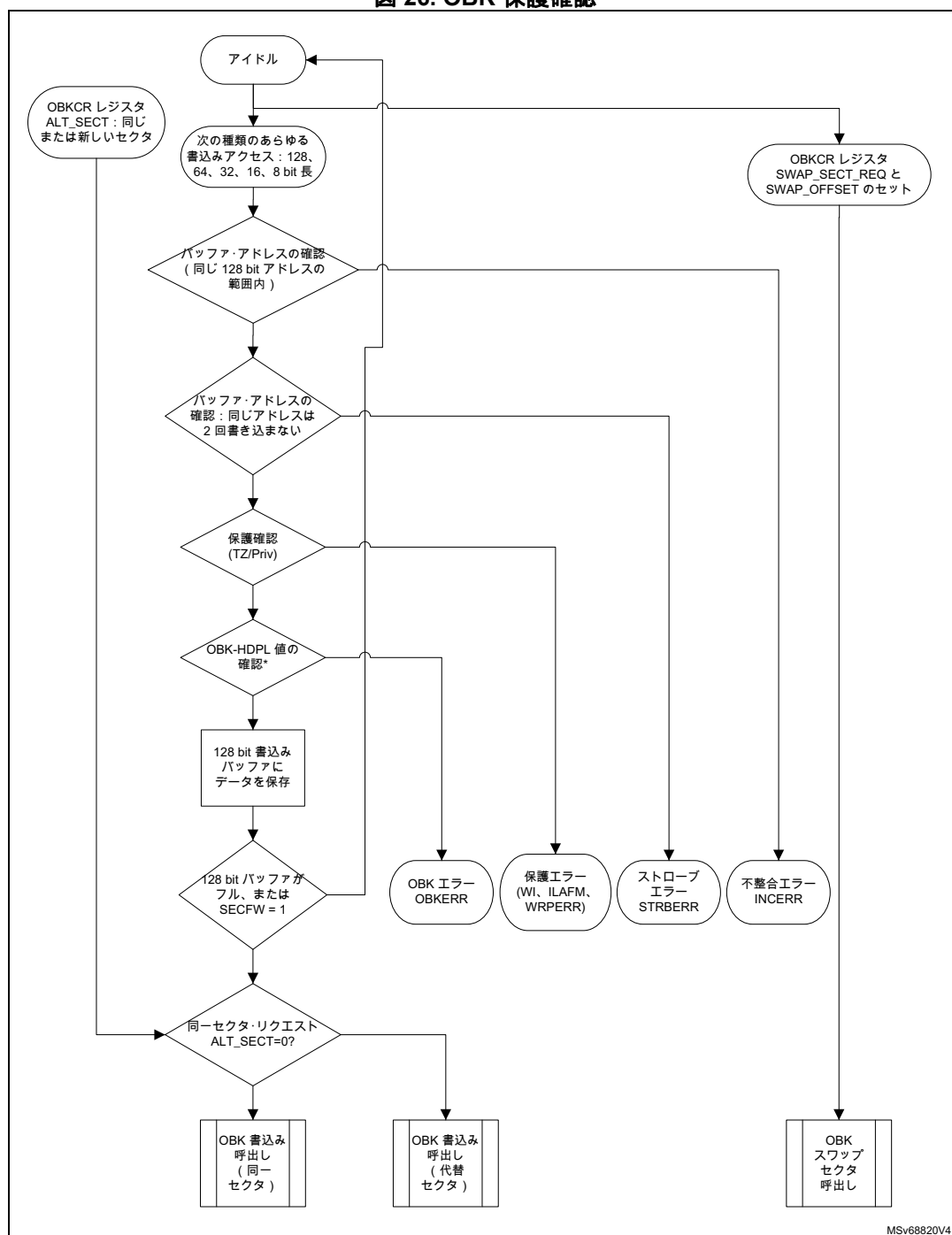
注： WBNE フラグは 128 bit 書込みバッファが新しいデータを待っていることを示します。

注： 最初の書込みから Flash 書込み操作の終了までの間は、消去リクエスト、オプション変更リクエスト、OBK スワップセクタ、あるいは OBK 代替セクタの消去ができません。

8. 対応する FLASH_SECCR レジスタの BSY ビットがクリアされるのを待ちます。
9. そのバンクで、プログラミング・リクエストがこれ以上ない場合、FLASH_SECCR レジスタの PG ビットをクリアします。

ステップ 7 が順に (バイト単位) に実行される場合、書込みバッファが部分的に満たされている可能性があります。この場合、アプリケーション・ソフトウェアは、FLASH_SECCR レジスタの FW ビットを使用して書込みバッファに格納されているものを強制書込みすることができます。この特殊な例では、書き込まれないビットは自動的に 1 にセットされます。書込みバッファのどのビットも 0 にクリアされない場合、FW ビットは影響しません。

図 26. OBK 保護確認



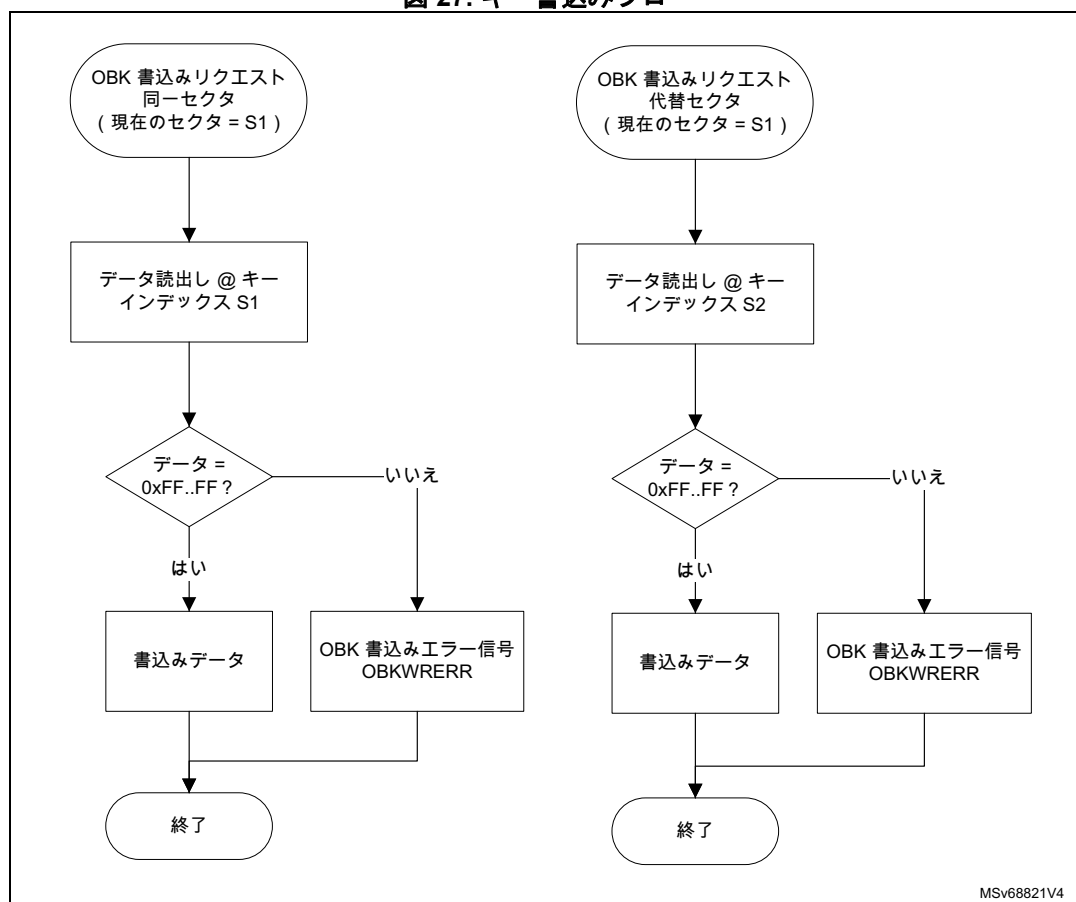
MSv68820V4

7.5.4 OBK プログラミング有限ステート・マシン

OBK の書き込みバッファがフィルされている場合、OBK プログラミングが ALT_SECT ビットに応じて、現在の OBK セクタまたは代替 OBK セクタのいずれかで自動的に立ち上がります。

アドレスが未使用ではない場合、OBKWRERR が立ちます。

図 27. キー書き込みフロー



MSv68821V4

7.5.5 OBK スワップ・セクタ

ユーザは、NS/SECOBKCFGR レジスタの SWAP_SECT_REQ ビットを設定することで、オプション・バイト・キーのスワップをリクエストすることが可能です。スワップされたキーの数は NS/SECOBKCFGR レジスタの SWAP_OFFSET で定義されます。

スワップは OBK-HDPL 値で制限されます。次の SWAP_OFFSET 値が許容されています。

OBK_HDPL0_OFFSET = 16

OBK_HDPL1_OFFSET = 144

OBK_HDPL2_OFFSET = 192

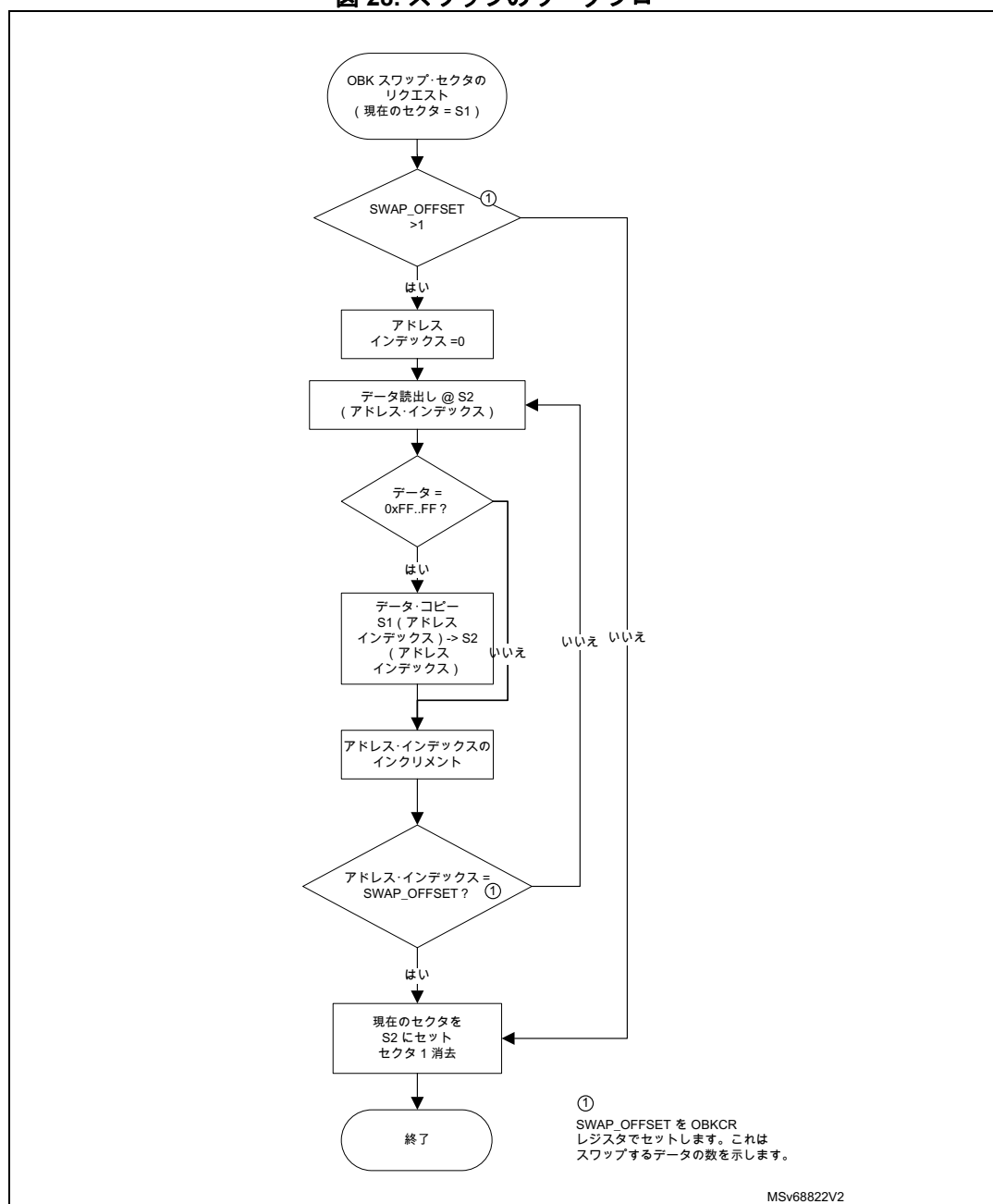
OBK_HDPL3SEC_OFFSET = 384

OBK_HDPL3NS_OFFSET = 511

プログラミングシーケンスは、次のとおりです。

1. FLASH メモリ操作が何も行われていないことを FLASH_SEC/NSSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_SEC/NSSR レジスタの WBNE ビットで確認します。
2. 以前の操作によるセキュア／非セキュア・エラー・フラグをすべて確認し、クリアします。
3. FLASH_NS/SECOBKKEYR により既にアンロックされている場合を除き、NS/SECOBKCFGR レジスタをアンロックします。
4. NS/SECOBKCFGR の SWAP_OFFSET フィールドを利用してスワップするキーの数を定義します。SWAP_OFFSET は $OBK_HDPL<N-1>_OFFSET$ 以上でなくてはなりません。
5. NS/SECOBKCFGR の SWAP_SECT_REQ ビットを設定します。
6. BSY ビットがクリアされるのを待ちます。エラーの場合は OBKERR フラグがセットされます。
7. SWAP_SECT_REQ ビットは操作の最後またはエラーの場合に自動的にクリアされます。

図 28. スワップのワークフロー



OBK SWAP 操作が進行中は、BSY フラグがセットされ他の操作（書込み、消去、ユーザ・オプション変更）を並行して開始することはできません。

7.5.6 OBK 代替セクタ消去

ユーザは、FLASH_NS/SECOBKCFGR レジスタの ALT_SECT_ERASE ビットを設定して、代替 OBK セクタの消去をリクエストすることができます。

プログラミング・シーケンスは次のとおりでなくてはなりません。

1. FLASH メモリ操作が何も行われていないことを FLASH_SEC/NSSR レジスタの BSY ビットと DBNE ビットで確認し、書込みバッファが空になっていることを FLASH_SEC/NSSR レジスタの WBNE ビットで確認します。
2. 以前の操作によるセキュア／非セキュア・エラー・フラグをすべて確認し、クリアします。
3. NS/SECOBKCFGR の ALT_SECT_ERASE ビットを設定します。
4. BSY ビットがクリアされるのを待ちます。
5. ALT_SECT_ERASE ビットは操作の最後またはエラーの場合に自動的にクリアされます。

7.6 FLASH のセキュリティと保護

Flash メモリには機密情報が格納されるため、秘密領域の読出し、変更できないセクタの不正プログラミング、Flash メモリの不正消去などの好ましくない操作から保護することが重要です。

そのため FLASH は次の保護メカニズムを実装しています。

- TrustZone によるウォーターマークとブロック・セキュリティの保護
- 一時隔離保護 (HDP)
- 設定保護
- ユーザ Flash メモリ書込み保護
- デバイスの不揮発性メモリのセキュリティ・ライフ・サイクルとアプリケーション・ブート・ステート・マネージメント
- OTP ロッキング

マイクロコントローラは、TrustZone を有効 (TZ_STATE = 0xB4) または 無効 (TZ_STATE = 0xC3) にすることができます。

TrustZone が有効なときに、Flash メモリは次の 4 つの基本モードでアクセスが可能です。

- 非セキュア／非特権
- 非セキュア／特権
- セキュア／特権
- セキュア／非特権

TrustZone が無効になっているときは、次の 2 つのモードのみが可能です。

- 非セキュア／非特権
- 非セキュア／特権

Flash メモリ・インタフェースは、次の順番でアクセス制限を評価します。

1. TrustZone セキュリティ
2. 書込み保護 (書込みアクセスに対して)
3. HDP
4. 特権

7.6.1 TrustZone のセキュリティ保護

TrustZone グローバル・セキュリティ・システムは、FLASH_OPTSR2_PRG レジスタの TZEN オプションを設定することによって有効になります。ただし、TZ の実際の状態は SBS と TZ_STATE の設定で決定されます（[セクション 14.3.5 : SBS ブート制御](#)参照）。

TrustZone が有効（TZ_state = 0xB4）な場合、次のような追加のセキュリティ機能が使用可能です。

- セキュア領域を定義するウォーターマークベースのセキュアなユーザ・オプション・バイト
- リセット後、動作中に設定可能なブロックベースのセキュア／非セキュア領域これは揮発性のセキュア領域です。
- **closed_secure**に関連した追加の製品状態は、外部アクセス（デバッグまたはブートローダ）に対してセキュア・ドメインの保護を提供します。
- 関連の設定ビットを使用して、セキュアまたは非セキュアモードで実行可能な消去／プログラム操作TrustZone が無効な場合（TZ_state = 0xC3）、上記の機能は無効で、すべてのセキュアレジスタは RAZ/WI です。

TrustZone セキュリティの有効化

TrustZone が有効な場合（TZ_STATE = 0xB4）、SECBOOTADD とセキュア・ウォーターマーク領域はセキュアなデフォルト値に設定されます。設定が破損した場合もデフォルト設定を使用します。このデフォルトの状態では、ユーザ Flash メモリは下の表に示すとおりセキュアです。

表 46. デフォルトのセキュア・ウォーターマーク

設定エラーの場合のセキュア・ウォーターマーク	セキュリティ属性
SECWMx_STRT = 0 SECWMx_END = 0x7F	すべての Flash メモリはセキュア

不正アクセスの生成

セキュアな Flash メモリ領域への非セキュア・アクセスは RAZ/WI で、不正アクセス・イベントを生成します。TZIC_IER2 レジスタで FlashIE 不正アクセス割込みを有効にすると、不正アクセス割込みが生成されます。

Flash のセキュアレジスタへの非セキュア・アクセスは、不正アクセス・イベントを生成します。TZIC_IER2 レジスタで Flash_REGIE 不正アクセス割込みを有効にすると、不正アクセス割込みが生成されます。

表 47. Flash メモリ TZ 保護のまとめ

TZ 保護 (TZ_STATE = 0xB4)		メイン Flash メモリ	
		非セキュア・セクタ	セキュア・セクタ
セキュア	フェッチ	バス・エラー	OK
	読出し	RAZ、ILAFM	
	書込み	WI、WRPERR、ILAFM	WRP ではない場合 : OK その他の場合 : WI、WRPERR
	消去		
非セキュア	フェッチ	OK	バス・エラー
	読出し		RAZ、ILAFM
	書込み	WRP ではない場合 : OK その他の場合 : WI、WRPERR	WI、WRPERR、ILAFM
	消去		

表 48. TZ 保護とバンクまたは全体消去のまとめ

TZ 保護 (TZ_STATE = 0xB4)	メイン Flash メモリ		
	非セキュア Flash メモリ	セキュア Flash メモリ	セキュアと非セキュアの混合
MES	WI, WRPERR, ILAFM	OK	WI, WRPERR, ILAFM
MENS	OK	WI, WRPERR, ILAFM	WI, WRPERR, ILAFM

TrustZone セキュリティの無効化

OB (TZEN) での TZ 設定の反転は、製品状態が ST-RoT-Ready またはプロビジョニングのときに可能です。他の状態から TZEN を無効化するには、完全な回帰 (回帰状態) が必要です。

オプション・バイト・ローディング後に TrustZone を無効化 (TZ_state = 0xC3) する場合、次のセキュリティ機能が無効化されます。

- ウォーターマークベースのセキュア領域 ([ウォーターマークベースのセキュア Flash メモリ領域の保護](#)を参照してください)
- ブロックベースのセキュア領域 ([セクション 7.6.3](#)を参照してください)
- セキュア割込み ([セクション 7.9.11](#)を参照してください)
- すべてのセキュアレジスタは RAZ/WI

ウォーターマークベースのセキュア Flash メモリ領域の保護

TrustZone セキュリティが有効 (TZ_state = 0xB4) な場合、Flash メモリの一部を、非セキュア読みおよび書き込みアクセスから保護することができます。最大 2 つの異なる不揮発性のセキュア領域はオプション・バイトで定義することができ、セキュア・アクセスでのみ読み出したり書き込んだりすることが可能です。バンクごとに 1 つの領域を、セクタの粒度で選択できます。

セキュア領域は SECWMx_STRT および SECWMx_END オプション・バイトを使って、開始セクタ・オフセットと終了セクタ・オフセットで定義されます。これらのオフセットは、セキュア・ウォーターマーク・レジスタのアドレス・レジスタ [バンク 1 の FLASH セキュリティ・ウォーターマーク \(FLASH_SECWM1R_CUR\)](#)、[バンク 2 の FLASH セキュリティ・ウォーターマーク \(FLASH_SECWM2R_CUR\)](#) で定義され、対応する FLASH_XXX_PRG レジスタで変更されます。

SECWMx_STRT と SECWMx_END のオプション・バイトはセキュア・ファームウェアでのみ変更可能です。

表 49. ウォーターマークベースセキュア領域

ウォーターマークベースのセキュアオプション・バイトの値 (x = 1, 2)	ウォーターマークベースセキュア保護領域
SECWMx_STRT > SECWMx_END	セキュア領域なし。
SECWMx_STRT = SECWMx_END	SECWMx_STRT で定義される 1 つのセクタをセキュアなウォーターマークベースの保護が行われます。
SECWMx_STRT < SECWMx_END	SECWMx_STRT と SECWMx_END の間の領域はセキュアなウォーターマークベースの保護が行われます。

注意 : メモリ領域のセキュアから非セキュアへの切替えでは、その内容は消去されません。あるセキュア領域を非セキュア属性に切り替える前に必要に応じてその領域を消去するために、セキュアユーザソフトウェアによって必要な操作を実行する必要があります。また、命令キャッシュを一掃することも推奨します。

7.6.2 非表示保護 (HDP)

HDP 領域は、Flash メモリのウォーターマークベースのセキュア領域とは独立しています。バンクごとの 1 つの不揮発性セキュア非表示保護 (HDP) 領域は、セクタの粒度で定義が可能です。非表示保護領域へのアクセスは、SBS の HDPL レベルを進めることで拒否できます。

HDPL = 1 の場合、ユーザ Flash メモリは HDP では保護されません。HDPL ≥ 2 の場合、ユーザ OB が定義する各バンクの HDP 領域はクローズドです。読み出し、書込み、フェッチ、あるいは消去は、この HDP 領域では許可されません。

HDP 領域は拡張することが可能で、拡張設定はレジスタ値のみで OB には保存されません。HDPL = 3 の場合、HDP 領域は有効のまま拡張が追加されます。ベースとなるユーザ OB 定義領域が有効でないときに拡張が追加されると、拡張領域が 1 つまたはそれ以上のセクタをカバーすることになります。これは HDPx_END セクタも保護されているためです。

HDPL レベルはシステム・リセットのみによりクリアすることができ、HDP 領域を無効化する方法はありません。

保護された HDP 領域は、開始および終了セクタを使い、セキュア・ウォーターマークと類似の方法でそのサイズを設定することで定義します。

アクセス不可領域のサイズは、レジスタ FLASH_HDPEXTR の値 HDPx_EXT を使って拡張することができます。HDPx_EXT の値は、(HDPx_END セクタを超えて) HDP 領域に追加されたセクタの数です。レジスタの揮発性の HDPx_EXT 値を増やすことは可能ですが、減らすことはできません。

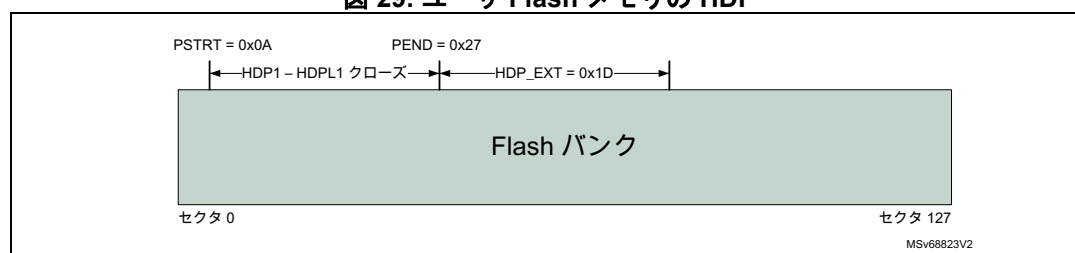
HDPx_STRT > HDPx_END の場合、拡張は HDPx_END と HDPx_END + HDPx_EXT によりマークされたセクタ間の (HDPx_EXT + 1) のサイズの領域をカバーします。

デフォルトで両方の HDPx_END はゼロに設定され、HDPx_STRT は 1 に設定されます。これは、ユーザ Flash メモリの HDP 領域のサイズは 0 で、Flash の開始アドレス (最初のセクタ) から拡張したものであることを意味しています。

たとえば、アドレス 0x0801 4000 (含む) からアドレス 0x0804 FFFF (含む) までの領域を HDP で保護するには、次のようにします。

- 物理バンク 1 では、オプション・バイト・レジスタは次のようにプログラムしなくてはなりません。
 - HDP1_STRT = 0x0A
 - HDP1_END = 0x27

図 29. ユーザ Flash メモリの HDP



- 一旦、HDPL を 2 まで増やすと、マークした領域がアクセスできなくなります。HDPL = 2 のコードは、次に拡張サイズを 29 にセットします。HDPL を 3 まで増やすと、やはり拡張した領域がアクセスできなくなります。

- その代わり、もし、
 $HDP1_END = 0$
 $HDP1_START = 1$
 $HDPx_EXT = 1$
 の場合、セクタ 0 とセクタ 1 は HDP 保護されます。
 あるいは、少し変わった例として、
 $HDP1_END = 2$
 $HDP1_START = 7$
 $HDPx_EXT = 3$
 の場合、セクタ 2 から 5 は HDP 保護されます。
 ここで、ユーザは自分のアプリケーションがセクタ 2 から 4 を単に HDP 領域として必要としているかどうかを知らなくてはならず、 $HDPx_EXT$ を 3 ではなく 2 でプログラムしなくてはなりません。
- 2 つのバンクがスワップされた場合、物理バンク 1 に定義された保護は物理バンク 1 に残り、スワッピングには影響されません。物理バンク 2 には別の保護が適用され、オプション・バイト・レジスタも次のとおりプログラムしなくてはなりません。
 - $HDP2_STRT = 0x0A$
 - $HDP2_END = 0x27$

注： バンク・スワッピング・メカニズムの詳細については、[セクション 7.6.6](#) を参照してください。

表 50. セキュア非表示保護

HDPx ウォーターマーク・オプション・バイトの値 (x = 1, 2)		非表示保護領域
HDPL ≤ 1	-	ユーザ Flash メモリのアクセス不可ではない HDP 領域
HDPL = 2	$HDPx_END < HDPx_STRT$	ユーザ Flash メモリのアクセス不可ではない HDP 領域
	$HDPx_END = HDPx_STRT$	厳密に 1 つのセクタが HDP により保護されます。
	$HDPx_END > HDPx_STRT$	$HDPx_STRT$ と $HDPx_END$ の間の領域は HDP 保護領域です。
HDPL = 3	$HDPx_END < HDPx_STRT$ および $HDPx_EXT = 0$	ユーザ Flash メモリのアクセス不可ではない HDP 領域
	$HDPx_END \geq HDPx_STRT$ または $HDPx_EXT > 0$	最小値 ($HDPx_STRT$ 、 $HDPx_END$) と ($HDPx_END + HDPx_EXT$) の間の領域は HDP 保護領域です。

表 51. HDP 保護のまとめ

HDP 保護	ユーザ Flash メモリ			
	OB HDP 領域へのアクセス		EXT HDP 領域へのアクセス	
	HDPL 2 または 3	HDPL0 または HDPL1	HDPL3	HDPL0、HDPL1、 あるいは、HDPL 2
フェッチ	バス・エラー	OK	バス・エラー	OK
読出し	RAZ		RAZ	
書込み	WI、WRPERR	WRP ではない場合 : OK	WI、WRPERR	WRP ではない場合 : OK
消去 (全体消去)		その他の場合 : WI、 WRPERR		その他の場合 : WI、 WRPERR

7.6.3 ブロックベースのセキュア Flash メモリの領域保護

すべてのセクタは、ブロックベースの設定レジスタを使用して、セキュアまたは非セキュアとして動作中に設定可能です。FLASH_SECBB1x レジスタと FLASH_SECBB2x レジスタは、それぞれバンク 1 とバンク 2 のセクタのセキュリティ属性を設定するのに使用されます。

セクタのセキュリティ属性ビット SECBByx[i] がセットされている場合、セキュリティ属性はウォーターマークベースのセキュア領域のものと同じです。セキュア・セクタはセキュア・アクセスによってのみアクセスできます。

SECBByx[i] ビットがウォーターマークベースのセキュア領域に既に含まれているセクタにセットまたはリセットされる場合、セクタはウォーターマークベースの保護セキュリティ属性をキープします。

ブロックベース・セクタのセキュリティ属性を変更するには、次の手順を推奨します。

- 関連セクタ上で進行中の Flash 操作がないことを確認します。
- セクタのセキュリティ属性ビット SECBByx[i] を変更した後で、ISB 命令を追加します。

注意： セクタのセキュアから非セキュアへの切替えでは、関連するセクタの内容は消去されません。非セキュア属性に切り替える前に、ユーザのセキュア・ソフトウェアは次の必要な操作を実行しなくてはなりません。

- セクタの内容の消去
- 命令キャッシュの無効化

注： SECBByx[i] のアクセス制御に関しては、表 52 を参照してください。

S-バス (AHB レジスタ・バス) の保護は、Flash メモリ・インタフェースそのものではなく、AHB デコーダで行われます。

表 52. セキュア設定、ブロック・ベースのレジスタ・アクセス条件

アクセス			対応するセクタ特権ステータス	SECBBxy のセクタ設定
フェッチ 読出し	セキュア／ 非セキュア	特権／ 非特権	-	バス・エラー
			-	OK
書込み	セキュア	特権	-	OK
		非特権	0	OK
			1	WI
	非セキュア	特権／ 非特権	-	WI、ILAP

7.6.4 ブロックベースの特権 Flash メモリの領域保護

あらゆるセクタはブロックベースの設定レジスタを使用して、特権または非特権として動作中に設定可能です。FLASH_PRIVBB1x レジスタと FLASH_PRIVBB2x レジスタは、それぞれバンク 1 とバンク 2 のセクタの特権属性を設定するのに使用されます。

セクタの特権属性ビット PRIVBByx[i] が設定されると、そのセクタは特権アクセスのみが可能となります。非特権セクタは特権アクセス、非特権アクセスのどちらも可能です。

ブロックベースの特権属性を変更するには、次の手順を推奨します。

- 関連セクタ上で進行中の Flash 操作がないことを確認します。
- セクタのセキュリティ属性ビット PRIVBByx[i] を変更した後で、ISB 命令を追加します。

注意： セクタの特権から非特権への切り替えでは、関連するセクタの内容は消去されません。

表 53. 特権保護のまとめ

アクセス (TZ は関係なし)		メイン Flash メモリ	
		非特権セクタ	特権セクタ
特権	フェッチ	OK	
	読出し		
	書込み		
	消去		
非特権	フェッチ	OK	RAZ
	読出し		WI、WRPERR
	書込み		
	消去		

表 54. 特権および全体またはバンク消去

アクセス (TZ 状態は関係なし)	メイン Flash メモリ		
	非特権 FLASH	特権 FLASH	非特権および特権 FLASH の組み合わせ
MEP	OK		
MENP	OK	WI、WRPERR	

注： PRIVBByx[i]のアクセス制御に関しては、表 55 と表 56 を参照してください。

表 55. 特権設定レジスタ・アクセス条件 (TZ 有効)

アクセス			対応するセクタの セキュア・ステータス ⁽¹⁾	PRIVBBxy のセクタ設定アクセス
フェッチ	特権／非特権	セキュア／ 非セキュア	-	バス・エラー
読出し			-	OK (全セクタ)
書込み	特権	セキュア	-	OK (全セクタ)
		非セキュア	NS	OK (そのビットに対応したセクタのみ)
			S	WI (そのビットに対応したセクタのみ)
	非特権	セキュア／ 非セキュア	-	WI

1. ウォーターマークベースまたはブロックベースのいずれか

表 56. 特権設定レジスタ・アクセス条件 (TZ 無効)

アクセス		PRIVBBRxy のセクタ設定アクセス
フェッチ	特権／非特権	バス・エラー
読出し		OK
書込み	特権	OK
	非特権	WI

7.6.5 Flash メモリ・レジスタの特権および非特権モード

Flash メモリ・レジスタは、Flash 特権設定レジスタ (FLASH_PRIVCFGR) の SPRIV ビットおよび NSPRIV ビットに応じて特権および非特権アクセスで読み書きができます。

- SPRIV (NSPRIV) ビットがリセットされているときは、セキュア (非セキュア) の Flash メモリ・レジスタがすべて特権、非特権アクセスのどちらでも読み書きができます。
- SPRIV (NSPRIV) ビットがセットされているとき、すべてのセキュア (非セキュア) Flash メモリ・レジスタは特権アクセスのみで読み書きができます。特権レジスタへの非特権アクセスは RAZ/WI です。

キー・ストレージに関連したレジスタ (FLASH_NSIOBKCFGR、FLASH_SECOBKCFGR、FLASH_NSIOBKKEYR、FLASH_SECOBKKEYR) はこの規則の例外で、他の設定に関係なく特権アクセスのみが可能です。

次の表は Flash メモリ・レジスタへのアクセス制御をまとめたものです。

表 57. Flash メモリ・レジスタへのアクセス⁽¹⁾

アクセス			非セキュアレジスタ		セキュアレジスタ	
			NSPRIV = 1	NSPRIV = 0	SPRIV = 1	SPRIV = 0
X	セキュア / 非セキュア	特権 / 非特権	バスエラー			
読出し / 書込み	セキュア	特権	OK			
読出し / 書込み		非特権	RAZ、WI	OK	RAZ、WI	OK
読出し / 書込み	非セキュア	特権	OK		RAZ、WI、ILAP	
読出し / 書込み		非特権	RAZ、WI	OK		

1. 一部のレジスタは特権アクセスのみで設定不可

7.6.6 バンク・スワップの際のFlash メモリ・バンク属性

SWAP_BANK オプション・ビットがメモリ・マップの各バンクのアドレスを変更します。SWAP_BANK をリセットすると、Flash メモリのバンク 1 がより小さなアドレス範囲にマッピングされます。SWAP_BANK がセットされると、Flash メモリのバンク 1 がより大きなアドレス範囲にマッピングされます。Flash メモリのバンク属性はバンクの内容に従うので、バンクをスワップしたときに設定レジスタを修正する必要はありません。

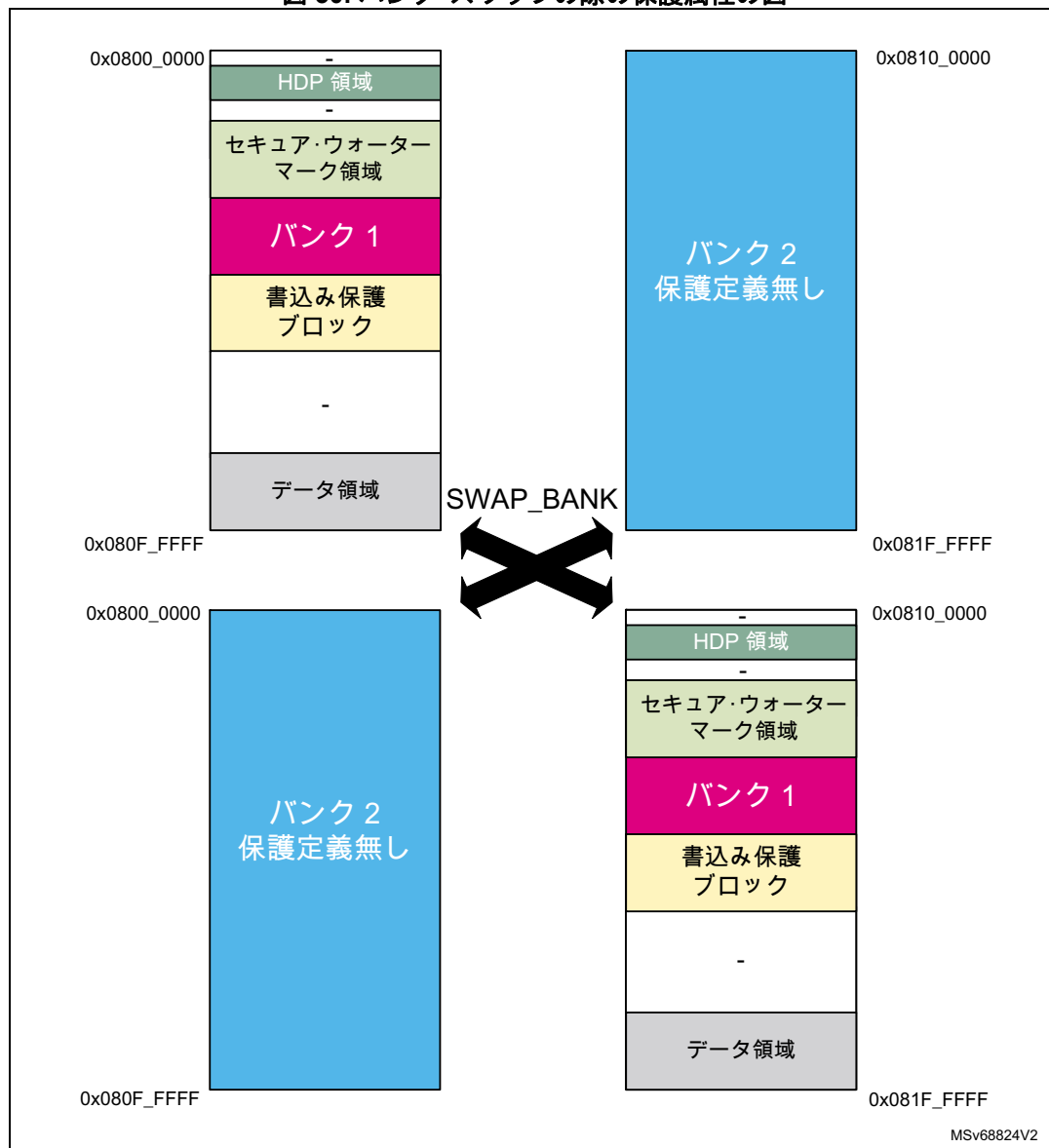
- FLASH のセキュア・ウォーターマーク × FLASH_SECWMx
- Flash メモリの書込み保護セクタ・グループ FLASH_WRPSG (セクション 7.6.8 参照)
- FLASH_HDPx レジスタの非表示保護
- Flash メモリのセキュア・ブロック・ベース・バンク x レジスタ y FLASH_SECBBxy
- Flash メモリの特権・ブロック・ベース・バンク x レジスタ y FLASH_PRIVBBxy
- データ領域設定 FLASH_PRIVBBxy

SWAP_BANK は、ユーザ OB の BOOT_LOCK 設定により変更不可にレンダリングされます。TZEN が有効になると、SECBOOT_LOCK がオプションをロックします。TZEN が無効になると、NSBOOT_LOCK が SWAP_BANK オプションをロックします。

注: **FLASH ECC 検出レジスタ (FLASH_ECCDETR)** と **FLASH ECC 訂正レジスタ (FLASH_ECCCORR)** の BK_ECC ビット、**Flash 非セキュア制御レジスタ (Flash_NSCR)** の BKSEL ビット、**Flash セキュア制御レジスタ (Flash_SECCR)** の BKSEL ビットは、SWAP_BANK 値に関係なく、低 (高) の場合は常にバンク 1 (バンク 2) を参照します。

バンク・スワップの際にセキュリティ属性と保護がどのように動作するかを図 30 に示します。

図 30. バンク・スワップの際の保護属性の図



7.6.7 Flash メモリ設定保護

このメモリは、ハードウェアメカニズムを使用して、不要な変更や誤った変更（ソフトウェアのバグなど）から次の資産を保護します。

- オプション・バイトの変更
- 書込み操作
- 消去コマンド
- 割込みマスク

メモリ設定レジスタの保護の概要を [表 58](#) に示します。この表に示されていないレジスタはキーでは保護されません。

表 58. Flash インタフェースレジスタ保護の概要

レジスタ名	アンロックするレジスタ	保護資産
Flash_NSCLR	Flash_NSKEYR	書込み／消去制御
Flash_SECCR	Flash_SECKEYR	セキュア書込み／消去制御
FLASH_OPTCR + all _PRG レジスタ	Flash_OPTKEYR	Flash バンク・オプション・バイトの変更
FLASH_NSOKCFGR	FLASH_NSOKKEYR	Flash OBkey ストレージ操作
FLASH_SECOBKFGR	FLASH_SECOBKEYR	

7.6.8 書込み保護

書込み保護の目的は、コードやデータに対する不要な変更を防止することです。

連続する 4 つの 8 KB セクタの Flash グループは、FLASH_WRP1R/2 レジスタの対応する WRPSGn1/2 ビットをクリア／セットすることで、個別に書込み保護または保護解除することができます（[バンク 1 の FLASH 書込みセクタ・グループ保護\(FLASH_WRP1R_CUR\)](#)と[バンク 2 の FLASH 書込みセクタ・グループ保護\(FLASH_WRP2R_CUR\)](#)参照）。

書込み保護されたセクタのグループは消去することも、プログラミングすることもできません。その結果、セクタの 1 つのグループが書込み保護されていると、NVSTATE の OPEN への遷移がトリガされない限り（ユーザ Flash メモリ全体の消去）、バンク消去を実行することができません。

注： 書込み保護エラーは、[セクション 7.9](#) に記載されています。

7.6.9 Flash の高サイクル・データ保護

内蔵 Flash メモリは、高サイクル能力（100K サイクル）を持った 96 KB のメモリ領域を持つように設定可能で、データを格納したり EEPROM をエミュレートしたりします。[セクション 7.3.10](#) を参照ください。

ユーザ Flash メモリのオプション・バイト制御のセキュリティ機能は、セクタが高サイクル・データ領域として使われることを防いでいます。設定の矛盾があった場合でも OPTCHANGEERR は立たず、セクタを使用できなくするだけです。[表 59](#) に詳細を列記します。

FLASH_SECBBxx と FLASH_PRIVBBxx を使った揮発性設定が適用できます。この場合、[図 23](#)、[図 24](#)、[図 25](#) に示すとおり、ユーザ Flash メモリ・セクタ 127 のセキュリティ設定が、Flash の高サイクル・データ・セクタ 0 などに適用されます。

データ領域のセクタにはみ出す HDP 拡張の揮発性設定も、効果的にデータ領域をアクセス不可にします。

表 59. 高サイクル領域保護のまとめ：データ領域のアドレス範囲へのアクセス

TZ 保護 (TZ_STATE = 0xB4 のとき) ⁽¹⁾		データ・セクタ有効					データ・セクタ無効	
		HDP 保護	OB セキュア 領域	SecBB 保護		NS セクタ		
				16/32 ビット・アクセス	その他	16/32 ビット・アクセス		その他
セキュア	フェッチ	バス・エラー	バス・エラー					バス・エラー
	読出し		OK	バス・エラー	RAZ、ILAC			
	書込み		WRP ではない場合 : OK、あるいは WI、WRPERR		WI、WRPERR、ILAFM			
	消去	対応するクラシック・ユーザ Flash セクタと同じ保護						
非セキュア	フェッチ	バス・エラー	バス・エラー					
	読出し		RAZ、ILAC		OK	バス・エラー		
	書込み		WI、WRPERR、ILAFM		WRP ではない場合 : OK、あるいは WI、WRPERR			
	消去	対応するクラシック・ユーザ Flash セクタと同じ保護						

1. TZ_STATE = 0xC3 の場合、セキュア・ステートアクセスやSecBB、セキュア領域がありません。HDP と NS アクセスに関しては規則に変更はありません。

表 60. HDP 保護の定義

ユーザ Flash メモリ	OB HDP 領域有効	HDP 拡張子有効
HDPL0 または HDPL1	HDP protected = 0	HDP protected = 0
HDPL2	HDP protected = 1	HDP protected = 0
HDPL3	HDP protected = 1	HDP protected = 1

表 61. 特権セクタとデータ領域：データ領域のアドレス範囲へのアクセス

アクセス		データ領域のアドレス範囲へのアクセス	
		非特権セクタ	特権セクタ
特権	フェッチ	バス・エラー	
	読出し	OK	OK
	書込み	OK	OK
非特権	フェッチ	バス・エラー	
	読出し	OK	RAZ、ILAC
	書込み	OK	WI、WRPERR

7.6.10 ライフ・サイクル・マネージメント

不揮発性状態またはデバッグ状態は、ユーザ・オプション・バイトで設定される製品状態により決まります。デバッグの可能性と、選択したセキュリティ設定変更の可能性はこの状態に関連しています。状態は OB に保存されていますが、デバッグ方針を制御するのは SBS です。

OBK ストレージ・アクセス条件は PRODUCT_STATE の影響は受けませんが、回帰遷移でキーはエビクションされる可能性があります。

HDP はデバッグ状態に関係なく均等に稼働します。デバッグでも HDPL インクリメント時には、HDP 領域が非表示になります。

次の状態が定義されます。

表 62. 製品状態、デバッグ状態、デバッグ・ポリシー

PRODUCT_STATE	OB のコード	説明
オープン	0xED	ユーザ Flash オープン (TZ セキュアおよび非セキュア・オープン) 外部アクセス (デバッグ) N および NS を有効化 (~RDP0)。
プロビジョニング	0x17	プロビジョニング: 不変の信頼の基点 (ルート・オブ・トラスト) が実装されています。 TZEN = 0xC3 の時オープン
iROT プロビジョン	0x2E	不変の信頼の基点 (ルート・オブ・トラスト) が実装されています (~RDP 0.5)。
TZ クローズド	0xC6	NS アプリケーションのデバッグ状態。 デバッグは非セキュア領域に制限されています。
クローズド	0x72	セキュア・アプリケーションの実行状態。 デバッグは無効、回帰は可能。
ロック	0x5C	他の状態への遷移、ポリシー変更、またはデバッグは許されていません (1)。
回帰	0x9A	オープンへの遷移で、デバッグ認証システムにより引き起こされた一時的な状態。
NS 回帰	0xA3	TZ クローズドへの遷移でのデバッグ認証システムにより引き起こされた一時的な状態。

1. 誤った、または定義されていないデバッグ証明書のクローズド状態も同じ状況です。

1. デバッグ保護なし

Flash メモリのメイン・メモリ領域に、読出し、プログラム、および消去操作を実行できます。オプション・バイトへの最低限の制約。

2. 非セキュア・デバッグのみ

非セキュアな Flash メモリに対するすべての読出し／書込み操作が可能です (書込み保護がセットされていない場合)。セキュア領域へのデバッグアクセスは禁止されます。非セキュア領域へのデバッグアクセスは可能なままです。

次の規則が適用されます。

- ユーザ・モード：ユーザ・モードで実行するコード（ブート Flash）は、すべての操作（読み出し、消去、プログラム）で Flash メモリのメイン・メモリおよびオプション・バイトにアクセスが可能です。
- 非セキュアデバッグ・モード：Flash メモリのセキュア領域はデバッグ目的ではアクセスできませんが、非セキュア領域のアクセスは可能なままです。
- ブート RAM モード：SRAM からのブートはできません。
- オープン状態への遷移は、Flash 全体消去とキー・スロット・リボケーションの結果として可能です。デバッグのアンロック・ポリシーに依存します（以下を参照してください）。

3. 製品の全保護

すべてのデバッグ機能は無効になっており、次の規則が適用されます。

- ユーザ・モード：ユーザ・モードで実行するコード（ブート Flash）は、すべての操作（読み出し、消去、プログラム）で Flash メモリのメイン・メモリおよび選択したオプション・バイトにアクセスが可能です。
- ブート RAM モード：SRAM からのブートはできません。
- TZ クローズドまたはオープン状態への遷移は、Flash 全体消去とキー・スロット・リボケーションの結果として可能です。デバッグのアンロック・ポリシーに依存します（以下を参照してください）。

7.6.11 製品状態の遷移

よりクローズドな状態へ進んでいくのが通常の製品のライフ・サイクルです。ここではセキュリティの対策は不要です。オープン状態への遷移は回帰で、デバッグ認証制御によりコントロールされます。デバッグ・アンロック・ポリシーが「ロック」に設定されると、回帰を受け付けなくなります。ロックされた状態は、デバッグ認証証明書を無効化することで強化されます。

この章に記載がないすべての遷移は不正です。

現在の製品状態の読み出しには制約はありません。

遷移の進む方向

オープンからクローズド状態へのあらゆる遷移は、正しい製品設定とプロビジョニングの問題です。遷移は正しい順番で行われなくてはなりません。

通常は次のように進みます。

- オープン、プロビジョニング、または iROT プロビジョン
- iROT プロビジョンへのプロビジョニング
- iROT プロビジョンからクローズド、ロックまたはクローズドへ
- クローズドからロックまたはクローズへ

遷移はデバイスで実行されているソフトウェアか、あるいはデバッグ・インタフェースを用いて直接管理されます。クローズドからの遷移はデバイスで実行されているソフトウェアでのみ可能です。ソフトウェアでは、遷移はブートローダまたは信頼の基点（ルート・オブ・トラスト）サービスによりトリガされますが、通常、デバイスで実行されているソフトウェアは進行遷移を行うことができます。

オープン状態への遷移

この遷移は完全な回帰です。開始状態はオープンとロックを除くあらゆる状態です。デバッグ・ツールをデバッグ認証ライブラリと共に、デバイスの HDPL1 で実行されているデバッグ回帰のアクセス権の認証に使用します（[セクション 14.3.6 : SBS デバッグ制御](#)参照）。認証情報を検証した後、デバイスは中間状態の回帰になります。この状態からデバイスは HDPL0 でオープンへとセキュアに復帰します。

遷移は次のような結果をもたらします。

- エポック・タイムのセキュア・カウンタがインクリメントします。
- キー・スロット・リボケーション
- 製品状態の更新
- ユーザ・オプション更新（セキュリティ設定リセット、TZEN リセット）
- Flashメモリの全体消去
- バックアップ RAM 消去のリクエスト

クローズドからクローズド-セキュア・デバッグ状態への遷移

この遷移は、ROT 機能のセキュリティを損なうことなく、非セキュア・アプリケーション・ソフトウェアのデバッグを認証するためにデバイスをオープンします。

開始状態はクローズドです。デバッグ・ツールを、デバイスの HDPL1 で実行されているデバッグ認証ライブラリと共にデバッグ回帰のアクセス権の認証に使用します（[セクション 14.3.6: SBS デバッグ制御](#)参照）。認証情報を検証した後、デバイスは中間状態の NS-回帰になります。この状態からデバイスは HDPL0 でクローズドへとセキュアに復帰します。

遷移は次のような結果をもたらします。

- エポック・タイムの非セキュア・カウンタがインクリメントします。
- 非セキュア・スロット・リボケーション
- 製品状態の更新
- セキュア・ウォーターマークでカバーされていない Flash メモリのセクタは消去されます。
- バックアップ RAM 消去のリクエスト

表 63. PRODUCT_STATE 遷移

開始状態	終了状態 ⁽¹⁾							
	オープン	プロビジョニング	iROT プロビジョン	TZ クローズド ⁽²⁾	クローズド	ロック	回帰	NS 回帰 ⁽²⁾
オープン	-	OK	OK	X	X	X	X	X
プロビジョニング	X	-	OK	OK	OK	OK	OK (HDPL1)	X
iROT プロビジョン	X	X	-	OK	OK	OK	OK (HDPL1)	X
TZ クローズド	X	X	X	-	OK	OK	OK (HDPL1)	X
クローズド	X	X	X	X	-	X	OK (HDPL1)	OK (HDPL1)
ロック	X	X	X	X	X	-	X	X
回帰	OK (HDPL0)	X	X	X	X	X	-	X
NS 回帰	X	X	X	OK (HDPL0)	X	X	X	-

1. X: 遷移が禁止 (OPTCHANGEERR が立つ)、-: 変化なし、OK: 有効な遷移。

2. この状態に遷移するには、TZEN の現在の値とプログラムされた値の両方が有効 (TZEN = 0xB4) でなくてはならず、そうでない場合は変更が無視され OPTCHANGEERR が立ちます。

表 63 の X は許されていない遷移を示し、ダッシュは変化なし、OK は有効な遷移を示します。いくつかの遷移は正しい HDPL でのみ可能で、括弧内に示しています。不正な遷移を行おうとすると OPTCHANGEERR が立ちます。

間違った PRODUCT_STATE (OBL ではない) はロックとして解釈されます。

7.6.12 OBK 保護

OBK アクセスを受け取ると (読出し/書込み/実行)、セキュア特権属性が最初に確認され、その後 OBK-HDPL を確認します。OBK-HDPL 値はキーの場所に割り当てられた HDPL と厳密に一致なくてはならず、一致しない場合は OBKERROR フラグが立ちます。Flash メモリのインタフェース・レスポンスは 表 64 で説明します。

表 64. TZ OBK 保護のまとめ

TZ 保護 (TZ_STATE = 0xB4 のとき)	OBK アクセス		
	タンパなし		タンパ
	HDPL0/1/2/3	OBK セレクタ	
XS	バス・エラー	バス・エラー	RAZ、WI
RS	特権の場合 OK、その他は RAZ	RAZ	
WS	特権の場合 OK、その他は WI、WRPERR	特権でない場合は WI および WRPERR	
XNS	バス・エラー	バス・エラー	
RNS	RAZ、ILAFM	RAZ、ILAFM	
WNS	WI、WRPERR、ILAFM	WI、WRPERR、ILAFM	
消去	該当なし	該当なし	該当なし

表 65. TZ 無効時の OBK 保護のまとめ

TZ 保護 (TZ_STATE = 0xC3 のとき)	OBK アクセス		
	タンパなし		タンパ
	HDPL0/1/2/3	OBK セレクタ	
XNS	バス・エラー	バス・エラー	RAZ、WI
RNS	特権の場合 OK、その他は RAZ	RAZ	
WNS	特権の場合 OK、その他は WI、WRPERR	特権の場合 WI、その他は WI、WRPERR	
消去	該当なし	該当なし	該当なし

注： キャッシュ可能なアクセスは flitf ではなくシステムレベルで管理されています。キャッシュ可能なアクセスが好ましくない場合、MPU はキャッシュを無効化するように設定する必要があります。

タンパの場合、すべてのキーは 0x00s として読み出されます。

また、OBKFGCR および OBKKEYR レジスタにアクセスする規則もあります。

表 66. セキュアな制御レジスタへのアクセス条件

-	S/NS	P/NP	TZ の状態	SECOBKCFGR および SECOBKKEYR アクセス
X	-	-	-	バスエラー
読出し/ 書込み	S	P	アクティブ	OK
読出し/ 書込み	S	NP	アクティブ	RAZ、WI
読出し/ 書込み	NS	P	アクティブ	RAZ、WI、ILAP
読出し/ 書込み	NS	NP	アクティブ	RAZ、WI、ILAP
読出し/ 書込み	-	-	非アクティブ	RAZ、WI

表 67. 非セキュアな制御レジスタへのアクセス条件

	S/NS	P/NP	TZ の状態	NSOBKCFGR および NSOBKKEYR アクセス
X	-	-	-	バスエラー
読出し/ 書込み	-	-	アクティブ	RAZ、WI
読出し/ 書込み	-	P	非アクティブ	OK
読出し/ 書込み	-	NP	非アクティブ	RAZ、WI

7.6.13 一度だけプログラム可能な（ワンタイムプログラマブル）、読出し専用メモリの保護

Flash メモリの OTP/RO セクションは [セクション 7.3.9](#) で説明しています。専用の書込み保護以外には、Flash メモリ・インタフェースにより提供される保護はありません。

製造時に作り込まれると RO 領域には書込みは不可能です。[OTP 書込み保護](#)に OTP 領域のロック・アウト方法を示します。

アクセス条件を 表 68 にまとめます。

表 68. OTP/RO アクセスの制約

-	RO/OTP 専用の 8 KB セクタ			
	0x08FF_E000 - E7FF	0x08FF_E800 - EFFF 予約済みです。	0x08FF_F000 - F7FF OTP	0x08FF_F800 - FFFF RO
XS	バス・エラー	バス・エラー		
RS		RAZ、ILAC		
WS		WI、WRPERR、ILAFM		
XNS	バス・エラー	バス・エラー		
RNS		サイズが正しい場合は ⁽¹⁾ RAZ、それ以外は BUS ERROR	サイズが正しい場合は OK、それ以外は BUS ERROR	
WNS		サイズが正しい場合は WI、 それ以外は BUS ERROR	サイズが正しくない場合は BUS ERROR ブロックが LOCKBL により保護さ れている場合は WRPERR、それ以 外は OK。	サイズが正しい場合は WI、 それ以外は BUS ERROR
消去	WI			

1. ワードサイズは 16 bit です。32 bit アクセスが可能です。これ以外のアクセスを行おうとするとバス・エラーとなります。

7.7 システムメモリ

7.7.1 概要

システム・メモリは、ST が製造時にプログラムした RSS (ルート・セキュア・サービス) ファームウェアを格納しています。RSS はユーザ・ファームウェアにランタイム・サービスを提供します。

TrustZone が有効な場合 (ユーザが TZEN ビット・フィールドに 0xB4 をセットした場合)、RSS はセキュアなユーザ・ファームウェアのみにセキュア・サービスを提供します。TrustZone が無効になっている場合 (ユーザが TZEN ビット・フィールドに 0xC3 をセットした場合)、RSS はユーザ・ファームウェアにサービスを提供します。

7.7.2 RSS ユーザ機能

RSS は、RSS ライブラリを利用してランタイム・サービスを提供でき、その機能は、STM32CubeH5 ファームウェア・パッケージにより提供される CMSIS デバイス・ヘッダ・ファイルの中でユーザにエクスポートされます (詳細は UM3065「STM32H5 シリーズの STM32CubeH5 を始める (Getting started with STM32CubeH5 for STM32H5 Series)」参照)。

RSS は、RSSLIB と NSSLIB というセキュアなユーザ・ファームウェア専用の 2 つの異なるライブラリを通じ、それぞれシステムが TrustZone 有効 (TZEN ビット・フィールドを 0xB4 にセット) および TrustZone 無効 (TZEN ビット・フィールドを 0xC3 にセット) に設定されている場合にサービスを提供します。

表 69. RSS サービス向け Macros

TrustZone	C で定義されたマクロ	Flash メモリの中の場所
有効	RSSLIB_PFUNC	0xBF9FB78
無効	NSSLIB_PFUNC	0xBF9FB6C

RSSLIB

セキュアなユーザ・ファームウェアが C で定義されたマクロ RSSLIB_PFUNC を使って RSSLIB 機能呼び出します。これは非セキュアなシステム・メモリの中の場所を指定します。RSSLIB 機能呼び出す前にセキュアなユーザ・ファームウェアは、RSSLIB_SYS_FLASH_NS_PFUNC_START (0xBF9FB78) から始まり、RSSLIB_SYS_FLASH_NS_PFUNC_END (0xBF9FB84) で終わる Cortex[®]-M33 の SAU 内にあるこの場所より上の非セキュア領域を定義しなくてはなりません。この最後のアドレスは CMSIS デバイスのヘッダ・ファイルに用意されています。

ユーザはこの非セキュア領域を、CMSIS システム・パーティション・ヘッダ・ファイルを使うか、SAU セットアップ用の自分のコードを実装することで設定することができます。CMSIS システム・パーティション・ヘッダ・ファイルは、STM32CubeH5 ファームウェア・パッケージの一部です。

表 70. RSS ライブラリ・インタフェース機能

ライブラリ	機能	属性
RSSLIB_PFUNC	JumpHDPLv2	セキュアな呼出し可能な機能
	JumpHDPLv3	
	JumpHDPLv3NS	
	RSSLIB_DataProvisioning	非セキュアな呼出し可能な機能

RSSLIB_DataProvisioning

セキュリティ属性：非セキュアな呼出し可能な機能。

プロトタイプ：

```
uint32_t RSSLIB_DataProvisioning_(RSSLIB_DataProvisioningConf_t *pConfig)
```

引数：

pConfig：入力パラメータ。RSSLIB_DataProvisioningConf_t

C 構造定義を下に説明します。

```
typedef struct
{
    uint32_t *pSource;
    uint32_t *pDestination;
    uint32_t Size;
    uint32_t DoEncryption;
    uint32_t Crc;
} RSSLIB_DataProvisioningConf_t;
```

構造要素

- pSource** プロビジョニングするデータのアドレスを提供します。SRAM3 のアドレス範囲になくはありません (非セキュア・エイリアス)
- pDestination** プロビジョニングするデータを格納するアドレスを提供します。OBKeys アドレス範囲になくはありません。16 バイトに合わせるものとします。
- サイズ** プロビジョニングするデータのサイズを提供します (バイト数で 16 の倍数でなくてはなりません)。
- DoEncryption** RSSLIB_DataProvisioning 機能に OBKeys 内のデータを暗号化しなくてはならないかどうかを通知します。
DoEncrypt は 0xF5F5A0AAU か 0xCACA0AA0U のいずれかです (STM32H563xx デバイスでは 1 つのみが許されています)。
– 0xF5F5A0AAU : OBKeys 内でプログラミングをする前に適切な DHUK でデータを暗号化しよう、RSSLIB_DataProvisioning に通知します。DHUK は pDestination 値に従って選択します。
– 0xCACA0AA0U : RSSLIB_DataProvisioning に OBKeys 内のクリア・データでプログラムするように通知します。
- Crc** フルのソース・データ・バッファの CRC、pConfig->pDestination 値、pConfig->Size 値、そして最終的に pConfig->DoEncryption 値。
CRC の計算には CRC-32 (イーサネット) を使用します。
– CRC 多項式: 0x04C11DB7U
– 初期値: 0xFFFFFFFFU

戻り値 :

- 0xEAEAEAEAU 成功
- 0xF5F5E0E0U エラー :
– pConfig または pSource が SRAM3 にありません。
– pDestination が 16 B に揃っていません。
– pDestination + Size が OBKey セクションに合っていない。
- 0xF5F50E0EU エラー : **サイズ** が 16 の倍数になっていません。
- 0xF5F58080U エラー : 計算された CRC が pConfig->CRC へのプロビジョンから期待されるものではない。
- 0xF5F58008U エラー : OBKeys の転送先セクション内でデータをプログラムできない。
- 0xF5F5E00EU エラー : DoEncryption パラメータ値が誤っている。
- 0xF5F50880U エラー : 暗号化がリクエストされたがプラットフォームがサポートしていない。
- 0xF5F50EE0U エラー : 暗号化エラー
- 0xF5F50808U. エラー : OBKeys プログラミングエラー

RSSLIB_DataProvisioning がインプット・データ・バッファで受け取り、OBKeys 内でプログラムします。CRC があらゆるデータおよびパラメータのタンパリング問題を回避します。

プロビジョニングするデータのサイズ (Size) に加算した転送先アドレス (pDestination) は、下に示す OBKeys セクションの境界を越えてはなりません。

OBKeys レベル 1	開始アドレス : 0x0FFD0100UL
	終了アドレス : 0x0FFD08FFUL
OBKeys レベル 2	開始アドレス : 0x0FFD0900UL
	終了アドレス : 0x0FFD0BFFUL

OBKeys レベル 3 セキュア	開始アドレス : 0x0FFD0C00UL
	終了アドレス : 0x0FFD17FFUL
OBKeys レベル 3 非セキュア	開始アドレス : 0x0FFD1800UL
	終了アドレス : 0x0FFD1FEFUL

pConfig->DoEncryption パラメータでリクエストされた場合、RSSLIB_DataProvisioning がデータを暗号化してから OBKeys 内でプログラミングします。

RSSLIB_DataProvisioning は AES CBC 128 bit を次のように使用します。

- IV : (C 定義フォーマットを使って) 次のように定義されます。
uint32_t IV = {0x8001D1CEU, 0xD1CED1CEU, 0xD1CE8001U, 0xCED1CED1U};
- キー : 次のように定義される、ターゲットの OBKeys セクションに対応する DHUK。
pConfig->pDestination パラメータ。

注 : OBKeys 内のデータ暗号化は STM32H573xx デバイスでのみサポートされています。

JumpHDPLv12

セキュリティ属性 : セキュアな呼出し可能な機能

プロトタイプ :

```
uint32_t JumpHDPLv12(uint32_t VectorTableAddr, uint32_t MPUIndex)
```

ユーザ・コード機能呼出しの例

```
RSSLIB_PFUNC->S.JumpHDPLv12((uint32_t)NextVectorTableAddr, 1U );
```

引数 :

- VectorTableAddr:
 - 入力パラメータ、適用する次のベクトル表のアドレス。
 - ベクトル表の型式は Cortex-M33 コアが使っているものです。
- MPU インデックス :
 - 入力パラメータ、MPU 領域インデックス。呼出し側の機能は、対応する MPU 領域を定義してから (ただし、無効状態を維持したまま)、JumpHDPLv12 を呼び出さなくてはなりません。この機能は、MPU 領域を有効にしてから、ベクトル表のリセット・ハンドラにジャンプします。ベクトル表リセット・ハンドラ機能は MPU 領域に属します。

ユーザは JumpHDPLv12 を呼び出し HDPL を 2 にインクリメントすることで、ユーザ Flash メモリの HDPL1 領域をクローズし、その後ベクトル表に埋め込んであるリセット・ハンドラにジャンプします。このアドレスは入力パラメータとして受け渡されます。

HDPL1 をクローズした後、JumpHDPLv12 により、インプット・パラメータとして用意された MPU 領域を有効にします。一旦 MPU が有効になると、この機能は受け渡されたベクトル表により提供されたアドレスに SP をセットし、これによりサポートされるリセット・ハンドラ機能にジャンプします。JumpHDPLv12 は新しいベクトル表を設定しません。

実行に成功すると、この機能は LR を戻さず、スタックにプッシュすることはありません。

失敗した場合（悪いインプット・パラメータ値）、RSSLIB_Sec_JumpHDPLv2 は 0xF5F5F5F5 を戻します。

JumpHDPLv3

セキュリティ属性：セキュアな呼出し可能な機能

プロトタイプ：

```
uint32_t JumpHDPLv3(uint32_t VectorTableAddr, uint32_t MPUIndex)
```

ユーザ・コード機能呼出しの例

```
RSSLIB_PFUNC->S.JumpHDPLv3((uint32_t)NextVectorTableAddr, 1U );
```

引数：

- VectorTableAddr:
 - 入力パラメータ、適用する次のベクトル表のアドレス。
 - ベクトル表の型式は Cortex-M33 コアが使っているものです。
- MPU インデックス：
 - 入力パラメータ、MPU 領域インデックス。呼出し側の機能は JumpHDPLv3 を呼び出す前に、対応する MPU 領域を定義しながらも、無効状態を維持したままにしなくてはなりません。この機能は、ベクトル表のリセット・ハンドラにジャンプする前に MPU 領域を有効にします。この機能は MPU 領域に属しています。

ユーザは JumpHDPLv3 を呼び出し HDPL を最大 3 までインクリメントすることで、ユーザ Flash メモリの HDPL1 および HDPL2 領域をクローズし、その後ベクトル表に埋め込んであるリセット・ハンドラにジャンプします。このアドレスは入力パラメータとして受け渡されます。

HDPL1/2 をクローズした後、JumpHDPLv3 により、インプット・パラメータとして用意された MPU 領域を有効にします。一旦 MPU が有効になると、この機能は受け渡されたベクトル表により提供されたアドレスに SP をセットし、ベクトル表によりサポートされるリセット・ハンドラ機能にジャンプします。JumpHDPLv3 は新しいベクトル表を設定しません。

実行に成功すると、この機能は LR を戻さず、スタックにプッシュすることはありません。

失敗した場合（悪いインプット・パラメータ値）、JumpHDPLv3 は 0xF5F5F5F5 を戻します。

JumpHDPLv3NS

セキュリティ属性：セキュアな呼出し可能な機能

プロトタイプ：

```
uint32_t JumpHDPLv3NS(uint32_t VectorTableAddr)
```

ユーザ・コード機能呼出しの例

```
RSSLIB_PFUNC->S.JumpHDPLv3NS((uint32_t)NextVectorTableAddr, 1U );
```

引数 :

- VectorTableAddr:
 - 入力パラメータ、適用する次のベクトル表のアドレス。
 - ベクトル表の型式は Cortex-M33 コアが使っているものです。

ユーザは JumpHDPLv3NS を呼び出し HDPL を最大 3 までインクリメントすることで、ユーザ Flash メモリの HDPL1 および HDPL2 領域をクローズし、セキュア・ドメインから非セキュア・ドメインに移動して、その後ベクトル表に埋め込んである非セキュア・リセット・ハンドラにジャンプします。このアドレスは入力パラメータとして受け渡されます。

HDPL1/2 をクローズしたのち、JumpHDPLv3 はベクトル表がサポートする非セキュア・リセット・ハンドラ機能にジャンプします。JumpHDPLv3NS は新しいベクトル表を設定しません。

実行に成功すると、この機能は LR を戻さず、スタックにプッシュすることはありません。

失敗した場合（悪いインプット・パラメータ値）、JumpHDPLv3NS は 0xF5F5F5F5 を戻します。

NSSLIB

TrustZone が無効になっている場合（TZEN ビットフィールドが 0xC3 に設定）、ユーザ・ファームウェアは NSSLIB のみを呼出します。

ユーザ・ファームウェアは、C で定義されたマクロ NSSLIB_PFUNC を使って NSSLIB 機能と呼出します。これはシステム・メモリ内の場所を示します。

表 71. NSS ライブラリ・インタフェース機能

ライブラリ	機能	属性
NSSLIB_PFUNC	JumpHDPLv2	非セキュア機能
	JumpHDPLv3	

JumpHDPLv2

プロトタイプ :

uint32_t JumpHDPLv2 (uint32_t VectorTableAddr, uint32_t MPUIndex)

ユーザ・コード機能呼出しの例

NSSLIB_PFUNC->JumpHDPLv2((uint32_t)NextVectorTableAddr, 1U);

引数 :

- VectorTableAddr:
 - 入力パラメータ、適用する次のベクトル表のアドレス。
 - ベクトル表の型式は Cortex-M33 コアが使っているものです。
- MPU インデックス :
 - 入力パラメータ、MPU 領域インデックス。呼出し側の機能は JumpHDPLv2 を呼び出す前に、対応する MPU 領域を、その無効状態を維持したまま定義するものとします。この機能は、MPU 領域を有効にしてから、ベクトル表のリセット・ハンドラにジャンプします。ベクトル表リセット・ハンドラ機能は MPU 領域に属します。



ユーザは JumpHDPLv2 を呼び出し HDPL を 2 までインクリメントすることで、ユーザ Flash メモリの HDPL1 領域をクローズし、ベクトル表に埋め込んであるリセット・ハンドラにジャンプします。このアドレスは入力パラメータとして受け渡されます

HDPL1 をクローズした後、JumpHDPLv2 により、インプット・パラメータとして用意された MPU 領域を有効にします。一旦 MPU が有効になるとこの機能は受け渡されたベクトル表により提供されたアドレスに SP をセットし、やはりベクトル表によりサポートされるリセット・ハンドラ機能にジャンプします。JumpHDPLv2 は新しいベクトル表を設定しません。

実行に成功すると、この機能は LR を戻さず、スタックにプッシュすることはありません。

失敗した場合（悪いインプット・パラメータ値）、RSSLIB_Sec_JumpHDPLv2 は 0xF5F5F5F5 を戻します。

JumpHDPLv3

プロトタイプ :

```
uint32_t JumpHDPLv3(uint32_t VectorTableAddr, uint32_t MPUIndex)
```

ユーザ・コード機能呼出しの例

```
RSSLIB_PFUNC->JumpHDPLv3((uint32_t)NextVectorTableAddr, 1U );
```

引数 :

- VectorTableAddr:
 - 入力パラメータ、適用する次のベクトル表のアドレス。
 - ベクトル表の型式は Cortex-M33 コアが使っているものです。
- MPU インデックス :
 - 入力パラメータ、MPU 領域インデックス。呼出し側の機能は JumpHDPLv3 を呼び出す前に、対応する MPU 領域を、その無効状態を維持したまま定義するものとします。この機能は、MPU 領域を有効にしてから、ベクトル表のリセット・ハンドラにジャンプします。ベクトル表リセット・ハンドラ機能は MPU 領域に属します。

ユーザは JumpHDPLv3 を呼び出し HDPL を最大 3 までインクリメントすることで、ユーザ Flash メモリの HDPL1 および HDPL2 領域をクローズし、その後ベクトル表に埋め込んであるリセット・ハンドラにジャンプします。このアドレスは入力パラメータとして受け渡されます。

HDPL1/2 をクローズした後、JumpHDPLv3 により、インプット・パラメータとして用意された MPU 領域を有効にします。一旦 MPU が有効になると、この機能は受け渡されたベクトル表により提供されたアドレスに SP をセットし、これによりサポートされるリセット・ハンドラ機能にジャンプします。JumpHDPLv3 は新しいベクトル表を設定しません。

実行に成功すると、この機能は LR を戻さず、スタックにプッシュすることはありません。

失敗した場合（悪いインプット・パラメータ値）、JumpHDPLv3 は 0xF5F5F5F5 を戻します。

7.8 FLASH 低電力モード

表 72 は、STM32 低電力モードにおけるメモリの振る舞いをまとめたものです。内蔵 Flash メモリは、コア・ドメインに含まれます。

表 72. 低電力モードが内蔵 Flash メモリ に与える影響

電力モード	コアドメイン 電圧範囲	FLASH ビジー時に 可能	FLASH 電力モード
RUN	VOS0/1/2/3	はい	RUN
STOP1 (クロック停止)	SVOS3/4/5	いいえ	SVOS5 の場合、クロックがゲートされるか 停止します
STANDBY	オフ	いいえ	オフ

システム状態が変化した場合、あるいはある所定のシステム状態の中で、メモリはアプリケーションに従ってさまざまな電源電圧範囲 (VOS) で動作する場合があります。メモリを電力モード (RUN、CLOCK GATED、STOP、OFF) に切り替える手順を以下に示します。

注： マイクロコントローラの電力状態の詳細は [セクション 10：電源制御 \(PWR\)](#) を参照してください。

FLASH ドメインの STOP または STANDBY への切り替えの管理

表 72 で説明したとおり、メモリがリセットやクロック制御 (RCC) がビジーである (BSY、DBNE、WBNE がセットされている) と通知すると、マイクロコントローラはコア・ドメインを STOP または STANDBY モードへ切り替えることができません。

メモリを解放するには、2 つの方法があります。

- 次のいずれかの操作で FLASH_NS/SECSR レジスタにある WBNE ビジー・フラグをリセットします。
 - データが欠落している書込みバッファを完了します。
 - FLASH_NS/SECCR レジスタの FW ビットを有効にして、欠落しているデータを入力せずに書込み操作を強制的に実行します。これにより、すべての欠落しているデータは強制的に「ハイ」値に設定されます。
- FLASH_NS/SECCR レジスタの BSY ビジービットがクリアされるまでポーリングします。これは、記録されたすべての書込み、消去、およびオプションの変更操作が完了したことを示します。

これにより、マイクロコントローラはドメインを STOP、または STANDBY モードに切り替えることができます。

7.9 FLASH エラー管理

7.9.1 概要

メモリは、読出し、プログラム、あるいは消去操作中にエラーが発生した場合、自動的に報告を行います。以下のような広範囲なエラーが報告されます。

- 非セキュア書込み保護エラー (WRPERR)
- セキュア書込み保護エラー (WRPERR)
- 非セキュア・プログラミング・シーケンス・エラー (PGSERR)
- セキュア・プログラミング・シーケンス・エラー (PGSERR)
- セキュア・ストローブ・エラー (STRBERR)
- エラー訂正コード・エラー (ECCC、ECCD)
- 不正アクセス (ILAFM/ILAP)
- オプション・バイト変更エラー (OPTCHANGEERR)
- OBK 非セキュア一般エラー (OBKERR)
- OBK セキュア一般エラー (OBKERR)
- OBK 非セキュア書込みエラー (OBKWERR)
- OBK セキュア書込みエラー (OBKWERR)

アプリケーション・ソフトウェアは、[セクション 7.10](#) に詳細が示されているように、エラーごとに個別に割込みを有効化することができます、

Flash メモリ・インタフェースはイベントをトリガするためにさまざまな割込みラインを使用します。NVIC には 2 つのダイレクト・ラインがあります。1 つはセキュア割込みで、もう 1 つは非セキュア割込みです。3 つめのラインは TZIC に ILAFM 割込みをもたらします。

注： すべてのエラーで、アプリケーション・ソフトウェアは、新しい変更操作を試みる前にエラー・フラグをクリアする必要があります。

書込みバッファは 1 つしかなく、1 度に 1 つの操作しか許されないので、書込み制御ビットとステータスビットは両方のバンクで共有されます。

- 書込み制御ビット (PG、FW、EOPIE、WRPERRIE、PGSERRIE、STRBERRIE、INCERRIE、CLR_EOP、CLR_WRPERR、CLR_STRBERR、CLR_INCERR、CLR_PGSERR) は、バンク 1 とバンク 2 を同時に制御します。
- 書込みステータスフラグ (WBNE、DBNE、EOP、WRPERR、PGSERR、STRBERR、INCERR、BSY) はバンク 1 とバンク 2 のエラーを同時に報告します。

7.9.2 非セキュア書込み保護エラー (WRPERR)

不正な非セキュア消去／プログラム操作を不揮発性メモリに行おうとすると、Flash メモリのインタフェースが、FLASH_NSSR レジスタの書込み保護エラー・フラグ WRPERR を立てます。

以下のメモリ領域のいずれかを標的とする場合、消去操作は拒否され、不正としてフラグが立てられます。

- WRPSGn で書込み保護されているセクタ
- HDPL がアクセス不可にした HDP 領域
- 消去されたメモリ属性と合致しないセキュア (TZ) 状態
- 非特権モードで特権セクタの消去を試みる

以下のメモリ領域のいずれかを標的とする場合、プログラム操作は無視され、不正としてフラグが立てられます。

- システム Flash メモリ
- WRPSGn で書き込み保護されているユーザ・セクタ
- LOCKBL でロックされている OTP ブロック
- 読出し専用セクション
- 予約済み領域
- セキュア領域
- 非特権モードからの特権領域
- OBK アクセス条件確認エラー

WRPERR フラグが立てられると、操作は拒否され、対応するバンクは何も変化しません。このエラーが検出されると書き込みバッファが無効になります。

注： あらゆる消去／プログラム操作の前に WRPERR フラグをクリアしなくてはなりません。

FLASH_NSCCR レジスタの CLR_WRPERR ビットを 1 にセットして、WRPERR フラグをクリアします。

FLASH_NSCR レジスタの WRPERRIE ビットが 1 にセットされている場合、WRPERR フラグが立てられたときに割込みが生成されます（詳細は [セクション 7.10](#) を参照）。

7.9.3 セキュア書き込み保護エラー (WRPERR)

不正なセキュア消去／プログラム操作を不揮発性メモリに行おうとすると、Flash メモリのインタフェースが、FLASH_SECSR レジスタの書き込み保護エラー・フラグ WRPERR をセットします。

以下のメモリ領域のいずれかを標的とする場合、消去操作は拒否され、不正としてフラグが立てられます。

- WRPSGn で書き込み保護されているセクタ
- HDPL がアクセス不可にした HDP 領域
- 消去されたメモリ属性と合致しないセキュア (TZ) 状態
- 非特権モードで特権セクタの消去を試みる

以下のメモリ領域のいずれかを標的とする場合、プログラム操作は無視され、不正としてフラグが立てられます。

- システム Flash メモリ
- WRPSGn で書き込み保護されているユーザ・セクタ
- LOCKBL でロックされている OTP ブロック
- 読出し専用セクション
- 予約済み領域
- 非セキュア領域
- 非特権モードからの特権領域
- OBK アクセス条件確認エラー

WRPERR フラグが立てられると、操作は拒否され、対応するバンクは何も変化しません。

このエラーが検出されると書き込みバッファが無効になります。

注： あらゆる消去／プログラム操作の前に WRPERR フラグをクリアしなくてはなりません。
FLASH_SECCR レジスタの CLR_WRPERR ビットを 1 にセットして、WRPERR フラグをクリアします。

FLASH_SECCR レジスタの WRPERRIE ビットが 1 にセットされている場合、WRPERR フラグが立てられたときに割込みが生成されます（詳細は[セクション 7.10](#)を参照）。

7.9.4 非セキュア・プログラミング・シーケンス・エラー (PGSERR)

非セキュア・プログラミング・シーケンスが正しくない場合、Flash メモリのインタフェースが FLASH_NSSR レジスタのプログラミング・シーケンス・エラー・フラグ PGSERR をセットします。

より具体的には、次の条件のいずれかが満たされると PGSERR フラグがセットされます。

- INCERR、WRPERR、PGSERR、STRBERR、OPTCHANGEERR、OBKERR、OBKWERR のようなエラーは、新しい書き込みまたは消去操作、OBK 操作、あるいはオプション変更をリクエストするまではクリアされません。
- 消去では、次のケースで PGSERR がセットされます。
 - 消去操作の欠如 (FLASH_NSCR) : (MER = 0、SER = 0、および BER = 0) で STRT = 1
 - セクタとバンクの消去が同時にリクエストされた (NSSTRT と MER、SER、BER の 2 つ以上)
 - 消去操作の間に PG がセットされた (消去リクエストと書き込みリクエストが同時には起こらないようにしている) : (PG = 1) で (STRT = 1)
 - 書き込みバッファが次のデータを待っている間に消去操作が始まった。WBNE = 1 で (STRT = 1)
 - セキュア・アクセスにより STRT = 1 が設定。
 - DBNE = 1 の間に消去操作が始まった。
- プログラミングでは、次のケースで PGSERR がセットされます。
 - 書き込みフラグが無い : 書き込み操作がリクエストされたが、そのリクエストの前に FLASH_NSCR レジスタでプログラム有効化ビット PG がセットされていない。
 - Flash メモリの高サイクル・データへの書き込み操作がリクエストされたが、PG = 0 である。
 - AHB 書き込みリクエストを受け取り (SER = 1、BER = 1 あるいは MER = 1) である。
 - WBNE = 1 の間に 16 bit データ・アクセスがリクエストされた。
- オプション変更では、次のケースで PGSERR がセットされます。
 - 書き込みバッファが次のデータを待っている間にオプション変更が始まった。WBNE = 1 で OPTSTRT = 1
 - OPTSTRT が非セキュア DBNE = 1 でセットされた。
- オプション・バイト・キー・ストレージ・アクセスでは、次のケースで PGSERR がセットされます。
 - 前の操作でのエラー・フラグがクリアされていない。
 - FLASH_NSOKCFGR の ALT_SECT_ERASE と SWAP_SECT_REQ が同時にセットされた。
 - SWAP_OFFSET 値が間違っている。SWAP_SECT_REQ が 1 に設定された場合、SWAP_OFFSET は OBK_HDPL < N - 1 > _SWAP_OFFSET 以上でなくてはなりません。
 - DBNE = 1 のときに ALT_SECT_ERASE または OBK_SWAP がセットされた。
 - OBK-HDPL が間違っており、非セキュア OBK_SWAP がリクエストされた。
 - 書き込みバッファが新しいデータを待っている間 (WBNE = 1) に、OBKSWAP または OBK の代替消去が始まった。

書き込みの試行 (Flash プログラミング) が失敗したために PGSERR フラグが立った場合、現在のプログラム操作がアボートし、対応するバンクは何も変化しません。書き込みデータバッファも無効になります。

注： 上記保護メカニズムのため、PGSERR フラグが立てられた場合、アプリケーションによって最後に行われた書き込み操作が失われているリスクが存在します。PGSERR に対する割込みを生成し、Flash メモリの値を読み直して、最後の書き込み操作が成功しているか割込みハンドラを確認することを推奨します。

PGSERR フラグによって、新しいプログラム操作もブロックされます。これは新しいプログラム操作が始まる前に PGSERR をクリアしなくてはならないことを意味しています。

FLASH_NSCCR レジスタの CLR_PGSERR ビットを 1 にセットして、PGSERR フラグをクリアします。

FLASH_NSCR レジスタの PGSERRIE ビットが 1 にセットされている場合、PGSERR フラグが立てられたときに割込みが生成されます。詳細については、[セクション 7.10](#) を参照してください。

7.9.5 セキュア・プログラミング・シーケンス・エラー (PGSERR)

セキュア・プログラミング・シーケンスが正しくない場合、Flash メモリのインタフェースが Flash_SECSR レジスタのプログラミング・シーケンス・エラー・フラグ PGSERR をセットします。

より具体的には、次の条件のいずれかが満たされると PGSERR フラグがセットされます。

- INCERR、WRPERR、PGSERR、STRBERR、OBKERR、OBKWERR のようなエラーは、新しい書き込みまたは消去操作、OBK 操作、あるいはオプション変更をリクエストするまではクリアされません。
- 消去では、次のケースで PGSERR がセットされます。
 - 消去操作の欠如 (Flash_SECCR) : (SER = 0、および BER = 0、MER = 0) で STRT = 1
 - セクタとバンクの消去が同時にリクエストされた (SECSTRT と MER、SER、BER の 2 つ以上)
 - 消去操作の間に PG がセットされた (消去リクエストと書き込みリクエストが同時には起こらないようにしている) : (PG = 1) で (STRT = 1)
 - 書き込みバッファが次のデータを待っている間に消去操作が始まった。WBNE = 1 で (STRT = 1)
 - セキュア DBNE = 1 の時に操作が始まった。
- プログラミングでは、次のケースで PGSERR がセットされます。
 - 書き込みフラグが無い : 書き込み操作がリクエストされたが、そのリクエストの前に FLASH_SECCR レジスタのプログラム有効化ビット PG がセットされていない。
 - Flash メモリの高サイクル・データへの書き込み操作がリクエストされたが、PG = 0 である。
 - AHB 書き込みリクエストを受け取り (SER = 1, SBER = 1、あるいは MER = 1) である。
 - WBNE = 1 の間に 16 bit データ・アクセスがリクエストされた。
- オプション変更では、次のケースで PGSERR がセットされます。
 - 書き込みバッファが次のデータを待っている間にオプション変更が始まった。WBNE = 1 で OPTSTRT = 1
 - OPTSTRT がセキュア DBNE = 1 でセットされた
- オプション・バイト・キー・ストレージ・アクセスでは、次のケースで SECPGSERR がセットされます。
 - 前の操作でのエラー・フラグがクリアされていない。
 - ALT_SECT_ERASE と SWAP_SECT_REQ が同時にセットされた。
 - SWAP_OFFSET 値が誤っている : SWAP_SECT_REQ = 1 の場合、SWAP_OFFSET は OBK_HDPL <N-1> SWAP_OFFSET 以上でなくてはなりません。
 - DBNE = 1 のときに SWAP_SECT_REQ または OBK_SWAP がセットされた。
 - OBK-HDPL が間違っており、セキュア OBK_SWAP がリクエストされた。

- 書き込みバッファが新しいデータを待っている間に、OBKSWAP または OBK の代替消去が始まった (WBNE =1)

書き込みの試行 (Flash プログラミング) が失敗したために PGSERR フラグが立った場合、現在のプログラム操作がアボートし、対応するバンクは何も変化しません。書き込みデータバッファも無効になります。

注： 上記保護メカニズムのため、PGSERR フラグが立てられた場合、アプリケーションによって最後に行われた書き込み操作が失われているリスクが存在します。PGSERR に対する割込みを生成し、Flash メモリの値を読み直して、最後の書き込み操作が成功しているか割込みハンドラを確認することを推奨します。

PGSERR フラグによって、新しいプログラム操作もブロックされます。これは新しいプログラム操作が始まる前に PGSERR をクリアしなくてはならないことを意味しています。

FLASH_SECCCR レジスタの CLR_SECPGSERR ビットを 1 にセットして、PGSERR フラグをクリアします。

FLASH_SECCR レジスタの PGSERRIE ビットが 1 にセットされている場合、PGSERR フラグが立てられたときに割込みが生成されます。詳細については、[セクション 7.10](#) を参照してください。

7.9.6 非セキュア・ストローブ・エラー (STRBERR)

非セキュア・アプリケーション・ソフトウェアが書き込みバッファの同一のバイトに数回書き込みを行った場合、書き込みアクセスのターゲット・バンクに関係なく、Flash メモリのインタフェースがストローブ・エラー・フラグ STRBERR (FLASH_NSSR) を立てます。

STRBERR フラグが立つと、現在のプログラム操作がアボートして書き込みバッファが無効になります。

FLASH_NSCCR レジスタの CLR_STRBERR ビットを 1 にセットして、STRBERR フラグをクリアします。

FLASH_NSCR レジスタの STRBERRIE ビットが 1 にセットされている場合、STRBERR フラグが立てられたときに割込みが生成されます。詳細については、[セクション 7.10](#) を参照してください。

7.9.7 セキュア・ストローブ・エラー (STRBERR)

セキュア・アプリケーション・ソフトウェアが書き込みバッファの同一のバイトに数回書き込みを行った場合、書き込みアクセスのターゲット・バンクに関係なく、メモリがストローブ・エラー・フラグ STRBERR (FLASH_SECSR) を立てます。

STRBERR フラグが立つと、現在のプログラム操作がアボートして書き込みバッファが無効になります。

FLASH_SECCCR レジスタの CLR_STRBERR ビットを 1 にセットして、STRBERR フラグをクリアします。

FLASH_SECCR レジスタの STRBERRIE ビットが 1 にセットされている場合、STRBERR フラグが立てられたときに割込みが生成されます。詳細については、[セクション 7.10](#) を参照してください。

7.9.8 非セキュア不整合エラー (INCERR)

非セキュア・アクセスのプログラミング不整合が検出された場合、Flash メモリのインタフェースがレジスタ FLASH_NSSR の不整合エラーフラグ INCERR をセットします。

より具体的には、次の条件のいずれかが満たされると INCERR フラグがセットされます。

- 例えば次のように、前の書き込み操作が完了する前に書き込み操作が試行された：
 - アプリケーション・ソフトウェアが 128 bit 書き込みバッファを満たすために書き込み操作を開始したが、バッファがフルになる前に、別の Flash メモリアドレスに対して新しいバースト・リクエストを送信した。
 - 1 つのマスタが書き込み操作を開始したものの、バッファがフルになる前に、同一アドレスまたは別のアドレスに対する新しい書き込み操作を別のマスタが開始した。
 - 書き込みバッファをフィルしている間に、FLASH_NSCKCFGR の ALT_SECT が変わった。

注： 新しい書き込み操作を開始する前に INCERR フラグをクリアする必要があります。そうしないと、シーケンスエラー (PGSERR) が立てられます。

不整合エラーの発生時にデータが失われるのを回避するため、以下の手順に従うことを推奨します。

1. INCERR フラグが立てられたときに、ハンドラ・ルーチンを実行します。
2. メモリに対するすべての書き込みリクエストを停止します。
3. INCERR ビットをクリアします。
4. 割り込まれた場所から書き込み操作を再開します。

FLASH_NSSCR レジスタの CLR_INCERR ビットを 1 にセットして、INCERR フラグをクリアします。

FLASH_NSSCR レジスタの INCERRIE ビットが 1 にセットされている場合、INCERR フラグが立てられたときに割込みが生成されます (詳細は [セクション 7.10](#) を参照)。

7.9.9 セキュア不整合エラー (INCERR)

プログラミング不整合が検出された場合、Flash メモリのインタフェースがレジスタ Flash_SECSR の不整合エラー・フラグ INCERR をセットします。

より具体的には、次の条件のいずれかが満たされると INCERR フラグがセットされます。

- 例えば次のように、前の書き込み操作が完了する前に書き込み操作が試行された：
 - アプリケーション・ソフトウェアが 128 bit 書き込みバッファを満たすために書き込み操作を開始したが、バッファがフルになる前に、別の Flash メモリアドレスに対して新しいバースト・リクエストを送信した。
 - 1 つのマスタが書き込み操作を開始したものの、バッファがフルになる前に、同一アドレスまたは別のアドレスに対する新しい書き込み操作を別のマスタが開始した。

注： 新しい書き込み操作を開始する前に INCERR フラグをクリアする必要があります。そうしないと、シーケンスエラー (PGSERR) が立てられます。

不整合エラーの発生時にデータが失われるのを回避するため、以下の手順に従うことを推奨します。

1. INCERR フラグが立てられたときに、ハンドラ・ルーチンを実行します。
2. Flash メモリに対するすべての書き込みリクエストを停止します。
3. INCERR ビットをクリアします。
4. 割り込まれた場所から書き込み操作を再開します。

FLASH_SECCCR レジスタの CLR_INCERR ビットを 1 にセットして、INCERR フラグをクリアします。

Flash_SECCCR レジスタの INCERRIE ビットが 1 にセットされている場合、INCERR フラグが立てられたときに割込みが生成されます (詳細は [セクション 7.10](#) を参照)。

7.9.10 エラー訂正コード・エラー (ECCC、ECCD)

読み出し中にシングルエラー訂正が検出された場合、Flash メモリのインタフェースが FLASH_ECCCORR レジスタのシングルエラー訂正フラグ ECCC をセットします。

読み出し中に 2 つの ECC エラーが検出された場合、Flash メモリのインタフェースが FLASH_ECCDETR レジスタのダブル・エラー検出フラグ ECCD をセットします。

ECCC フラグが立てられると、訂正された読み出しデータが返されます。アプリケーションはエラーを無視して、新しい読み込み操作を要求することができます。ECCD フラグが立つと NMI が生成され、データアクセス用 (OTP、データ領域、RO データ) SBS レジスタ ([SBS flift ECC NMI マスクレジスタ \(SBS_ECCNMIR\)](#)) によりマスクすることが可能です。ECCD フラグがセットされたら、ソフトウェアは NMI 割込みサービス・ルーチンの命令キャッシュ (CACHEINV = 1) を無効にしなくてはなりません。

ECCC または ECCD フラグが立てられると、エラーが発生した Flash ワードのアドレスが FLASH_ECCCORR (FLASH_ECCDETR) レジスタに保存されます。このアドレスが読み出し専用領域または OTP 領域に対応するか、あるいは Flash メモリの高サイクル・データの場合、FLASH_ECCCORRR (FLASH_ECCDETR) レジスタの OTP_ECC ビットも 1 にセットされます。エラーを生成した関連フラグがリセットされると、このレジスタは自動的にクリアされます。

BK_ECC フラグはどの Flash バンクでエラーが発生したかを示します。

SYSF_ECC フラグはシステム Flash 領域で検出されたエラーを示します。

表 73. ECC 障害の場所

OTP_ECC	SYSF_ECC	BK_ECC	EDATA_ECC	OBK_ECC	Flash領域	ADDR_ECC 最小	ADDR_ECC 最大
0	0	0/1	0	0	ユーザ Flash メモリ	0x0000	0xFFFF
0	1	0/1	0	0	システム Flash メモリ	0x0000	0x0FFF
1	0	0	0	0	OTP	0x0600	0x07FF
0	0	1	0	1	OBKeys	0x0000	0x03FF
0	0	0/1	1	0	データ領域、セクタ 7	0xF000	0xF1FF
0	0	0/1	1	0	データ領域、セクタ 6	0xF200	0xF3FF
0	0	0/1	1	0	データ領域、セクタ 5	0xF400	0xF5FF
0	0	0/1	1	0	データ領域、セクタ 4	0xF600	0xF7FF
0	0	0/1	1	0	データ領域、セクタ 3	0xF800	0xF9FF
0	0	0/1	1	0	データ領域、セクタ 2	0xFA00	0xFBFF
0	0	0/1	1	0	データ領域、セクタ 1	0xFC00	0xFDFF
0	0	0/1	1	0	データ領域、セクタ 0	0xFE00	0xFFFF

注： シングル訂正エラーまたはダブル検出エラーが連続した場合、最初のエラーに対応するアドレスのみが FLASH_ECCCORR (FLASH_ECCDETR) レジスタに格納されます。

新しい読み出し操作を開始する前に、ECCC フラグまたは ECCD フラグを必ずクリアしなくてはなりません。

ECC インタフェースを 2 つのバンクで共有しているため、同じエラーが同時に登録された場合、物理バンク 1 のエラーが記録されます。1 つのバンクが ECCD を報告し、もう 1 つのバンクが ECCC を報告した場合は両方のエラーが記録されます。

ECCC (ECCD) フラグは、FLASH_ECCCORR (FLASH_ECCDETR) レジスタの ECCC ビット (ECCD ビット) を 1 にセットすることでクリアします。

FLASH_ECCCORR レジスタの ECCC ビットが 1 にセットされている場合、ECCC フラグが立てられたときに割込みが生成されます。ECCD には NMI のみが生成されます。詳細については、[セクション 7.10](#) を参照してください。

7.9.11 不正アクセス (ILAFM/ILAP)

不正アクセスは TZIC への信号で、そこでセキュア割込みをトリガします。これは他の割込みを補完するもので、TZ が有効な場合 (TZ_STATE = 0xB4) にのみ生成されます。

GTZC レベルでのみマスク可能です。

次のように、セキュアな Flash メモリ・アクセスの規則に違反した場合に ILAFM が生成されます。

- 非セキュア・モードでセキュア・メモリの位置にアクセスを試みた。
- セキュア・モードで非セキュア・メモリの位置にアクセスを試みた。

ILAP は次の場合に生成されます。

- 非セキュア・モードでセキュア・レジスタにアクセスを試みた。

書き込み中に ILAFM が検出されると、書き込みバッファが無効になります。

注： ILAFM と ILAP は Flash メモリ・コントローラには、操作を再開するためにクリアするフラグがありません。これは TZIC で対処されます。

7.9.12 オプション・バイト変更エラー (OPTCHANGEERR)

オプション変更操作の間に Flash メモリのインタフェースがエラーを発見すると、その操作をアボートして、Flash_NSSR レジスタにオプション・バイト変更エラー・フラグ OPTCHANGEERR をセットします。

次の場合、OPTSTRT ビットがセットされた後にエラーが立ちます。

- TZEN がセットされておらず、TZ クローズドまたは NS 回帰状態への遷移がリクエストされた。その他のあらゆる禁止されている PRODUCT_STATE 遷移 ([表 63](#))
- TZEN と BOOT_LOCK の組み合わせで SWAP_BANK をロック・アウトした。
- 間違った HDPL で OB 変更をしようとした (選択された PRODUCT_STATE 遷移)。
- 不正な値で OB を変更しようとした。
 - 現在の値以下の新しいエポック・タイム値。
 - 仕様で認識されていない値 (列記されているマジック・ナンバーのみ)。
- 許されていない製品状態の OB を変更しようとした (通常、デバッグに対して開かれている状態が変更には必要)。

注： OB の説明の例外と詳細 ([セクション 7.4.7](#) と [表 43](#) 参照)。

FLASH_NSSCR レジスタで CLR_OPTCHANGEERR ビットを 1 にセットすると、OPTCHANGEERR フラグがクリアされます。

FLASH_NSCR レジスタの OPTCHANGEERRIE ビットが 1 にセットされている場合、OPTCHANGEERR フラグが立てられたときに割込みが生成されます (詳細は [セクション 7.10](#) を参照)。

新しい操作 (オプション変更、消去または書き込み) を開始する前に、OPTCHANGEERR フラグを必ずクリアしなければなりません。

7.9.13 その他のハードフォルトエラー

次のイベントによって、対応するバスインタフェースにバスエラーが発生します。

- ターゲット・コードおよび 9 bit ECC データにアクセスする、AHB メイン・システム・バス上。
 - 非セキュア・モードでのセキュア・ユーザ Flash メモリからのフェッチ。
 - セキュア・モードでの非セキュア・ユーザ Flash メモリからのフェッチ。
 - (コードへの使用が禁止されているデータ・アドレスを含む) 不正なアドレスへのアクセス。
 - 正しくない HDPL 値による HDP 領域からのフェッチ。
 - 非特権モードでの特権領域からのフェッチ。
- AHB 設定、またはターゲット OTP/RO にアクセスするためのシステム・バス上 (6 bit ECC を使うすべてのアドレス)。
 - FLASH_NS/SECKEYR または FLASH_OPTKEYR に対する誤ったキー入力。
 - システム AHB インタフェースに対する 8 ビットアクセス。
 - レジスタの間違ったアンロック・シーケンス。

7.9.14 OBK エラー事例 (OBKERR、OBKWERR)

OBK 書込みバッファを埋める際に、次の 4 種類のエラーが検出される場合があります : STRBERR、INCERR、WRPERR、OBKERR。OBKERR は OBK アクセス特有で、他の Flash メモリ資産へのアクセスで立つことはありません。

注 : STRBERR、INCERR、WRPERR に関しては、AHB アクセスの種類 (セキュアまたは非セキュア) により、非セキュアまたはセキュア・フラグが使用されます。非セキュア・アクセスを受け取った場合は、非セキュア・エラー・フラグが使用されます。セキュア・アクセスを受け取った場合は、セキュア・エラー・フラグが使用されます。

OBKERR と OBKWERR に関しては、セキュア状態の値に応じて非セキュアまたはセキュア・フラグが使用されます。SOC がセキュア状態の場合 (TZ_STATE が有効)、セキュア・エラー・フラグ (OBKERR と OBKWERR) が使用されます。それ以外は非セキュア・エラー・フラグ (OBKERR と OBKWERR) が使用されます。

次の場合に OBKERR フラグが立ちます。

- OBK-HDPL (SBS からの入力信号) が、読取りまたは書込みアクセスの際のキーに関連する HDPL に合致せず、TZ と PRIV 属性が正しい (詳細は [セクション 7.5.2](#) 参照)。

書込みアクセス中に OBKERR が立てられると、書込みバッファが一掃されます。読出しアクセス中に OBKERR が立てられても、書込みバッファは変わりません。両方のケースで、フラグをクリアして Flash メモリへの書込み/消去アクセスを有効にしなくてはなりません。

次の場合、書込みバッファを埋めた後に OBKWERR が検出されて立ちます。

- 書込みアクセスの際のアドレスが未使用ではない。
- SWAP 操作中の代替セクタの OBK セレクタが未使用ではない。

OBKWERR が立った場合、書込みバッファが一掃されます。フラグをクリアして Flash メモリへの書込み/消去アクセスを有効にしなくてはなりません。

7.10 FLASH 割込み

Flash メモリのインタフェースは、所定のバンクに以下のイベントをシグナル送信するためにマスク可能な割込みを生成可能です。

- 読出しエラーと書き込みエラー（[セクション 7.9](#) を参照）
 - 読出し操作中のシングル ECC エラー訂正
 - 書き込み不整合エラー
 - 誤ったプログラミングシーケンス
 - 書き込み操作中のストローブエラー
 - オプション変更操作エラー
- セキュリティエラー（[セクション 7.9](#) を参照）
 - 書き込み保護エラー
 - 不正アクセス・エラー：TZIC への信号
- その他のイベント（下記）
 - プログラムの終了

読出し操作中のダブル ECC エラー検出で、NMI が立てられます。

2 つの割込みラインがすべてのエラー・ソースを集めます。1 つは非セキュア・アクセスのエラーで 2 番目はセキュア・アクセスで発生したエラーです。

ユーザは、FLASH_NS/SECCR、FLASH_ECCCORR と FLASH_ECCDETR レジスタのマスク・ビットを変更することで、個別に Flash メモリのインタフェース割込みソースを有効、無効にすることができます。適切なマスクビットを 1 にセットすることで、その割込みが有効になります。

注： [セクション 7.6.7](#) に説明されているように、書き込みを行う前には FLASH_NS/SECCR レジスタをアンロックする必要があります。

[表 74](#) に、使用可能な Flash メモリ・インタフェース割込み機能の概要を示します。下表に説明されているように、新しい操作をトリガする前に一部のフラグをクリアする必要があります。

表 74. Flash 割込みリクエスト

割込みイベント イベントフラグ	レジスタのエラー・ フラグ・ラベル	有効制御ビット	操作再開 クリアフラグ
操作イベントの終了	EOP	EOPIE	N/A
書き込み保護エラー	WRPERR	WRPERRIE	はい
プログラミングシーケンスエラー	PGSERR	PGSERRIE	はい
ストローブエラー	STRBERR	STRBERRIE	はい
OBK 書き込みエラー	OBKWERR	OBKWERRIE	はい
OBK 一般エラー	OBKERR	OBKERRIE	はい
不整合エラー	INCERR	INCERRIE	はい
不正アクセス・エラー	ILAFM	TZEN	いいえ
オプション・バイト・エラー	OPTCHANGEERR	-	はい
ECC シングルエラー訂正イベント	ECCC	ECCCIE	いいえ
ECC ダブルエラー検出イベント ⁽¹⁾	ECCD	無効	いいえ

1. NMI.

表 74 に記載されている個別のマスク可能な割込みソース（オプション・バイト・エラーと ECC を除く）の状態は、FLASH_NS/SECSR レジスタから読み出すことができます。これは FLASH_NS/SECCCR レジスタの適切なビットを 1 にセットすることでクリアできます。

注： 割込みをクリアするには、アウンロックメカニズムは必要ありません。

操作イベントの終了

FLASH_NS/SECCR レジスタの操作終了割込みイネーブル・ビット（EOPIE）をセットすることで、消去操作、プログラム操作、あるいはオプション・バイト変更の最後に割込みを生成することができます。

OBKey ストレージ領域の管理にあたって、EOP は OBK スワップ操作と代替の領域消去操作とも関連しています。これらのイベントのどれかが発生すると、FLASH_NS/SECSR レジスタの EOP ビットもセットされます。

FLASH_NS/SECCCR レジスタの CLR_EOP ビットを 1 に設定することで EOP フラグをクリアすることができます。

7.11 Flash レジスタ

各レジスタにはオフセット・アドレスとリセット値が割り当てられています。オプション・バイト値を表すレジスタの場合、リセット値は OBL プロセスで決定されます。成功した場合、リセット値が OB からロードされます。OBL が失敗した場合は、高度に制約的なデフォルト値がセットされます。

7.11.1 Flashアクセス制御レジスタ (Flash_ACR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

詳細については、[セクション 7.3.4](#) および[セクション 7.3.5](#) を参照してください。

アドレスオフセット : 0x000

リセット値 : 0x0000 0013

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRF TEN	Res.	Res.	WRHIGHFREQ [1:0]		LATENCY[3:0]			
							rw			rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PRFTEN** : プリフェッチ有効化。ビット値が変更された場合、ユーザは ACR レジスタを読み戻して PRFTEN が考慮されていることを確認する必要があります。

プリフェッチを制御するために使われるビット。

0 : プリフェッチ無効。

1 : 遅延が最低 1 ウェイト・ステートの場合、プリフェッチ有効。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **WRHIGHFREQ[1:0]** : Flash信号遅延

これらのビットは、プログラム操作中に不揮発性メモリの信号間の遅延を制御するために使用されます。アプリケーション・ソフトウェアは、それらをメモリのインタフェース周波数に応じた正しい値にプログラムする必要があります。詳細については、[表 37](#) を参照してください。

注 : 設定が正しいかどうかを検証するための確認は行われません。

一部の周波数には 2 つの WRHIGHFREQ 値を選択できます。

ビット 3:0 **LATENCY[3:0]** : 読み出し遅延

これらのビットは両方の不揮発性メモリバンク上で読み出し操作中に使用する、ウェイト・ステート数の制御に使用されます。アプリケーション・ソフトウェアは、それらをメモリのインタフェース周波数および電圧条件に応じた正しい値にプログラムする必要があります。

0000 : 不揮発性メモリからワードを読み出すために、0 ウェイト・ステートを使用します。

0001 : 不揮発性メモリからワードを読み出すために、1 ウェイト・ステートを使用します。

0010 : 不揮発性メモリからワードを読み出すために、2 ウェイト・ステートを使用します。

.....

0111 : 不揮発性メモリからワードを読み出すために、7 ウェイト・ステートを使用します。

1111 : 不揮発性メモリからワードを読み出すために、15 ウェイト・ステートを使用します。

注 : 設定が正しいかどうかを検証するためのハードウェアによる確認は行われません。

7.11.2 Flash 非セキュア・キー・レジスタ (Flash_NSKEYR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも書込みができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

FLASH_NSKEYR は書込み専用レジスタです。FLASH_NSCR レジスタのロックを解除し、プログラミングや消去を許可するには、次の値を連続してプログラムする必要があります。シーケンスを誤ると、次のシステム・リセットまで FLASH_NSCR レジスタがロックされます。

1 番目のキー = 0x4567 0123

2 番目のキー = 0xCDEF 89AB

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **NSKEY[31:0]** : 不揮発性メモリ、非セキュア設定アクセス・アンロック・キー

7.11.3 Flash セキュア・キー・レジスタ (Flash_SECKEYR)

このレジスタはセキュアです。これはセキュア・アクセスによってのみ書込み可能です。非セキュア読み出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

FLASH_SECKEYR は書込み専用レジスタです。FLASH_SECCR レジスタのロックを解除し、プログラミングや消去を許可するには、次の値を連続してプログラムする必要があります。シーケンスを誤ると、次のシステム・リセットまで FLASH_SECCR レジスタがロックされます。

1 番目のキー = 0x4567 0123

2 番目のキー = 0xCDEF 89AB

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **SECKEY[31:0]** : 不揮発性メモリ、セキュア設定アクセス・アンロック・キー

7.11.4 Flash オプションキーレジスタ (Flash_OPTKEYR)

このレジスタは非セキュアです。これは、セキュアおよび非セキュア・アクセスのどちらでも書込みが可能です。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

Flash_OPTKEYR は書込み専用レジスタです。FLASH_OPTCR レジスタをアンロックするには、次の値を連続でプログラムする必要があります。シーケンスを誤ると、次のシステム・リセットまで FLASH_OPTCR レジスタがロックされます。

1 番目のキー = 0x0819 2A3B

2 番目のキー = 0x4C5D 6E7F

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **OPTKEY[31:0]** : FLASH オプション・バイト制御アクセスアンロックキー

7.11.5 Flash 非セキュア・OBKキー・レジスタ (FLASH_NSObKKEYR)

FLASH_NSObKKEYR は書込み専用レジスタです。FLASH_NSObKCFGR レジスタをアンロックするには、次の値を連続でプログラムする必要があります。シーケンスを誤ると、次のシステム・リセットまで FLASH_NSObKCFGR レジスタがロックされます。

このレジスタは非セキュアです。このレジスタは、特権アクセスによってのみ書込みができます。

1 番目のキー = 0x192A 083B

2 番目のキー = 0x6E7F 4C5D

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSObKKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSObKKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **NSObKKEY[31:0]** : FLASH の非セキュア・オプション・バイト・キー制御アクセス・アンロック・キー

7.11.6 Flash セキュア OBK キー・レジスタ (FLASH_SECObKKEYR)

FLASH_SECObKKEYR は書き込み専用レジスタです。FLASH_SECObKCFGR レジスタをアンロックするには、次の値を連続でプログラムする必要があります。シーケンスを誤ると、次のシステム・リセットまで FLASH_SECObKCFGR レジスタがロックされます。

1 番目のキー = 0x192A 083B

2 番目のキー = 0x6E7F 4C5D

このレジスタはセキュアです。このレジスタは、特権アクセスでのみ書き込むことができます。

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECObKKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECObKKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **SECObKKEY[31:0]** : FLASH のセキュア・オプション・バイト・キー制御アクセス・アンロック・キー

7.11.7 Flash 操作ステータス・レジスタ (FLASH_OPsR)

このレジスタは非セキュアです。このレジスタは、セキュアおよび非セキュア・アクセスのどちらでも読み出すことができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x0018

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CODE_OP[2:0]			Res.	Res.	Res.	Res.	OTP_OP	SYSF_OP	BK_OP	DATA_OP	Res.	ADDR_OP[19:16]			
r	r	r					r	r	r	r		r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_OP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:29 **CODE_OP[2:0]** : Flash メモリ操作コード

- 000 : 前のリセット時に進行中の Flash メモリ操作なし
- 001 : 1 回の書き込み操作に対する割込みあり
- 010 : OBK 代替セクタ消去
- 011 : セクタ消去操作に対する割込みあり
- 100 : バンク消去操作に対する割込みあり
- 101 : 全体消去操作に対する割込みあり
- 110 : オプション変更操作に対する割込みあり
- 111 : OBK スワップ・セクタのリクエスト

ビット 28:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OTP_OP** : OTP 操作に対する割込みあり

OTP 領域（または OBKeys 領域）で進行中の操作に、リセットによる割込みがあったことを示します。

ビット 23 **SYSF_OP** : システム Flash メモリ操作に対する割込み

システム Flash メモリで進行中の操作に、リセットによる割込みがあったことを示します。

ビット 22 **BK_OP** : 割込みがあった操作バンク

どのバンクが操作に関係しているかを示します。

ビット 21 **DATA_OP** : Flash メモリの高サイクル・データ領域での操作に対する割込み

Flash メモリの高サイクル・データ領域が操作に関係していたかどうかを示します。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ADDR_OP[19:0]** : 割込みがあった操作アドレス

7.11.8 Flash オプション制御レジスタ (Flash_OPTCR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アクセス : Flash メモリ操作が進行中ではない場合、ウェイト・ステート無し BSY ビットがセットされていると、書き込みモードでは FLASH_OPTCR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込みもうとすると、BSY ビットがクリアされるまで AHB バスはストールします。

アドレスオフセット : 0x01C

リセット値 : 0xX000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_BANK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPT STRT	OPT LOCK
														rw	rs

ビット 31 **SWAP_BANK** : バンクスワッピングオプション設定ビット

SWAP_BANK はバンク 1 とバンク 2 がスワップするかどうかを制御します。このビットは、リセットか POR の後に限り、FLASH_OPTSR_CUR レジスタの SWAP_BANK ビットとともにロードされます。

0 : バンク 1 とバンク 2 のスワップ無し

1 : バンク 1 とバンク 2 のスワップ有り

ビット 30:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OPTSTRT** : オプション・バイト開始変更オプション設定ビット

OPTSTRT は、オプション・バイトの変更操作をトリガします。ユーザは、OPTLOCK ビットが 0 にクリアされている場合にのみ OPTSTRT をセットできます。これはソフトウェアによってのみセットされ、オプション・バイトの変更が完了したかエラーが起きたとき (PGSERR または OPTCHANGEERR) にクリアされます。これは、BSY ビットと同時にリセットされます。

オプション・バイトの変更操作が完了するまで、ユーザ・アプリケーションはどの FLASH_XXX_PRG Flash インタフェース・レジスタも変更できません。

このビットをセットする前に、ユーザは Flash_XXX_PRG レジスタに必要な値を書き込む必要があります。Flash_XXX_PRG レジスタは、不揮発性メモリでオプション・バイトの変更操作が実行されるまでロックされます。

ビット 0 **OPTLOCK** : FLASH_OPTCR ロックオプション設定ビット

OPTLOCK ビットは、FLASH_OPTCR レジスタとすべての _PRG レジスタをロックします。FLASH_OPTKEYR レジスタに対する正しい書き込みシーケンスにより、このビットがアンロックされます。誤ったシーケンスが実行された場合、あるいは FLASH_OPTKEYR に対するアンロックシーケンスが 2 回実行された場合、このビットは次のシステムリセットまでロックされたままとなります。

OPTLOCK は 1 にプログラムすることでセットすることができます。1 にセットすると、アンロックするには新しいアンロックシーケンスが必要になります。OPTLOCK を 0 から 1 に変更するときに、FLASH_OPTCR レジスタのその他のビットは変更されません。

0 : Flash_OPTCR レジスタをアンロックします。

1 : Flash_OPTCR レジスタをロックします。

7.11.9 Flash 非セキュアステータスレジスタ (Flash_NSSR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x020

リセット値 : 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPTCH ANGE ERR	OBKW ERR	OBK ERR	INC ERR	STRB ERR	PGS ERR	WRP ERR	EOP
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBNE	Res.	WBNE	BSY
												r		r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **OPTCHANGEERR** : オプション・バイト変更エラーフラグ

OPTCHANGEERR フラグは、オプション・バイトの変更操作中にエラーが発生したことを示します。OPTCHANGEERR が 1 にセットされた場合、オプション・バイトの変更操作は正常に完了していません。FLASH_NSCR レジスタの OPTCHANGEERRIE ビットが 1 にセットされている場合、このフラグが立てられると割込みが生成されます。

FLASH_NSCCR レジスタの CLR_OPTCHANGEERR に 1 を書き込むと OPTCHANGEERR はクリアされます。

0 : オプション・バイトの変更エラーは発生していません。

1 : オプション・バイトの変更操作中に 1 つ以上のエラーが発生しました。

注 : FLASH_OPTCR レジスタの OPTSTRT ビットは、OPTCHANGEERR がセットされている間はセットできません。

ビット 22 **OBKWERR** : OBK 書き込みエラーフラグ

OBK ストレージへの書き込みアクセス時にそのアドレスが未使用ではない場合、あるいはスワップ操作中の代替セクタの OBK セクタが未使用ではない場合に OBKWERR フラグが立てられます。

0 : OBK 書き込みエラーは発生していません

1 : OBK 書き込みエラーが発生しました

ビット 21 **OBKERR** : OBK 一般エラー・フラグ

キー・ロケーションへのアクセス中に、SBS からの OBK-HDPL 信号が、そのキー・スロットに関連した HDPL 値と一致しない場合、あるいは書き込みバッファがフルになっているときに、予期せぬ ALT_SECT の変更があった場合に OBKERR フラグが立ちます。

0 : OBK 一般エラーは発生していません

1 : OBK 一般エラーが発生しました

ビット 20 **INCERR** : 不整合エラー・フラグ

不整合エラーが発生した場合、NSINCERR フラグが立てられます。INCERRIE が 1 にセットされていると、割込みが生成されます。FLASH_NSCCR レジスタの CLR_INCERR ビットに 1 を書き込むと NSINCERR はクリアされます。

0 : 不整合エラーは発生していません

1 : 不整合エラーが発生しています

ビット 19 **STRBERR** : ストローブ・エラー・フラグ

ストローブ・エラーが発生した場合（マスタが、書き込みバッファの同一のバイトに数回、書き込みをしようとした場合）、STRBERR フラグが立てられます。STRBERRIE ビットが 1 にセットされていると、割込みが生成されます。FLASH_NSCCR レジスタの CLR_STRBERR ビットに 1 を書き込むと STRBERR はクリアされます。

0 : ストローブ・エラーは発生していません

1 : ストローブ・エラーが発生しました

ビット 18 **PGSERR** : プログラミング・シーケンス・エラー・フラグ

シーケンス・エラーが発生した場合、PGSERR フラグが立てられます。PGSERRIE ビットが 1 にセットされている場合、割込みが生成されます。FLASH_NSCCR レジスタの CLR_PGSERR ビットに 1 を書き込むと PGSERR はクリアされます。

0 : シーケンス・エラーは発生していません

1 : シーケンス・エラーが発生しました

ビット 17 **WRPERR** : 書き込み保護エラー・フラグ

プログラム操作中に保護エラーが発生した場合、WRPERR フラグが立てられます。WRPERRIE が 1 にセットされている場合には、割込みも生成されます。FLASH_NSCCR レジスタの CLR_WRPERR ビットに 1 を書き込むと WRPERR はクリアされます。

0 : 書き込み保護エラーは発生していません

1 : 書き込み保護エラーが発生しました

ビット 16 **EOP** : 演算終了フラグ

操作（プログラム／消去）が終了すると EOP フラグがセットされます。EOPIE が 1 にセットされている場合、割込みが生成されます。新しい操作を開始する前に EOP をリセットする必要はありません。EOP ビットは、FLASH_NSCCR レジスタの CLR_EOP ビットに 1 を書き込むことによりクリアされます。

0 : 操作は完了していません

1 : 操作が完了しました

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **DBNE** : データ・バッファが空ではないことを示すフラグ

Flash インタフェースが専用バッファで 6 bit ECC データを処理しているときに DBNE フラグがセットされます。このビットをソフトウェアによって 0 にすることはできません。バッファが空になると、ハードウェアがこれをリセットします。

0 : データ・バッファは使用していません

1 : データ・バッファを使用しています、ウェイト状態です

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **WBNE** : 書き込みバッファが空ではないことを示すフラグ

WBNE フラグは、新しいデータが書き込みバッファを完了するのを Flash インタフェースが待っている場合にセットされます。この状態では、書き込みバッファは空ではありません。WBNE は、書き込みバッファが完了するか、次に示すイベントのどれかの後に書き込みバッファが空になると、毎回ハードウェアによってリセットされます。

- アプリケーション・ソフトウェアが Flash_NSCR の FW ビットを使用して書き込み操作を強制
- メモリがデータ損失を伴うエラーを検出

このビットはソフトウェアが 0 を書き込むことによって直接リセットすることはできません。リセットするには、上記のいずれかのアクションを実行して、書き込みバッファをクリアするか、欠落しているデータを送信します。

0 : 書き込みバッファが空またはフルです。

1 : 書き込みバッファがデータの書き込み完了を待っています

ビット 0 **BSY** : ビジー・フラグ

BSY フラグは、Flash メモリが操作（書き込み、消去、オプション・バイトの変更、OBK 操作）によりビジーであることを示します。これは、Flash メモリ操作の開始時にセットされ、操作が終了するかエラーが発生するとクリアされます。

0 : 実行中のプログラミング、消去、またはオプション・バイトの変更操作はありません

1 : プログラミング、消去、またはオプション・バイトの変更操作が実行中です

7.11.10 **FLASH セキュア・ステータス・レジスタ (Flash_SECSR)**

このレジスタはセキュアです。これはセキュア・アクセスによってのみ読み出し可能です。非セキュア読み出し／書き込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x024

リセット値 : 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBKW ERR	OBK ERR	INC ERR	STRB ERR	PGS ERR	WRP ERR	EOP
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBNE	Res.	WBNE	BSY
												r		r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **OBKWERR** : OBK 書き込みエラーフラグ

OBK ストレージへの書き込みアクセス時にそのアドレスが未使用ではない場合、あるいはスワップ操作中の代替セクタの OBK セクタが未使用ではない場合に OBKWERR フラグが立てられます。

0 : OBK 書き込みエラーは発生していません

1 : OBK 書き込みエラーが発生しました

ビット 21 **OBKERR** : OBK 一般エラー・フラグ

キー・ロケーションへのアクセス中に、SBS からの OBK-HDPL 信号が、そのキー・スロットに関連した HDPL 値と一致しない場合、あるいは書き込みバッファがフルになっているときに、予期せぬ ALT_SECT の変更があった場合に OBKERR フラグが立ちます。

0 : OBK 一般エラーは発生していません

1 : OBK 一般エラーが発生しました

ビット 20 INCERR : 不整合エラー・フラグ

不整合エラーが発生した場合、INCERR フラグが立てられます。INCERRIE が 1 にセットされていると、割込みが生成されます。FLASH_SECCCR レジスタの CLR_INCERR ビットに 1 を書き込むと INCERR はクリアされます。

0 : 不整合エラーは発生していません

1 : 不整合エラーが発生しました

ビット 19 STRBERR : ストローブ・エラー・フラグ

ストローブ・エラーが発生した場合 (マスタが、書き込みバッファの同一のバイトに数回、書き込みをしようとした場合)、STRBERR フラグが立てられます。STRBERRIE ビットが 1 にセットされていると、割込みが生成されます。FLASH_SECCCR レジスタの CLR_STRBERR ビットに 1 を書き込むと STRBERR はクリアされます。

0 : ストローブ・エラーは発生していません

1 : ストローブ・エラーが発生しました

ビット 18 PGSERR : プログラミング・シーケンス・エラー・フラグ

シーケンス・エラーが発生した場合、PGSERR フラグが立てられます。PGSERRIE ビットが 1 にセットされている場合、割込みが生成されます。FLASH_SECCCR レジスタの CLR_PGSERR ビットに 1 を書き込むと PGSERR はクリアされます。

0 : シーケンス・エラーは発生していません

1 : シーケンス・エラーが発生しました

ビット 17 WRPERR : 書き込み保護エラー・フラグ

プログラム操作中に保護エラーが発生した場合、WRPERR フラグが立てられます。WRPERRIE が 1 にセットされている場合には、割込みも生成されます。FLASH_SECCCR レジスタの CLR_WRPERR ビットに 1 を書き込むと WRPERR はクリアされます。

0 : 書き込み保護エラーは発生していません

1 : 書き込み保護エラーが発生しました

ビット 16 EOP : 演算終了フラグ

操作 (プログラム/消去) が終了すると EOP フラグがセットされます。EOPIE が 1 にセットされている場合、割込みが生成されます。新しい操作を開始する前に EOP をリセットする必要はありません。EOP ビットは、FLASH_SECCCR レジスタの CLR_EOP ビットに 1 を書き込むことによりクリアされます。

0 : 操作は完了していません

1 : 操作が完了しました

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 DBNE : データ・バッファが空ではないことを示すフラグ

メモリ・インタフェースが専用バッファで 6 bit ECC データを処理しているときに DBNE フラグがセットされます。このビットをソフトウェアによって 0 にすることはできません。バッファが空になると、ハードウェアがこれをリセットします。

0 : データ・バッファは使用していません

1 : データ・バッファを使用しています、ウェイト状態です

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **WBNE** : 書込みバッファが空ではないことを示すフラグ

WBNE フラグは、新しいデータが書込みバッファを完了するのを Flash インタフェースが待っている場合にセットされます。この状態では、書込みバッファは空ではありません。WBNE は、書込みバッファが完了するか、次に示すイベントのどれかの後に書込みバッファが空になると、毎回ハードウェアによってリセットされます。

- アプリケーション・ソフトウェアが FLASH_SECCR の FW ビットを使用して書込み操作を強制
- Flash インタフェースがデータ損失を伴うエラーを検出

このビットはソフトウェアで直接 0 を書き込んでリセットすることはできません。リセットするには、上記のいずれかのアクションを実行して、書込みバッファをクリアするか、欠落しているデータを送信します。

0 : 書込みバッファが空またはフルです。

1 : 書込みバッファがデータの書込み完了を待っています

ビット 0 **BSY** : ビジー・フラグ

BSY フラグは、Flash メモリがビジーである（書込み、消去、オプション・バイトの変更、OBK 操作）ことを示します。これは、Flash メモリ操作の開始時にセットされ、操作が終了するかエラーが発生するとクリアされます。

0 : 実行中のプログラミング、消去、またはオプション・バイトの変更操作はありません

1 : プログラミング、消去、またはオプション・バイトの変更操作が実行中です

7.11.11 Flash 非セキュア制御レジスタ (Flash_NSCR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アクセス : Flash メモリ操作が進行中ではない場合、ウェイト・ステート無し BSY ビットがセットされていると、書込みモードでは FLASH_NSCR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込もうとすると、BSY ビットがクリアされるまで AHB バスはストールします。

アドレスオフセット : 0x028

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPT CHANGE ERRIE	OBKW ERRIE	OBK ERRIE	INC ERRIE	STRB ERRIE	PGS ERRIE	WRP ERRIE	EOPIE
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MER	Res.	Res.	SNB[6:0]							STRT	FW	BER	SER	PG	LOCK
rw			rw	rw	rw	rw	rw	rw	rw	rs	rw	rw	rw	rw	rs

ビット 31 **BKSEL** : バンク・セクタ・ビット

BKSEL は、LOCK が 0 にクリアされたときのみプログラム可能です。このビットは物理バンクを選択し、SWAP_BANK 設定は無視されます。

0 : バンク消去／セクタ消去／割込みイネーブルにバンク 1 が選択されています。

1 : BER / SER にバンク 2 が選択されています。

ビット 30:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 OPTCHANGEERRIE : オプション・バイト変更エラー割込みイネーブルビット

このビットは、オプション・バイトの変更中にエラーが発生した場合、割込みを生成する必要があるかどうかを制御します。これは、LOCK ビットが 0 にクリアされている場合にのみプログラムすることができます。

0 : オプション・バイトの変更中にエラーが発生したときに、割込みを生成しません。

1 : オプション・バイトの変更中にエラーが発生したときに、割込みを生成します。

ビット 22 OBKWERRIE : OBK 書込みエラー割込みイネーブル・ビット

OBKWERRIE は、OBK 特有の書込みエラーに対して割込みの生成を可能します。このビットは LOCK ビットが 0 にクリアされている場合にのみプログラムすることができます。

0 : OBK 書込みエラー時に割込みを生成しません

1 : OBK 書込みエラー時に割込みを生成します

ビット 21 OBKERRIE : OBK 一般エラー割込みイネーブル・ビット

OBKERRIE は、OBK 特有のアクセス・エラーに対して割込みの生成を可能します。このビットは LOCK ビットが 0 にクリアされている場合にのみプログラムすることができます。

0 : OBK 一般アクセス・エラー時に割込みを生成しません

1 : OBK 一般アクセス・エラー時に割込みを生成します

ビット 20 INCERRIE : 不整合エラー割込みイネーブル・ビット

INCERRIE ビットを 1 にセットすると、書込み操作中に不整合エラーが発生したときに割込みが生成されます。LOCK が 0 にクリアされている場合のみ INCERRIE をプログラムすることができます。

0 : 不整合エラーが発生したときに割込みを生成しません

1 : 不整合エラーが発生したときに割込みを生成します

ビット 19 STRBERRIE : ストローブ・エラー割込みイネーブル・ビット

STRBERRIE ビットを 1 にセットすると、書込み操作中にストローブ・エラーが発生したとき（マスタが書込みバッファの同一のバイトに数回プログラムを行う）に割込みが生成されます。LOCK が 0 にクリアされている場合のみ STRBERRIE をプログラムすることができます。

0 : ストローブ・エラーが発生したときに割込みを生成しません

1 : ストローブ・エラーが発生したときに割込みを生成します

ビット 18 PGSERRIE : プログラミング・シーケンス・エラー割込みイネーブル・ビット

このビットを 1 にセットすると、プログラム操作中にシーケンス・エラーが発生したときに、割込みが生成されます。LOCK が 0 にクリアされている場合のみ PGSERRIE をプログラムすることができます。

0 : シーケンス・エラーが発生したときに割込みを生成しません

1 : シーケンス・エラーが発生したときに割込みを生成します

ビット 17 WRPERRIE : 書込み保護エラー割込みイネーブル・ビット

このビットを 1 にセットすると、プログラム操作中に保護エラーが発生したときに割込みが生成されます。LOCK が 0 にクリアされている場合のみ WRPERRIE をプログラムすることができます。

0 : 保護エラーが発生したときに割込みを生成しません

1 : 保護エラーが発生したときに割込みを生成します

ビット 16 EOPIE : 操作終了割込み制御ビット

EOPIE ビットを 1 にセットすると、プログラムまたは消去操作の終了時に割込み生成が可能になります。LOCK が 0 にクリアされている場合のみ EOPIE をプログラムすることができます。

0 : 操作の終了時に割込みを生成しません

1 : 操作の終了時に割込みを有効にします

ビット 15 MER : 全体消去リクエスト

MER ビットを 1 にセットすると、全体消去操作を要求します（ユーザ Flash メモリのみ）。LOCK が 0 にクリアされている場合のみ MER をプログラムすることができます。

BER と SER が両方セットされていると、PGSERR が立ちます。

0 : 全体消去がリクエストされていません

1 : 全体消去がリクエストされています

全体消去が要求されているのに一部のセクタが保護されている場合には、エラーがトリガされます。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:6 **SNB[6:0]** : セクタ消去選択番号

これらのビットは、消去操作の対象のセクタを選択するために使用します（そうしないと使用されません）。LOCK が 0 にクリアされている場合のみ SNB をプログラムすることができます。

0x00 : セクタ 0 が選択されます

0x01 : セクタ 1 が選択されます

...

0x7F : セクタ 127 が選択されます

ビット 5 **STRT** : 消去開始制御ビット

STRT ビットは、セクタ消去またはバンク消去操作を開始するために使用します。LOCK が 0 にクリアされている場合のみ STRT をプログラムすることができます。

STRT は操作の終わり、あるいはエラー発生時にリセットされます。ソフトウェアによってリセットすることはできません。

ビット 4 **FW** : 強制書込み制御ビット

FW は書込みバッファがフルでなくても書込み操作を強制します。この場合、書き込まれていないすべてのビットはハードウェアによって 1 にセットされます。LOCK が 0 にクリアされている場合のみ FW をプログラムすることができます。

該当する操作を確認すると、メモリは FW をリセットします。

注 : 強制書込み操作を使用すると、後にアプリケーションによって、欠落しているビットが 1 とは異なる値で更新されることが防止されます（恒久的 ECC エラーが発生する可能性があり得るため）。

書込みの強制は、書込みバッファが空ではなく、非セキュア・アクセスでフィルされている場合にのみ有効です（特に強制書込み操作が連続して行われている場合、FW ではいくつかの書込み操作を開始しません）。

書込みバッファは 1 つしかないため、FW はバンク 1 またはバンク 2 での強制書込みが可能です。

ビット 3 **BER** : 消去リクエスト

BER ビットを 1 にセットすると、バンク消去操作を要求します（ユーザ Flash メモリのみ）。LOCK が 0 にクリアされている場合のみ BER をプログラムすることができます。

MER と SER もセットされていると、PGSERR が立ちます。

0 : バンク消去がリクエストされていません

1 : バンク消去がリクエストされています

注 : 書込み保護エラーは、バンク消去が必要であり、一部のセクタが保護されている場合にトリガされます。

ビット 2 **SER** : セクタ消去リクエスト

SER ビットを 1 にセットすると、セクタ消去が要求されます。LOCK が 0 にクリアされている場合のみ SER をプログラムすることができます。

MER と SER もセットされていると、PGSERR が立ちます。

0 : セクタ消去がリクエストされていません

1 : セクタ消去がリクエストされています

ビット 1 **PG** : プログラミング制御ビット

LOCK が 0 にクリアされている場合のみ PG をプログラムすることができます。

PG はバンク 1 とバンク 2 のプログラミングを可能にします。

0 : プログラミングは無効になっています

1 : プログラミングが有効になっています

ビット 0 **LOCK** : 設定ロック・ビット

このビットで FLASH_NSCR レジスタをロックします。FLASH_NSKEYR レジスタに対する正しい書込みシーケンスにより、このビットがアンロックされます。誤ったシーケンスが実行された場合、あるいは FLASH_NSKEYR に対するアンロック・シーケンスが 2 回実行された場合、このビットは次のシステム・リセットまでロックされたままとなります。

LOCK は 1 にプログラムすることでセットすることができます。1 にセットすると、アンロックするには新しいアンロックシーケンスが必要になります。LOCK を 0 から 1 に変更しても、FLASH_NSCR レジスタのその他のビットは変更されません。

0 : FLASH_NSCR レジスタがアンロックされています

1 : FLASH_NSCR レジスタがロックされています

7.11.12 Flash セキュア制御レジスタ (Flash_SECCR)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読みし／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アクセス : Flash メモリ操作が進行中ではない場合、ウェイト・ステート無し BSY ビットがセットされていると、書込みモードでは FLASH_SECCR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込もうとすると、BSY ビットがクリアされるまで AHB バスはストールします。

アドレスオフセット : 0x02C

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKSEL	Res.	INV	Res.	Res.	Res.	Res.	Res.	Res.	OBKW ERRIE	OBK ERRIE	INC ERRIE	STRB ERRIE	PGS ERRIE	WRP ERRIE	EOPIE
rw		rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MER	Res.	Res.	SNB[6:0]							STRT	FW	BER	SER	PG	LOCK
rw			rw	rw	rw	rw	rw	rw	rw	rs	rw	rw	rw	rw	rs

ビット 31 **BKSEL** : バンク・セクタ・ビット

BKSEL は、LOCK が 0 にクリアされたときのみプログラム可能です。このビットは物理バンクを選択し、SWAP_BANK 設定は無視されます。

0 : バンク消去／セクタ消去／割込みイネーブルにバンク 1 が選択されています。

1 : BER / SER にバンク 2 が選択されています。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **INV** : Flash メモリ・セキュリティ状態反転。

このビットは、Flash メモリのセキュリティ状態を反転します。

ビット 28:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **OBKWERRIE** : OBK 書込みエラー割込みイネーブル・ビット

OBKWERRIE は、OBK 特有の書込みエラーに対して割込みの生成を可能します。LOCK が 0 にクリアされている場合のみ OBKWERRIE をプログラムすることができます。

0 : OBK 書込みエラー時に割込みを生成しません

1 : OBK 書込みエラー時に割込みを生成します

ビット 21 **OBKERRIE** : OBK 一般エラー割込みイネーブル・ビット

OBKERRIE は、OBK 特有のアクセス・エラーに対して割込みの生成を可能します。LOCK が 0 にクリアされている場合のみ OBKERRIE をプログラムすることができます。

- 0 : OBK 一般アクセス・エラー時に割込みを生成しません
- 1 : OBK 一般アクセス・エラー時に割込みを生成します

ビット 20 **INCERRIE** : 不整合エラー割込みイネーブル・ビット

INCERRIE ビットを 1 にセットすると、書き込み操作中に不整合エラーが発生したときに割込みが生成されます。LOCK が 0 にクリアされている場合のみ INCERRIE をプログラムすることができます。

- 0 : 不整合エラーが発生したときに割込みを生成しません
- 1 : 不整合エラーが発生したときに割込みを生成します

ビット 19 **STRBERRIE** : ストローブ・エラー割込みイネーブル・ビット

STRBERRIE ビットを 1 にセットすると、書き込み操作中にストローブ・エラーが発生したとき（マスタが書き込みバッファの同一のバイトに数回プログラムを行う）に割込みが生成されます。LOCK が 0 にクリアされている場合のみ STRBERRIE をプログラムすることができます。

- 0 : ストローブ・エラーが発生したときに割込みを生成しません
- 1 : ストローブ・エラーが発生したときに割込みを生成します

ビット 18 **PGSERRIE** : プログラミング・シーケンス・エラー割込みイネーブル・ビット

PGSERRIE ビットを 1 にセットすると、プログラム操作中にシーケンス・エラーが発生したときに割込みが生成されます。LOCK が 0 にクリアされている場合のみ PGSERRIE をプログラムすることができます。

- 0 : シーケンス・エラーが発生したときに割込みを生成しません
- 1 : シーケンス・エラーが発生したときに割込みを生成します

ビット 17 **WRPERRIE** : 書き込み保護エラー割込みイネーブル・ビット

WRPERRIE ビットを 1 にセットすると、プログラム操作中に保護エラーが発生したときに割込みが生成されます。LOCK が 0 にクリアされている場合のみ WRPERRIE をプログラムすることができます。

- 0 : 保護エラーが発生したときに割込みを生成しません
- 1 : 保護エラーが発生したときに割込みを生成します

ビット 16 **EOPIE** : 操作終了割込み制御ビット

EOPIE ビットを 1 にセットすると、プログラム／消去操作の終了時に割込み生成が可能になります。LOCK が 0 にクリアされている場合のみ EOPIE をプログラムすることができます。

- 0 : 操作の終了時に割込みを生成しません
- 1 : 操作の終了時に割込みを有効にします

ビット 15 **MER** : 全体消去リクエスト

MER ビットを 1 にセットすると、全体消去操作を要求します（ユーザ Flash メモリのみ）。LOCK が 0 にクリアされている場合のみ MER をプログラムすることができます。

BER または SER が両方セットされていると、PGSERR が立ちます。

- 0 : 全体消去がリクエストされていません
- 1 : 全体消去がリクエストされています

全体消去が要求されているのに一部のセクタが保護されている場合には、エラーがトリガされます。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:6 **SNB[6:0]** : セクタ消去選択番号

これらのビットは、消去操作の対象のセクタを選択するために使用します（そうしないと使用されません）。LOCK が 0 にクリアされている場合のみ SNB をプログラムすることができます。

- 0x00 : セクタ 0 が選択されます
- 0x01 : セクタ 1 が選択されます
- ...
- 0x7F : セクタ 127 が選択されます

ビット 5 STRT : 消去開始制御ビット

STRT ビットは、セクタ消去またはバンク消去操作を開始するために使用します。LOCK が 0 にクリアされている場合のみ STRT をプログラムすることができます。

STRT は操作の終わり、あるいはエラー発生時にリセットされます。ソフトウェアによってリセットすることはできません。

ビット 4 FW : 強制書き込み制御ビット

FW は書き込みバッファがフルでなくても書き込み操作を強制します。この場合、書き込まれていないすべてのビットはハードウェアによって 1 にセットされます。LOCK が 0 にクリアされている場合のみ FW をプログラムすることができます。

該当する操作を確認すると、メモリは FW をリセットします。

注： 強制書き込み操作を使用すると、後にアプリケーションによって、欠落しているビットが 1 とは異なる値で更新されることが防止されます (恒久的 ECC エラーが発生する可能性があり得るため)。

書き込みの強制は、書き込みバッファが空ではなく、セキュア・アクセスでフィルされている場合にのみ有効です (特に強制書き込み操作が連続して行われている場合、FW ではいくつかの書き込み操作を開始しません)。

書き込みバッファは 1 つしかないため、FW はバンク 1 またはバンク 2 での強制書き込みが可能です。

ビット 3 BER : 消去リクエスト

BER ビットを 1 にセットすると、バンク消去操作を要求します (ユーザ Flash メモリのみ)。LOCK が 0 にクリアされている場合のみ BER をプログラムすることができます。

MER と SER もセットされていると、PGSERR が立ちます。

0 : バンク消去がリクエストされていません

1 : バンク消去がリクエストされています

注： 書き込み保護エラーは、バンク消去が必要であり、一部のセクタが保護されている場合にトリガされます。

ビット 2 SER : セクタ消去リクエスト

SER ビットを 1 にセットすると、セクタ消去が要求されます。LOCK が 0 にクリアされている場合のみ SER をプログラムすることができます。

BER および MER、もセットされていると、PGSERR が立ちます。

0 : セクタ消去がリクエストされていません

1 : セクタ消去がリクエストされています

ビット 1 PG : プログラミング制御ビット

LOCK が 0 にクリアされている場合のみ PG をプログラムすることができます。

PG はバンク 1 とバンク 2 のプログラミングを可能にします。

0 : プログラミングは無効になっています

1 : プログラミングが有効になっています

ビット 0 LOCK : 設定ロック・ビット

このビットは FLASH_SECCR レジスタをロックします。FLASH_SECKEYR レジスタに対する正しい書き込みシーケンスにより、このビットがアンロックされます。誤ったシーケンスが実行された場合、あるいは FLASH_NSKEYR に対するアンロック・シーケンスが 2 回実行された場合、このビットは次のシステム・リセットまでロックされたままとなります。

LOCK は 1 にプログラムすることでセットすることができます。1 にセットすると、アンロックするには新しいアンロックシーケンスが必要になります。LOCK を 0 から 1 に変更しても、FLASH_SECCR レジスタのその他のビットは変更されません。

0 : FLASH_SECCR レジスタはアンロックされています

1 : FLASH_SECCR レジスタはロックされています

7.11.13 FLASH 非セキュア・クリア制御レジスタ (FLASH_NSCCR)

このレジスタは非セキュアです。これは、セキュアおよび非セキュア・アクセスのどちらでも書込みが可能です。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR_OPTCHANGEERR	CLR_OBKWERR	CLR_OBKERR	CLR_INCERR	CLR_STRBERR	CLR_PGSERR	CLR_WRPERR	CLR_EOP
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **CLR_OPTCHANGEERR** : このビットに書き込むことで FLASH_NSSR のフラグに対応するフラグをクリアします。

ビット 22 **CLR_OBKWERR** : OBKWERR フラグ・クリア・ビット。

このビットを 1 にセットすると、FLASH_NSSR レジスタの OBKWERR フラグが 0 にリセットされます。

ビット 21 **CLR_OBKERR** : OBKERR フラグ・クリア・ビット。

このビットを 1 にセットすると、FLASH_NSSR レジスタの OBKERR フラグが 0 にリセットされます。

ビット 20 **CLR_INCERR** : INCERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_NSSR レジスタの INCERR フラグが 0 にリセットされます。

ビット 19 **CLR_STRBERR** : STRBERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_NSSR レジスタの STRBERR フラグが 0 にリセットされます。

ビット 18 **CLR_PGSERR** : PGSERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_NSSR レジスタの PGSERR フラグが 0 にリセットされます。

ビット 17 **CLR_WRPERR** : WRPERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_NSSR レジスタの WRPERR フラグが 0 にリセットされます。

ビット 16 **CLR_EOP** : EOP フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_NSSR レジスタの EOP フラグが 0 にリセットされます。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

7.11.14 FLASH セキュア クリア 制御レジスタ (FLASH_SECCCR)

このレジスタはセキュアです。これはセキュア・アクセスによってのみ書込み可能です。非セキュア読出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR_ OBK WERR	CLR_ OBK ERR	CLR_ INC ERR	CLR_ STRB ERR	CLR_ PGS ERR	CLR_ WRP ERR	CLR_ EOP
									w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **CLR_OBKWERR** : OBKWERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの OBKWERR フラグが 0 にリセットされます。

ビット 21 **CLR_OBKERR** : OBKWERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの OBKWERR フラグが 0 にリセットされます。

ビット 20 **CLR_INCERR** : INCERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの INCERR フラグが 0 にリセットされます。

ビット 19 **CLR_STRBERR** : STRBERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの STRBERR フラグが 0 にリセットされます。

ビット 18 **CLR_PGSERR** : PGSERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの PGSERR フラグが 0 にリセットされます。

ビット 17 **CLR_WRPERR** : WRPERR フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの WRPERR フラグが 0 にリセットされます。

ビット 16 **CLR_EOP** : EOP フラグ・クリア・ビット

このビットを 1 にセットすると、FLASH_SECSR レジスタの EOP フラグが 0 にリセットされます。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

7.11.15 Flash 特権設定レジスタ (FLASH_PRIVCFGR)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

このレジスタは、特権および非特権アクセスのどちらでも読み出すことができます。NSPRIV は非セキュア・ビットです。SPRIV はセキュア・ビットです。特権モードでのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSPRIV	SPRIV
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **NSPRIV** : 非セキュア・レジスタの特権属性

0 : 非セキュア・レジスタへのアクセスは常に許されます

1 : 非セキュア・レジスタへのアクセスは、非特権アクセスの場合、拒否されます

ビット 0 **SPRIV** : セキュア・レジスタの特権属性

0 : セキュア・レジスタへのアクセスは常に許されます

1 : セキュア・レジスタへのアクセスは、非特権アクセスの場合、拒否されます

7.11.16 FLASH 非セキュア・OBK設定・レジスタ (FLASH_NSOBKCFGR)

TZ_STATE = 0xC3 のときのみレジスタが利用可能です。このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは特権コードでのみアクセスすることができます。

BSY ビットがセットされていると、書込みモードでは FLASH_NSOBKCFGR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込もうとすると、BSY ビットがクリアされるまで AHB バスはストールします。

アドレスオフセット : 0x040

リセット値 : 0x01FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWAP_OFFSET[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALT_SECT_ERASE	ALT_SECT	SWAP_SECT_REQ	LOCK
												rw	rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:16 **SWAP_OFFSET[8:0]** : キー・インデックス (オフセット /16 bits) は次のスワップをポイントしています。

0x00 は SWAP 操作中、現在の OBK から代替 OBK セクタへの OBK データのコピーが行われないことを意味します。

0x01 は最初の OBK データ (128 bit) のみが現在の OBK から代替 OBK セクタにコピーされることを意味します。

0x02 は最初の 2 つの OBK データが現在の OBK から代替 OBK セクタにコピーされることを意味します。

...

0x1FF はすべての OBK データ (511) がコピーされることを意味します。

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ALT_SECT_ERASE** : 代替セクタ消去ビット

0 : OBK セクタには影響を与えません

1 : 代替 OBK セクタを消去します

ALT_SECT ビットがセットされている場合、このビットを使って OBK 代替セクタの消去コマンドを生成します。これはソフトウェアによってのみセットされ、OBK スワップ操作が完了したかエラーが発生したとき (PGSERR) にクリアされます。これは、BUSY ビットと同時にリセットされます。

ビット 2 **ALT_SECT** : 代替セクタ・ビット

0 : 現在の OBK セクタが、アクセスのために OBK アドレス範囲にマッピングされています

1 : 代替 OBK セクタが、アクセスのために OBK アドレス範囲にマッピングされています。

このビットは書き込みバッファをフィルしている間に変更してはなりません。変更するとエラー (OBKERR) が生成されます。

ビット 1 **SWAP_SECT_REQ** : OBK スワップ・セクタのリクエスト・ビット

0 : スワップはリクエストされていません

1 : セクタ・スワップを開始します

セットされている場合、代替セクタで更新されていないすべての OBK が、現在のセクタから代替セクタにコピーされます。

SWAP_OFFSET 値は、OBK-HDPL \neq 0 でスワップを開始するために、ある最低値でなくてはなりません。OBK-HDPL = 1 の最低値は 16、OBK-HDPL = 2 の最低値は 144、OBK-HDPL = 3 の最低値は 192 です。

ビット 0 **LOCK** : OBKCFGR ロック・オプション設定ビット

このビットは、FLASH_NSObKCFGR レジスタをロックします。FLASH_NSObKKEYR レジスタに対する正しい書き込みシーケンスにより、このビットがアンロックされます。誤ったシーケンスが実行された場合、あるいは FLASH_NSObKKEYR に対するアンロック・シーケンスが 2 回実行された場合、このビットは次のシステム・リセットまでロックされたままとなります。LOCK は 1 にプログラムすることでセットすることができます。1 にセットすると、アンロックするには新しいアンロックシーケンスが必要になります。LOCK を 0 から 1 に変更しても、FLASH_NSCR レジスタのその他のビットは変更されません。

0 : FLASH_NSObKCFGR レジスタをアンロックします

1 : FLASH_NSObKCFGR レジスタをロックします

7.11.17 FLASH セキュア・OBK設定・レジスタ (FLASH_SECOBKCFGR)

このレジスタはセキュア特権モードでのみアクセス可能です。BSY ビットがセットされていると、書込みモードでは FLASH_SECOBKCFGR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込もうとすると、BSY ビットがクリアされるまで AHB バスはストールします。

アドレスオフセット : 0x044

リセット値 : 0x01FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWAP_OFFSET[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALT_SECT_ERASE	ALT_SECT	SWAP_SECT_REQ	LOCK
												rw	rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:16 **SWAP_OFFSET[8:0]** : キー・インデックス (オフセット /16 bits) は次のスワップをポイントしています。

0x0 : SWAP 操作中、現在の OBK から代替 OBK セクタへの OBK データのコピーは行われません。

0x1 : 最初の OBK データ (128 bit) のみが現在の OBK から代替 OBK セクタにコピーされます。

0x2 : 最初の 2 つの OBK データが現在の OBK から代替 OBK セクタにコピーされます。

...

0x1FF : 最初の 511 の OBK データが現在の OBK から代替 OBK セクタにコピーされます

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ALT_SECT_ERASE** : 代替セクタ消去ビット

0 : OBK セクタには影響を与えません

1 : 代替 OBK セクタを消去します

ALT_SECT ビットがセットされている場合、このビットを使って OBK 代替セクタの消去コマンドを生成します。これはソフトウェアによってのみセットされ、OBK スワップ操作が完了したかエラーが発生したとき (PGSEERR) にクリアされます。これは、BUSY ビットと同時にリセットされます。

ビット 2 **ALT_SECT** : 代替セクタ・ビット

0 : 現在の OBK セクタが、アクセスのために OBK アドレス範囲にマッピングされています

1 : 代替 OBK セクタが、アクセスのために OBK アドレス範囲にマッピングされています。

このビットは書込みバッファをフィルしている間に変更してはなりません。変更するとエラーが生成されます。

ビット 1 **SWAP_SECT_REQ** : OBK スワップ・セクタのリクエスト・ビット

0 : スワップはリクエストされていません

1 : セクタ・スワップを開始します

セットされている場合、代替セクタで更新されていないすべての OBK が、現在のセクタから代替セクタにコピーされます。

SWAP_OFFSET 値は、OBK-HDPL ≠ 0 でスワップを開始するために、ある最低値でなくてはなりません。OBK-HDPL = 1 の最低値は 16、OBK-HDPL = 2 の最低値は 144、OBK-HDPL = 3 の最低値は 192 です。

ビット 0 **LOCK** : OBKCFGR ロック・オプション設定ビット

このビットは、FLASH_OBKCFGR レジスタをロックします。FLASH_SECOBKKEYR レジスタに対する正しい書き込みシーケンスにより、このビットがアンロックされます。誤ったシーケンスが実行された場合、あるいは FLASH_SECOBKKEYR に対するアンロック・シーケンスが 2 回実行された場合、このビットは次のシステム・リセットまでロックされたままとなります。LOCK は 1 にプログラムすることでセットすることができます。1 にセットすると、アンロックするには新しいアンロックシーケンスが必要になります。LOCK を 0 から 1 に変更しても、FLASH_NSCR レジスタのその他のビットは変更されません。

0 : FLASH_OBKCFGR レジスタをアンロックします

1 : FLASH_OBKCFGR レジスタをロックします

7.11.18 **FLASH HDP 拡張レジスタ (FLASH_HDPEXTR)**

このレジスタのすべてのビットは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

レジスタは HDPL<=2 でのみ変更可能です。

このレジスタの値はデクリメントできません。現在の値より低い値を書き込もうとしても無視されます。

アドレスオフセット : 0x048

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_EXT [6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_EXT [6:0]						
									rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HDP2_EXT[6:0]** : バンク 2 の 8 KB セクタの HDP 領域拡張。拡張は HDP2_END セクタの後（このセクタを含む）に追加されます。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **HDP1_EXT[6:0]** : バンク 1 の 8 KB セクタの HDP 領域拡張。拡張は HDP1_END セクタの後（このセクタを含む）に追加されます。

7.11.19 **FLASH オプションステータスレジスタ (FLASH_OPTSR_CUR)**

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

デフォルト値 : 0x0030 5CD8 (OBL 中にダブル ECC が発行された場合の値)

この読出し専用レジスタは、対応するオプションビットの現在値を反映します。

アドレスオフセット : 0x050

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_BANK	Res.	BOOT_UBE[7:0]								IWDG_STDBY	IWDG_STOP	Res.	Res.	IO_VDDIO2_HSLV	IO_VDD_HSLV
r		r	r	r	r	r	r	r	r	r	r			r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRODUCT_STATE[7:0]								NRST_STDBY	NRST_STOP	Res.	WWDG_SW	IWDG_SW	BORH_EN	BOR_LEV[1:0]	
r	r	r	r	r	r	r	r	r	r		r	r	r	r	r

ビット 31 **SWAP_BANK** : バンクスワッピングオプションステータスビット

SWAP_BANK はバンク 1 とバンク 2 がスワップされているかどうかを反映します。
 SWAP_BANK はリセット後に FLASH_OPTCR の SWAP_BANK にロードされます。
 0 : バンク 1 とバンク 2 のスワップ無し
 1 : バンク 1 とバンク 2 のスワップ有り

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:22 **BOOT_UBE[7:0]** : 暗号化が有効になっているデバイスでのみ使用可能です。
 唯一のブート・エントリ制御で、セキュア・ブートに ST か OEM iRoT のいずれかを選択します。
 0xC3 : ST-iRoT (システム Flash) を選択
 0xB4 : OEM-iRoT (ユーザ Flash) を選択。Open PRODUCT_STATE でこの値がブートローダを選択します。デフォルト値。

ビット 21 **IWDG_STDBY** : IWDG STANDBY モード凍結オプションステータスビット

セットされると、独立型ウォッチドッグ IWDG はシステム STANDBY モードで凍結します。
 0 : 独立型ウォッチドッグが STANDBY モードで凍結しています。
 1 : 独立型ウォッチドッグが STANDBY モードで動作し続けています。

ビット 20 **IWDG_STOP** : IWDG STOP モード凍結オプションステータスビット

セットされると、独立型ウォッチドッグ IWDG はシステム STOP モードになります。
 0 : 独立型ウォッチドッグがシステム STOP モードで凍結しています。
 1 : 独立型ウォッチドッグがシステム STOP モードで動作し続けています。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **IO_VDDIO2_HSLV** : 低い V_{DDIO2} 電圧設定ビットでのハイスピード IO。

このビットは V_{DDIO2} が 2.7 V 未満の時のみセットできます。

0 : 低い V_{DDIO2} 電圧でのハイスピード IO 機能は無効です (V_{DDIO2} は 2.7 V を超えることができます)
 1 : 低い V_{DDIO2} 電圧でのハイスピード IO 機能は有効です (V_{DDIO2} は 2.7 V 未満にとどまります)

ビット 16 **IO_VDD_HSLV** : 低い V_{DD} 電圧設定ビットでのハイスピード IO。

このビットは V_{DD} が 2.7 V 未満の時のみセットできます。

0 : 低い V_{DD} 電圧でのハイスピード IO 機能は無効です (V_{DD} は 2.7 V を超えることができます)
 1 : 低い V_{DD} 電圧でのハイスピード IO 機能は有効です (V_{DD} は 2.7 V 未満にとどまります)

ビット 15:8 **PRODUCT_STATE[7:0]** : 寿命状態コード (ハミング 8,4 に基づく) 詳しくは [セクション 7.6.11 : 製品状態の遷移](#) をご覧ください。

ビット 7 **NRST_STDBY** : コアドメイン STANDBY 移行リセットオプションステータスビット

0 : コア・ドメインの STANDBY モードに移行するときにリセットを生成します。
 1 : コアドメインの STANDBY モードに移行するときにリセットを生成しません。

ビット 6 **NRST_STOP** : コア・ドメイン STOP 移行リセットオプション・ステータス・ビット

0 : コア・ドメインの STOP モードに移行するときにリセットを生成します。
 1 : コア・ドメインの STOP モードに移行するときにリセットを生成しません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **WWDG_SW** : WWDG 制御モード・オプション・ステータス・ビット
 0 : WWDG ウォッチドッグはハードウェアによって制御されます。
 1 : WWDG ウォッチドッグはソフトウェアによって制御されます。

ビット 3 **IWDG_SW** : IWDG 制御モード・オプション・ステータス・ビット
 0 : IWDG ウォッチドッグはハードウェアによって制御されます。
 1 : IWDG ウォッチドッグはソフトウェアによって制御されます。

ビット 2 **BORH_EN** : ブラウンアウト・ハイ有効
 0 : 無効
 1 : 有効

ビット 1:0 **BOR_LEV[1:0]** : ブラウンアウトレベルオプションステータスビット
 これらのビットは、システムリセットを生成するパワーレベルを反映します。
 00 または 11 : BOR レベル 1、閾値レベルは低 (約 2.1V) です。
 01 : BOR レベル 2、閾値レベルは中 (約 2.4V) です。
 10 : BOR レベル 3、閾値レベルは高 (約 2.7V) です。

7.11.20 FLASH オプション・ステータス・レジスタ (FLASH_OPTSR_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。リセット後の値は、対応するオプションビットの現在値を反映します。

アドレスオフセット : 0x054

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_BANK	Res.	BOOT_UBE[7:0]								IWDG_STDBY	IWDG_STOP	Res.	Res.	IO_VD_DIO2_HSLV	IO_VD_D_HSLV
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRODUCT_STATE[7:0]								NRST_STDBY	NRST_STOP	Res.	WWDG_SW	IWDG_SW	BORH_EN	BOR_LEV[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31 **SWAP_BANK** : バンクスワッピングオプション設定ビット

SWAP_BANK オプション・ビットはバンク 1 およびバンク 2 がスワップされるかどうかの設定に使用されます。このビットは、リセットの後、FLASH_OPTSR_CUR レジスタの SWAP_BANK ビットとともにロードされます。

0 : バンク 1 とバンク 2 のスワップ無し
 1 : バンク 1 とバンク 2 のスワップ有り

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:22 **BOOT_UBE[7:0]** : 暗号化が有効になっているデバイスでのみ使用可能です。

唯一のブート・エントリ制御で、セキュア・ブートに ST か OEM iROT のいずれかを選択します。

0xC3 : ST-iRoT (システム Flash) を選択

0xB4 : OEM-iRoT (ユーザ Flash) を選択。

Open PRODUCT_STATE でこの値がブートローダを選択します。デフォルト値。

ビット 21 **IWDG_STDBY** : IWDG STANDBY モード凍結オプションステータスビット

セットされると、独立型ウォッチドッグ IWDG はシステム STANDBY モードで凍結します。

0 : 独立型ウォッチドッグが STANDBY モードで凍結しています。

1 : 独立型ウォッチドッグが STANDBY モードで動作し続けています。

ビット 20 **IWDG_STOP** : IWDG STOP モード凍結オプションステータスビット

セットされると、独立型ウォッチドッグ IWDG はシステム STOP モードになります。

0 : 独立型ウォッチドッグがシステム STOP モードで凍結しています。

1 : 独立型ウォッチドッグがシステム STOP モードで動作し続けています。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **IO_VDDIO2_HSLV** : 低い VDDIO2 電圧設定ビットでのハイスピード IO。

このビットは VDDIO2 が 2.7 V 未満の時のみセットできます。

0 : 低い VDDIO2 電圧でのハイスピード IO 機能は無効です (VDDIO2 は 2.7 V を超えることができます)

1 : 低い VDDIO2 電圧でのハイスピード IO 機能は有効です (VDDIO2 は 2.7 V 未満にとどまります)

ビット 16 **IO_VDD_HSLV** : 低い VDD 電圧設定ビットでのハイスピード IO。

このビットは VDD が 2.7 V 未満の時のみセットできます。

0 : 低い VDD 電圧でのハイスピード IO 機能は無効です (VDD は 2.7 V を超えることができます)

1 : 低い VDD 電圧でのハイスピード IO 機能は有効です (VDD は 2.7 V 未満にとどまります)

ビット 15:8 **PRODUCT_STATE[7:0]** : 寿命状態コード (ハミング 8,4 に基づく) 詳しくは [セクション 7.6.11 : 製品状態の遷移](#) をご覧ください。

ビット 7 **NRST_STDBY** : コアドメイン STANDBY 移行リセットオプション設定ビット

0 : コアドメインの STANDBY モードに移行するときにリセットを生成します。

1 : コアドメインの STANDBY モードに移行するときにリセットを生成しません。

ビット 6 **NRST_STOP** : コアドメイン STOP 移行リセット・オプション設定ビット

0 : コアドメインの STOP モードに移行するときにリセットを生成します。

1 : コアドメインの STOP モードに移行するときにリセットを生成しません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **WWDG_SW** : WWDG 制御モード・オプション設定ビット

0 : WWDG ウォッチドッグはハードウェアによって制御されます。

1 : WWDG ウォッチドッグはソフトウェアによって制御されます。

ビット 3 **IWDG_SW** : IWDG 制御モードオプション設定ビット

0 : IWDG ウォッチドッグはハードウェアによって制御されます。

1 : IWDG ウォッチドッグはソフトウェアによって制御されます。

ビット 2 **BORH_EN** : ブラウンアウト・ハイ有効設定ビット

0 : 無効

1 : 有効

ビット 1:0 **BOR_LEV[1:0]** : ブラウンアウトレベルオプション設定ビット

これらのビットは、システムリセットを生成するパワーレベルを反映します。

00 または 11 : BOR レベル 1、閾値レベルは低 (約 2.1V) です。

01 : BOR レベル 2、閾値レベルは中 (約 2.4V) です。

10 : BOR レベル 3、閾値レベルは高 (約 2.7V) です。

7.11.21 FLASH 非セキュア・エポック・タイム・レジスタ (FLASH_NSEPOCHR_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x060

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NS_EPOCH[23:16]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NS_EPOCH[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **NS_EPOCH[23:0]** : 不揮発性、非セキュア・エポック・タイム・カウンタ

7.11.22 FLASH セキュア・エポック・タイム・レジスタ (FLASH_SECEPOCHR_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x068

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC_EPOCH[23:16]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_EPOCH[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **SEC_EPOCH[23:0]** : 不揮発性、セキュア・エポック・タイム・カウンタ

7.11.23 FLASH オプション・ステータス・レジスタ 2 (FLASH_OPTSR2_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

デフォルト値 : 0xB400 0170

この読出し専用レジスタは、対応するオプションビットの現在値を反映します。

アドレスオフセット : 0x070

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TZEN[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPD _DIS	Res.	SRAM2 _ECC	SRAM3 _ECC	BKPRAM _ECC	SRAM2 _RST	SRAM13 _RST	Res.	Res.
							r		r	r	r	r	r		

ビット 31:24 **TZEN[7:0]** : TrustZone イネーブル設定ビット

このビットはオプション・バイトの変更時にデバイスが TrustZone モードに移行できるようにします。

0xC3 : TrustZone は無効です。

0xB4 : TrustZone は有効です。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **USBPD_DIS** : USB Power Delivery 設定オプション・ビット

0 : 有効

1 : 無効

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SRAM2_ECC** : SRAM2 ECC 検出および訂正無効

0 : SRAM2 ECC チェックは有効です

1 : SRAM2 ECC チェックは無効です

ビット 5 **SRAM3_ECC** : SRAM3 ECC 検出および訂正無効

0 : SRAM3 ECC チェックは有効です

1 : SRAM3 ECC チェックは無効です

ビット 4 **BKPRAM_ECC** : バックアップ RAM ECC 検出および訂正無効

0 : BKPRAM ECC チェックは有効です

1 : BKPRAM ECC チェックは無効です

ビット 3 **SRAM2_RST** : システムリセット時の SRAM2 消去

0 : システムリセットが発生した際に SRAM2 を消去します。

1 : システム・リセットが発生した際に SRAM2 を消去しません。

ビット 2 **SRAM13_RST** : システム・リセット時に SRAM1 と SRAM3 を消去

0 : システム・リセットが発生した際に SRAM1 および SRAM3 を消去します

1 : システム・リセットが発生した際に SRAM1 および SRAM3 を消去しません

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

7.11.24 FLASH オプション・ステータス・レジスタ 2 (FLASH_OPTSR2_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。リセット後の値は、対応するオプションビットの現在値を反映します。

アドレスオフセット : 0x074

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TZEN[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBPD _DIS	Res.	SRAM2 _ECC	SRAM3 _ECC	BKPRAM _ECC	SRAM2 _RST	SRAM1 _3_RS T	Res.	Res.
							rw		rw	rw	rw	rw	rw		

ビット 31:24 **TZEN[7:0]** : TrustZone イネーブル設定ビット

このビットはオプション・バイトの変更時にデバイスが TrustZone モードに移行できるようにします。

0xC3 : TrustZone は無効です。

0xB4 : TrustZone は有効です。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **USBPD_DIS** : USB Power Delivery 設定オプション・ビット

0 : 有効

1 : 無効

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SRAM2_ECC** : SRAM2 ECC 検出および訂正無効

0 : SRAM2 ECC チェックは有効です

1 : SRAM2 ECC チェックは無効です

ビット 5 **SRAM3_ECC** : SRAM3 ECC 検出および訂正無効

0 : SRAM3 ECC チェックは有効です

1 : SRAM3 ECC チェックは無効です

ビット 4 **BKPRAM_ECC** : バックアップ RAM ECC 検出および訂正無効

0 : BKPRAM ECC チェックは有効です

1 : BKPRAM ECC チェックは無効です

ビット 3 **SRAM2_RST** : システムリセット時の SRAM2 消去

0 : システムリセットが発生した際に SRAM2 を消去します。

1 : システム・リセットが発生した際に SRAM2 を消去しません。

注 : この機能が有効になっていると、SRAM 消去がオプション・バイトの変更によりトリガされます。

ビット 2 **SRAM1_3_RST** : システム・リセット時に SRAM1 と SRAM3 を消去

0 : システム・リセットが発生した際に SRAM1 および SRAM3 を消去します

1 : システム・リセットが発生した際に SRAM1 および SRAM3 を消去しません

注 : この機能が有効になっていると、SRAM 消去がオプション・バイトの変更によりトリガされます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

7.11.25 FLASH 非セキュア・ブート・レジスタ (FLASH_NSBOOTR_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x080

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSBOOTADD[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSBOOTADD[7:0]								NSBOOT_LOCK[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:8 **NSBOOTADD[23:0]** : 非セキュア・ユニーク・ブート・エントリ・アドレス
このビットは非セキュア UBE アドレスを反映しています。

ビット 7:0 **NSBOOT_LOCK[7:0]** : SWAP_BANK および NSBOOTADD 設定の値をロックするフィールド。
0xC3 : SWAP_BANK と NSBOOTADD はそれぞれの規則に従い変更することも可能です。
0xB4 : NSBOOTADD は凍結されています。SWAP_BANK は 0xB4 に設定された (有効な) TZEN でのみ変更が可能です。

7.11.26 FLASH 非セキュア・ブート・レジスタ (FLASH_NSBOOTR_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x084

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSBOOTADD[23:8]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSBOOTADD[7:0]								NSBOOT_LOCK[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 **NSBOOTADD[23:0]** : 非セキュア・ユニーク・ブート・エントリ・アドレス

このビットは非セキュア BOOT アドレスの設定を可能にします。

ビット 7:0 **NSBOOT_LOCK[7:0]** : SWAP_BANK および NSBOOTADD 設定の値をロックするフィールド。

0xC3 : SWAP_BANK と NSBOOTADD はそれぞれの規則に従い変更することも可能です。

0xB4 : NSBOOTADD は凍結されています。SWAP_BANK は 0xB4 に設定された (有効な) TZEN でのみ変更が可能です。

7.11.27 FLASH セキュア・ブート・レジスタ (FLASH_SECBOOTR_CUR)

このレジスタはセキュアです。これはセキュア・アクセスによってのみ読み出し可能です。非セキュア読み出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x088

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECBOOTADD[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECBOOTADD[7:0]								SECBOOT_LOCK[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:8 **SECBOOTADD[23:0]** : ユニーク・ブート・エントリ・セキュア・アドレス

このビットはセキュア UBE アドレスを反映しています。

ビット 7:0 **SECBOOT_LOCK[7:0]** : UBE、SWAP_BANK、および SECBOOTADD 設定の値をロックするフィールド。

0xC3 : BOOT_UBE、SWAP_BANK、SECBOOTADD はそれぞれの規則に従い変更することも可能です。

0xB4 : BOOT_UBE と SECBOOTADD は凍結されています。SWAP_BANK は 0xC3 に設定された (無効な) TZEN でのみ変更が可能です。

7.11.28 FLASH セキュア・ブート・レジスタ (FLASH_BOOTR_PRG)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読みし／書き込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x08C

リセット値 : 0xFFFF XXXX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECBOOTADD[23:8]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECBOOTADD[7:0]								SECBOOT_LOCK[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 **SECBOOTADD[23:0]** : セキュア・ユニーク・ブート・エントリ・アドレス。

このビットはセキュア UBE アドレスの設定を可能にします。

ビット 7:0 **SECBOOT_LOCK[7:0]** : UBE、SWAP_BANK、および SECBOOTADD 設定の値をロックするフィールド。

0xC3 : BOOT_UBE、SWAP_BANK、SECBOOTADD はそれぞれの規則に従い変更することも可能です。

0xB4 : BOOT_UBE と SECBOOTADD は凍結されています。SWAP_BANK は 0xC3 に設定された (無効な) TZEN でのみ変更が可能です。

7.11.29 FLASH 非セキュア OTP ブロックのロック (FLASH_OTPBLR_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読み出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x090

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCKBL[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCKBL[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **LOCKBL[31:0]** : OTP ブロックのロック

ブロック n は、OTP 16 ビットワード $32 \times n$ から $32 \times n + 31$ までに対応します。

LOCKBL[n] = 1 は、OTP ブロック n のすべての OTP 16 bit ワードがロックされており、プログラムしようとする WRPERR となることを示します。

LOCKBL[n] = 0 は、OTP ブロック n のすべての OTP 16 bit ワードがロックされていないことを示します。

1 つのブロックがロックされると、書き込み保護を解除できなくなります。

また、ロックされていないと、OTP ワードを消去することができません。

7.11.30 FLASH 非セキュア OTP ブロックのロック (FLASH_OTPBLR_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x094

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCKBL[31:16]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCKBL[15:0]															
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

ビット 31:0 **LOCKBL[31:0]** : OTP ブロックのロック

ブロック n は、OTP 16 ビットワード $32 \times n$ から $32 \times n + 31$ までに対応します。

LOCKBL[n] = 1 は、OTP ブロック n のすべての OTP 16 bit ワードがロックされており、プログラムしようとする WRPERR となることを示します。

LOCKBL[n] = 0 は、OTP ブロック n のすべての OTP 16 bit ワードがロックされていないことを示します。

1 つのブロックがロックされると、書き込み保護を解除できなくなります。

LOCKBL ビットは、FLASH_OTPBLR_CUR の対応するビットがクリアされている場合にセットできます。

7.11.31 バンク 1 のFLASH セキュア・ブロック・ベースのレジスタ (FLASH_SECBB1Rx)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読みし／書き込みアクセスは RAZ/WI です。

アドレスオフセット : $0x0A0 + 0x004 \times (x-1)$, ($x = 1 \sim 4$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECBB1 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECBB1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **SECBB1[31:0]** : セキュア／非セキュア 8 KB Flash バンク 1 セクタ属性 ($y = 0$ から 31)

0 : バンク 1 のセクタ ($32 \times (x-1)+y$) は非セキュアです。

1 : バンク 1 のセクタ ($32 \times (x-1)+y$) はセキュアです。

7.11.32 バンク 1 のFLASH 特権ブロック・ベースのレジスタ (FLASH_PRIVBB1Rx)

このレジスタは特権です。これは特権アクセスによってのみ読出し、書込みが可能です。このレジスタは、非セキュア書込みアクセスから保護できます。

アドレスオフセット : $0x0C0 + 0x004 \times (x-1)$, ($x = 1 \sim 4$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIVBB1 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIVBB1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **PRIVBB1[31:0]** : 特権／非特権 8 KB Flash バンク 1 セクタ属性 ($y = 0$ から 31)

0 : バンク 1 のセクタ ($32 \times (x-1)+y$) は非特権です。

1 : バンク 1 のセクタ ($32 \times (x-1)+y$) は特権です。

7.11.33 バンク 1 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM1R_CUR)

このレジスタは非セキュアです。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x0E0

リセット値 : 0x00XX : 00XX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_END [6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_STRT [6:0]						
									r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **SECWM1_END[6:0]** : バンク 1 セキュリティ WM 領域 1 の終了セクタ

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **SECWM1_STRT[6:0]** : バンク 1 セキュリティ WM 領域 1 の開始セクタ

7.11.34 バンク 1 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM1R_PRG)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x0E4

リセット値 : 0x00XX : 00XX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_END [6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM1_STRT [6:0]						
									rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **SECWM1_END[6:0]** : バンク 1 セキュリティ WM 領域 1 の終了セクタ

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **SECWM1_STRT[6:0]** : バンク 1 セキュリティ WM 領域 1 の開始セクタ

7.11.35 バンク 1 の FLASH 書込みセクタ・グループ保護(FLASH_WRP1R_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x0E8

リセット値 : 0XXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPSG1 [31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPSG1 [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **WRPSG1[31:0]** : バンク 1 セクタグループ保護オプションステータスバイト

各 FLASH_WRP1R_CUR ビットは、バンク 1 の対応する 4 つの連続セクタからなるグループの書込み保護ステータスを反映します (0 : グループは書込み保護されています。1 : グループは書込み保護されていません)。

ビット 0 : セクタ 0~3 が組み込まれているグループ

ビット 1 : セクタ 4~7 が組み込まれているグループ

ビット N : セクタ 4 x N から 4 x N + 3 ままで組み込まれているグループ

ビット 31 : セクタ 124~127 が組み込まれているグループ

7.11.36 バンク 1 の FLASH 書込みセクタ・グループ保護(FLASH_WRP1R_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x0EC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPSG1 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPSG1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **WRPSG1[31:0]** : バンク 1 セクタ・グループ保護オプション・ステータス・バイト

WRPSG1 ビットを 0 にセットすると、バンク 1 の対応する 4 つの連続セクタからなるグループが書込み保護されます (0 : グループは書込み保護されます。1 : グループは書込み保護されません)。

ビット 0 : セクタ 0~3 が組み込まれているグループ

ビット 1 : セクタ 4~7 が組み込まれているグループ

ビット N : セクタ 4 x N から 4 x N + 3 ままで組み込まれているグループ

ビット 31 : セクタ 124~127 が組み込まれているグループ

7.11.37 FLASH データ・セクタ設定、バンク 1 (FLASH_EDATA1R_CUR)

このレジスタは非セキュアで、セキュアおよび非セキュア・アクセスのどちらでも読み出すことができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読み出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x0F0

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDATA1_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_STRT [2:0]		
r													r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EDATA1_EN** : バンク 1 の Flash 高サイクル・データ・イネーブル

0 : Flash の高サイクル・データ領域無し

1 : Flash の高サイクル・データが使用されています

ビット 14:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **EDATA1_STRT[2:0]** :

EDATA1_STRT はFlash のバンク 1 にある高サイクル・データ領域の開始セクタを含んでいます。ハードウェアはこれらのビットに影響を与えません。これはST のツールで管理します。

000 : バンク 1 の最後のセクタは、Flash の高サイクル・データに予約されています。

001 : バンク 1 の最後の 2 つのセクタはFlash の高サイクル・データに予約されています。

010 : バンク 1 の最後の 3 個のセクタはFlash の高サイクル・データに予約されています。

.....

111 : バンク 1 の最後の 8 つのセクタはFlash の高サイクル・データに予約されています。

7.11.38 FLASH データ・セクタ設定、バンク 1 (FLASH_EDATA1R_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0xF4

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDATA1_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_STRT [2:0]		
rw													rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EDATA1_EN** : バンク 1 のFlash 高サイクル・データ・イネーブル

0 : Flash の高サイクル・データ領域無し

1 : Flash の高サイクル・データが使用されています

ビット 14:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **EDATA1_STRT[2:0]** :

EDATA1_STRT はFlash のバンク 1 にある高サイクル・データ領域の開始セクタを含んでいます。ハードウェアはこれらのビットに影響を与えません。これは ST のツールで管理します。

000 : バンク 1 の最後のセクタはFlash の高サイクル・データに予約されています。

001 : バンク 1 の最後の 2 つのセクタはFlash の高サイクル・データに予約されています。

010 : バンク 1 の最後の 3 個のセクタはFlash の高サイクル・データに予約されています。

.....

注 : 111 : バンク 1 の最後の 8 つのセクタはFlash の高サイクル・データに予約されています。

7.11.39 FLASH HDP バンク 1 設定 (FLASH_HDP1R_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタには、初めに対応するオプション・ビットの値がロードされています。

アドレスオフセット : 0x0F8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_END [6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_STRT [6:0]						
									r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HDP1_END[6:0]** : HDPL バリア終了設定 (8 KB セクタの数)

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **HDP1_STRT[6:0]** : HDPL バリア開始設定 (8 KB セクタの数)

7.11.40 FLASH HDP バンク 1 設定 (FLASH_HDP1R_PRG)

このレジスタは非セキュアです。これはセキュアおよび非セキュアアクセスにより、読出し、書込みが可能です。FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 のときに非特権アクセスから保護することができます。

このレジスタは、HDPL0 および HDPL1 でのみアクセス可能です。HDPL2 または HDPL3 では、これは WI、RAZ です。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x0FC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_END [6:0]						
									rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_STRT [6:0]						
									rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HDP1_END[6:0]** : HDPL バリア終了設定 (8 KB セクタの数)

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **HDP1_STRT[6:0]** : HDPL バリア開始設定 (8 KB セクタの数)

7.11.41 FLASH ECC 訂正レジスタ (FLASH_ECCCORR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x100

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ECCC	Res.	Res.	Res.	Res.	ECCCI E	OTP_E CC	SYSF _ECC	BK _ECC	EDATA _ECC	OBK _ECC	Res.	Res.	Res.	Res.
	rw					rw	r	r	r	r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ECCC** : 1 つの ECC エラーが検出され訂正されたときに、ハードウェアによってセットされる ECC 訂正。1 を書き込むとクリアされます。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **ECCCI E** : ECC シングル訂正エラー割込みイネーブル・ビット

ECCCI E ビットを 1 にセットすると、読出し操作中に ECC シングル訂正エラーが発生したときに割込みが生成されます。

0 : ECC シングル訂正エラーが発生したときに、割込みを生成しません。

1 : ECC シングル訂正エラーが発生したときに、非セキュア割込みを生成します。

ビット 24 **OTP_ECC** : OTP ECC エラービット

このビットは、最後に成功した読出し専用／OTP 領域からの読出し操作中に、シングル ECC 訂正が発生すると 1 にセットされます。ECC エラーのアドレスは、ADDR_ECC ビットフィールドの中にあります。

ビット 23 **SYSF_ECC** : システム Flash メモリでの訂正された ECC エラーの ECC 失敗

これは、システム Flash メモリが ECC エラーに関与しているかどうかを示します。

ビット 22 **BK_ECC** : 訂正された ECC エラーの ECC 失敗のバンク

これは、どのバンクが ECC エラーに関与しているかどうかを示します。

ビット 21 **EDATA_ECC** : Flash の高サイクル・データ領域での訂正された ECC エラーの ECC 失敗

これは、Flash メモリの高サイクル・データ領域が ECC エラーに関与しているかどうかを示します。

ビット 20 **OBK_ECC** : Flash OB キー・ストレージ領域で訂正されたシングル ECC エラーこれは ECC エラーに関与している OBK ストレージを示します。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ADDR_ECC[15:0]** : ECC エラー・アドレス

読出し操作中に（シングル訂正の）ECC エラーが発生すると、ADDR_ECCにはエラーを生成したアドレスが含まれます。

フラグ・エラーがリセットされると ADDR_ECC もリセットされます。

Flash インタフェースでは、ECC エラー・フラグがセットされていない場合にのみ、このレジスタのアドレスをプログラムします。つまり、ECC エラーを生成した最初のアドレスのみが保存されるということです。

ADDR_ECC のアドレスは、エラーが発生した Flash メモリ領域（ユーザ Flash メモリ、システム Flash メモリ、データ領域、読出し専用／OTP 領域）に対する相対値です。

7.11.42 FLASH ECC 検出レジスタ (FLASH_ECCDETR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x104

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCD	Res.	Res.	Res.	Res.	Res.	Res.	OTP_ECC	SYSF_ECC	BK_ECC	EDATA_ECC	OBK_ECC	Res.	Res.	Res.	Res.
rw							r	r	r	r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 ECCD : ECC 検出

2 つの ECC エラーが検出されたときに、ハードウェアによってセットされます。

このビットがセットされた場合、NMI が生成されます。

1 を書き込むとクリアされます。続いて起きるダブル ECC エラーを検出するためにクリアする必要があります。

ビット 30:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 OTP_ECC : OTP ECC エラービット

このビットは、読み出し専用/OTP 領域からの最後の読み出し操作中に、ダブル ECC 検出が発生すると 1 にセットされます。ECC エラーのアドレスは、ADDR_ECC ビットフィールドの中にあります。

ビット 23 SYSF_ECC : システム Flash メモリでのダブル ECC エラーの ECC 失敗

これは、システム Flash メモリが ECC エラーに関与しているかどうかを示します。

ビット 22 BK_ECC : ダブル ECC エラーの ECC 失敗のバンク

これは、どのバンクが ECC エラーに関係しているかを示します。

ビット 21 EDATA_ECC : Flash の高サイクル・データ領域でのダブル ECC エラーの ECC 失敗

これは、Flash メモリの高サイクル・データ領域が ECC エラーに関係しているかどうかを示します。

ビット 20 OBK_ECC : Flash OB キー・ストレージ領域でのダブル ECC エラーの ECC 失敗。これは ECC エラーに関係している OBK ストレージを示します。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 ADDR_ECC[15:0] : ECC エラー・アドレス

読み出し操作中に (ダブル検出の) ECC エラーが発生すると、ADDR_ECC にはエラーを生成したアドレスが含まれます。

フラグ・エラーがリセットされると ADDR_ECC もリセットされます。

Flash インタフェースでは、ECC エラー・フラグがセットされていない場合にのみ、このレジスタのアドレスをプログラムします。つまり、ダブル ECC エラーを生成した最初のアドレスのみが保存されるということです。

ADDR_ECC のアドレスは、エラーが発生した Flash メモリ領域 (ユーザ Flash メモリ、システム Flash メモリ、データ領域、読み出し専用/OTP 領域) に対する相対値です。

7.11.43 FLASH ECC データ (FLASH_ECCDR)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x108

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **DATA_ECC[15:0]** : ECC エラー・データ

16 bit データの 6 bit ECC を行っている特別な領域（データ領域、読出し専用／OTP 領域）でダブル検出 ECC エラーが発生すると、失敗データがこのレジスタに読み込まれます。

失敗が実際のデータで起こっているのか、初期化されていないメモリへのアクセスが原因なのかを決定できるかチェックします。

7.11.44 バンク 2 の FLASH セキュア・ブロックベースのレジスタ (FLASH_SECBB2Rx)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x1A0 + 0x004 × (x-1), (x = 1~4)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SECBB2 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECBB2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **SECBB2[31:0]** : セキュア／非セキュア Flash バンク 2 セクタ属性 (y = 0 から 31)

0 : バンク 2 のセクタ (32 × (x-1)+y) は非セキュアです。

1 : バンク 2 のセクタ (32 × (x-1)+y) はセキュアです。

7.11.45 バンク 2 の FLASH 特権ブロックベースのレジスタ (FLASH_PRIVBB2Rx)

このレジスタは特権です。これは特権アクセスによってのみ読出し、書込みが可能です。このレジスタは、ウォーターマークにより非セキュア書込みアクセスから保護できます。

アドレスオフセット : $0x1C0 + 0x004 \times (x-1)$, ($x = 1 \sim 4$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIVBB2 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIVBB2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **PRIVBB2[31:0]** : 特権/非特権 8 KB Flash バンク 2 セクタ属性 ($y = 0$ から 31)

0 : バンク 2 のセクタ ($32 \times (x-1) + y$) は非特権です。

1 : バンク 2 のセクタ ($32 \times (x-1) + y$) は特権です。

7.11.46 バンク 2 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM2R_CUR)

このレジスタは非セキュアです。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : $0x1E0$

リセット値 : 0x00XX : 00XX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM_END2 [6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM_STRT2 [6:0]						
									r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **SECWM_END2[6:0]** : バンク 2 セキュリティ WM の終了セクタ

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **SECWM_STRT2[6:0]** : バンク 2 セキュリティ WM 領域の開始セクタ

7.11.47 バンク 2 の FLASH セキュリティ・ウォーターマーク (FLASH_SECWM2R_PRG)

このレジスタはセキュアです。セキュアアクセスによってのみ読み書きができます。非セキュア読出し／書込みアクセスは RAZ/WI です。このレジスタは、FLASH_PRIVCFGR レジスタの SPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x1E4

リセット値 : 0x00XX : 00XX

(レジスタ・ビット 0 から 31 に Flash メモリの OBL の値がロードされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM_END2 [6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM_STRT2 [6:0]						
									rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **SECWM_END2[6:0]** : バンク 2 セキュリティ WM 領域の終了セクタ

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **SECWM_STRT2[6:0]** : バンク 2 セキュリティ WM 領域の開始セクタ

7.11.48 バンク 2 の FLASH 書込みセクタ・グループ保護(FLASH_WRP2R_CUR)

このレジスタは非セキュアで、セキュアおよび非セキュアアクセスにより、読出しが可能です。FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 のときに非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x1E8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPSG2 [31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPSG2 [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **WRPSG2[31:0]** : バンク 2 セクタ・グループ保護オプション・ステータス・バイト

各ビットは、バンク 2 の対応する 4 つの連続セクタからなるグループの書込み保護ステータスを反映します (0 : グループは書込み保護されています。1 : グループは書込み保護されていません)。

ビット 0 : セクタ 0~3 が組み込まれているグループ

ビット 1 : セクタ 4~7 が組み込まれているグループ

ビット N : セクタ 4 x N から 4 x N + 3 ままで組み込まれているグループ

ビット 31 : セクタ 124~127 が組み込まれているグループ

7.11.49 バンク 2 の FLASH 書込みセクタ・グループ保護(FLASH_WRP2R_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x1EC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRPSG2 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRPSG2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **WRPSG2[31:0]** : バンク 2 セクタグループ保護オプションステータスバイト

WRPSGn2 ビットを 0 にセットすると、バンク 2 の対応する 4 つの連続セクタからなるグループが書込み保護されます (0 : グループは書込み保護されます。1 : グループは書込み保護されません)。

ビット 0 : セクタ 0~3 が組み込まれているグループ

ビット 1 : セクタ 4~7 が組み込まれているグループ

ビット N : セクタ 4 x N から 4 x N + 3 ままで組み込まれているグループ

ビット 31 : セクタ 124~127 が組み込まれているグループ

7.11.50 FLASH データ・セクタ設定、バンク 2 (FLASH_EDATA2R_CUR)

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

この読出し専用レジスタは、対応するオプション・ビットの現在値を反映します。

アドレスオフセット : 0x1F0

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDATA2_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_STRT [2:0]		
r													r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EDATA2_EN** : バンク 2 の Flash 高サイクル・データ・イネーブル

0 : Flash の高サイクル・データ領域無し

1 : Flash の高サイクル・データが使用されています

ビット 14:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **EDATA2_STRT[2:0]** :

EDATA2_STRT は Flash のバンク 2 にある高サイクル・データ領域の開始セクタを含んでいます。ハードウェアはこれらのビットに影響を与えません。これは ST のツールで管理します。

000 : バンク 2 の最後のセクタは Flash の高サイクル・データに予約されています。

001 : バンク 2 の最後の 2 個のセクタは Flash の高サイクル・データに予約されています。

010 : バンク 2 の最後の 3 個のセクタは Flash の高サイクル・データに予約されています。

.....

注 : 111 : バンク 2 の最後の 8 つのセクタは Flash の高サイクル・データに予約されています。

7.11.51 FLASH データ・セクタ設定、バンク 2 (FLASH_EDATA2R_PRG)

このレジスタは非セキュアです。セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x1F4

リセット値 : 0XXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDATA2_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_STRT [2:0]		
rw													rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EDATA2_EN** : バンク 2 の Flash 高サイクル・データ・イネーブル

0 : Flash の高サイクル・データ領域無し

1 : Flash の高サイクル・データが使用されています

ビット 14:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **EDATA2_STRT[2:0]** :

EDATA2_STRT は Flash のバンク 2 にある高サイクル・データ領域の開始セクタを含んでいます。ハードウェアはこれらのビットに影響を与えません。これは ST のツールで管理します。

000 : バンク 2 の最後のセクタは Flash の高サイクル・データに予約されています。

001 : バンク 2 の最後の 2 個のセクタは Flash の高サイクル・データに予約されています。

010 : バンク 2 の最後の 3 個のセクタは Flash の高サイクル・データに予約されています。

.....

注 : 111 : バンク 2 の最後の 8 個のセクタは Flash の高サイクル・データに予約されています。

7.11.52 FLASH HDP バンク 2 設定 (FLASH_HDP2R_CUR)

このレジスタには、初めに対応するオプション・ビットの値がロードされています。

このレジスタは非セキュアです。セキュアおよび非セキュア・アクセスのどちらでも読出しができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

アドレスオフセット : 0x1F8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_END [6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_STRT [6:0]						
									r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HDP2_END[6:0]** : HDPL バリア終了設定 (8 KB セクタの数)

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **HDP2_STRT[6:0]** : HDPL バリア開始設定 (8 KB セクタの数)

7.11.53 FLASH HDP バンク 2 設定 (FLASH_HDP2R_PRG)

このレジスタは非セキュアです。これは、セキュアおよび非セキュアアクセスのどちらでも読み書きができます。このレジスタは、FLASH_PRIVCFGR レジスタの NSPRIV ビットが 1 の場合に非特権アクセスから保護することができます。

レジスタは、HDPL0 および HDPL1 でのみアクセス可能です。HDPL2 または HDPL3 では、これは WI、RAZ です。

このレジスタは、対応するオプションビットの値のプログラムに使用されます。

アドレスオフセット : 0x1FC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_END [6:0]						
									r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_STRT [6:0]						
									r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HDP2_END[6:0]** : HDPL バリア終了設定 (8 KB セクタの数)

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **HDP2_STRT[6:0]** : HDPL バリア開始設定 (8 KB セクタの数)

7.12 Flash のレジスタ・マップとリセット値

表 75. レジスタマップとリセット値の表

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	Flash_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTEN	Res.	Res.	Res.	WRHIGHFREQ [1:0]	LATENCY [3:0]				
	リセット値																							0			0	1	0	0	1	1	
0x004	Flash_NSKEYR	NSKEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x008	Flash_SECKEYR	SECKEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	Flash_OPTKEYR	OPTKEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	Flash_NSGBKKEYR	NSGBKKEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	Flash_SECOBKKEYR	SECOBKKEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	FLASH_OPSR	CODE_OP [2:0]			Res.	Res.	Res.		OTP_OP	SYSF_OP	BK_OP	DATA_OP	Res.	ADDR_OP[19:0]																			
	リセット値	X	X	X					X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x01C	Flash_OPTCR	SWAP_BANK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0																													0	1	
0x020	Flash_NSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPTCHANGERR	OBKWERR	OBKERR	INCERR	STRBERR	PGSERR	WRPERR	EOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDBNE	WBNE	BSY
	リセット値									0	0	0	0	0	0	0	0													0	0	0	
0x024	Flash_SECSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBKWERR	OBKERR	INCERR	STRBERR	PGSERR	WRPERR	EOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBNE	WBNE	BSY	
	リセット値										0	0	0	0	0	0	0													X	X	X	
0x028	Flash_NSCR	BKSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPTCHANGERR	OBKWIE	OBKIE	INCERR	STRBERR	PGSERR	WRPERR	EOPIE	MER	Res.	Res.	SNB[6:0]						STRT	FW	BER	SER	PG	LOCK	
	リセット値	0									0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	1
0x02C	Flash_SECCR	BKSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBKWIE	OBKERR	INCERR	STRBERR	PGSERR	WRPERR	EOPIE	MER	Res.	Res.	SNB[6:0]						STRT	FW	BER	SER	PG	LOCK	
	リセット値	0	0								0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	1

表 75. レジスタマップとリセット値の表 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x030	FLASH_NSCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR_OPTCHANGEERR	CLR_OBKWERR	CLR_OBKERR	CLR_INCCRR	CLR_STRBERR	CLR_PGSERR	CLR_WRPERR	CLR_LEOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値									0	0	0	0	0	0	0	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
0x034	Flash_SECCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR_OBKWERR	CLR_OBKERR	CLR_INCCRR	CLR_STRBERR	CLR_PGSERR	CLR_WRPERR	CLR_EOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値										0	0	0	0	0	0	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
0x03C	Flash_PRIVCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSPRIV	SPRIV				
	リセット値																		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0					
0x040	Flash_NSOKBCKFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWAP_OFFSET[8:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALT_SECT_ERASE	ALT_SECT	SWAP_SECT_REQ	LOCK	
	リセット値								1	1	1	1	1	1	1	1	1		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0		
0x044	Flash_SECOBKCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWAP_OFFSET[8:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALT_SECT_ERASE	ALT_SECT	SWAP_SECT_REQ	LOCK
	リセット値								1	1	1	1	1	1	1	1	1		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0		
0x048	FLASH_HDPEXTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_EXT [6:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_EXT [6:0]							
	リセット値									0	0	0	0	0	0	0	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0		
0x04C	予約済みです。	予約済みです。																																			
0x050	Flash_OPTSR_CUR	SWAP_BANK	Res.	BOOT_UBE[7:0]								IWDG_STDBY	IWDG_STOP	Res.	Res.	IO_VDDIO2_HSLV	IO_VDD_HSLV	PRODUCT_STATE [7:0]								NRST_STDBY	NRST_STOP	Res.	WWDG_SW	IWDG_SW	BORH_EN	BOR_LEV[1:0]					
	リセット値	X		X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
0x054	Flash_OPTSR_PRG	SWAP_BANK	Res.	BOOT_UBE[7:0]								IWDG_STDBY	IWDG_STOP	Res.	Res.	IO_VDDIO2_HSLV	IO_VDD_HSLV	PRODUCT_STATE [7:0]								NRST_STDBY	NRST_STOP	Res.	WWDG_SW	IWDG_SW	BORH_EN	BOR_LEV[1:0]					
	リセット値	X		X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			
0x058	予約済みです。	予約済みです。																																			
0x05C	予約済みです。	予約済みです。																																			
0x060	Flash_NSEPOCHR_CUR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NS_EPOCH[23:0]																												
	リセット値								X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			

表 75. レジスタマップとリセット値の表 (続き)

[illegible]

表 75. レジスタマップとリセット値の表 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0E8	FLASH_WRP_CUR1R	WRPSG1 [31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x0EC	FLASH_WRP_PRG1	WRPSG1 [31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x0F0	FLASH_EDATA_CUR1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_STRT(2:0)		
	リセット値																	X													X	X	X	
0x0F4	FLASH_EDATA_PRG1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA1_STRT(2:0)		
	リセット値																	X													X	X	X	
0x0F8	FLASH_HDP_CUR1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_END [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_STRT [6:0]						
	リセット値										0	0	0	0	0	0	0	0										0	0	0	0	0	0	1
0x0FC	FLASH_HDP_PRG1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_END [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP1_STRT [6:0]					
	リセット値										X	X	X	X	X	X	X	0										X	X	X	X	X	X	X
0x100	FLASH_ECCCORR	Res.	ECCC	Res.	Res.	Res.	Res.	Res.	ECC0IE	OTP_ECC	SYSF_ECC	BK_ECC	EDATA_ECC	OBK_ECC	Res.	Res.	Res.	Res.	ADDR_ECC[15:0]															
	0x0000 0000		0						0	0	0	0		0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x104	FLASH_ECCDETR	ECCD	Res.	Res.	Res.	Res.	Res.	Res.	OTP_ECC_FAIL	SYSF_ECC	BK_ECC	EDATA_ECC	OBK_ECC	Res.	Res.	Res.	Res.	ADDR_ECC[15:0]																
	リセット値	0							0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x108	FLASH_ECCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA_ECC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1A0 + 0x004 * x (x = 1 ~ 4)	Flash_SECBB2R_x	SECBB2 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1A0 + 0x004 * x (x = 1 ~ 4)	Flash_PRIVBB2R_x	PRIVBB2 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1D0 - 0x1DC	予約済みです。	予約済みです。																																
0x1E0	FLASH_SECWM_CUR2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_END [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_STRT [6:0]								
	リセット値									X	X	X	X	X	X	X	X										X	X	X	X	X	X	X	

表 75. レジスタマップとリセット値の表 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x1E4	FLASH_SECWM _PRG2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_END1 [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECWM2_STRT [6:0]					
	リセット値										X	X	X	X	X	X	X											X	X	X	X	X	X	X
0x1E8	FLASH_WRP _CUR2R	WRPSG2 [31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x1EC	FLASH_WRP _PRG2R	WRPSG2 [31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x1F0	FLASH_EDATA _CUR2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_STRT [2:0]	
	リセット値																	X													X	X	X	
0x1F4	FLASH_EDATA _PRG2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EDATA2_STRT [2:0]	
	リセット値																	X													X	X	X	
0x1F8	FLASH_HDP _CUR2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_END [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_STRT [6:0]						
	リセット値										X	X	X	X	X	X	X											X	X	X	X	X	X	X
0x1FC	FLASH_HDP _PRG2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_END [6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDP2_STRT [6:0]						
	リセット値										X	X	X	X	X	X	X											X	X	X	X	X	X	X

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

8 命令キャッシュ (ICACHE)

8.1 ICACHE の概要

命令キャッシュ (ICACHE)は、内部および外部メモリから命令およびデータをフェッチするときのパフォーマンスを高めるために Cortex-M33 プロセッサの C-AHB コードバスに導入されました。

デュアル・マスタ・ポート、ヒットアンダーミス、クリティカルワードファースト・リフィル・ポリシーなど、いくつかの特定の機能により、ほとんどのユースケースにおいてゼロに近いウェイトステート性能が実現可能です。

8.2 ICACHE の主な機能

ICACHE の主な機能は次の通りです。

- バスインタフェース
 - 1 つの 32 ビット AHB スレーブポート、実行ポート (Cortex-M33 C-AHB コードインタフェースからの入力)
 - 2 つの AHB マスタポート: マスタ 1 およびマスタ 2 ポート (それぞれメイン AHB バスマトリックスの高速バスと低速バスへの出力)
 - 制御用の 32 ビット AHB スレーブポート x 1 (AHB ペリフェラル相互接続からの入力、ICACHE レジスタへのアクセス用)
- キャッシュアクセス
 - ヒット時のゼロウェイトステートを実現します。
 - ヒットアンダーミス機能: 以前のキャッシュミスによるラインリフィル中にプロセッサリクエスト (キャッシュデータへのアクセス) を処理する機能です。
 - デュアルマスタアクセス: 対象メモリに応じてトラフィックを切り離すために使用する機能です。たとえば、ICACHE は高速トラフィック (Flash および SRAM メモリのアドレス指定) を AHB マスタ 1 ポートに割り当て、低速トラフィック (外部メモリのアドレス指定) を AHB マスタ 2 ポートに割り当てることで、外部メモリからのラインリフィル時にプロセッサがストールするのを防ぎます。これにより、ISR (割込みサービスルーチン) による内部 Flash メモリのフェッチを外部メモリからのキャッシュラインリフィルと並行して行うことができます。
 - 2 つのマスタがあることで、割込み遅延への影響が最小限に抑えられます。
 - キャッシュラインのサイズの WRAPw バースト (キャッシュラインサイズに合わせた 32 ビットワードサイズ w) により、最適なキャッシュラインリフィルが行えます。
 - 1 ウェイに設定される可能性がある n ウェイセットアソシアティブのデフォルト設定とは、超低電力消費プロファイルを必要とする用途向けのダイレクトマップ方式のキャッシュを意味します。
- メモリアドレス再配置
 - 入力アドレスを最大 4 つのメモリ領域に再配置することができます (外部メモリにあるエイリアスコードを Code 領域に再配置して、C-AHB コードインタフェースから実行するために使用されます)。
- 書換えとリフィル
 - pLRU-t 書換え方式 (疑似最低使用頻度、バイナリツリーベース)、複雑性と性能のベストバランスを提供するアルゴリズムを実装しています。
 - リフィルのクリティカルワードファースト方式がプロセッサのストールを最小化します。

- 再配置した領域に対して AHB メモリトランザクションのバーストタイプ、INCRw または WRAPw を設定することができます（サイズ w は、キャッシュラインのサイズに整列したもの）。
- 性能カウンタ
ICACHE は次の 2 つの性能カウンタを実装しています。
 - ヒットモニタカウンタ（32 ビット）
 - ミスモニタカウンタ（16 ビット）
- エラー管理
 - 予期しないキャッシュ格納可能な書込みアクセスの検出、エラーフラグのセット、および割込み生成（オプション）が可能です。
- TrustZone セキュリティをサポート
- メンテナンス操作
 - キャッシュの無効化：キャッシュの完全無効化、高速コマンド、割込み不可

8.3 ICACHE の実装

表 76. ICACHE の機能

機能	ICACHE
ウェイ数	2
キャッシュサイズ	8 KB
キャッシュライン幅	16 バイト
再配置されるメモリ領域の範囲精度	2 MB
再配置する領域数	4
AHB スレーブインタフェースのデータサイズ	32 ビット
AHB 高速マスタ 1 インタフェースのデータサイズ	128 ビット
AHB 低速マスタ 2 インタフェースのデータサイズ	32 ビット

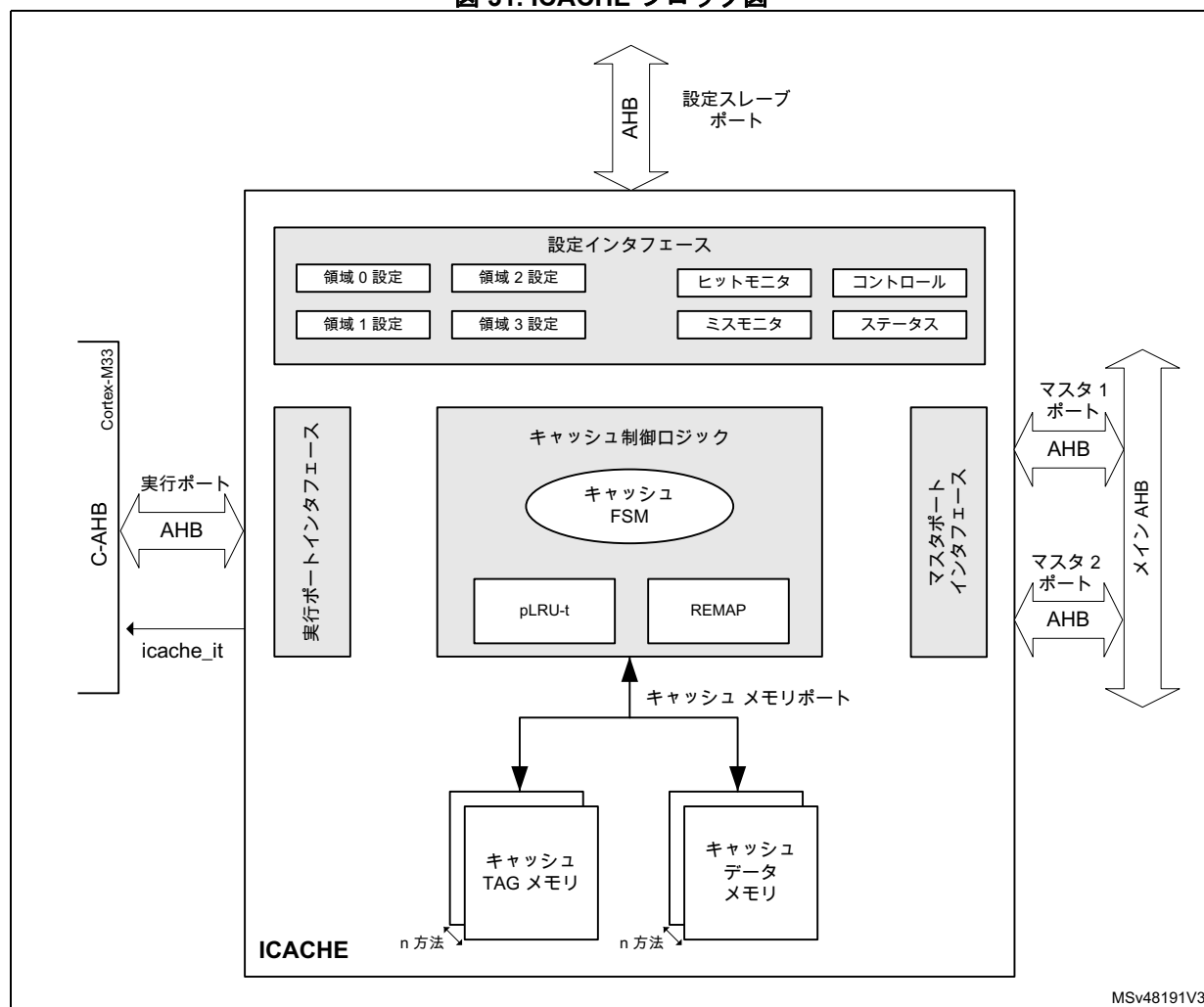
8.4 ICACHE の機能説明

命令キャッシュの目的は、プロセッサからの命令フェッチや命令メモリロードをキャッシュすることです。そのため、ICACHE は読出しトランザクションのみを管理し、書込みトランザクションは管理しません。

エラー管理の目的で、キャッシュ可能な書込みトランザクションを実行する場合（ソフトウェアプログラミングが正しくない場合のみ）、ICACHE はエラーフラグをセットし、有効な場合はプロセッサに割込みを生成します。

8.4.1 ICACHE ブロック図

図 31. ICACHE ブロック図



MSv48191V3

8.4.2 ICACHE のリセットおよびクロック

ICACHE は、Cortex-M33 C-AHB バスクロックでクロック供給されます。

ICACHE リセット信号が解除されると、キャッシュ無効化手順が自動的に起動し、ICACHE をビジーにします (ICACHE_SR = 0x0000 0001)。

この手順の終了後

- ICACHE は無効化されます。「コールドキャッシュ」状態になり、すべてのキャッシュライン有効ビット = 0 になります (ICACHE は完全にフィルされる必要があります)。
- ICACHE_SR = 0x0000 0002 (キャッシュがビジーでなくなったことを反映しています)。
- ICACHE は無効: ICACHE_CR の EN ビットはリセット状態 (= 0) を保持します。

注: 無効な場合、機能継続中の再配置メカニズムを除き、ICACHE はバイパスされ、スレーブ入力リクエスト (再配置される/されない) はマスタポートに転送されるのみです。

8.4.3 ICACHE TAG メモリ

ICACHE のタグメモリは以下を含みます。

- どのデータがキャッシュデータメモリに入っているかを示すアドレスタグ
- 有効性ビット

キャッシュラインごと（ウェイごと）に 1 つの有効ビットがあります。

有効ビットは、キャッシュラインのリフィル時（ミス後）にセットされます。

有効ビットは次のタイミングでリセットされます。

- ICACHE のリセットが解除されたあと
- ICACHE_CR の EN = 0 をセットすることによって（ソフトウェアによって）、キャッシュが無効になったとき
- ICACHE_CR の CACHEINV = 1 をセットすることによって（ソフトウェアによって）、ICACHE 無効化コマンドを実行するとき

キャッシュ可能トランザクションが実行入力ポートで受信され、その AHB アドレス (HADDR_in) が以下のフィールドに分割されているとき (B および W の定義については 表 77 を参照)

- HADDR_in[B-1:0]: アドレスバイトオフセット。キャッシュライン内で選択するバイトを示します。
- HADDR_in[B+W-1:B]: アドレスウェイインデックス。各ウェイ内で選択するキャッシュラインを示します。
- HADDR_in[31:B+W]: タグアドレス。リクエストされたデータがすでに ICACHE 内で使用可能（有効）かどうかを確認するために、TAG メモリアドレスと比較されます。

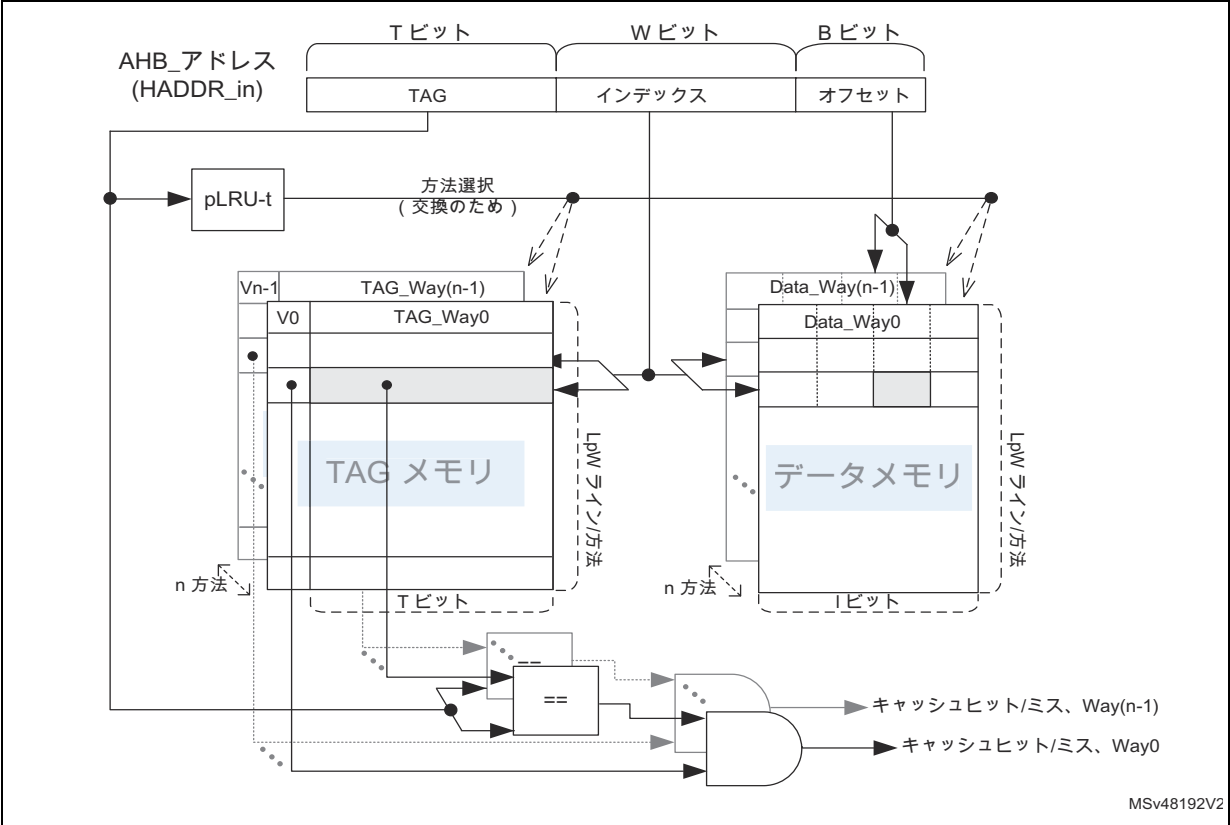
下の表に、TAG メモリのディメンショニングに関する ICACHE の主なパラメータの概要を示します。

図 32 は、n ウェイセットアソシアティブ ICACHE について、TAG およびデータメモリの機能図を示しています。

表 77. n ウェイセットアソシアティブ動作モード（デフォルト）の TAG メモリディメンショニングパラメータ

パラメータ	値	例
キャッシュサイズ	$S \text{ KB} = s \text{ バイト} (s = 1024 \times S)$	8 KB = 8192 バイト
キャッシュのウェイ数	n	2
キャッシュラインサイズ	$L \text{ バイト} = l \text{ ビット} (l = 8 \times L)$	16 バイト = 128 ビット
キャッシュライン数(ウェイあたり)	$LpW = s / (n \times L) \text{ ライン / ウェイ}$	256 ライン / ウェイ
アドレスバイトのオフセットサイズ	$B = \log_2(L) \text{ ビット}$	4 ビット
アドレスウェイのインデックスサイズ	$W = \log_2(LpW) \text{ ビット}$	8 ビット
タグアドレスサイズ	$T = (32 - W - B) \text{ ビット}$	20 ビット

図 32. ICACHE TAG およびデータメモリの機能図



8.4.4 ダイレクトマップ ICACHE (1 ウェイキャッシュ)

デフォルト設定(リセット時)はnウェイセットアソシアティブキャッシュ(ICACHE_CRのWAYSEL = 1)ですが、ユーザはWAYSEL = 0を書き込むことによってICACHEをダイレクトマップとして設定できます(キャッシュが無効、つまりICACHE_CRのEN = 0のときのみ可能)。

下の表に、ダイレクトマップキャッシュ動作モードが選択されたときのTAGメモリに関するICACHEの主なパラメータの概要を示します。

表 78. ダイレクトマップキャッシュモードのTAGメモリディメンショニングパラメータ

パラメータ	値	例
キャッシュサイズ	$S \text{ KB} = s \text{ バイト} (s = 1024 \times S)$	8 KB = 8192 バイト
キャッシュのウェイ数	1	1
キャッシュラインサイズ	$L \text{ バイト} = l \text{ ビット} (l = 8 \times L)$	16 バイト = 128 ビット
キャッシュライン数	$LpW = s / L \text{ ライン}$	512 本のライン
アドレスバイトのオフセットサイズ	$B = \log_2(L) \text{ ビット}$	4 ビット
アドレスウェイのインデックスサイズ	$W = \log_2(LpW) \text{ ビット}$	9 ビット
タグアドレスサイズ	$T = (32 - W - B) \text{ ビット}$	19 ビット

すべてのキャッシュ操作(読出し、リフィル、再配置、無効化など)は、ダイレクトマップ設定でも同じままです。唯一の違いは、データのリフィルに使用できるウェイは1つのみ(固有のもの)なので、ラインエビクション(セクション 8.4.8 で説明)が発生した場合の置換アルゴリズムがないことです。

8.4.5 ICACHE 有効化

ICACHE を有効にするには、ICACHE_CR の EN ビットを 1 にセットする必要があります。

無効な場合、ICACHE はバイパスされ、すべてのトランザクションはスレーブポートから同じクロックサイクルでマスタポートにコピーされます。

ICACHE が無効なときに、メインメモリの内容（後でキャッシュされる領域）を初期化または変更し、この領域が変更されないときにのみ、ICACHE を有効にすることをお勧めします（有効な ICACHE はキャッシュ可能な書込みトランザクションをエラーとして検出します）。

パフォーマンスの確実性を確実にするためには、潜在的なキャッシュ無効化手順の終了を待ってから、ICACHE を有効にすることをお勧めします。無効化手順は、ハードウェアリセット信号が解除されたとき、CACHEINV がセットされたとき、または ICACHE_CR で EN がクリアされたときに発生します。この手順中、ICACHE_SR の BUSYF がセットされ、手順が終了すると、BUSYF がクリアされ、同じレジスタの BSYENDF がセットされます（そのようなビジー終了条件では、有効な場合、ICACHE 割込みが生成されます）。

ソフトウェアは、ICACHE を有効にする前に、BUSYF および BSYENDF の値をテストする必要があります。無効化手順が終了する前に ICACHE が有効化された場合、キャッシュアクセス（BUSYF = 1 のとき）はキャッシュ不可として扱われ、パフォーマンスはメインメモリのアクセス時間に依存することになります。

入力トランザクションアドレスが、ICACHE_CRRx で定義され、有効化されたメモリ領域に属する場合、ICACHE が有効かどうかに関係なく、アドレス再配置が行われます（図 33 を参照）。

ICACHE はデフォルトではブート時に無効化されます。

8.4.6 キャッシュ格納可および不可のトラフィック

ICACHE は、Cortex-M33 コア用に開発されています。C-AHB バス上に配置されているため、コードメモリ領域（メモリマップの 0x0000 0000 から 0x1FFF FFFF まで）をキャッシュします。

その他のメモリ領域をキャッシュ可能にするために、ICACHE はメモリ領域再配置機能をサポートしています。これは最大 4 つの外部メモリ領域を定義するために使用され、これらのアドレスはコード領域でエイリアスを持ちます。これらの外部メモリ領域をコードエイリアスアドレスを通じてアドレス指定することで、メモリリクエストを C-AHB バスにルーティングし、ICACHE によって管理できます。

[0x6000 0000:0xAFFF FFFF] の範囲にある任意のアドレスに物理的に配置されている外部メモリ空間は、[0x0000 0000:0x07FF FFFF] または [0x1000 0000:0x1FFF FFFF] の範囲にあるアドレスでエイリアスを作成することができます。

コード領域にある特定のメモリリクエストに対して、ICACHE はまずアドレス再配置機能を実装します。エイリアスが作成されると、それは次にキャッシュされるアドレスに再配置され、必要な場合は、メイン AHB バスバスマトリックスのアドレスを指定するためにマスタポートに提供されます。デスティネーション物理アドレスは、AHB バスに対するそれ以上の操作を必要としません。

再配置機能は、キャッシュ不可トラフィックや、キャッシュが無効なときにも使用できます。

アドレス再配置の詳細については、[セクション 8.4.7](#) を参照してください。

ICACHE へのメモリリクエストは、その AHB トランザクションメモリルックアップ属性に応じて（表 79 を参照）、キャッシュ可能と定義されます。この AHB 属性は、アドレス指定された領域の MPU（メモリ保護ユニット）プログラミングに依存します。

表 79. AHB トランザクションの ICACHE のキャッシュ可能性

AHB ルックアップ属性	キャッシュ可能性
1	キャッシュ可能
0	キャッシュ不可

キャッシュ不可アクセスの場合、ICACHE はバイパスされます。すなわち、AHB トランザクションが変更されずにそのままマスタ出力ポートに伝搬されます。ただし、アドレス再配置機能（[セクション 8.4.7](#) を参照）によって変更される可能性があるトランザクションアドレスを除きます。

バイパス、およびその結果としての再配置ロジックにより、対象メモリへのアクセス遅延時間が長くなることはありません。

キャッシュ可能アクセスの場合、ICACHE は [セクション 8.4.8](#) で説明されているように動作します。

キャッシュ可能メモリ領域は、ユーザによって MPU で定義され、プログラムされます。MPU は特定の領域をアドレス指定するトランザクションについて、AHB 属性信号の生成を行います。

次の表に、さまざまなメモリのプログラム可能な設定の概要を示します。

表 80. メモリ設定

メモリ	キャッシュ可能 (MPU プログラミング)	ICACHE で再配置 (ICACHE_CRRx プログラミング)
Flashメモリ	はい／いいえ	不要
SRAM	推奨せず	不要
外部メモリ	はい	必要
	不可	ユーザが外部メモリのコードを C-AHB バスでフェッチした場合は必須（そうでない場合は S-AHB バスでフェッチ）

8.4.7 アドレス再配置

ICACHE により、最大 4 つの 外部メモリ領域について、コード領域でエイリアスアドレスを定義できます。

アドレス再配置はコードエイリアスアドレスに適用されて、デスティネーション外部物理アドレスに変換されます。

再配置操作は、ICACHE_CRRx（再配置領域数 $x = 0 \sim 3$ ）をプログラムすることで、完全にソフトウェアによって設定可能です。このプログラミングは、ICACHE が無効なときのみ行うことができます。

各領域 x は、ICACHE_CRRx の REN で個別に有効化できます。有効化されると、ICACHE が無効な場合や、トランザクションがキャッシュ不可の場合でも、再配置操作が発生します。

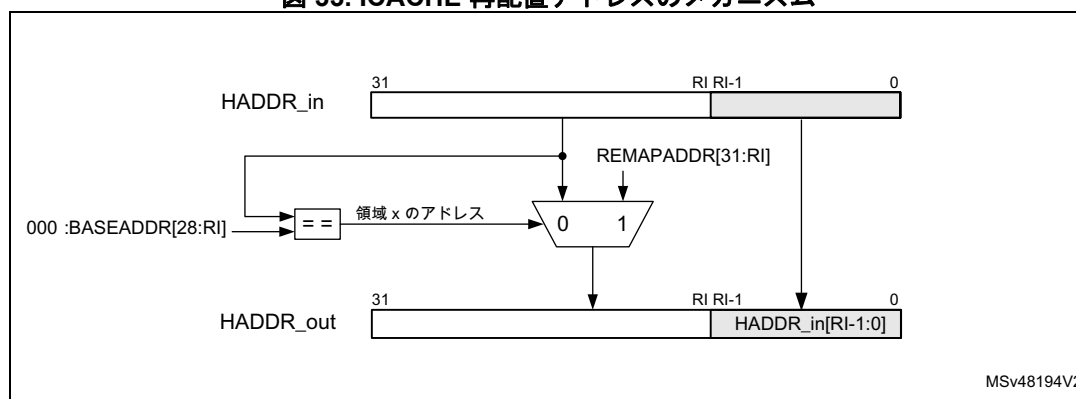
再配置領域のサイズは同じでなくても構いません。各領域のサイズはその ICACHE_CRRx の RSIZE でプログラムできます。各領域のサイズは範囲の最小単位（2 MB）の 2 の累乗値であり、最小領域サイズは 2 MB、最大領域サイズは 128 MB です。

アドレス再配置のメカニズムは、受信 AHB アドレス (HADDR_in) と特定のコードサブ領域のベースアドレスが一致することと、以下に示すように、このアドレスを (再配置された) 外部物理アドレスに変更することに基づきます。

- HADDR_in[31:RI] = 000:BASEADDR[28:RI] の場合、HADDR_in は領域 x に属します。ここで、
 - 000:BASEADDR は、ICACHE_CRRx の BASEADDR でプログラムされたコードサブ領域のベースアドレスです。
 - RI には考慮すべき上位ビット数が定義されます。RI = log2 (領域サイズ) であり、最小値は 21 (2 MB 領域用)、最大値は 27 (128 MB 領域用) です。
- 領域 x が有効な場合、マスタポート出力 AHB のアドレス (HADDR_out) は次の 2 つの部分で構成されます。
 - MSB として、ICACHE_CRRx の REMAPADDR[31:RI] フィールド
 - LSB として、HADDR_in[RI-1:0]

下の図は、一致および出力アドレス生成について示しています。

図 33. ICACHE 再配置アドレスのメカニズム



次の表に、RSIZE に応じて、ICACHE_CRRx の BASEADDR および REMAPADDR (上位 MSB の数) の可能なすべての設定を示します。

表 81. ICACHE 再配置領域のサイズ、ベースアドレス、および再配置アドレス

領域サイズ (MB)	ベースアドレスサイズ (MSB)	再配置アドレス (MSB)
2	8	11
4	7	10
8	6	9
16	5	8
32	4	7
64	3	6
128	2	5

ICACHE_CRRx の BASEADDR および REMAPADDR をプログラムするときには注意が必要です。プログラムされた値が予想値より大きい場合 (MSB 数については 表 81 を参照)、不要かつ余分な LSB は無視されます。

代表的な再配置の例：外部アドレス範囲 [0x6800 0000:0x6FFF FFFF] に物理的に配置されている128 MB の FMC 領域 (NOR/SRAM) を、コードセクション範囲 [0x1000 0000:0x17FF FFFF] に再配置します。

- REMAPADDR[31:21] = 0x340
- BASEADDR[28:21] = 0x80
- HADDR_in[31:27] は 000:BASEADDR[28:27], と比較され、HADDR_in/BASEADDR[26:21] は比較では無視されます。

比較して一致する場合

- HADDR_out[31:27] は REMAPADDR[31:27] になります (HADDR_in[31:27] の代わりに)。
- HADDR_out[26:0] は HADDR_in[26:0] になります。

ソフトウェアは、ICACHE_CRRx の HBURST をセットすることによって、バスマトリックスの ICACHE マスタポートによって生成される AHB バースト (キャッシュラインのリフィルのため) の種類を次のようにプログラムできます。

- WRAP バーストモードをサポートできるインタフェースを通じてアクセスされる再配置された外部メモリについては WRAP。これにより、クリティカルワードファースト機能のパフォーマンス上のメリットが得られます。
 - ラップバーストサイズ = キャッシュラインサイズ
 - ラップバースト開始アドレス = コアによってリクエストされる最初のデータのワードアドレス
- INCR: WRAP バーストモードをサポートしていないインタフェースを通じてアクセスされる外部メモリについては、INCR バーストモード (クリティカルワードファースト機能のメリットは失われます)。
 - INCR バーストサイズ = キャッシュラインサイズ
 - INCR バースト開始アドレス = リクエストされたワードを含むキャッシュラインの境界に整列されたアドレス

注： 外部領域とそれらのエイリアスコードサブ領域の両方の SAU (セキュア属性ユニット) および MPU (メモリ保護ユニット) 属性をプログラムするときには、一貫性が必要です。

8.4.8 キャッシュ可能アクセス

ICACHE がキャッシュ可能トランザクションを Cortex-M33 から受信すると、ICACHE はリクエストされたアドレスが TAG メモリにあるかどうか、また、対応するキャッシュラインが有効であるかどうかを確認します。

次の 3 通りの方法があります。

- アドレスが TAG メモリ内にあり、キャッシュラインが有効な場合：**キャッシュヒット**。データはキャッシュから読み出され、同じサイクル内でプロセッサに提供されます。
- アドレスが TAG メモリにない場合：**キャッシュミス**。データはメインメモリから読み出されてプロセッサに提供され、キャッシュラインのリフィルが行われます。

クリティカルワードファーストポリシーによってプロセッサの最小ウェイトサイクルが保証されます。キャッシュがキャッシュラインリフィルを実行している間に読出しデータを提供できるためです (関連する遅延は、メインメモリから 1 ワードをフェッチする際の遅延です)。

アドレスの再配置が発生しない場合、ICACHE マスタバスで生成されるバーストは WRAPw です (w は、ワード数によるキャッシュライン幅です)。アドレスの再配置が発生した場合、バーストの種類は、対応する ICACHE_CRRx でプログラムされた HBURST に依存します。

AHB トランザクションの属性も、ラインリフィルのために選択されたマスタポートのメイン AHB バスマトリックスに伝搬されます。

- アドレスは TAG メモリにないが、現在行われているメインメモリからのリフィルバーストに属する場合：**キャッシュヒット**（ヒットアンダーミス機能）に属します。

これは、キャッシュラインのリフィル中に発生します。ICACHE は、リクエストされたデータを、それがマスタインタフェースで使用可能になるとすぐに提供できるため、ミス（メインメモリからのデータのフェッチ）を回避できます。

（キャッシュミスにより）キャッシュリフィルが発生した場合、ICACHE はリフィルデータを書き込むキャッシュラインを次のように選択します。

- **ダイレクトマップ（1 ウェイ）モード**では、1 ラインのみを使用してリフィルデータ、すなわち入力アドレスのインデックスで指し示されたラインを格納することができます。
- **n ウェイセットアソシアティブモード**では、n ライン中の 1 ライン（n 個のウェイのそれぞれにおいてアドレスインデックスで指し示されたライン）を使用できます。ウェイの選択は、pLRU-t 置換アルゴリズムに基づき、これは、各インデックスについて、次のリフィルのためのウェイ候補を指し示します。

リフィルデータを書き込む必要があるキャッシュラインがすでに有効な場合でも、ターゲットのキャッシュラインを最初に無効化する必要があります。ダイレクトマップモードの場合でも、n ウェイセットアソシアティブキャッシュモードの場合でも同様です。

8.4.9 デュアルマスタキャッシュ

ICACHE は、メイン AHB バスマトリックス上にデュアルポート AHB マスタ、すなわちマスタ 1 およびマスタ 2 ポートを実装できます。これは、別々のデスティネーションメモリへのトラフィックを分割するために使用されます。

再配置されないトラフィックは自動的にマスタ 1 ポートになります。外部メモリに再配置されたトラフィックは、（領域単位で）ICACHE_CRRx の MSTSEL をプログラムすることによって、マスタ 2 ポートにルーティングする必要があります。

コードは一般に、次のようにフェッチできます。

- マスタ 1 ポート（高速バス）の内部 Flash メモリおよび内部 SRAM
- マスタ 2 ポート（低速バス）の外部 Flash メモリ/RAM

外部メモリを実装していないシステムの場合、内部 Flash メモリへのトラフィックを内部 SRAM へのトラフィックから切り離すことができます（ICACHE によって再配置された場合）。この機能は、ミス発生時のさらなるプロセッサのストールを防ぐために使用されます。

ヒットアンダーミス機能とならび、このデュアルマスタ機能により、プロセッサは異なるメモリからのフェッチ時に代替バスを使用できます。

8.4.10 ICACHE のセキュリティ

ICACHE は、Armv8-M TrustZone を実装します。

ICACHE 設定レジスタは、システムレベルで保護されます。

8.4.11 ICACHE のメンテナンス

ソフトウェアは、ICACHE_CR レジスタの CACHEINV をプログラムすることによって、ICACHE の全内容を無効化できます。

CACHEINV = 1 のとき、ICACHE 制御ロジックは ICACHE_SR の BUSYF フラグをセットし、キャッシュ無効化操作を開始して、各 TAG の有効ビットを 0 にリセットします（キャッシュラインごとに 1 つの有効ビット）。CACHEINV は自動的にクリアされます。

無効化操作が終了すると、ICACHE は BUSYF を自動的にクリアして、ICACHE_SR レジスタの BSYENDF をセットします。

このフラグ条件で有効化された場合 (ICACHE_IER の BSYENDIE = 1)、ICACHE 割込みが生成されます。すると、(空の) キャッシュを再び使用できるようになります。

8.4.12 ICACHE 性能の監視

ICACHE は、パフォーマンス分析のために以下のモニタを備えています。

- 32 ビットのヒットモニタは、ICACHE の内容にヒットしたスレーブキャッシュポート上のキャッシュ可能 AHB トランザクションをカウントします。
また、このモニタは、アドレスが TAG メモリまたはリフィルバッファにある (以前のミスのため、そのデータがキャッシュマスターポートから受信されるか、まもなく受信される) すべてのアクセスも考慮します (セクション 8.4.8 を参照)。
- 16 ビットのミスモニタは、ICACHE の内容にミスしたスレーブキャッシュポート上のキャッシュ可能 AHB トランザクションをカウントします。
また、このモニタは、アドレスが TAG メモリにもリフィルバッファにも存在しないすべてのアクセスも考慮します。

最大値に達しても、これらのモニタは最小値に戻りません。

ヒットおよびミスモニタは、特定のコードの分析が可能なソフトウェアによって有効化およびリセットすることができます。

ソフトウェアは次のタスクを実行できます。

- ヒットモニタの有効化/停止は、ICACHE_CR の HITMEN で行います。
- ヒットモニタをリセットするには、ICACHE_CR の HITMRST をセットします。
- ミスモニタの有効化/停止は、ICACHE_CR の MISSMEN で行います。
- ミスモニタをリセットするには、ICACHE_CR の MISSMRST をセットします。

消費電力を低減するために、これらのモニタはデフォルトでは無効化 (停止) されています。

8.4.13 ICACHE ブート

ICACHE はブート時に無効化されます (ICACHE_CR の EN = 0)。

Cortex-M33 は 0x0 以外のブート開始アドレス定義を可能にする VTOR (ベクタテーブル) を実装しているため、ブート時にコードを再配置する必要はありません。

ブートが終了すると、ICACHE を有効にできます (ソフトウェアで ICACHE_CR の EN = 1 をセット)。

8.5 ICACHE 低電力モード

デバイスレベルでは、ICACHE の使用により、たいていの場合、サイズが大きく、そのために消費電力も多いメインメモリからではなく、内部 ICACHE から命令をフェッチすることによって、消費電力が低減されます。キャッシュされるメインメモリが外部メモリの場合、低減量はさらに大きくなります。

より低性能プロファイル (ヒット率の点で) のアプリケーションや、厳しい低消費電力制約を持つアプリケーションの場合、ダイレクトマップとして設定された ICACHE の低電力消費が有効かもしれません。シングルウェイキャッシュ設定は、ICACHE_CR の WAYSEL = 0 をプログラムすることで取得されます (図 32 を参照)。消費電力は、リクエストのたびに、タグおよびデータメモリの必要な一部分だけにアクセスすることによって低減されます。その間も、キャッシュ効果によってフェッチパフォーマンスは向上します。それでも、ほとんどのコード実行の場合、n ウェイセットアソシティブキャッシュモードよりは、効率がやや低くなります。

8.6 ICACHE のエラー管理と割込み

サポートされていないキャッシュ可能な書込みリクエストが検出された場合（機能エラー）、ICACHE は ICACHE_SR の ERRF フラグをセットすることによってエラーを生成します。対応する割込みイネーブルビットがセットされている場合（ICACHE_IER の ERRIE = 1）、割込みが生成されます。

その他にキャッシュ無効化操作の終わりにも、割込みが生成される可能性があります。キャッシュビジー状態が終了すると、ICACHE は ICACHE_SR の BSYENDF フラグをセットします。対応する割込みイネーブルビットがセットされている場合（ICACHE_IER の BSYENDIE = 1）、割込みが生成されます。

すべての ICACHE 割込みソースは、同一かつ一意な割込み信号 `icache_it` を生成し、同じ割込みベクタを使用します。

表 82. ICACHE 割込み

割込みベクタ	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法
ICACHE	機能エラー	ICACHE_SR の ERRF	ICACHE_IER の ERRIE	ICACHE_FCR の CERRF を 1 にセット
	ビジー状態終了（無効化終了）	ICACHE_SR の BSYENDF	ICACHE_IER の BSYENDIE	ICACHE_FCR の CBSYENDF を 1 にセット

また、ICACHE は、すべての AHB バスエラー（セキュリティ問題、アドレスデコード問題など）を マスタ 1 または マスタ 2 ポートから実行ポートに再伝播します。

8.7 ICACHE レジスタ

8.7.1 ICACHE 制御レジスタ (ICACHE_CR)

アドレスオフセット：0x000

リセット値：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MISSMRST	HITMRST	MISSMEN	HITMEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WAYSEL	CACHEINV	EN
													rw	w	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **MISSMRST**：ミスモニタのリセット

0：影響なし。

1：キャッシュミスモニタをリセットします。

ビット 18 **HITMRST**：ヒットモニタのリセット

0：影響なし。

1：キャッシュヒットモニタをリセットします。

ビット 17 **MISSMEN** : ミスモニタの有効化

- 0 : キャッシュミスモニタのスイッチを切ります。モニタは停止してもリセットされません。
- 1 : キャッシュミスモニタを有効にします。

ビット 16 **HITMEN** : ヒットモニタの有効化

- 0 : キャッシュヒットモニタのスイッチを切ります。モニタは停止してもリセットされません。
- 1 : キャッシュヒットモニタを有効にします。

ビット 15:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **WAYSEL** : アソシアティブキャッシュモード選択

このビットによって、ICACHE セットアソシアティブを選択することができます。キャッシュが無効 (EN = 0) の場合のみ、ソフトウェアで書き込むことができます。

- 0 : ダイレクトマップキャッシュ (1 ウェイキャッシュ)
- 1 : n-ウェイセットアソシアティブキャッシュ (リセット値)

ビット 1 **CACHEINV** : キャッシュ無効化 (インバリデート)

BUSYF フラグがセットされたときに (キャッシュのメンテナンス操作中)、ソフトウェアによってセットされ、ハードウェアによってクリアされます。0 を書き込んでも、ビットの値は変化しません。

- 0 : 影響なし。
- 1 : キャッシュ全体を無効化 (インバリデート) します (すべてのキャッシュラインの有効ビット = 0)。

ビット 0 **EN** : 有効化 (イネーブル)

- 0 : キャッシュは無効です。(ディセーブル)
- 1 : キャッシュは有効です。(イネーブル)

8.7.2 ICACHE ステータスレジスタ (ICACHE_SR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRF	BSYEN DF	BUSYF
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **ERRF** : キャッシュエラーフラグ

- 0 : エラーはありません。
- 1 : 操作 (キャッシュ可能な書込み) 中にエラーが発生しました。

ビット 1 **BSYENDF** : ビジー終了フラグ

- 0 : キャッシュはビジーです。
- 1 : 完全無効化 CACHEINV 操作が終了しました。

ビット 0 **BUSYF** : ビジーフラグ

- 0 : キャッシュは CACHEINV 操作でビジーではありません。
- 1 : キャッシュは完全無効化 CACHEINV 操作を実行中です。

8.7.3 ICACHE 割込み有効化レジスタ (ICACHE_IER)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRIE	BSYEN DIE	Res.
													rw	rw	

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **ERRIE** : キャッシュエラー時の割込み有効化

キャッシュ機能エラー（キャッシュ可能な書込みアクセス）発生時に割込み生成を有効にするために、ソフトウェアによってセットされます。

0 : エラー時の割込みは無効です。

1 : エラー時の割込みは有効です。

ビット 1 **BSYENDIE** : ビジー終了時の割込み有効化

キャッシュ無効化（インバリデート）操作の終了時に割込み生成を有効にするために、ソフトウェアによってセットされます。

0 : ビジー終了時の割込みは無効です。

1 : ビジー終了時の割込みは有効です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

8.7.4 ICACHE フラグクリアレジスタ (ICACHE_FCR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERRF	CBSYE NDF	Res.
													w	w	

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CERRF** : キャッシュエラーフラグのクリア

ソフトウェアによってセットされます。

0 : 影響なし。

1 : ICACHE_SR の ERRF フラグをクリアします。

ビット 1 **CBSYENDF** : ビジー終了フラグのクリア

ソフトウェアによってセットされます。

0 : 影響なし。

1 : ICACHE_SR の BSYENDF フラグをクリアします。

ビット 0 予約済みであり、リセット値に保持する必要があります。

8.7.5 ICACHE ヒットモニタレジスタ (ICACHE_HMONR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HITMON[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HITMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **HITMON[31:0]** : キャッシュヒットモニタのカウンタ

8.7.6 ICACHE ミスモニタレジスタ (ICACHE_MMONR)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MISSMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **MISSMON[15:0]** : キャッシュミスモニタのカウンタ

8.7.7 ICACHE 領域 x 設定レジスタ (ICACHE_CRRx)

アドレスオフセット : $0x020 + 0x4 * x$, ($x = 0 \sim 3$)

リセット値 : $0x0000\ 0200$

他の領域のコード領域でエイリアスアドレスを定義して、キャッシュ可能にします。

ICACHE_CR の EN = 1 のとき、BASEADDR および REMAPADDR フィールドは書き込みロックされています (読み出し専用)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HBURST	Res.	Res.	MSTSEL	Res.	REMAPADDR[31:21]										
rw			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REN	Res.	Res.	Res.	RSIZE[2:0]			Res.	BASEADDR[28:21]							
rw				rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **HBURST** : 領域 x の出力バーストタイプ

0 : WRAP

1 : INCR

ビット 30:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **MSTSEL** : 領域 x の AHB キャッシュマスタ選択

0 : 選択なし (デフォルトでマスタ 1 に選択されています)。

1 : マスタ 2 を選択します。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:16 **REMAPADDR[31:21]** : 領域 x の再配置アドレス

このフィールドには、BASEADDR フィールドで定義したエイリアスアドレスが入ります。

唯一の有用なビットは [31:RI] です。RI は $21 \leq RI \leq 27$ の範囲にある、RSIZE のビット数です (セクション 8.4.7 を参照)。プログラムされた値に LSB の方が多い場合、無用なビットは無視されます。

ビット 15 **REN** : 領域 x の有効化

0 : 無効

1 : 有効

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:9 **RSIZE[2:0]** : 領域 x のサイズ

000 : 予約済み

001 : 2 MB

010 : 4 MB

011 : 8 MB

100 : 16 MB

101 : 32 MB

110 : 64 MB

111 : 128 MB

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **BASEADDR[28:21]** : 領域 x のベースアドレス

このエイリアスアドレスは REMAPADDR フィールドによって置き換えられます。

唯一の有用なビットは [28:RI] です。RI は $21 \leq RI \leq 27$ の範囲にある、RSIZE のビット数です (セクション 8.4.7 を参照)。プログラムされた値に LSB の方が多い場合、無用なビットは無視されます。

8.7.8 ICACHE レジスタマップ

表 83. ICACHE レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	ICACHE_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	MISSMRST	HITMRST	MISSMEN	HITMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WAYSEL	CACHEINV	EN
	リセット値														0	0	0	0														1	0	0
0x004	ICACHE_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRF	BSYENDF	BUSYF
	リセット値																															0	0	1
0x008	ICACHE_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRIE	BSYENDIE	Res.
	リセット値																															0	0	
0x00C	ICACHE_FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERRF	BSYENDF	Res.
	リセット値																															0	0	
0x010	ICACHE_HMONR	HITMON[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	ICACHE_MMONR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MISSMON[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018 - 0x01C	予約済み	予約済み																																
0x020	ICACHE_CRR0	HBURST	Res.	Res.	MSTSEL	Res.	Res.	REMAPADDR[31:21]										REN	Res.	Res.	Res.	Res.	RSIZE [2:0]	Res.	Res.	BASEADDR[28:21]								
	リセット値	0			0			0	0	0	0	0	0	0	0	0	0	0					0	0	1			0	0	0	0	0	0	0
0x024	ICACHE_CRR1	HBURST	Res.	Res.	MSTSEL	Res.	Res.	REMAPADDR[31:21]										REN	Res.	Res.	Res.	Res.	RSIZE [2:0]	Res.	Res.	BASEADDR[28:21]								
	リセット値	0			0			0	0	0	0	0	0	0	0	0	0	0					0	0	1			0	0	0	0	0	0	0
0x028	ICACHE_CRR2	HBURST	Res.	Res.	MSTSEL	Res.	Res.	REMAPADDR[31:21]										REN	Res.	Res.	Res.	Res.	RSIZE [2:0]	Res.	Res.	BASEADDR[28:21]								
	リセット値	0			0			0	0	0	0	0	0	0	0	0	0	0					0	0	1			0	0	0	0	0	0	0
0x02C	ICACHE_CRR3	HBURST	Res.	Res.	MSTSEL	Res.	Res.	REMAPADDR[31:21]										REN	Res.	Res.	Res.	Res.	RSIZE [2:0]	Res.	Res.	BASEADDR[28:21]								
	リセット値	0			0			0	0	0	0	0	0	0	0	0	0	0					0	0	1			0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

9 データキャッシュ (DCACHE)

9.1 DCACHE の概要

データキャッシュ (DCACHE) は、外部メモリとの間のデータトラフィックのパフォーマンスを高めるために、Cortex-M33 プロセッサの S-AHB システムバスに導入されました。

ヒットアンダーミスやクリティカルワードファーストリフィルポリシーなど、いくつかの特定の機能により、外部メモリデータアクセスで最適なパフォーマンスが得られます。

9.2 DCACHE の主な機能

DCACHE の主な機能は次の通りです。

- バスインタフェース
 - 1 つの 32 ビット AHB スレーブポート、システムポート (Cortex-M33 S-AHB システムインタフェースからの入力)
 - 1 つの 32 ビット AHB マスタポート (メイン AHB バスマトリックスへの出力)
 - 制御用の 1 つの 32 ビット AHB スレーブポート (DCACHE レジスタにアクセスするための AHB ペリフェラル相互接続からの入力)
- キャッシュアクセス
 - ヒット時のゼロウェイトステート
 - ヒットアンダーミス機能: 以前のキャッシュミスによるラインリフィル中にプロセッサリクエスト (キャッシュデータへのアクセス) を処理する機能です。
 - キャッシュラインのサイズの WRAP バーストにより、最適化されたキャッシュラインリフィル (128 ビットのキャッシュラインの WRAP4 など)
 - 2 ウェイセットアソシアティブ
 - ライトバックとライトスルーの両方のポリシーをサポートします (AHB バッファ可能属性で選択可能)
 - 読出しおよびライトバックは常に割り当て
 - ライトスルーは常に非割り当て (write-around)
 - バイト、ハーフワード、およびワードの書込みをサポート
- 書換えとリフィル
 - pLRU-t 書換え方式 (疑似最低使用頻度、バイナリツリーベース)、複雑性と性能のベストバランスを提供するアルゴリズムを実装しています。
 - 読出しトランザクションのクリティカルワードファーストリフィルポリシーにより、プロセッサのストールを最小化します。
 - すべての AHB メモリトランザクションのバーストタイプの設定可能性: (サイズ w は、キャッシュラインのサイズに整列したもの)。
- 性能カウンタ

DCACHE は次の 4 つのパフォーマンスカウンタを実装しています。

 - 2 つのヒットモニタカウンタ (32 ビット): 読出しヒット数、書込みヒット数
 - 2 つのミスモニタカウンタ (16 ビット): 読出しミス数、書込みミス数

- エラー管理
 - DCACHE 自体によって開始されたマスタポートリクエストについて、エラーを検出し(キャッシュラインがエビクションまたはクリーン操作によりメインメモリに書き戻される)、このエラーにフラグを立て、オプションで割込みをレイズする能力
- TrustZone セキュリティサポート
- メンテナンス操作
 - キャッシュの無効化：キャッシュの完全無効化、高速コマンド、割込み不可
 - キャッシュ無効化範囲：指定されたアドレス範囲に属するキャッシュラインを無効化します(リセット有効ビット=0)。バックグラウンドタスク、割込み可能。
 - キャッシュクリーン範囲：指定されたアドレス範囲に属するキャッシュラインをクリーンします(ダーティビット=1の場合、ラインを書き戻し、ダーティビットをクリア)。バックグラウンドタスク、割込み可能。
 - キャッシュクリーンおよび無効化範囲：指定されたアドレス範囲に属するキャッシュラインをクリーンして無効化します(ダーティビット=1の場合、ラインを書き戻し、有効ビットをクリア)。バックグラウンドタスク、割込み可能。

9.3 DCACHE の実装

DCACHE1 は Cortex-M33 S-AHB バスにあり、外部 RAM メモリ領域 (OCTOSPI および FMC) のみをキャッシュします (メモリマップのアドレス範囲 [0x6000 0000:0x9FFF FFFF])。

実際、バスマトリックスでマルチプレキシングノードを DCACHE1 の前に置くことによって、SRAM 領域またはペリフェラル領域をアドレス指定した S-AHB バスメモリリクエスト (それぞれ [0x2000 0000:0x3FFF FFFF] と [0x4000 0000:0x5FFF FFFF] の範囲) は、メイン AHB バスマトリックスに直接ルーティングされ、DCACHE1 はバイパスされます。

表 84. DCACHE の機能

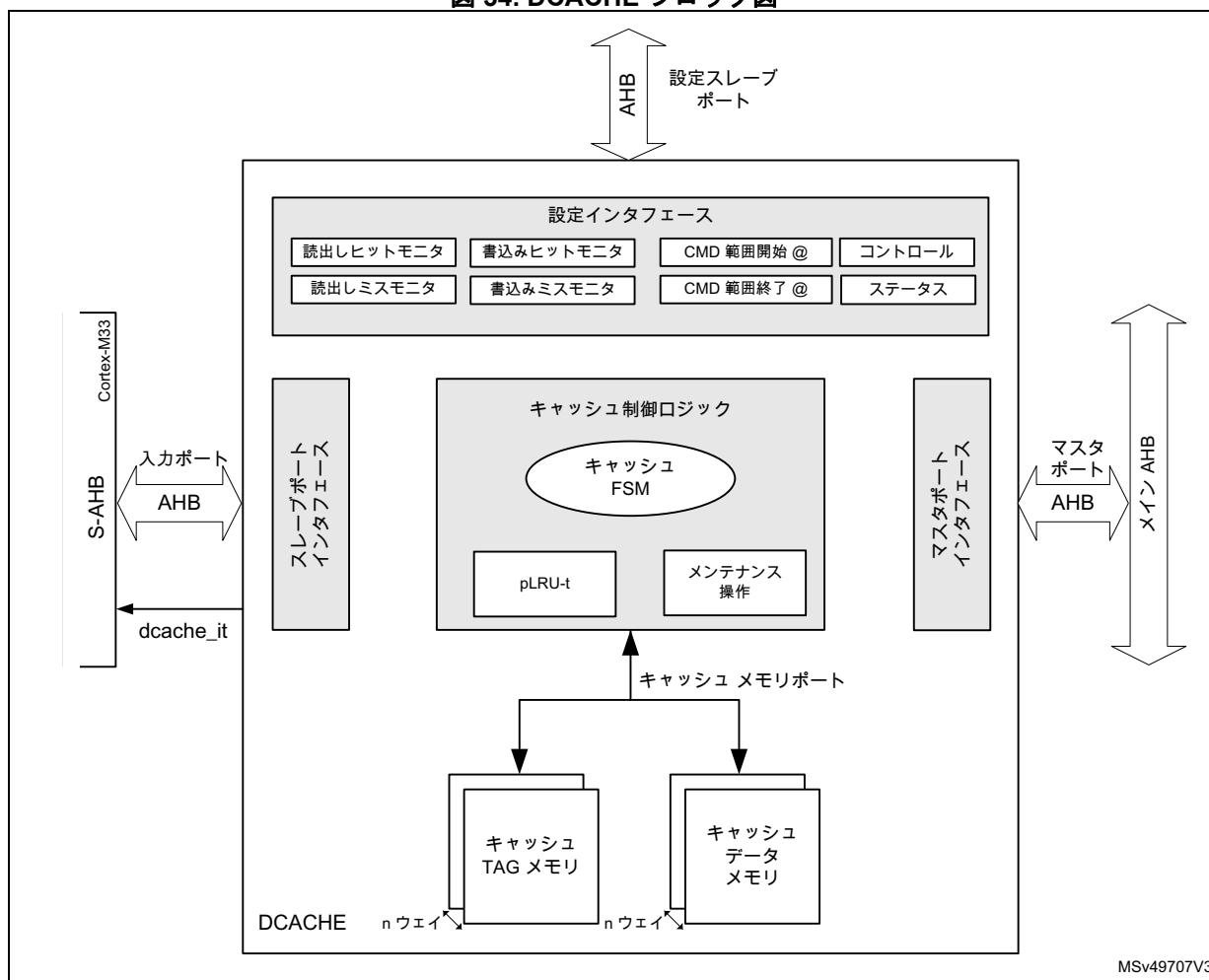
機能	DCACHE1
ウェイ数	2
キャッシュサイズ	4 KB
キャッシュライン幅	16 バイト
AHB マスタインタフェースのデータサイズ	32 ビット

9.4 DCACHE の機能説明

データキャッシュの目的は、プロセッサ から着信する外部メモリデータのロードおよびストアをキャッシュすることです。これらのアクセスには、外部メモリアドレスで発生する可能性のある命令フェッチが含まれます。DCACHE は、読出しと書込みの両方のトランザクションを管理します。

9.4.1 DCACHE ブロック図

図 34. DCACHE ブロック図



9.4.2 DCACHE のリセットおよびクロック

DCACHE は、Cortex-M33 S-AHB バスクロックでクロック供給されます。

DCACHE リセット信号が解除されると、キャッシュ無効化手順が自動的に開始され、DCACHE はビジーになります (DCACHE_SR = 0x0000 0001)。

この手順の終了後

- DCACHE は無効化されます。「コールドキャッシュ」と呼ばれ、すべてのキャッシュラインが有効、ダーティおよび特権ビットがゼロになります (DCACHE は完全にフィルされる必要があります)。
- DCACHE_SR = 0x0000 0002 (キャッシュがビジーでなくなったことを反映しています)
- DCACHE は無効 : DCACHE_CR の EN ビットはリセット状態 (= 0) を保持します。

注 : 無効なとき、DCACHE はバイパスされます。スレーブ入力リクエストはマスターポートに転送されるだけです。

9.4.3 DCACHE TAG メモリ

DCACHE のタグメモリは以下を含みます。

- どのデータがキャッシュデータメモリに入っているかを示すアドレスタグ
- 有効性ビット
- ダーティビット
- 特権ビット

キャッシュラインごとに（ウェイごとに）1 つの有効ビット、1 つのダーティビット、および 1 つの特権ビットがあります。

有効ビットは、データキャッシュラインへのアクセスを有効/無効にします。ラインが有効でない場合、データアクセス（読出しまたは書込み）はメインメモリで行われます。

有効ビットは、キャッシュラインが書き込まれるとき（読出しミスまたはライトバックミスのいずれかによってリフィルされるとき）にセットされます。

有効ビットは次のタイミングでリセットされます。

- DCACHE のリセットが解除されたあと
- DCACHE_CR の EN = 0 をセットすることによって（ソフトウェアによって）、キャッシュが無効になったとき
- DCACHE 無効化コマンドの 1 つを実行して、ソフトウェアによって DCACHE_CR の CACHEINV = 0、または CACHECMD = 0b010 あるいは 0b011 にセットされたとき（[セクション 9.4.8](#) を参照）。

ダーティビットは、キャッシュラインがメインメモリの内容に関して最新の値を持っていることを示します（キャッシュに最新の正しい値があり、メインメモリは最新ではない）。

ダーティビットは、キャッシュラインがスレーブポート書込みトランザクションによって書き込まれたときにセットされます（ライトバック属性を持つアクセスの場合のみ）。

ダーティビットは、以下のいずれかの場合にリセットされます。

- DCACHE のリセットが解除されたあと
- 読出しミス時にラインリフィルが実行されたとき（ライトバックミス時、リフィルされるキャッシュラインは書込みデータによって変更され、ダーティビット = 1）
- キャッシュ無効化が実行されたとき
- DCACHE クリーン操作の 1 つが実行され（キャッシュラインがメインメモリに書き戻される）、ソフトウェアによって DCACHE_CR の CACHECMD = 0b001 または 0b011 にセットされたとき（[セクション 9.4.8](#) を参照）。

特権ビットは、データが特権エンティティによって管理されるかどうかを示します。このラインへの最初のアクセスについて、入カスレーブポートで AHB 特権属性の値に従って割り当てられます（ラインリフィル、読出しミス、またはライトバックミス時にのみ書き込まれます）。

特権ビットは、特権属性と同じ極性を保持します。特権アクセスの場合は 1、非特権アクセスの場合は 0 です。

特権ビットは、キャッシュが無効化されたとき、また、DCACHE リセットが解除された後にリセットされます。

キャッシュ可能トランザクションが入カスレーブポートで受信され、その AHB アドレス (HADDR_in) が以下のフィールドに分割されているとき (B および W の値については、次の表を参照) :

- HADDR_in[B-1:0] : アドレスバイトオフセット。キャッシュライン内で選択するバイトを示します。
- HADDR_in[B+W-1:B] : アドレスウェイインデックス。各ウェイ内で選択するキャッシュラインを示します。
- HADDR_in[31:B+W] : タグアドレス。リクエストされたデータがすでに DCACHE 内で使用可能 (有効) かどうかを確認するために、TAG メモリアドレスに比較されます。

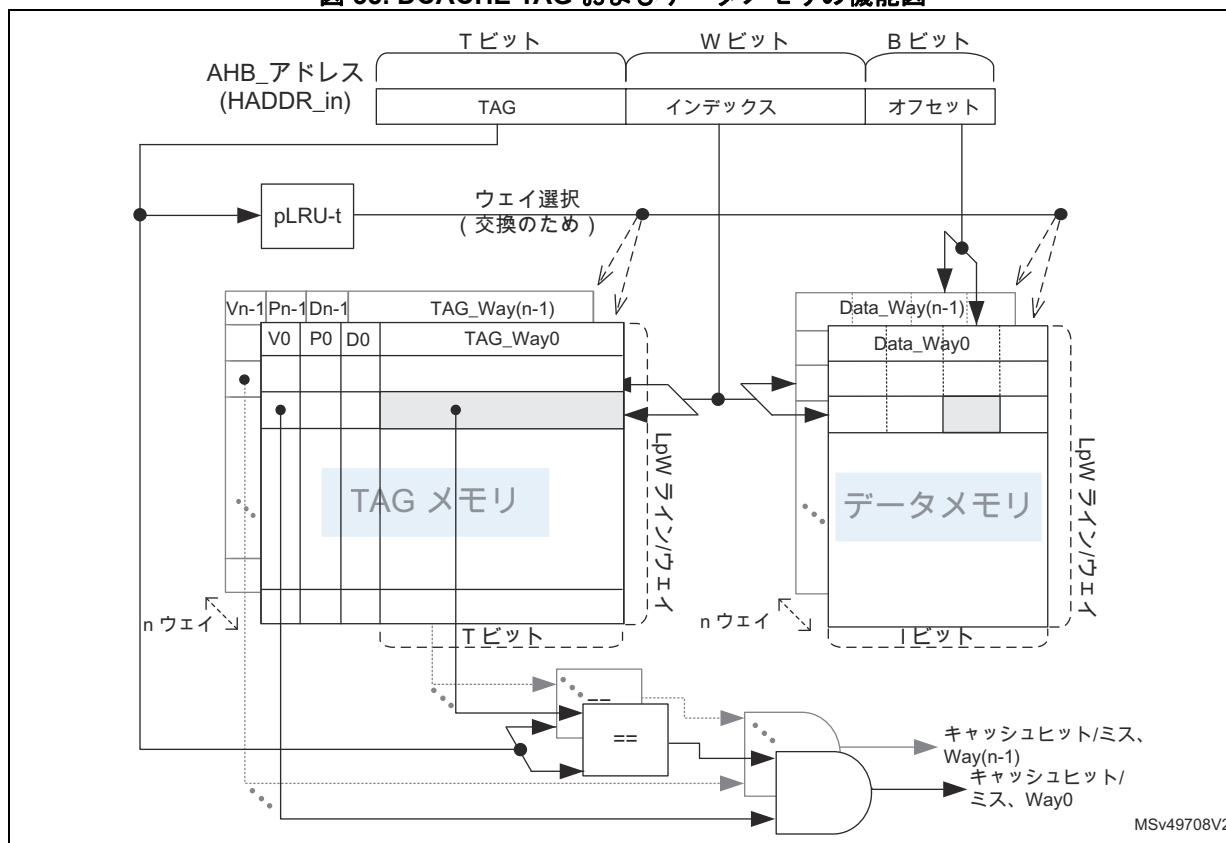
次の表に、TAG メモリのディメンショニングに関する DCACHE の主なパラメータを示します。

図 35. は、n ウェイセットアソシアティブ DCACHE について、TAG およびデータメモリの機能図を示しています。

表 85. TAG メモリディメンショニングパラメータ

パラメータ	値	例
キャッシュサイズ	$S \text{ KB} = s \text{ バイト} (s = 1024 \times S)$	4 KB = 4096 バイト
キャッシュのウェイ数	n	2
キャッシュラインサイズ	$L \text{ バイト} = l \text{ ビット} (l = 8 \times L)$	16 バイト = 128 ビット
キャッシュライン数(ウェイあたり)	$LpW = s / (n \times L) \text{ ライン/ウェイ}$	128 ライン/ウェイ
アドレスバイトのオフセットサイズ	$B = \log_2(L) \text{ ビット}$	4 ビット
アドレスウェイのインデックスサイズ	$W = \log_2(LpW) \text{ ビット}$	7 ビット
タグアドレスサイズ	$T = (32 - W - B) \text{ ビット}$	21 ビット

図 35. DCACHE TAG およびデータメモリの機能図



9.4.4 DCACHE 有効化

DCACHE の機能を有効にするには、DCACHE_CR 制御レジスタの EN ビットをセットする必要があります。

無効な場合、DCACHE はバイパスされ、すべてのトランザクションはスレーブポートから同じクロックサイクルのマスタポートにコピーされ、TAG アドレスとの比較は行われません。

DCACHE はデフォルトではブート時に無効化されます。

9.4.5 キャッシュ格納可および不可のトラフィック

DCACHE は Cortex-M33 コア用に開発され、これに接続された AHB バスによってアドレス指定されたメモリ領域をキャッシュします。

さらに、メモリ領域への AHB バストラフィックは、キャッシュ可能な場合とキャッシュ不可場合があります。DCACHE に着信するメモリリクエストは、AHB トランザクションメモリルックアップ属性に応じて、キャッシュ可能として定義されます。

書込みトランザクションの場合、DCACHE 書込みポリシーは、AHB トランザクションメモリバッファ可能属性に従って、ライトスルーまたはライトバックとして定義されます (次の表を参照)。

これらの AHB 属性は、アドレス領域のメモリ保護ユニット (MPU) のプログラミングに依存します。

表 86. AHB トランザクションの DCACHE のキャッシュ可能性

AHB ルックアップ属性	AHB バッファ可能属性	キャッシュ可能性
0	x	読出しおよび書込み：キャッシュ不可
1	0	読出し：キャッシュ可能 書込み：(キャッシュ可能) ライトスルー
1	1	読出し：キャッシュ可能 書込み：(キャッシュ可能) ライトバック

キャッシュ不可アクセスの場合、DCACHE はバイパスされます。すなわち、AHB トランザクションはマスタ出力ポートにそのまま伝播されます。

バイパスによって対象メモリへのアクセス遅延時間が長くなることはありません。

キャッシュ格納可アクセスの場合、DCACHE は[セクション 9.4.6](#)で説明されているように動作します。

キャッシュ可能メモリ領域は、ユーザによって MPU で定義され、プログラムされます。MPU は、特定の領域をアドレス指定するトランザクションについて、AHB 属性信号の生成を行います。

9.4.6 キャッシュ可能アクセス

DCACHE がキャッシュ可能トランザクションを Cortex-M33 からスレーブポートで受信すると、DCACHE はリクエストされたアドレスが TAG メモリにあるかどうか、また、対応するキャッシュラインが有効であるかどうかを確認します。

読出しトランザクションの場合、次の 3 通りの方法があります。

- アドレスが TAG メモリ内にあり、キャッシュラインが有効な場合：**キャッシュ読出しヒット**。データはキャッシュから読み出され、同じサイクル内でプロセッサに提供されます。
- アドレスが TAG メモリにない場合：**キャッシュ読出しミス**。データはメインメモリから読み出されてプロセッサに提供され、キャッシュラインのリフィルが行われます。

クリティカルワードファーストポリシーにより、プロセッサの最小ウェイトサイクルが保証されます。キャッシュがまだキャッシュラインのリフィルを実行している間に読出しデータを提供することができるためです（関連する遅延は、メインメモリから 1 ワードをフェッチする際の遅延です）。

DCACHE マスタバスで生成されるバーストの種類は、DCACHE_CR の HBURST ビットによって決まり、INCRw または WRAPw です（w はワード単位のキャッシュライン幅です）。

AHB トランザクション属性も、スレーブ入力（ミスした）リクエストからマスタ出力リフィルリクエストに伝播されます。

- アドレスは TAG メモリにないが、現在行われているメインメモリからのリフィルバーストに属する場合：**キャッシュ読出しヒット**（ヒットアンダーミス機能）。

ラインリフィルが読出しと書込みのいずれの（ミスした）トランザクションであっても、DCACHE は、データがマスタインタフェースで使用可能になるとすぐに、リクエストされた読出しデータを提供できるため、ミス（メインメモリからのデータフェッチ）を回避できます。

ライトバックトランザクションの場合も（ライトバックバッファ可能属性を持つ書込みトランザクション）、3通りの方法があります。

- アドレスが TAG メモリ内にあり、キャッシュラインが有効な場合：**キャッシュライトバックヒット**。データはキャッシュに書き込まれます。
- アドレスが TAG メモリにない（またはキャッシュラインが有効でない）：**キャッシュライトバックミス**。

まず、キャッシュライン全体のデータをメインメモリから読み出すことによって、ライン割り当てが行われます。このラインリフィルのために DCACHE マスタバスで生成されるバーストの種類は、DCACHE_CR の HBURST ビットによって決まり、INCRw または WRAPw であり（w はワード単位のキャッシュライン幅）、AHB トランザクション属性がスレーブポートの最初のリクエストから伝播されます。

リフィルされたラインが DCACHE に書き込まれると、スレーブポートで提供された最初のデータが、この DCACHE ラインに書き込まれます（直前にリフィルされたキャッシュラインのデータ部分を上書きします）。

- アドレスは TAG メモリにないが、現在行われているメインメモリからのリフィルバーストに属する場合：**キャッシュライトバックヒット**（ヒットアンダーミス機能）。

ラインリフィルが読出しと書込みのいずれの（ミスした）トランザクションによる場合でも、DCACHE は、着信したデータをリフィルされたラインに直接書き込んで、ミス（メインメモリからのリフィル）を回避できます。

ライトスルートランザクション（ライトスルーバッファ可能属性を持つ書込みトランザクション）の場合、2通りの方法があります。

- アドレスが TAG メモリ内にあり、キャッシュラインが有効な場合：**キャッシュライトスルーヒット**。データはキャッシュとメインメモリ（マスタポートを通じて）の両方に書き込まれます。
- アドレスが TAG メモリにない（またはキャッシュラインが有効でない）：**キャッシュライトスルーミス**。スレーブポートに着信したデータは、メインメモリにのみ書き込まれます（ライトバックミスと違って、ライン割り当てはなく、データがキャッシュに書き込まれることもありません）。

（キャッシュミスによる）キャッシュリフィルの場合、DCACHE は、リフィルデータを書き込むキャッシュラインを選択します。2ウェイセットアソシアティブキャッシュとして、2つのラインのうちの1つを使用できます（2ウェイのそれぞれのアドレスインデックスで指し示されたライン）。ウェイ選択は、pLRU-t 置換アルゴリズムに基づき、これは、各インデックスについて、次のリフィルのためのウェイ候補を指し示します。

リフィルデータを書き込む必要があるキャッシュラインがすでに有効な場合でも、ターゲットのキャッシュラインを最初に無効化する必要があります。

- このラインのダーティタグが 0 に等しい場合（クリーンデータ）、ラインは単に無効化されます。
- 1 に等しい場合（ダーティデータ）、ラインをメインメモリに書き戻す必要があります。

DCACHE はマスタポートにバースト書込みトランザクションを生成します。これは、バーストタイプが INCRw（w はワード単位のキャッシュライン幅）に設定され、AHB メモリトランザクション属性信号が次のように設定されています。

- データ（命令ではない）
- 特権 = TAG 特権ビット
- ライトバック（関係ない場合でも）
- 通常メモリ
- キャッシュ可能
- 割り当て（関係ない場合でも）

– 共有不可

これらの AHB 属性をスレーブポートから伝播することはできません (DCACHE によって送信される他のすべてのトランザクションの場合と違って)。無効化トランザクションは最初のミス トランザクションとは関係がないためです。AHB 属性の設定は固定されますが、特権ビットは無効化されたラインの TAG 特権ビットからコピーされます。

9.4.7 DCACHE セキュリティ

DCACHE は、Armv8-M TrustZone を実装します。

DCACHE の設定レジスタは、システムレベルで保護されます。

9.4.8 DCACHE のメンテナンス

DCACHE はいくつかのメンテナンス操作を備え、ソフトウェアが DCACHE_CR 制御レジスタでプログラムできます。

- **完全検証**：キャッシュ全体を検証します。割り込み不可タスクです。

ソフトウェアは、DCACHE_CR の CACHEINV をプログラムすることによって、DCACHE の内容全体を検証できます。

CACHEINV = 1 のとき、DCACHE 制御ロジックは DCACHE_SR ステータスレジスタの BUSYF フラグをセットし、キャッシュ無効化操作を実行して、各 TAG 有効ビットを 0 にリセットします (キャッシュラインごとに 1 つの有効ビット)。次のキャッシュライン検証時の未知の値を避けるために、キャッシュ無効化時には、各ダーティおよび特権ビットもリセットされます。CACHEINV は自動的にクリアされます。

完全無効化操作が終了すると、DCACHE は BUSYF フラグを自動的にクリアして、DCACHE_SR の BSYENDF をセットします。

このフラグ条件で有効化された場合 (DCACHE_IER の BSYENDIE = 1)、DCACHE 割り込みがレイズされます。すると、(空の) キャッシュを再び使用できるようになります。

この完全無効化操作は割り込み不可です。つまり、BUSYF = 1 の間は、キャッシュはキャッシュ可能リクエストを扱いません。ただし、キャッシュ不可トラフィックは扱われ (リクエストアドレスが TAG アドレスと比較されないため)、DCACHE は同じクロックサイクルでバイパスされます (DCACHE が無効なときと同じ動作)。

- **範囲無効化**：キャッシュ内の特定の範囲のアドレスを無効化します。バックグラウンドタスク (割り込み可能)。

ソフトウェアは、アドレス範囲が DCACHE_CMDRSADDR (範囲開始アドレス) と DCACHE_CMDREADDR (範囲終了アドレス) にプログラムされた後、DCACHE_CR の STARTCMD = 1 および CACHECMD = 0b010 をプログラムすることによって、DCACHE 内の特定のデータ領域を無効化できます。

その場合、DCACHE 制御ロジックは TAG メモリ全体を解析します。読み出しラインアドレス (TAG アドレス + ラインインデックス) が、プログラムされたアドレス範囲に該当する場合 (DCACHE_CMDRSADDR ≤ ラインアドレス ≤ DCACHE_CMDREADDR)、対応するキャッシュラインは無効化されます (ラインの TAG ビットがクリアされ、有効ビット = ダーティビット = 特権ビット = 0 になります)。

STARTCMD がセットされると、DCACHE 制御ロジックは DCACHE_SR の BUSYCMDF をセットし、範囲無効化操作を開始します。STARTCMD も自動的にクリアされます。

操作が完了すると (すべての TAG メモリが解析されると)、DCACHE は DCACHE_SR の BUSYCMDF を自動的にクリアし、CMDENDF をセットします。

このフラグ条件で有効化されると (DCACHE_IER の CMDENDIE = 1)、DCACHE 割り込みがレイズされます。

範囲無効化操作中、DCACHE は割込み可能です。つまり、無効化プロセスより優先度の高い新しい着信リクエストを受け入れることができます。TAG メモリは、外部キャッシュリクエストによってすでにアクセスされていない場合のみ、範囲無効化操作のためにアクセスされます。このため、範囲無効化の実行は、通常、一气に行われるのではなく、割り込まれることがあります。

- **範囲クリーン:** キャッシュ内の特定の範囲のアドレスをクリーンします。バックグラウンドタスク（割込み可能）。

キャッシュラインのクリーンとは、メインメモリの内容が最新であり、キャッシュで変更された可能性のあるデータも含んでいることを確認することを意味します。クリーン操作では、「ダーティ」としてタグ付けされているキャッシュライン（TAG ダーティビットがセットされているもの）がメインメモリにライトバックされます。

ソフトウェアは、アドレス範囲が DCACHE_CMDRSADDRR（範囲開始アドレス）と DCACHE_CMDREADRR（範囲終了アドレス）にプログラムされた後、DCACHE_CR の STARTCMD = 1 および CACHEDCMD = 0b001 をプログラムすることによって、DCACHE 内の特定のデータ領域をクリーンできます。

その場合、DCACHE 制御ロジックは TAG メモリ全体を解析します。読出しラインアドレス（TAG アドレス + ラインインデックス）が、プログラムされたアドレス範囲に該当し

（DCACHE_CMDRSADDRR ≤ ラインアドレス ≤ DCACHE_CMDREADRR）、対応するラインがダーティの場合、このラインはクリーンされます。つまり、キャッシュライン全体が DCACHE マスタポートを通じてメモリに書き戻され、TAG ダーティビットがクリアされます。

STARTCMD がセットされると、DCACHE 制御ロジックは DCACHE_SR の BUSYCMDF をセットし、範囲クリーン操作を開始します。STARTCMD も自動的にクリアされます。

操作が完了すると（すべての TAG メモリが解析されると）、DCACHE は DCACHE_SR の BUSYCMDF を自動的にクリアし、CMDENDF をセットします。

このフラグ条件で有効化されると（DCACHE_IER の CMDENDIE = 1）、DCACHE 割込みがレイズされます。

範囲クリーン操作中、DCACHE は割込み可能です。つまり、クリーニングプロセスより優先度の高い新しい着信リクエストを受け入れることができます。TAG メモリは、外部キャッシュリクエストによってすでにアクセスされていない場合のみ、範囲クリーン操作のためにアクセスされます。このため、範囲クリーンの実行は、通常、一气に行われるのではなく、割り込まれることがあります。

範囲クリーンが完了するまで、クリーンされる領域の内容をバスイニシエータが変更を試みないようにするのは、ソフトウェアの責任です。そのため、ソフトウェアは、DCACHE_SR の BUSYCMDF フラグを利用でき、このフラグにポーリングして、クリーンされる領域への疑わしいアクセスを防ぐことができます。

または、コマンド終了フラグ（CMDENDF）あるいは DCACHE 割込みを利用して、範囲クリーンの実行終了を検出することもできます。

- **範囲のクリーンおよび無効化:** キャッシュ内の特定の範囲のアドレスをクリーンして無効化します。バックグラウンドタスク（割込み可能）。

この操作は、操作アドレス範囲（範囲クリーン操作と同じ）に属する「ダーティ」なキャッシュラインをクリーンして、このアドレス範囲に属するすべての（有効な）キャッシュラインを（ダーティかどうかに関係なく）無効化します。

ソフトウェアは、アドレス範囲が DCACHE_CMDRSADDRR（範囲開始アドレス）と DCACHE_CMDREADRR（範囲終了アドレス）にプログラムされた後、DCACHE_CR の STARTCMD = 1 および CACHEDCMD = 0b011 をプログラムすることによって、この範囲のクリーンおよび無効化操作を開始できます。

これは、範囲無効化操作または範囲クリーン操作と同じフラグ、および潜在的に同じ割込みをセットおよびクリアします。

9.4.9 DCACHE 性能の監視

DCACHE は、パフォーマンス分析のために以下のモニタを備えています。

- 2 つの 32 ビットの読出しヒットおよび書込みヒットモニタは、DCACHE 入力（スレーブポート）で、DCACHE 出力（マスタポート）でトランザクションを生成しない AHB トランザクションをカウントします。

これらのモニタは、アドレスが TAG メモリ、またはリフィルバッファにある（以前のミスのため、そのデータがキャッシュマスタポートから受信されるか、まもなく受信される）すべてのアクセスも考慮します（[セクション 9.4.6](#) を参照）。

- 2 つの 16 ビットの読出しミスおよび書込みミスモニタは、DCACHE 入力（スレーブポート）で、DCACHE 出力（マスタポート）でトランザクションを生成する AHB トランザクションをカウントします。

これらのモニタは、アドレスが TAG メモリにもリフィルバッファにも存在しないすべてのアクセスも考慮します。

最大値に達しても、これらのモニタは最小値に戻りません。

ソフトウェアは次のタスクを実行できます。

- DCACHE_CR の R(W)HITMEN を通じて、読出し（書込み）ヒットモニタを有効化/停止します。
- DCACHE_CR の R(W)HITMRST を通じて、読出し（書込み）ヒットモニタをリセットします。
- DCACHE_CR の R(W)MISSMEN を通じて、読出し（書込み）ミスモニタを有効化/停止します。
- DCACHE_CR の R(W)MISSMRST を通じて、読出し（書込み）ミスモニタをリセットします。

消費電力を低減するために、これらのモニタはデフォルトでは無効化（停止）されています。

9.4.10 DCACHE ブート

DCACHE はブート時に無効化されます（DCACHE_CR の EN = 0）。

ブートが終了すると、DCACHE を有効にできます（ソフトウェアで DCACHE_CR の EN = 1 をセット）。

9.5 DCACHE 低電力モード

製品レベルでは、DCACHE の使用により、たいいていの場合、サイズが大きく、そのために消費電力も多いメインメモリからではなく、内部 DCACHE との間でデータの読込み/保存を行うことにより、消費電力が低減されます。キャッシュされるメインメモリが外部メモリの場合、低減量はさらに大きくなります。

9.6 DCACHE のエラー管理と割込み

DCACHE マスタポートで開始されたトランザクションがエラーを返すことがあります（読出し専用メモリに書込みを試みた場合など）。マスタポートリクエストがスレーブポートリクエストに伝播された場合、エラーはスレーブポートに再伝播されます。マスタポートリクエストが DCACHE 自体によって開始された場合でも（エビクションまたはクリーン操作により、キャッシュラインがメインメモリに書き戻される）、DCACHE はこの機能エラーを受信し、DCACHE_SR の ERRF フラグをセットすることによって、内部でフラグを立てます。

そのような場合、対応する割込みイネーブルビットがセットされていた場合（DCACHE_IER の ERRIE = 1）、割込みが生成されます。

割込みが生成される可能性のある、もう 1 つのケースは、完全無効化操作の終了時です。つまり、キャッシュビジー状態が終了し、DCACHE が DCACHE_SR の BSYENDF フラグをセットしたときです。

対応する割込みイネーブルビットがセットされていた場合 (DCACHE_IER の BSYENDIE = 1)、割込みが生成されます。

最後のケースは、範囲無効化またはクリーン操作の終了時です。つまり、コマンドビジー状態が終了し、DCACHE が DCACHE_SR の CMDENDF フラグをセットしたときです。

対応する割込みイネーブルビットがセットされていた場合 (DCACHE_IER の CMDENDIE = 1)、割込みが生成されます。

すべての DCACHE 割込みソースは、同一かつ一意な割込み信号 dcache_it をレイズし、同じ割込みベクタを使用します。

表 87. DCACHE 割込み

割込みベクタ	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法
DCACHE	機能エラー	DCACHE_SR の ERRF	DCACHE_IER の ERRIE	DCACHE_FCR の CERRF を 1 にセット
	ビジー状態の終了 (完全無効化の終了)	DCACHE_SR の BSYENDF	DCACHE_IER の BSYENDIE	DCACHE_FCR の CBSYENDF を 1 にセット
	キャッシュ操作の終了 (アドレス範囲ベース)	DCACHE_SR の CMDENDF	DCACHE_IER の CMDENDIE	DCACHE_FCR の CCMDENDF を 1 にセット

また、DCACHE は、すべての AHB バスエラー (セキュリティ問題、アドレスデコード問題など) をマスタポートから S-AHB スレーブポートに再伝播します。

9.7 DCACHE レジスタ

9.7.1 DCACHE 制御レジスタ (DCACHE_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HBURST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WMISS MRST	WHITM RST	WMISS MEN	WHITM EN	RMISS MRST	RHITM RST	RMISS MEN	RHITM EN
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	START CMD	CACHECMD[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	CACHE INV	EN
				w	rw	rw	rw							w	rw

ビット 31 **HBURST** : キャッシュマスタポート読出しアクセスの出力バーストタイプ

EN = 0 のときのみ、ソフトウェアによってセットできます。

マスタポート書き込みアクセスは、常に INCR バーストタイプで行われます。

0 : WRAP

1 : INCR

ビット 30:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **WMISSMRST** : 書込みミスモニタのリセット

0 : 影響なし。

1 : キャッシュ書込みミスモニタをリセットします。

ビット 22 **WHITMRST** : 書込みヒットモニタのリセット

0 : 影響なし。

1 : キャッシュ書込みヒットモニタをリセットします。

ビット 21 **WMISSMEN** : 書込みミスモニタイネーブル

0 : キャッシュ書込みミスモニタのスイッチを切ります。モニタは停止してもリセットされません。

1 : キャッシュ書込みミスモニタを有効にします。

ビット 20 **WHITMEN** : 書込みヒットモニタイネーブル

0 : キャッシュ書込みヒットモニタのスイッチを切ります。モニタは停止してもリセットされません。

1 : キャッシュ書込みヒットモニタを有効にします。

ビット 19 **RMISSMRST** : 読出しミスモニタのリセット

0 : 影響なし。

1 : キャッシュ読出しミスモニタをリセットします。

ビット 18 **RHITMRST** : 読出しヒットモニタのリセット

0 : 影響なし。

1 : キャッシュ読出しヒットモニタをリセットします。

ビット 17 **RMISSMEN** : 読出しミスモニタイネーブル

0 : キャッシュ読出しミスモニタのスイッチを切ります。モニタは停止してもリセットされません。

1 : キャッシュ読出しミスモニタを有効にします。

ビット 16 **RHITMEN** : 読出しヒットモニタイネーブル

0 : キャッシュ読出しヒットモニタのスイッチを切ります。モニタは停止してもリセットされません。

1 : キャッシュ読出しヒットモニタを有効にします。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **STARTCMD** : メンテナンスコマンドを開始します (CACHECMD で定義されたメンテナンス操作)。

EN = 1、BUSYCMD = 0、BUSYF = 0、および CACHECMD = 0b001、0b010、または 0b011 のときのみ、ソフトウェアによって設定できます。

BUSYCMD フラグがセットされたときに (キャッシュメンテナンス操作中)、ハードウェアによってクリアされます。0 を書き込んでも、ビットの値は変化しません。

0 : コマンド操作 (キャッシュメンテナンス) 終了

1 : メンテナンスコマンド (キャッシュメンテナンス) 開始

ビット 10:8 **CACHECMD[2:0]** : キャッシュコマンドメンテナンス操作 (アドレス範囲をクリーンまたは無効化)

メンテナンスコマンドが進行中でないときのみ (BUSYCMD = 0)、ソフトウェアによってセットおよびクリアできます。

000 : 操作なし

001 : 範囲クリーン

010 : 範囲無効化

011 : 範囲クリーンおよび無効化

その他 : 予約済み

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CACHEINV** : 完全キャッシュ無効化

EN = 1 のときのみ、ソフトウェアによってセットできます。

BUSYF フラグがセットされたとき（完全キャッシュ無効化操作中）、ハードウェアによってクリアされます。0 を書き込んでも、ビットの値は変化しません。

0 : 影響なし。

1 : キャッシュ全体を無効化（インバリデート）します（すべてのキャッシュラインの有効ビット = 0）。

ビット 0 **EN** : イネーブル

0 : キャッシュは無効です。（ディセーブル）

1 : キャッシュは有効です。（イネーブル）

9.7.2 DCACHE ステータスレジスタ(DCACHE_SR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDE NDF	BUSYC MDF	ERRF	BSYEN DF	BUSYF
											r	r	r	r	r

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CMDENDF** : コマンド終了フラグ

DCACHE_FCR.CCMDENDF = 1 を書き込むことによってクリアされます。

0 : キャッシュはビジーまたはアイドルです。

1 : CACHEDCMD コマンドは終了しました。

ビット 3 **BUSYCMDF** : コマンドビジーフラグ

0 : キャッシュは CACHEDCMD コマンドでビジーではありません。

1 : キャッシュは CACHEDCMD コマンド（アドレス範囲のクリーンまたは無効化）でビジーです。

ビット 2 **ERRF** : キャッシュエラーフラグ

DCACHE_FCR.CERRF = 1 を書き込むことによってクリアされます。

0 : エラーはありません。

1 : 操作中にエラーが発生しました（エビクションまたはクリーン操作ライトバックエラー）。

ビット 1 **BSYENDF** : 完全無効化ビジー終了フラグ

DCACHE_FCR.CBSYENDF = 1 を書き込むことによってクリアされます。

0 : キャッシュはビジーまたはアイドルです。

1 : 完全無効化 CACHEINV 操作が終了しました。

ビット 0 **BUSYF** : 完全無効化ビジーフラグ

0 : キャッシュは CACHEINV 操作でビジーではありません。

1 : キャッシュは完全無効化 CACHEINV 操作を実行中です。

9.7.3 DCACHE 割込みイネーブルレジスタ(DCACHE_IER)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDE NDIE	Res.	ERRIE	BSYEN DIE	Res.
											rw		rw	rw	

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CMDENDIE** : コマンド終了時の割込みイネーブル

キャッシュコマンド（アドレス範囲のクリーンまたは無効化）の終了時に割込み生成を有効にするために、ソフトウェアによってセットされます。

0 : コマンド終了時の割込みは無効です。

1 : コマンド終了時の割込みは有効です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **ERRIE** : キャッシュエラー時の割込みイネーブル

キャッシュ機能エラー（エビクションまたはクリーン操作ライトバックエラー）発生時に割込み生成を有効にするために、ソフトウェアによってセットされます。

0 : エラー時の割込みは無効です。

1 : エラー時の割込みは有効です。

ビット 1 **BSYENDIE** : ビジー終了時の割込みイネーブル

キャッシュの完全無効化操作の終了時に割込み生成を有効にするために、ソフトウェアによってセットされます。

0 : ビジー終了時の割込みは無効です。

1 : ビジー終了時の割込みは有効です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

9.7.4 DCACHE フラグクリアレジスタ(DCACHE_FCR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCMD ENDF	Res.	CERRF	CBSYE NDF	Res.
											w		w	w	

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

- ビット 4 **CCMDENDF** : コマンド終了フラグのクリア
ソフトウェアによってセットされます。
0 : 影響なし。
1 : DCACHE_SR の CMDENDF フラグをクリアします。
- ビット 3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **CERRF** : キャッシュエラーフラグのクリア
ソフトウェアによってセットされます。
0 : 影響なし。
1 : DCACHE_SR の ERRF フラグをクリアします。
- ビット 1 **CBSYENDF** : 完全無効化ビジー終了フラグのクリア
ソフトウェアによってセットされます。
0 : 影響なし。
1 : DCACHE_SR の BSYENDF フラグをクリアします。
- ビット 0 予約済みであり、リセット値に保持する必要があります。

9.7.5 DCACHE 読出しヒットモニタレジスタ(DCACHE_RHMONR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RHITMON[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RHITMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RHITMON[31:0]** : キャッシュ読出しヒットモニタカウンタ

9.7.6 DCACHE 読出しミスモニタレジスタ(DCACHE_RMMONR)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMISSMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RMISSMON[15:0]** : キャッシュ読出しミスモニタカウンタ

9.7.7 DCACHE 書込みヒットモニタレジスタ(DCACHE_WHMONR)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WHITMON[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WHITMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **WHITMON[31:0]** : キャッシュ書込みヒットモニタカウンタ

9.7.8 DCACHE 書込みミスモニタレジスタ(DCACHE_WMMONR)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WMISSMON[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **WMISSMON[15:0]** : キャッシュ書込みミスモニタカウンタ

9.7.9 DCACHE コマンド範囲開始アドレスレジスタ(DCACHE_CMDRSADDRR)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDSTARTADDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDSTARTADDR[15:4]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:4 **CMDSTARTADDR[31:4]** : DCACHE_CR.CACHECMD フィールドで指定されたキャッシュメンテナンスコマンドが適用される範囲の開始アドレス

このレジスタは、DCACHE_CR.CACHECMD が書き込まれる前にセットする必要があります。
を参照してください。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

9.7.10 DCACHE コマンド範囲終了アドレスレジスタ(DCACHE_CMDREADDRR)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDENDADDR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDENDADDR[15:4]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

ビット 31:4 **CMDENDADDR[31:4]** : DCACHE_CR.CACHECMD フィールドで指定されたキャッシュメンテナンスコマンドが適用される範囲の終了アドレス

このレジスタは、DCACHE_CR.CACHECMD が書き込まれる前にセットする必要があります。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

9.7.11 DCACHE レジスタマップ

表 88. DCACHE レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DCACHE_CR	HBURST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WMISSMRST	WHITMRST	WMISSMEN	WHITMEN	RMISSMRST	RHITMRST	RMISSMEN	RHITMEN	Res.	Res.	Res.	Res.	STARTCMD	CACHECMD [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	CACHEINV	EN
	リセット値	0								0	0	0	0	0	0	0	0					0	0	0	0						0	0	
0x004	DCACHE_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDENDF	BUSYCMDF	ERRF	BSYENDF	BUSYF
	リセット値																												0	0	0	0	1
0x008	DCACHE_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDENDIE	ERRIE	BSYENDIE	Res.	
	リセット値																												0	0	0	0	
0x00C	DCACHE_FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDENDF	CERRF	CBSYENDF	Res.	
	リセット値																												0	0	0		
0x010	DCACHE_RHMONR	RHITMON[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	DCACHE_RMMONR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RMISSMON[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	DCACHE_WHMONR	WHITMON[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	DCACHE_WMMONR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WMISSMON[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 88. DCACHE レジスタのマップとリセット値（続き）

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x028	DCACHE_ CMDRSADDRR	CMDSTARTADDR[31:4]																												Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x02C	DCACHE_ CMDREADDRR	CMDENDADDR[31:4]																												Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

10 電源制御 (PWR)

10.1 概要

電源コントローラは、デバイスの電源と電力モードの遷移を管理します。

10.2 PWR の主な機能

電源コントローラ (PWR) の主な機能を以下に示します。

- 電源供給と電源ドメイン
 - コアドメイン (V_{CORE})
 - V_{DD} ドメイン
 - バックアップドメイン (V_{SW})
 - アナログドメイン (V_{DDA})
 - SMPS パワーステージ用の電源 (SMPS パッケージで使用可能)
 - 10 の I/O (PD6、PD7、PG9:14、PB8、PB9) の V_{DDIO2} 供給
 - USB トランシーバの V_{DDUSB}
- システム電源の電圧レギュレーション
 - SMPS ステップダウンコンバータ
 - リニア電圧レギュレータ (LDO)
- 電源供給の監視
 - POR/PDR モニタ
 - BOR モニタ
 - PVD モニタ
 - AVD モニタ
 - 機能範囲外温度モニタ
 - 機能範囲外バックアップドメイン電圧モニタ
- 電源管理
 - 動作モード
 - 電圧スケーリングの制御
 - 低電力モード
- V_{BAT} バッテリーの充電
- TrustZone セキュリティおよび特権保護

10.3 PWR ピンおよび内部信号

表 89. PWR の入出力ピン

ピン名	信号タイプ	説明
VDD	電源	メイン電源
GND	電源	メイングラウンド
VDDA	電源	アナログペリフェラル電源
VSSA	電源	アナログペリフェラルグラウンド
VDDIO2	電源	独立 I/O 電源
VDDUSB	電源	USB 電源
VCAP	電源	ロジック電源 (V_{CORE})
VBAT	電源	バックアップドメイン電源
VDDSMPS	電源	SMPS 電源
VSSSMPS	電源	SMPS グラウンド
VLXSMPS	電源	SMPS 出力
VREF+	電源	ADC/DAC 高基準電圧
VREF-	電源	ADC/DAC 低基準電圧
WKUPx (x = 1 ~ 8)	入力	ウェイクアップピン
CSLEEP	出力	SLEEP モードの MCU
CDSTOP	出力	STOP モードの CPU

表 90. PWR 内部入力／出力信号

内部信号名	信号タイプ	説明
WKUPx (x = 1 ~ 8)	入力	ウェイクアップイベントソース

10.4 PWR 電源供給と電源ドメイン

図 36. SMPS 付き電源装置

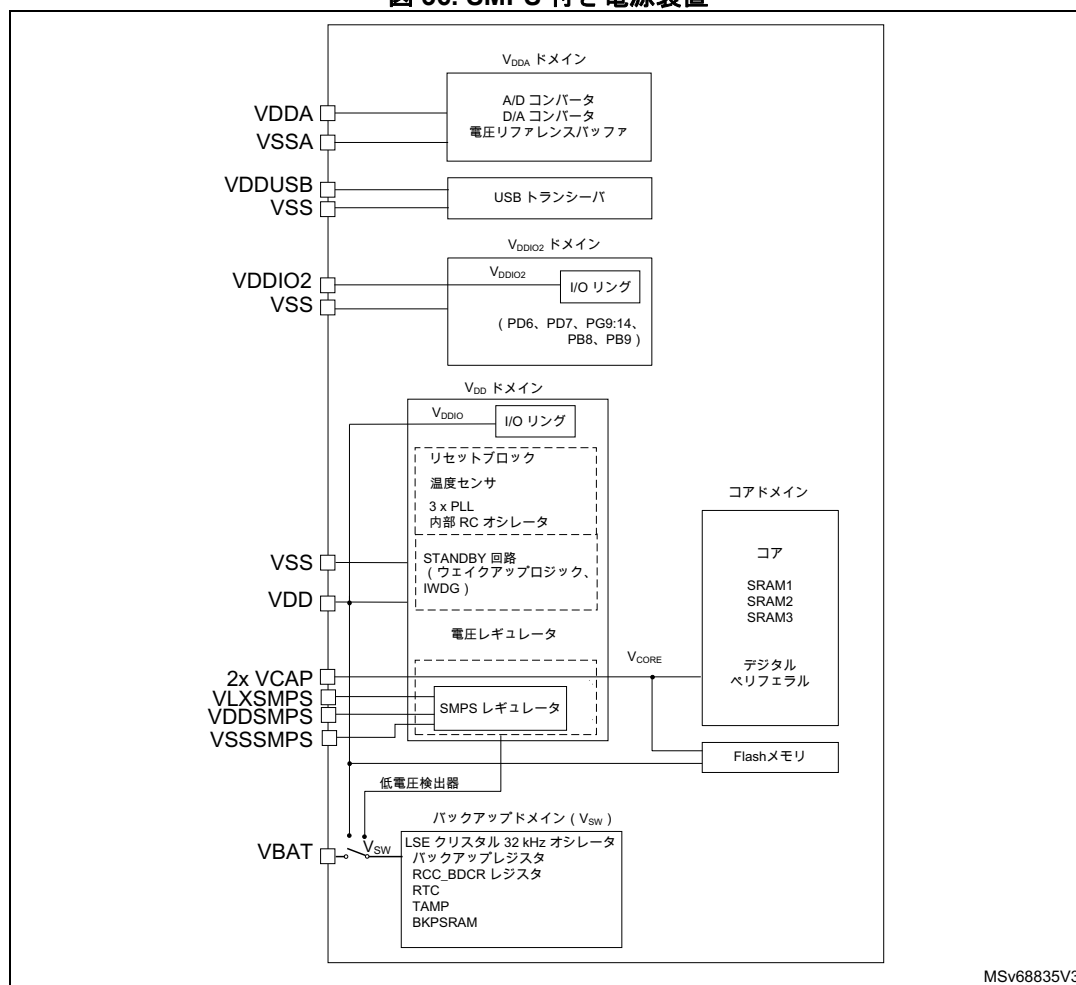
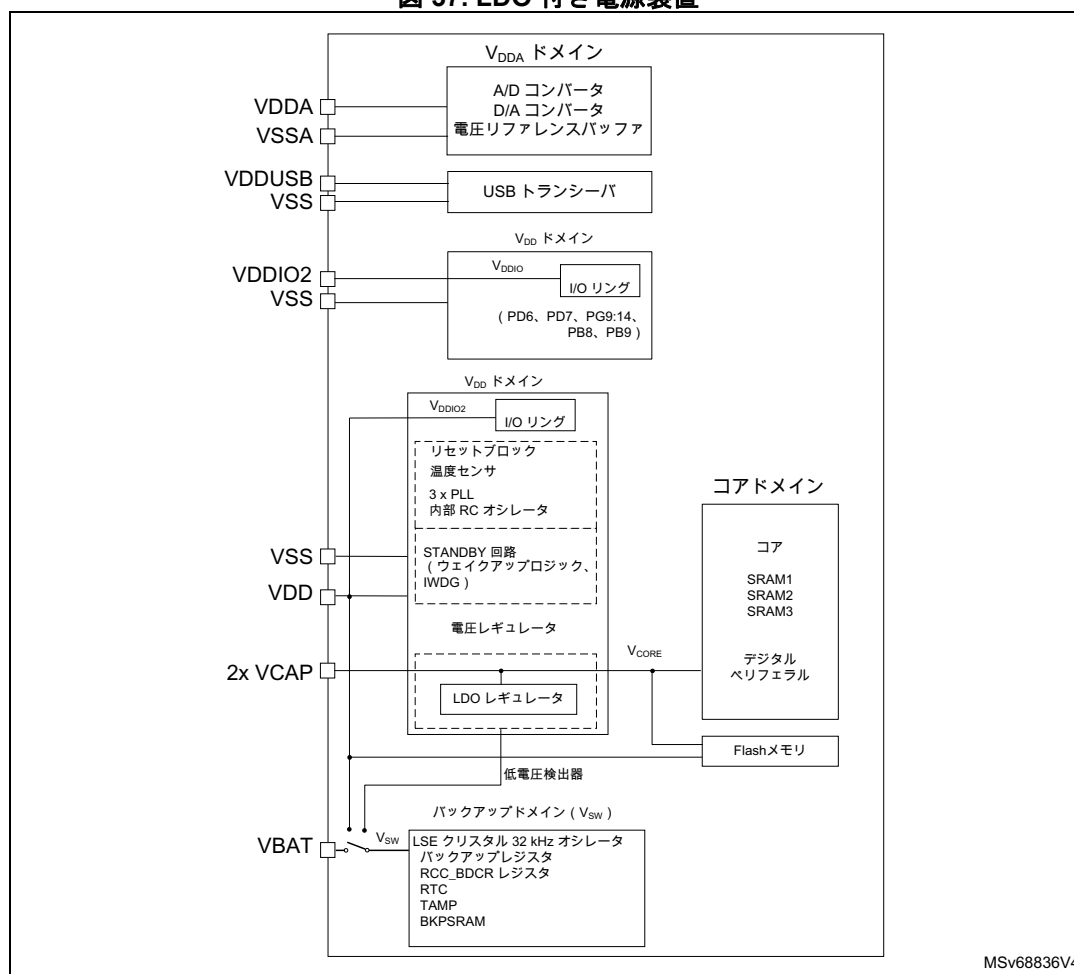


図 37. LDO 付き電源装置



10.4.1 外部電源

デバイスには、1.71 ~ 3.6 V の V_{DD} 動作電圧供給が必要です。特定のペリフェラルにいくつかの独立型電源を供給できますが、VDD ピンに有効な動作電源がない場合は供給しないでください。

- $V_{DD} = 1.71 \sim 3.6 \text{ V}$
 V_{DD} は、I/O、内部レギュレータ、およびシステムアナログ (リセット、電源管理、内部クロックなど) の外部電源です。VDD ピンを通じて外部から供給されます。
- $V_{DDA} = 1.62 \text{ V (ADC、DAC) / 2.1 (VREFBUF)} \sim 3.6 \text{ V}$
 V_{DDA} は、A/D コンバータ、D/A コンバータ、および電圧リファレンスバッファの外部アナログ電源です。 V_{DDA} の電圧レベルは、 V_{DD} 電圧から独立しており、これらのペリフェラルが使用されない場合は、 V_{DD} に接続することを推奨します。
- $V_{DDSMPS} = 1.71 \sim 3.6 \text{ V}$
 V_{DDSMPS} は、SMPS ステップダウンコンバータ用の外部電源です。 V_{DDSMPS} 電源ピンを通じて外部から供給され、VDD ピンと同じ電源に接続される必要があります。
- V_{LXSMPS} は、SMPS ステップダウンコンバータの切替え後の出力です。

注: SMPS 電源ピンは、SMPS ステップダウンコンバータオプション搭載の特定パッケージでのみ使用できます。

- $V_{DDUSB} = 3.0 \sim 3.6 \text{ V}$
 V_{DDUSB} は、USB トランシーバ用の独立した外部電源です。 V_{DDUSB} の電圧レベルは、 V_{DD} 電圧から独立しており、USB が使用されないときは、 V_{DD} に接続することを推奨します。
- $V_{DDIO2} = 1.08 \sim 3.6 \text{ V}$
 V_{DDIO2} は、10 の I/O (PD6、PD7、PG9:14、PB8、PB9) の外部電源です。 V_{DDIO2} の電圧レベルは、 V_{DD} 電圧から独立しており、これらのピンが使用されないときには、 V_{DD} に接続することを推奨します。
- $V_{CAP} = 1.0\text{V} \sim 1.35\text{V}$: デジタルコアドメイン電源
 この電源は、他のすべての電源から独立しています。
 - 電圧レギュレータが有効なとき、 V_{CORE} は内部電圧レギュレータによって供給されます。
 - 電圧レギュレータが無効なとき、 V_{CORE} は VCAP ピンを通じて外部電源によって、または SMPS によって供給されます。
- $V_{BAT} = 1.62 \sim 3.6 \text{ V}$
 V_{BAT} は、RTC、外部クロック 32 kHz オシレータ、バックアップレジスタ、およびオプションでバックアップ SRAM 用の (電源スイッチを通じて) V_{DD} がないときの電源です。
- V_{REF-} 、 V_{REF+}
 V_{REF+} は ADC および DAC の入力ファレンス電圧です。有効時、内部電圧リファレンスバッファの出力にもなります。
 ADC および DAC がアクティブでない場合、 V_{REF+} をアース接続できます。
 内部電圧リファレンスバッファは、 V_{REFBUF_CSR} レジスタの VRS ビットで設定された 4 つの出力電圧をサポートします。
 - $V_{REF+} \sim 1.8 \text{ V}$ ($V_{DDA} \geq 2.1 \text{ V}$ が必要)
 - $V_{REF+} \sim 2.048 \text{ V}$ ($V_{DDA} \geq 2.4 \text{ V}$ が必要)
 - $V_{REF+} \sim 2.5 \text{ V}$ ($V_{DDA} \geq 2.8 \text{ V}$ が必要) V_{REF-} ピンおよび V_{REF+} ピンはすべてのパッケージで使用できるわけではありません。使用可能でないとき、 V_{SSA} と V_{DDA} にそれぞれ結合されます。
 V_{REF+} がパッケージの V_{DDA} と二重結合されている場合、内部電圧リファレンスバッファは使用できず、無効にしておく必要があります。
 V_{REF-} は、常に V_{SSA} と同じである必要があります。

10.4.2 内部レギュレータ

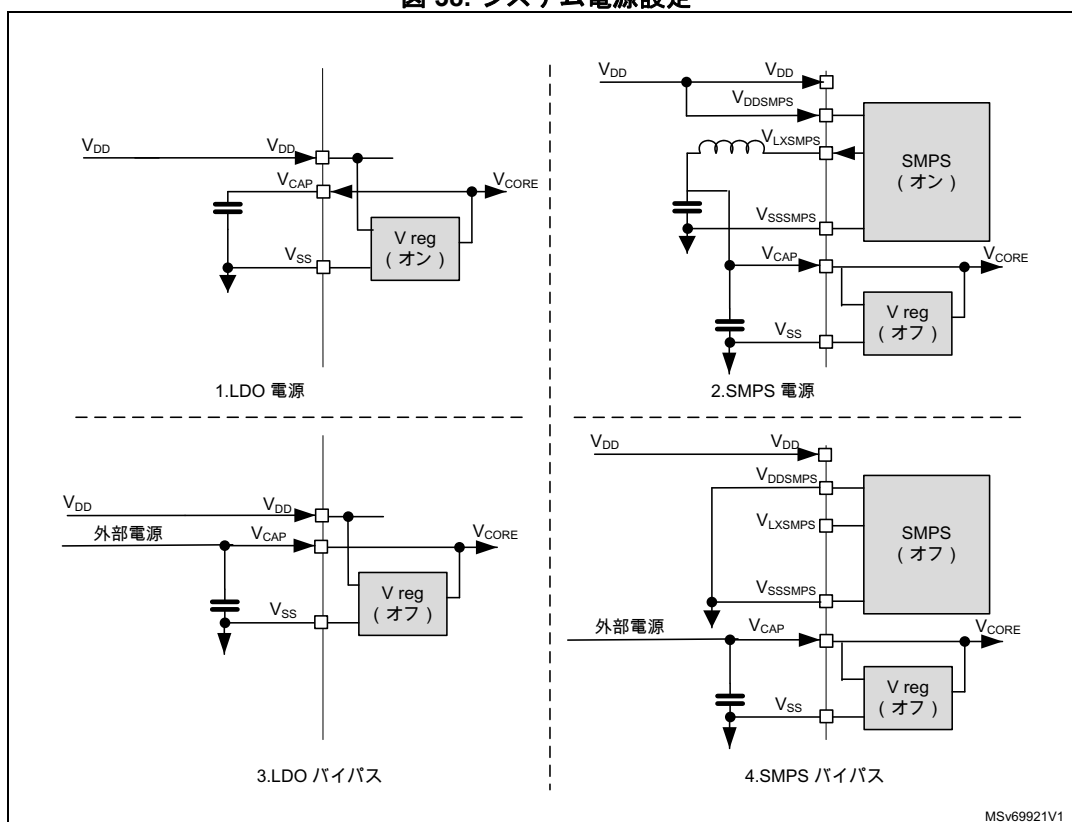
デバイスには 2 つのレギュレータが内蔵されています。デジタルペリフェラル、SRAM (BKPSRAM を除く)、および内蔵 Flash メモリに V_{CORE} 電源を提供する 1 つの LDO または 1 つの SMPS です。SMPS は、この電圧を VCAP (2 つのピン) で合計 10 μF (標準) の外部キャパシタで生成し、2.2 μH (標準) の外部コイルを必要とします。

LDO は、合計 4.7 μF (標準) の外部キャパシタンスで、この電圧を VCAP で生成します。

両方のレギュレータは 4 つの異なる電圧を提供でき (電圧スケーリング)、STOP モードで動作できます。

パッケージ設定 (SMPS または LDO) に応じて、レギュレータはハードウェアによって選択されます。SMPS および LDO レギュレータは、排他的に選択されます。図 38 に示されている電源設定は、 V_{CORE} ドメイン電源についてサポートされます。

図 38. システム電源設定



外部電源から供給される V_{CORE} での起動 (バイパス)

バイパスモードで電源供給されるとき、 V_{DD} が POR 閾値に達する前に、 V_{CORE} がデフォルトレベル ($\geq 1.1\text{ V}$) で安定する必要があります。

パワーアップ後の LDO のデフォルトの状態により (デフォルトで有効)、LDO がソフトウェアによって無効化されるまで、外部 V_{CORE} 電圧は 1.1 V より高く保たれる必要があります。LDO が無効化されると、ユーザアプリケーションのニーズに応じて外部 V_{CORE} 電圧を調整できます (V_{CORE} のレベルと最大動作周波数の詳細については、データシートの「一般動作条件」のセクションを参照)。

バイパスモードで動作するときには、アプリケーションは PWR_VOSCR レジスタのビット VOS[1:0] を使用して、VOS を調整する必要があります。VOS[1:0] は、外部で供給されるコア電圧と関連のパフォーマンスに応じて設定する必要があります。

VOS レベルを調整するには、ソフトウェアは中間レベルを順に選択する必要があります。

- パフォーマンスを高めるとき：
 - まず、電圧スケーリングを増分する必要があります (たとえば、VOS3 から VOS0 に変更するときには、VOS[1:0] ビットで、より低いレベルを選択する必要があります : VOS2、VOS1、次に VOS0)。
 - 外部電圧を増加できます。
 - システム周波数を増加できます。
- パフォーマンスを下げるとき：
 - システム周波数を下げる必要があります。
 - 外部電圧を下げる必要があります。
 - 電圧スケーリングを減分できます (たとえば、VOS1 から VOS3 に変更するときには、VOS[1:0] ビットで、より低いレベルを選択する必要があります : VOS2、次に VOS3)。

10.4.3 パワーアップおよびパワーダウン電源シーケンス

パワーアップおよびパワーダウンフェーズでは、以下の電源シーケンス要件を満たす必要があります。

- V_{DD} が 1 V 未満のときには、他の電源 (V_{DDA} 、 V_{DDIO2} 、 V_{DDUSB}) は $V_{DD} + 300$ mV 未満に保つ必要があります。
- V_{DD} が 1 V を上回るときには、すべての電源は独立となります。

パワーダウンフェーズでは、MCU に供給されるエネルギーが 1 mJ 未満の場合に限り、 V_{DD} を一時的に他の電源より下げることができます。これにより、パワーダウン過渡フェーズ時に外部デカップリングコンデンサを異なる時間定数で放電できます。

10.4.4 独立したアナログペリフェラルの電源

ADC および DAC 変換の精度を高め、電源の柔軟性を拡張するために、アナログペリフェラルには独立した電源が供給されます。この電源は、PCB のノイズを防ぐために、個別にフィルタシールドすることができます。

- アナログペリフェラルの電圧供給入力、個別の V_{DDA} ピンで使用できます。
- 分離された電源グラウンド接続は、VSSA ピンに用意されています。

V_{DDA} 電源電圧は V_{DD} とは異なる場合があります。 V_{DDA} の有無は、 V_{DDA} によって電源供給されるアナログペリフェラル (A/D コンバータ、D/A コンバータ、電圧リファレンスバッファ) のいずれかを有効にする前に確認する必要があります。

電源供給レベル監視は、PWR_VMSR レジスタの AVDO ビットを介して、 V_{DDA} で使用可能です。

単電源が使用されるときには、ノイズフリーの V_{DDA} 基準電圧を確保するために、外部フィルタ回路を通じて V_{DDA} を V_{DD} に外部接続することができます。

ADC および DAC の基準電圧

低電圧入力および出力時の精度を確保するため、 V_{DDA} より低い独立した基準電圧 V_{REF+} に接続できます。 V_{REF+} は、アナログ入力 (ADC) または出力 (DAC) 信号のフルスケール値に相当する最高電圧です。

V_{REF+} は、外部リファレンスまたは内部バッファ付き電圧リファレンス (V_{REFBUF}) のいずれかによって供給されます。内部電圧リファレンスは、設定可能な電圧、つまり、1.8、2.048、または 2.4 V を出力できます。また、内部電圧リファレンスは、 V_{REF+} ピンを通じて外部コンポーネントに電圧を供給できます。詳細については、製品データシートおよび[セクション 29 : 基準電圧バッファ \(\$V_{REFBUF}\$ \)](#)を参照してください。

注： VREF+ ピンおよび VREF- ピンはすべてのパッケージで利用できるわけではありません（それぞれ VDDA と VSSA に内部接続されています）。外部電源が VREF+ ピンに印加されている場合には、内部電圧基準バッファを有効にしないでください。

10.4.5 独立 I/O 電源レール

一部の I/O (PD6、PD7、PG9:14、PB8、PB9) は、個別の電源レールから供給されます。このレールの電源は 1.08 ~ 3.6 V であり、VDDIO2 ピンを通じて外部から供給されます。VDDIO2 の電圧レベルは、VDD または VDDA から完全に独立しています。VDDIO2 ピンは、一部のパッケージでのみ使用できます。I/O のリストについては、関連デバイスのデータシートのピン配置図または表を参照してください。

電源供給レベル監視は、PWR_VMSR レジスタの VDDIO2RDY ビットを介して、VDDIO2 で使用可能です。

10.4.6 独立した USB トランシーバの電源

USB トランシーバは、個別の VDDUSB 電源ピンから供給されます。VDDUSB の範囲は、3.0 ~ 3.6 V であり、VDD または VDDA から完全に独立しています。

電源供給レベル監視は、PWR_VMSR レジスタの USB33RDY ビットを介して、VDDUSB で使用可能です。

PWR USB 電源制御レジスタ (PWR_USBSCR) の USB33SV ビットをセットする前に、PWR 電圧モニタステータスレジスタ (PWR_VMSR) の USB33RDY ビットを監視することによって、VDDUSB が使用可能であることを確認してください。VDD33USB 供給レベル検出器は、PWR USB 電源制御レジスタ (PWR_USBSCR) の USB33DEN ビットを通じて有効化する必要があります。

USB33SV ビットのセットは、USBFS ペリフェラルを使用するために必須です。これは、電気的および論理的アイソレーション目的で VDDUSB 電源を検証するために使用されます。

10.4.7 バックアップドメイン

VDD がオフになったときにバックアップレジスタの内容を保持し、RTC 機能を提供するために、VBAT ピンをバッテリーやその他の電源から供給されるオプションのバックアップ電圧に接続することができます。

VBAT ピンは、RTC ユニット、LSE オシレータ、PI8、および PC13 から PC15 までの I/O に電源を供給するため、主電源がオフの場合でも RTC は動作できます。[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) で BREN ビットがセットされているとき、バックアップ SRAM はオプションで VBAT ピンによって電源供給されます。VBAT 電源への切り替えは、リセットブロックに組み込まれているパワーダウンリセット回路によって制御されます。

警告： t_{RSTTEMPO} (VDD 起動時の過渡期間) 中や PDR の検出後、VBAT と VDD の間の電源スイッチは VBAT に接続されたままになります。
起動フェーズ中、VDD が t_{RSTTEMPO} 以内に規定値に達し (t_{RSTTEMPO} の値については、データシートを参照)、かつ $V_{\text{DD}} > V_{\text{BAT}} + 0.6 \text{ V}$ である場合、電流は VBAT まで VDD と電源スイッチ (VBAT) の間に接続された内部ダイオードを通して注入されます。
VBAT ピンに接続された電源/バッテリーがこの電流注入に対応できない場合は、この電源と VBAT ピンの間に外部低電圧降下ダイオードを接続することを強く推奨します。

外部バッテリーを使用しないアプリケーションでは、 V_{BAT} を V_{DD} に外部接続し、デカップリング用の 100 nF の外部セラミックコンデンサを接続することを推奨します。

バックアップドメインが V_{BAT} (V_{BAT} に接続されたアナログスイッチ) によって電源供給されているときには、以下のピンを使用できます。

- PC13、PI8、PC14、および PC15。RTC または LSE によって設定できます ([セクション 46.3 : RTC の機能説明](#)を参照)。
- PC13、PI8、PA0、PA1、および PA2。TAMP ペリフェラルによってタンパピンとして設定されているとき。

注： アナログスイッチは限られた電流しか流すことできないため、出力モードでの GPIO PC13 ~ PC15 および PI8 の使用には制限があります。最大負荷 30 pF で最大速度 2 mHz に制限する必要があり、これらの I/O を電流ソースとして使用することはできません (たとえば、LED を駆動するなど)。

バックアップドメインアクセス

システムリセット後、バックアップドメイン (RCC バックアップドメイン制御レジスタ RCC_BDCR、RTC レジスタ、TAMP レジスタ、バックアップレジスタ、およびバックアップ SRAM) は、望ましくない書込みアクセスから保護されます。バックアップドメインへのアクセスを可能にするには、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) の DBP ビットをセットして、バックアップドメインへのアクセスを有効にします。

バックアップ RAM

バックアップドメインには、32 ビット、16 ビット、または 8 ビットデータモードでアクセス可能な 4KB のバックアップ RAM があります。バックアップ RAM は、バックアップドメインのバックアップレギュレータから供給されます。PWR_BDCR の BREN ビットを通じてバックアップレギュレータが有効化されると、STANDBY モードや VBAT モードでもバックアップ RAM の内容が保持されます (VBAT が常に存在している場合は、内部 EEPROM とみなすことができます)。

バックアップレギュレータは、アプリケーションが STANDBY モードまたは VBAT モードでバックアップ RAM 機能を必要とするかどうかに応じてオン/オフを切り替えることができます。

バックアップ RAM は読み出し保護され、タンパイイベントが発生すると全体消去されます。これは、機密データ (暗号私鍵など) へのアクセスを防止するためです。

バックアップ RAM は、次の方法で消去できます。

- 製品状態完全復帰後に Flash インタフェースを通じて
- タンパイイベント後
- バックアップドメインリセット後

V_{BAT} バッテリーの充電

V_{DD} が存在する場合、内部抵抗を通して VBAT で外部バッテリーを充電できます。

V_{BAT} の充電は、PWR_BDCR レジスタの VBRS ビットに応じて、5 k Ω または 1.5 k Ω の抵抗を通じて行われます。

バッテリー充電を有効にするには、PWR_BDCR レジスタの VBE ビットをセットします。 V_{BAT} モードで自動的に無効になります。

10.5 PWR システム電源電圧レギュレーション

10.5.1 SMPS および LDO 内蔵レギュレータ

デバイスには 2 つの内部レギュレータが組み込まれていて、パッケージ設定に応じて、ハードウェアによってのみ有効化されます。レギュレータは、パワーオンリセット時に有効化されます。外部ソースから V_{CORE} を供給するために、PWR_SCCR レジスタの BYPASS ビットをセットすることによって、レギュレータを無効にすることができます。

BYPASS ビットは、パワーオンリセット後に一度だけ書き込まれます。ライトワンスメカニズムにより、レジスタはロックされ、それ以後の書き込みアクセスは無視されます。新しい値を書き込む前に、システムを再起動する必要があります。

V_{CORE} が外部ソースから供給されるとき、外部から印加される電圧レベルが PWR_VOSCR レジスタの VOSx ビットに反映される必要があります。

両方のレギュレータは 4 つの異なる電圧を提供でき (電圧スケーリング)、STOP モードで動作できます。

10.5.2 V_{CORE} 電源対リセット、電圧スケーリング、および低電力モード

リセット後、 V_{CORE} は VOS3 です。

STOP または STANDBY モードを終了すると、電圧範囲は VOS3 です。

10.5.3 内蔵電圧レギュレータの動作モード

3 種類の電力モードがあります。RUN、STOP、および STANDBY モードです。

RUN モード

電圧レギュレータ (LDO または SMPS) は、 V_{CORE} ドメイン (コア、メモリ、およびデジタルペリフェラル) に完全な電力を供給します。レギュレータの出力電圧 (LDO または SMPS) は、ソフトウェアによって、PWR 電圧スケーリング制御レジスタ (PWR_VOSCR) の VOS ビットを通じて設定された異なる電圧レベル (VOS0、VOS1、VOS2、および VOS3) にスケーリングできます。

VOS 電圧スケーリングによって、システムが最大周波数より低速のクロックで動作しているときの消費電力を最適化することができます。デフォルトでは、システムリセット後は VOS3 が選択されます。

VOSx ビットは、システムに必要な性能に適應するように動作中に変更することができます。

STOP モード

電圧レギュレータ (LDO または SMPS) は、 V_{CORE} ドメインに電源を供給して、レジスタと内部メモリの内容を保持します。レギュレータのモードは、PWR 電力モード制御レジスタ (PWR_PMCR) の SVOS ビットを通じて選択されます。

STOP モードの消費電力は、SVOS4 (VOS3 より低い電圧レベル) を使用して削減でき、SVOS5 ではさらに削減できます。

STANDBY モード

レギュレータ (LDO または SMPS) はオフであり、 V_{CORE} ドメインはパワーダウンされます。STANDBY 回路とバックアップドメインを除き、レジスタとメモリの内容は失われます。

10.6 PWR 電源および温度監視

電源供給レベルの監視は、以下の電源で使われます。

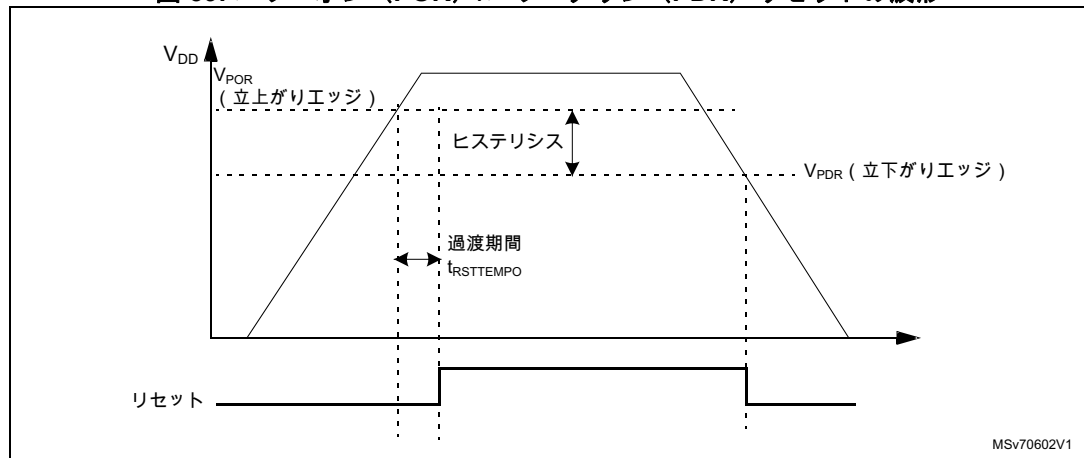
- V_{DD} : POR/PDR (セクション 10.6.1 を参照)、BOR (セクション 10.6.2 を参照)、および PVD モニタ (セクション 10.6.3 を参照) を介して。
- V_{DDA} : AVD モニタ (セクション 10.6.4 を参照) を介して。
- V_{BAT} : V_{BAT} 閾値 (セクション 10.6.6 を参照) を介して
- V_{DDIO2} : VDDIO2RDY ビット (セクション 10.11.10 を参照) を介して。
- 温度監視 (セクション 10.6.7 を参照)

10.6.1 パワーオンリセット (POR) / パワーダウンリセット (PDR)

このシステムには、適切な起動操作を確実に実行する POR/PDR 統合回路が搭載されています。

システムは、 V_{DD} が規定の V_{POR} の閾値を下回っている場合は、外部のリセット回路を必要とせずに、リセットモードを維持します。供給レベルが V_{POR} の閾値を上回ると、システムはリセットモードから解除されます (図 39 を参照)。リセット閾値の詳細については、データシートの電気的特性のセクションを参照してください。

図 39. パワーオン (POR) / パワーダウン (PDR) リセットの波形



1. 閾値とヒステリシス値については、データシートを参照してください。

10.6.2 ブラウンアウトリセット (BOR)

パワーオン時、ブラウンアウトリセット (BOR) は、 V_{DD} 電源電圧が規定の V_{BOR} の閾値に達するまで、システムをリセット状態に維持します。

V_{BOR} の閾値は、システムオプションバイトを介して設定されます。

- BOR OFF (BORH_EN = 0)
- BOR レベル 1 (V_{BOR1})
- BOR レベル 2 (V_{BOR2})
- BOR レベル 3 (V_{BOR3})

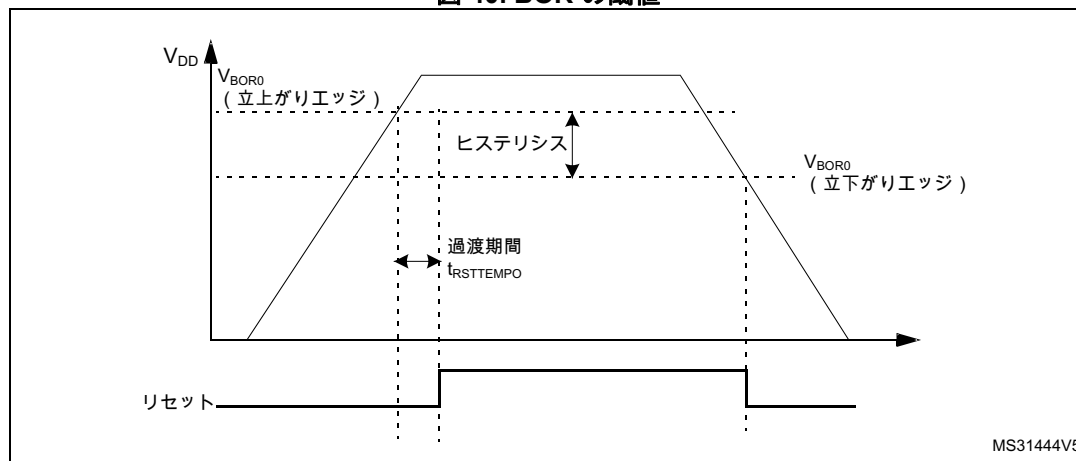
デフォルトでは、BOR は OFF であり、BORH_EN オプションビットをセットすることによって有効化できます。

ブラウンアウトリセットの閾値の詳細については、製品データシートの「電気的特性」セクションを参照してください。

BOR を有効にし、 V_{DD} 電源電圧が選択された V_{BOR} の閾値を下回ると、システムリセットが生成されます。

BOR は、BORH_EN オプションビットを 0 にプログラミングすることによって無効にできます。BOR 機能を無効にするには、システムオプションバイトのプログラミングシーケンスを開始するために、 V_{DD} を POR 閾値よりも高く設定する必要があります。BOR が無効化されると、パワーダウンは PDR によって監視されます。

図 40. BOR の閾値



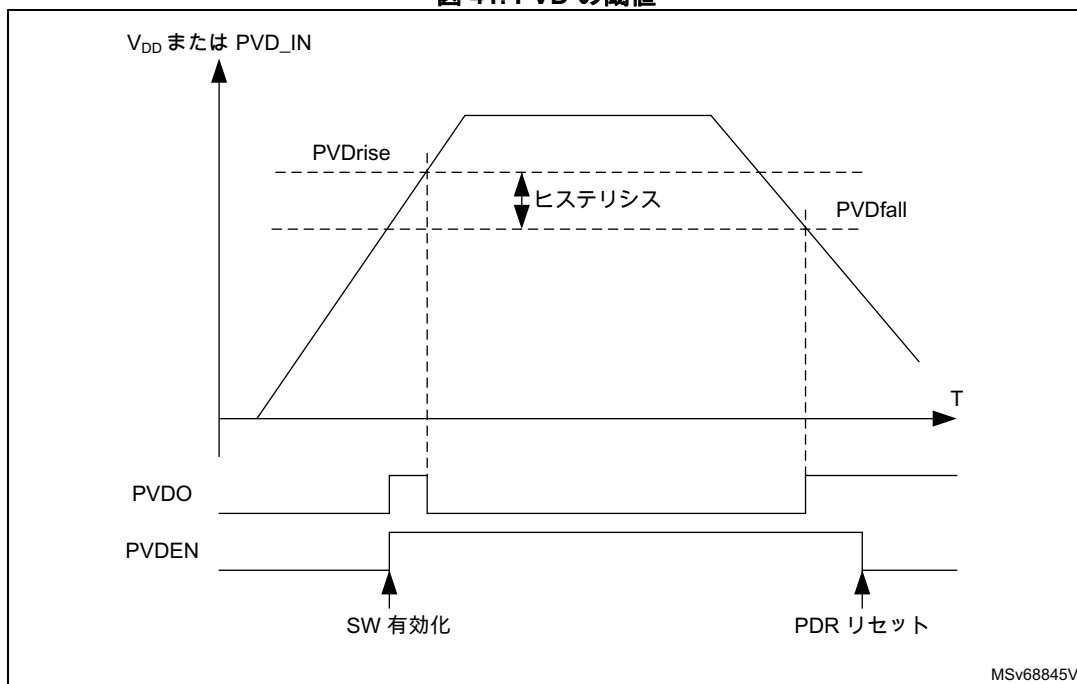
10.6.3 プログラム可能な電圧検出器 (PVD)

PVD を使用して、PWR 電圧モニタ制御レジスタ (PWR_VMCR) の PLS[2:0] ビットで選択した閾値と比較することで、 V_{DD} 電源を監視できます。PVD は、PVD_IN ピンの電圧レベルの監視にも使用できます。この場合、PVD_IN 電圧は内部 VREFINT レベルと比較されます。

PVD は、PWR 電圧モニタ制御レジスタ (PWR_VMCR) の PVDE ビットをセットすることで有効になります。

PWR 電圧モニタステータスレジスタ (PWR_VMSR) の PVDO フラグを使用して、 V_{DD} または PVD_IN 電圧が PVD の閾値よりも高いか低いかを示すことができます。このイベントは EXTI に内部接続され、また EXTI レジスタで有効化されている場合は、割込みを生成することができます。EXTI ラインの立ち上がり/立下がりエッジ感度は、PVD 出力動作に応じて設定する必要があります。たとえば、EXTI ラインが立ち上がりエッジ感度に設定されている場合、 V_{DD} または PVD_IN 電圧が PVD 閾値を下回ると、割込みが生成されます。その場合、サービスルーチンは、緊急シャットダウンを開始できます。

図 41. PVD の閾値



1. 閾値とヒステリシス値については、データシートを参照してください。

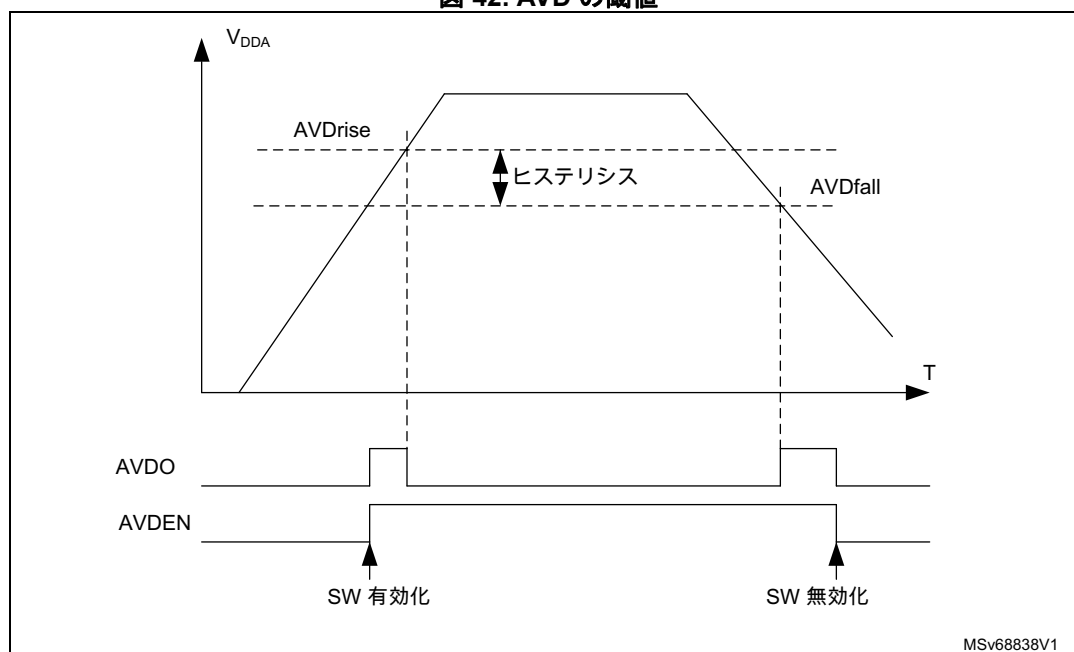
10.6.4 アナログ電圧検出器 (AVD)

AVD を使用して、[PWR 電圧モニタ制御レジスタ \(PWR_VMCR\)](#)の ALS[1:0] ビットで選択した閾値と比較することで、 V_{DDA} 電源を監視できます。

AVD は、[PWR 電圧モニタ制御レジスタ \(PWR_VMCR\)](#) の AVDEN ビットをセットすることで有効になります。

[PWR 電圧モニタステータスレジスタ \(PWR_VMSR\)](#)の AVDO フラグを使用して、 V_{DDA} が AVD の閾値よりも高いか低いかを示すことができます。このイベントは EXTI に内部接続され、EXTI レジスタで有効化されている場合は、割込みリクエストを生成させることができます。EXTI 立上がり/立下がりエッジの設定に応じて、 V_{DDA} が AVD 閾値を下回るか上回ったとき、あるいはその両方で、AVDO 割込みを生成することができます。たとえば、サービスルーチンは、 V_{DDA} 電源がいつ最小レベルを下回るかを示すことができます。

図 42. AVD の閾値



1. 閾値とヒステリシス値については、データシートを参照してください。

10.6.5 V_{DDIO2} 電圧モニタ (IO2VM)

IO2VM は、独立電源電圧 V_{DDIO2} を監視して、ペリフェラルが機能電源範囲にあることを確認します。 $V_{DDIO2RDY}$ フラグ ([PWR 電圧モニタ制御レジスタ \(PWR_VMCR\)](#) を参照) は、有効な電源が存在するかどうかを示します。

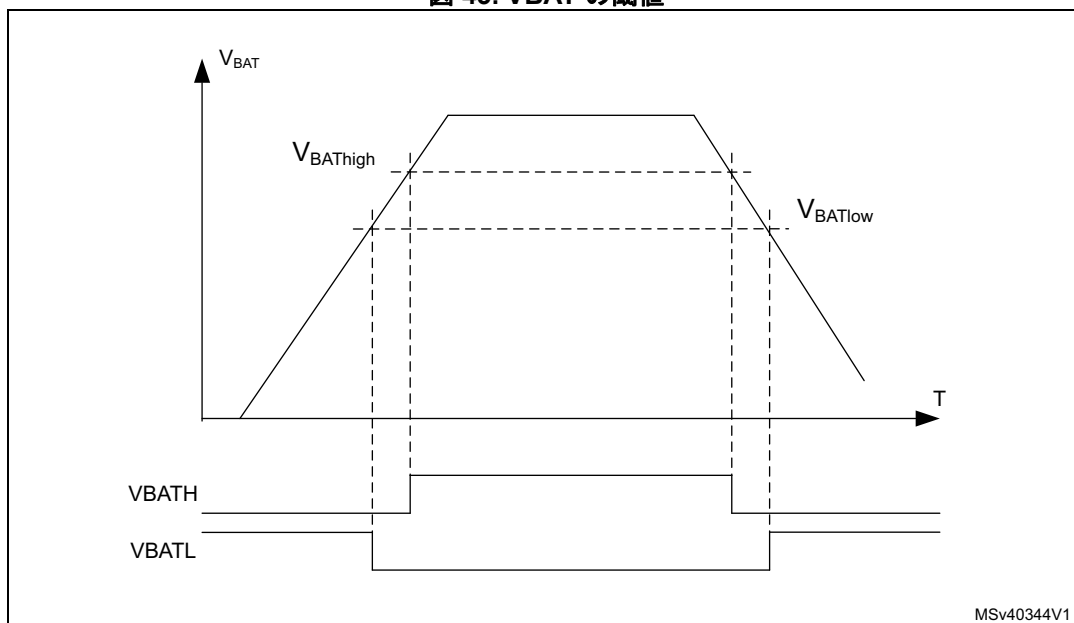
10.6.6 バックアップドメイン電圧監視

VBAT モードでは、バッテリー電源（バックアップドメイン）は、2 つの閾値レベルと比較することで監視できます。 $V_{BAThigh}$ と V_{BATlow} です。VBAT 電源監視は、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) の MONEN ビットによって有効 / 無効にできます。有効にすると、バッテリー電圧の閾値によって消費電力が増加します。

バックアップドメイン電圧監視内部タンパが TAMP ペリフェラルで有効な場合 (TAMP_CR1 レジスタの ITAMP1E = 1)、バッテリー電圧が機能範囲を上回ると、タンパイイベントが生成されます。

注： バックアップドメイン電圧は、存在するときには V_{DD} であり、そうでない場合は V_{BAT} です。

図 43. VBAT の閾値



MSv40344V1

1. 閾値とヒステリシス値については、データシートを参照してください。

10.6.7 温度監視

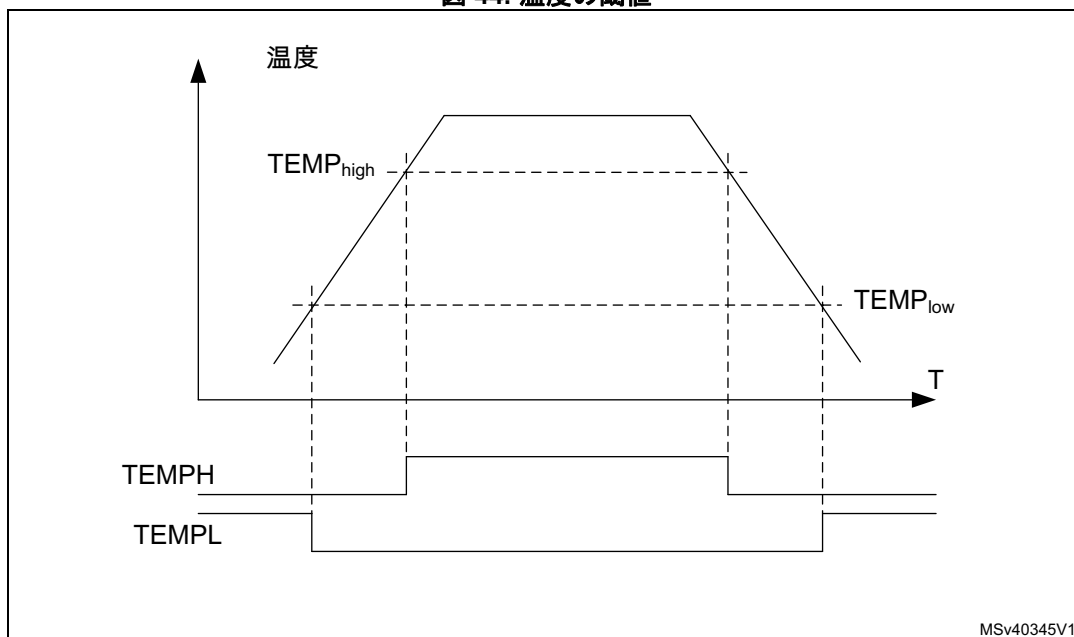
電源制御には専用の温度センサセルが内蔵されています。接合温度は、TEMPhigh と TEMPLOW の 2 つの閾値レベルと比較することで監視できます。[PWR バックアップドメインステータスレジスタ \(PWR_BDSR\)](#) の TEMPH および TEMPL フラグは、デバイスの温度が閾値よりも高いか低いを示します。温度監視は、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) の MONEN ビットによって有効/無効にすることができます。

有効にすると、温度の閾値によって消費電力が増加します。たとえば、このレベルを使用して温度制御タスクを実行するルーチンをトリガすることなどが可能です。

温度監視内部タンパが TAMP ペリフェラルで有効な場合 (TAMP_CR1 レジスタの ITAMP2E = 1)、温度が機能範囲を上回るか下回ると、タンパイベントが生成されます。

TEMPH および TEMPL ウェイクアップ割込みは、RTC タンパ信号で使用できます ([セクション 47: タンパおよびバックアップレジスタ \(TAMP\)](#) を参照)。

図 44. 温度の閾値



1. 閾値とヒステリシス値については、データシートを参照してください。

10.7 PWR 管理

10.7.1 電圧スケーリング

電圧スケーリングをサポートする電圧レギュレータは、以下の機能を備えています。

- RUN モードの電圧スケーリング
 - VOS0 : スケール 0
 - VOS1 : スケール 1
 - VOS2 : スケール 2
 - VOS3 : スケール 3
- STOP モードの電圧スケーリング
 - SVOS3 : スケール 3
 - SVOS4 : スケール 4
 - SVOS5 : スケール 5

電圧スケーリング値の詳細については、製品データシートを参照してください。

リセット後、システムは最低の RUN モード電圧スケーリング (VOS3) で開始します。電圧スケーリングは、必要なシステム性能に合わせて、PWR_VOSCR レジスタの VOS ビットをプログラムすることで、ソフトウェアによってオンザフライで変更できます。STOP モードまたは STANDBY モードを終了すると、RUN モードの電圧スケーリングはデフォルトの VOS3 値にリセットされます。

STOP モードに移行する前に、ソフトウェアは [PWR 電力モード制御レジスタ \(PWR_PMCR\)](#) で SVOS レベルを事前選択する必要があります。SVOS4 および SVOS5 の STOP モードの電圧スケーリングでは、消費電力をさらに低減するために、電圧レギュレータを低消費電力モードに設定します。

10.7.2 電源管理の例

RUN モードでの V_{CORE} 電圧スケーリングの動作の例。

図 45. RUN モードでのダイナミック電圧スケーリング

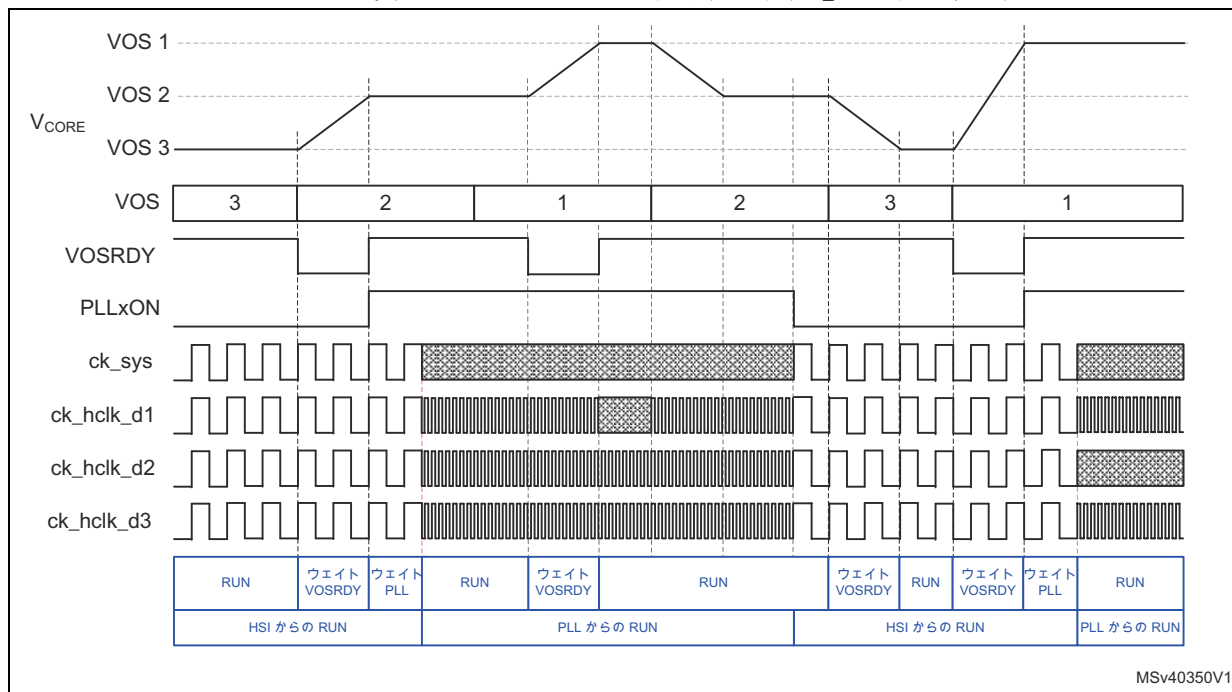


図 45 に、以下のシステム動作シーケンスの例を示します。

- リセット後、システムは VOS3 設定で HSI から開始します。
- システム性能は、最初に電圧スケーリング VOS2 の設定で PLL からミディアムスピードクロックに上げられます。このためには、次の操作を行います。
 - 電圧スケーリングを VOS2 にプログラムします。
 - V_{CORE} 電源が VOSRDY が示す必要なレベルに達したら、PLL を有効にしてクロック周波数を増加します。
 - PLL がロックされたら、システムクロックを切り替えます。
- 電圧スケーリング VOS1 の設定で PLL からハイスピードクロックにすることでシステム性能を向上できます。このためには、次の操作を行います。
 - 電圧スケーリングを VOS1 にプログラムします。
 - V_{CORE} 電源が VOSRDY によって示される必要なレベルに達すると、クロック周波数を増加します。
- その後、システム性能は、電圧スケーリング VOS2 の設定でミディアム・スピード・クロックまで低下します。このためには、次の操作を行います。
 - まず、システム周波数を下げます。
 - 次に、電圧スケーリングを VOS2 に下げます。

5. 次のステップでは、システム性能を電圧スケーリング VOS3 の設定で HSI クロックまで低下させます。このためには、次の操作を行います。
 - a) クロックを HSI に切り替えます。
 - b) PLL を無効にします。
 - c) 電圧スケーリングを VOS3 に下げます。
6. その後、システム性能は、PLL からハイスピードクロックに上げることができます。このためには、次の操作を行います。
 - a) 電圧スケーリングを VOS1 にプログラムします。
 - b) V_{CORE} 電源が VOSRDY が示す必要なレベルに達したら、PLL を有効にしてクロック周波数を増加します。
 - c) PLL がロックされたら、システムクロックを切り替えます。

システム性能（クロック周波数）が変わると、VOS を適切にセットする必要があります。セットされない場合、システムが信頼できなくなる場合があります。

10.8 電源モード

デフォルトでは、マイクロコントローラは、システムまたは電源のリセット後は RUN モードです。外部イベント待ちなど、CPU を実行状態に保つ必要がないときに電力を節減するために、いくつかの低電力モードを使用できます。ユーザは、消費電力の節減、スタートアップ時間の短縮、およびウェイクアップソースを考慮して、最適なモードを選択できます。

デバイスは、次の低電力モードを備えています。

- SLEEP モード
CPU クロックオフ。NVIC、SysTick など、Cortex-M33 コアを含むすべてのペリフェラルが実行でき、割込みやイベントが発生したときに CPU をウェイクアップできます。[セクション 10.8.4: SLEEP モード](#)を参照してください。
- STOP モード
STOP モードは、SRAM とレジスタの内容を保持したまま、最低の消費電力を達成します。コアドメインのすべてのクロックが停止します。PLL、HSE クリスタルオシレータ、HSI (HSIKERON がセットされている場合を除く)、HSI48、および CSI RC (CSIKERON がセットされている場合を除く) は無効になります。LSE や LSI は動作を継続します。
RTC をアクティブなままにできます (RTC ありの STOP モード、RTC なしの STOP モード)。
STOP モードを終了するとき、システムクロックはソフトウェア設定に応じて、最大 64 mHz の HSI または CSI になります。
[セクション 10.8.5 : STOP モード](#)を参照してください。
- STANDBY モード
このモードは、BOR によって最低の消費電力を達成します。内部レギュレータがオフになるため、コアドメインはパワーオフになります。PLL、HSI RC、HSI48、CSI RC、および HSE クリスタルオシレータもオフになります。
RTC をアクティブなままにできます (RTC ありの STANDBY モード、RTC なしの STANDBY モード)。
ブラウンアウトリセット (BOR) は、STANDBY モードでは常にアクティブです。
STANDBY モード中、I/O の状態を保持できます (STANDBY モードによって使用される I/O を除く)。
STANDBY モードに入った後、バックアップドメインおよび STANDBY 回路のレジスタとバックアップ SRAM を除き、SRAM とレジスタの内容は失われます。

デバイスは、外部リセット (NRST ピン)、IWDG リセット、WKUP ピンイベント (設定可能な立上がりまたは立下がりエッジ)、RTC イベントの発生 (アラーム、周期的ウェイクアップ、タイムスタンプ)、またはタンパ検出時に、STANDBY モードを終了します。タンパ検出は、外部ピンによって、または内部エラー検出によってレイズされます。

ウェイクアップ後のシステムクロックは、32 mHz の HSI です。

[セクション 10.8.6 : STANDBY モード](#)を参照してください。

表 91 に電力モードの概要を示します。

表 91. 低消費電力モードの概要

モード名	エントリ	ウェイクアップソース ⁽¹⁾	ウェイクアップシステムクロック	クロックへの影響	電圧レギュレータ
SLEEP (Sleep-now または Sleep-on-exit)	WFI または ISRからの復帰	割込み	SLEEP モードへの 移行前と同様	– CPU クロックオフ – 他のクロックまたは アナログクロック ソースには影響なし	VOS3、 VOS2、 VOS1、 または VOS0
	WFE	ウェイクアップ イベント			
STOP	LPMS = 0 + SLEEPDEEP ビット + WFI または ISR あるいは WFE からの復帰	– 任意の EXTI ライン (EXTI レジスタで 設定) – 特定の ペリフェラル イベント ⁽²⁾	– CSI。RCC_CFGR の STOPWUCK = 1 の とき。 – HSI。STOP モードに 入る前の周波数、 最大 64 mHz、 STOPWUCK = 0 の とき。	– LSI および LSE を除 くすべてのクロック オフ – HSI または CSI は、 ソフトウェアによっ てリクエストされた とき、一時的に有効 にできます。	SVOS3、 SVOS4、 または SVOS5
STANDBY	LPMS = 1 + SLEEPDEEP ビット + WFI または ISR あるいは WFE からの復帰	WKUP ピンのエッジ、 RTC イベント、 IWDG リセット、 NRST ピンの 外部リセット	64 mHz の HSI クロック	LSI および LSE を除 くすべてのクロックオフ	OFF

1. 表 92 を参照してください。

2. ペリフェラルはシステムを STOP モードからウェイクアップできます (これは、STOP モードに入る前に SVOS3 が選択されたときのみ可能です)。

表 92. 動作モードに応じた機能⁽¹⁾

ペリフェラル	RUN	SLEEP	STOP				STANDBY		VBAT
			利用可能	ウェイクアップ機能			利用可能	ウェイク アップ機能	
				SVOS3	SVOS4	SVOS5			
CPU	Y	-	-	-	-	-	-	-	-
Flashメモリ（2 MB）	O	O	(2)	-	-	-	-	-	-
SRAM1（256 KB）	Y ⁽³⁾	Y ⁽³⁾	O ⁽⁴⁾	-	-	-	-	-	-
SRAM2（64 KB）	Y ⁽³⁾	Y ⁽³⁾	O ⁽⁴⁾	-	-	-	-	-	-
SRAM3（320 KB）	Y ⁽³⁾	Y ⁽³⁾	O ⁽⁴⁾	-	-	-	-	-	-
BKPSRAM	O	O	O	-	-	-	O	-	O
FMC	O	O	-	-	-	-	-	-	-
OCTOSPI1	O	O	-	-	-	-	-	-	-

表 92. 動作モードに応じた機能⁽¹⁾ (続き)

ペリフェラル	RUN	SLEEP	STOP				STANDBY		VBAT
			利用可能	ウェイクアップ機能			利用可能	ウェイクアップ機能	
				SVOS3	SVOS4	SVOS5			
バックアップレジスタ	Y	Y	Y	-	-	-	Y	-	Y
ブラウンアウトリセット (BOR)	Y	Y	Y	-	-	-	Y	-	-
プログラム可能な電圧検出器 (PVD)	O	O	O	O	O	O	-	-	-
アナログ電圧検出器 (AVD)	O	O	O	O	O	O	-	-	-
GPDMA	O	O	-	-	-	-	-	-	-
高速内部 (HSI)	O	O	-	-	-	-	-	-	-
オシレータ HSI48	O	O	-	-	-	-	-	-	-
高速外部 (HSE)	O	O	-	-	-	-	-	-	-
低速内部 (LSI)	O	O	O	O	-	-	O	-	-
低速外部 (LSE)	O	O	O	-	-	-	O	-	O
低電力 RC オシレータ (CSI)	O	O	-	-	-	-	-	-	-
クロックセキュリティシステム (CSS)	O	O	-	-	-	-	-	-	-
LSE のクロックセキュリティシステム	O	O	O	O	O ⁽⁵⁾	O ⁽⁵⁾	O	O	-
バックアップドメイン電圧および温度監視	O	O	O	O	O ⁽⁵⁾	O ⁽⁵⁾	O	O	O
RTC/TAMP	O	O	O	O	O	O	O	O	O
TAMP タンパピンの数	8	8	8	-	-	-	4	-	2
USB FS、UCPD	O	O	O	O	-	-	-	-	-
USARTx	O	O	O	O	-	-	-	-	-
低電力 UART (LPUART)	O	O	O	O	-	-	-	-	-
I2Cx (x = 1、2、3、4)	O	O	O	O	-	-	-	-	-
I3C1	O	O	O	O	-	-	-	-	-
HDMI_CEC	O	O	O	O	-	-	-	-	-
SPIx (x = 1..6)	O	O	O	O	-	-	-	-	-
FDCANx	O	O	-	-	-	-	-	-	-
SDMMCx	O	O	-	-	-	-	-	-	-
Ethernet	O	O	O	O	-	-	-	-	-
SAIx	O	O	-	-	-	-	-	-	-
ADCx (x = 1、2)	O	O	-	-	-	-	-	-	-
DAC1 (2 つのコンバータ)	O	O	O	-	-	-	-	-	-
VREFBUF	O	O	O	-	-	-	-	-	-
温度センサ (DTS)	O	O	O	O	-	-	-	-	-
タイマ (TIMx)	O	O	-	-	-	-	-	-	-

表 92. 動作モードに応じた機能⁽¹⁾ (続き)

ペリフェラル	RUN	SLEEP	STOP				STANDBY		VBAT
			利用可能	ウェイクアップ機能			利用可能	ウェイクアップ機能	
				SVOS3	SVOS4	SVOS5			
低電力タイマ LPTIMxx = 1..6)	O	O	O	O	-	-	-	-	-
独立型ウォッチドッグ (IWDG)	O	O	O	O	O ⁽⁵⁾	O ⁽⁵⁾	O	O	-
ウィンドウ型ウォッチドッグ (WWDG)	O	O	-	-	-	-	-	-	-
SysTick タイマ (SYSTICK)	O	O	O	-	-	-	-	-	-
デジタルカメラインタフェース (DCMI)	O	O	-	-	-	-	-	-	-
パラレル同期スレーブインタフェース (PSSI)	O	O	-	-	-	-	-	-	-
CORDIC コプロセッサ (CORDIC)	O	O	-	-	-	-	-	-	-
フィルタ演算アクセラレータ (FMAC)	O	O	-	-	-	-	-	-	-
乱数発生器 (RNG)	O	O	-	-	-	-	-	-	-
AES およびセキュア AES (AES、SAES)	O	O	-	-	-	-	-	-	-
公開鍵アクセラレータ (PKA)	O	O	-	-	-	-	-	-	-
オンザフライ復号 (OTFDEC)	O	O	-	-	-	-	-	-	-
HASH アクセラレータ	O	O	-	-	-	-	-	-	-
CRC 計算ユニット	O	O	-	-	-	-	-	-	-
GPIO	O	O	-	-	-	-	O ⁽⁶⁾	O ⁽⁷⁾	-
EXTI	O	O	O	O	O	O	-	-	-

1. Y = yes (有効)。O = オプション (デフォルトでは無効ですが、ソフトウェアによって有効化可能)。- = 使用不可
HSI または CSI は、SVOS3 のみ、ペリフェラル用カーネルクロックとして使用可能です。
2. メモリは低電力モードで設定できます。デフォルトでは、STOP 時には低電力モードではありません (SVOS3、SVOS4)。
3. SRAM クロックは、個別にゲートオンまたはゲートオフできます。デフォルトでは、RUN モードおよび SLEEP モードではクロックは有効です。
4. SRAMは、個別にパワーオンまたはパワーオフできます。デフォルトでは、STOP 時、パワーオフモードではありません。
5. 内部タンバによるウェイクアップ。
6. GPIO 状態を、STANDBY モード中、保持できます。デフォルトでは、GPIO 状態は保持されません。
7. 8 ピンは、PA0、PA2、PB7、PC1、PC13、PD2、PD3、および PI8 を STANDBY モードからウェイクアップできます。

さらに、システムクロックを遅くし、電圧スケーリングを低電力範囲に設定することによって、また、APB および AHB ペリフェラルが使用されていないときにはクロックをゲーティングすることによって、RUN モードの消費電力を削減できます。

デバッグモード

デフォルトでは、デバッグ機能が使用されているときにアプリケーションが MCU を STOP または STANDBY モードにすると、デバッグ接続は失われます。これは、Cortex-M33 がクロック供給されなくなるためです。

ただし、DBGMCU 制御レジスタのいくつかの設定ビットをセットすることで、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。詳細については、[セクション 58.2.5：デバッグおよび低電力モード](#)を参照してください。

10.8.1 システムクロックの低速化

RUN モードでは、プリスケアラレジスタをプログラミングすることによって、システムクロック (SYSCLK、HCLK、PCLK) の速度を下げるができます。SLEEP モードに移行する前にペリフェラルの速度を下げるため、これらのプリスケアラを使用することもできます。

詳細については、[セクション 11：リセットおよびクロック制御 \(RCC\)](#) を参照してください。

10.8.2 ペリフェラルクロックゲーティング

RUN モードでは、消費電力を低減するため、個々のペリフェラルとメモリへの HCLK および PCLK をいつでも停止することができます。

SLEEP モードの消費電力をさらに削減するには、WFI または WFE 命令を実行する前に、ペリフェラルクロックを無効化できます。

ペリフェラルクロックゲーティングは、RCC_AHBxENR および RCC_APBxENR レジスタで制御されます。

SLEEP モードでのペリフェラルクロックの無効化は、RCC_AHBxLPENR および RCC_APBxLPENR レジスタの対応するビットをリセットすることによって自動的に行うことができます。

10.8.3 低電力モード

低電力モードへの移行

MCU は、WFI (割り込み待ち) あるいは WFE (イベント待ち) 命令を実行することによって、または ISR からの復帰時に Cortex-M33 システム制御レジスタの SLEEPONEXIT ビットがセットされたときに、低電力モードに移行します。

WFI または WFE による低電力モードへの移行は、ペンディング状態の割り込みやイベントがない場合にのみ実行されます。

低電力モードの終了

MCU は、低電力モードへの移行方法に応じて、SLEEP または STOP モードを終了します。

- WFI 命令または 割り込みサービスルーチン (ISR) 終了によって低電力モードに移行した場合、NVIC によって認識されたペリフェラル割り込みであればどれもデバイスをウェイクアップすることができます。
- WFE 命令によって低電力モードに移行した場合、MCU はイベントの発生直後に低電力モードを終了します。ウェイクアップイベントは、次のいずれかによって生成できます。
 - NVIC IRQ 割り込み：
 - > Cortex-M33 システム制御レジスタの SEVONPEND = 0 のとき
ペリフェラル制御レジスタおよび NVIC にて割り込みを有効にすることで行います。MCU が WFE からレジュームするときには、(NVIC 割り込みクリアペンディングレジスタの)ペリフェラル割り込みペンディングビットと NVIC ペリフェラル IRQ チャネルペンディングビットをクリアする必要があります。十分な優先度の NVIC 割り込みのみが MCU をウェイクアップし、割り込むことができます。
 - > Cortex-M33 システム制御レジスタの SEVONPEND = 1 のとき
ペリフェラル制御レジスタ (および 任意で NVIC の割り込み) にて割り込みを有効にする

ことで行います。MCU が WFE からレジュームするときには、(NVIC 割込みクリアペンディングレジスタの) ペリフェラル割込みペンディングビットと、有効であれば、NVIC ペリフェラル IRQ チャンネルペンディングビットをクリアする必要があります。すべての NVIC 割込み(無効化されているものも含む)によって MCU をウェイクアップします。十分な優先度で有効化された NVIC 割込みのみが MCU をウェイクアップし、割り込むことができます。

– イベント :

- > イベントモードの EXTI ラインの設定 CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、EXTI ペリフェラル割込みペンディングビットや NVIC IRQ チャンネルペンディングをクリアする必要はありません。ペリフェラルの割込みフラグをクリアする必要があるかもしれません。

MCU は、外部リセット (NRST ピン)、IWDG リセット、有効化された WKUPx ピンの 1 つの立上がりエッジ、または RTC/TAMP イベントによって、STANDBY モードを終了します (図 617 : RTC ブロック図を参照)。

STANDBY モードからウェイクアップした後、リセット後と同様に、プログラム実行が再開されます (ブートピンのサンプリング、オプションバイトのロード、リセットベクタのフェッチ)。

注意 : デバイスが STOP モードのとき、ペリフェラル割込みにより、内部オシレータがパワーオンされます。割込みによってデバイスが STOP モードを終了できるように、対応する NVIC 割込みチャンネルを有効にする必要があります。クロックがオンの状態でデバイスが STOP モードにとどまるおそれがあるため、ペリフェラル割込みを有効にしたままで、NVIC チャンネルのみを無効化することでペリフェラル割込みを無効化することはできません。

10.8.4 SLEEP モード

SLEEP モードにおける I/O の状態

SLEEP モードで、すべての I/O ピンは RUN モードと同じ状態を保持します。

SLEEP モードへの移行

MCU は、Cortex-M33 システム制御レジスタの SLEEPDEEP ビットがクリアされると、[低電力モードへの移行](#)で説明されているように SLEEP モードに移行します (SLEEP モードへの移行方法の詳細については、次の表を参照)。

SLEEP モードの終了

MCU は、[低電力モードの終了](#)で説明されているように SLEEP モードを終了します (SLEEP モードの終了方法の詳細については、次の表を参照)。

表 93. SLEEP モード

SLEEP モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0 – ペンディング中の割り込み (WFI) やイベント (WFE) なし Cortex-M33 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) からの復帰時 – SLEEPDEEP = 0 および – SLEEPONEXIT = 1 – ペンディング中の割り込みなし Cortex-M33 システム制御レジスタを参照。
モードの終了	WFI または ISR からの復帰を使用して移行した場合 : 割り込み (表 135 : STM32H563/H573および STM32H562 ベクタテーブルを参照) WFE 命令を使用して移行し、SEVONPEND = 0 の場合 : ウェイクアップイベント (セクション 18.3 : EXTI の機能説明を参照) WFE 命令を使用して移行し、SEVONPEND = 1 の場合 : NVIC で無効化されているときの割り込み (表 135:STM32H563/H573および STM32H562 ベクタテーブルを参照) またはウェイクアップイベント (セクション 18.3 : EXTI の機 能説明を参照)
ウェイクアップ 遅延時間	なし

10.8.5 STOP モード

STOP モードは、Cortex-M33 の DEEPSLEEP モードとペリフェラルクロックゲーティングの組み合わせに基づきます。電圧レギュレータは、SVOSx ビットによって設定されます (SVOS4 および SVOS5 レベルを選択すると、システムが STOP モードを終了するときのスタートアップ遅延が長くなります)。STOP モードでは、コアドメインのすべてのクロックが停止します。PLL、HSI、HSI48、CSI、および HSE オシレータは無効になります。

STOP モードでも HSI または CSI クロックを有効なままにして、ペリフェラルのカーネルクロックとしてすばやく使用可能にすることができます。

すべての SRAM とレジスタの内容は保持されますが、消費電力をさらに削減するために、SRAM は完全に、または部分的にオフになります。ユーザは、[PWR 電力モード制御レジスタ \(PWR_PMCRR\)](#) の xxSO ビットによって、.STOP モード時に破棄するメモリを選択できます。

表 94. メモリシャットオフブロックの選択

選択ビット	STOP モードでのシャットオフブロック
SRAM1SO	AHB SRAM1
SRAM2_48SO	AHB SRAM2 48 KB
SRAM2_16SO	AHB SRAM2 16 KB
SRAM3SO	AHB SRAM3
ETHERNETSO	ETHERNET RAM

BOR は常に STOP モードで使用できます。

STOP モードにおける I/O の状態

STOP モードでは、すべての I/O ピンは RUN モードと同じ状態を保持します。

STOP モードへの移行

MCU は、Cortex-M33 システム制御レジスタの SLEEPDEEP ビットがセットされたとき、[低電力モードへの移行](#)で説明されているように STOP モードに移行します (STOP モードへの移行方法の詳細については、[表 95](#) を参照)。

Flash メモリがプログラミング中の場合、メモリアクセスが終了してから、STOP モードに移行します。

APB ドメインにアクセス中の場合、APB アクセスが終了してから、STOP モードに移行します。

STOP モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) は、キーレジスタへの書き込みによって、またはハードウェアオプシオンによって起動します。起動後は、リセットによってのみ停止できます ([セクション 44 : 独立型ウォッチドッグ \(IWDG\)](#) を参照)。
- リアルタイムクロック (RTC) は、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN ビットによって設定されます。
- 内部 RC オシレータ LSI クロックは、RCC_BDCR の LSION ビットによって設定されます。
- 外部 32.768 kHz オシレータ (LSE) は、RCC_BDCR の LSEON によって設定されます。

AVD および PVD は、STOP モードで使用できます。これらが必要ではない場合、消費電力を低減するためにソフトウェアで無効にする必要があります。

ADCx (x = 1, 2)、DAC1 (2 つのチャネル)、温度センサ、および VREFBUF は、STOP モードに移行する前に無効にしない限り、STOP モードでも電力消費します。

STOP モードの終了

MCU は、低消費電力モードへの移行方法に応じて、EXTI 割込みまたはイベントを有効にすることで STOP モードを終了します。一部のペリフェラルは、システムを STOP モードからウェイクアップできます ([表 138 : EXTI ラインの接続](#) を参照)。これは、STOP モードに入る前に SVOS3 が選択されたときのみ可能です。

注 : ペリフェラルによる STOP モードからのウェイクアップが必要なときには、SVOS3 を選択する必要があります。

割込みまたはウェイクアップイベントを発行することによって STOP モードを終了するときには、RCC クロック設定レジスタ 1 (RCC_CFGR) で STOPWUCK ビットがセットされていた場合、CSI がシステムクロックとして選択されます。STOPWUCK がクリアされた場合、システムクロックとして HSI オシレータが選択されます。CSI がウェイクアップシステムクロックとして選択されると、ウェイクアップ時間が短くなります。HSI を選択すると、より高い周波数 (最大 64 MHz) でウェイクアップできます。

MCU は、低消費電力モードへの移行方法に応じて、EXTI 割込みまたはイベントを有効にすることで STOP モードを終了します。

STOP モードを終了するとき、MCU は VOS3 の RUN モードになります。

表 95. STOP モード

STOP モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – Cortex-M33 システム制御レジスタの SLEEPDEEP ビットをセット – ペンディング中の割り込み (WFI) やイベント (WFE) なし – PWR_CR1 の LPMS = 000
	次の条件下での割り込みサービスルーチン (ISR) からの復帰時 – Cortex-M33 システム制御レジスタの SLEEPDEEP ビットをセット – SLEEPONEXIT = 1 – ペンディング中の割り込みなし – PWR_PMCR の LPMS = 0
	注 : STOP モードに移行するには、すべての EXTI ラインペンディングビット (EXTI 立上がりエッジ・ペンディング・レジスタ 2 (EXTI_RPR2) 内) と、ウェイクアップ割り込みを生成するペリフェラルフラグをクリアする必要があります。そうでない場合、STOP モード移行手順が無視され、プログラムが実行され続けます。
モードの終了	WFI または ISR からの復帰を使用して移行した場合 – 割り込みモードに設定されている任意の EXTI ライン (対応する EXTI 割り込みベクタが NVIC で有効になっている必要があります)。割り込みソースは、ウェイクアップ機能を備えた外部割り込みまたはペリフェラルです (表 135: STM32H563/H573 および STM32H562 ベクタテーブルを参照)。 – 自律ペリフェラルクロックリクエストにより、AHB/APB クロックが存在するときにペリフェラル割り込みが発生しました (ペリフェラルベクタが NVIC で有効になっている必要があります)。 WFE 命令を使用して移行し、SEVONPEND = 0 の場合 : – イベントモードに設定されている任意の EXTI ライン (セクション 18.3 : EXTI の機能説明を参照)。 WFE 命令を使用して移行し、SEVONPEND = 1 の場合 : – 割り込みモードに設定されている任意の EXTI ライン (対応する EXTI 割り込みベクタが NVIC で無効になっている場合も含む)。割り込みソースは、ウェイクアップ機能を備えた外部割り込みまたはペリフェラルです (表 135: STM32H563/H573 および STM32H562 ベクタテーブルを参照)。 – イベントモードに設定されている任意の EXTI ライン (セクション 18.3 : EXTI の機能説明を参照)
	注 : このペリフェラルが STOP からのウェイクアップ割り込みを生成するためには、すべてのペリフェラルクロックが有効である必要があります (RCC で [PERIPH]EN および [PERIPH]LPEN ビットがセットされ、機能独立クロックが選択されている必要があります)。
ウェイクアップ遅延時間	次のいずれか長い方のウェイクアップ時間 : STOP モードからの HSI または CSI ウェイクアップ時間と Flash メモリウェイクアップ時間。

10.8.6 STANDBY モード

BOR がアクティブなとき、最も消費電力が少ないモードは、STANDBY モードです。これは、電圧レギュレータを無効にした状態の Cortex-M33 DEEPSLEEP モードに基づきます。PLL、HSI、HSI48、CSI、および HSE オシレータもオフになります。

バックアップドメインと STANDBY 回路のレジスタを除いて、SRAM とレジスタの内容は失われます (図 36 : SMPS 付き電源装置を参照)。

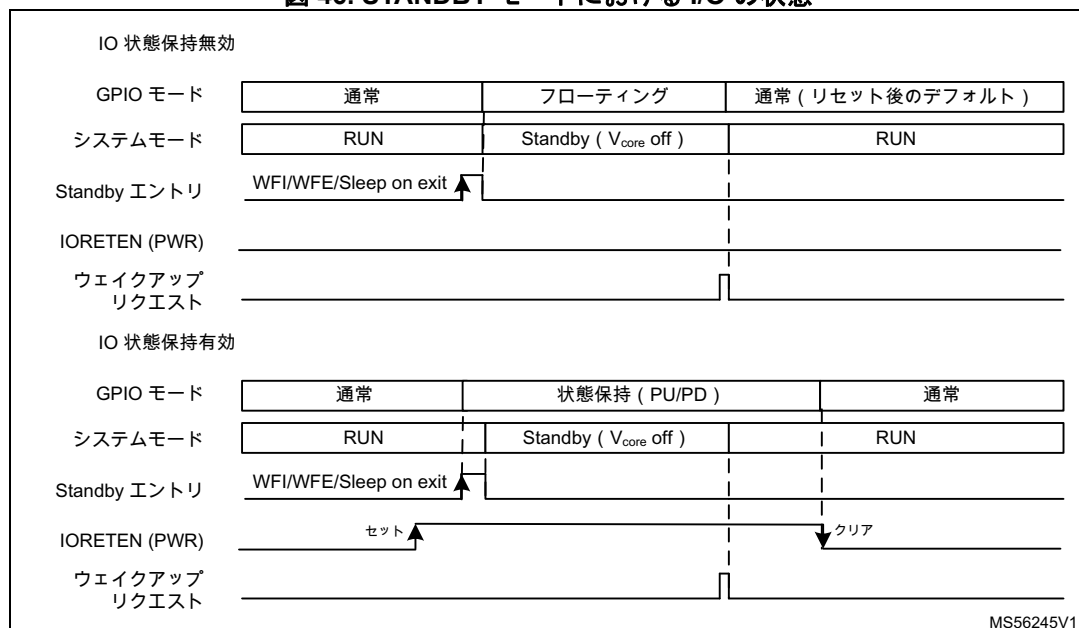
BOR は常に STANDBY モードで使用できます。

STANDBY モードにおける I/O の状態

STANDBY モードでは、I/O はデフォルトでフローティング状態です。PWR_IOPRETR レジスタの IOPRETEN ビットがセットされている場合、I/O の出力状態が保持されます。I/O 保持モードは、スタンバイ機能をサポートする I/O と JTAG IO (PA13、PA14、PA15、および PB4) を除くすべての I/O で有効です。STANDBY モードに移行するとき、出力の状態がサンプリングされ、STANDBY モード時に I/O 出力を維持するようにプルアップまたはプルダウンレジスタがセットされます。

PWR_IOPRETR レジスタの JTAGIOPRETEN ビットがセットされている場合、I/O の出力状態が保持されます。I/O 保持モードは、PA13、PA14、PA15、および PB4 について有効です (ウェイクアップ後のデフォルトの JTAG プルアップ/プルダウンは無効です)。

図 46. STANDBY モードにおける I/O の状態



STANDBY モードからのウェイクアップ後、IOPRETEN (または JTAG IO については JTAGIOPRETEN) がセットされている限り、保持された状態 (プルアップ/プルダウン) が適用され続けます。

GPIO_IDR レジスタで識別できる、STANDBY 前の GPIO ピンの状態 (GPIO ポートクロックと入力バッファの両方とも有効なとき)。

アプリケーションは、IOPRETEN (または JTAG IO については JTAGIOPRETEN) ビットをクリアすることによって、IO 状態を解除 (保持されたプルアップ/プルダウンをクリア) できます。これは、GPIO と関連のペリフェラルを再設定する前または後で行うことができます。このため、アプリケーションは、保持された状態を解除する前に、GPIO を既知の状態に設定できます。

PC13 および PI8 の RTC 出力は STANDBY モードで機能します。LSE に使用される PC14 と PC15 も機能します。8 つのウェイクアップピン (WKUPx, x = 1 ~ 8) と 4 つの RTC タンパピンを使用できます。

STANDBY モードへの移行

MCU は、Cortex-M33 システム制御レジスタの SLEEPDEEP ビットがセットされたとき、[低電力モードへの移行](#)で説明されているように STANDBY モードに移行します (STANDBY モードへの移行方法の詳細については、[表 96](#)を参照)。

STANDBY モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) は、キーレジスタへの書込みによって、またはハードウェアオプションによって起動します。起動後はリセットによってのみ停止できます ([セクション 44.4: IWDG の機能説明](#)を参照)。
- リアルタイムクロック (RTC) は、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN ビットによって設定されます。
- 内部 RC オシレータ LSI クロックは、RCC_BDCR の LSION ビットによって設定されます。
- 外部 32.768 kHz オシレータ (LSE) は、RCC_BDCR の LSEON ビットによって設定されます。
- IO の保持は、PWR_IOPRETR レジスタの IOPRETEN ビットで設定されます。

STANDBY モードの終了

MCU は、[低電力モードの終了](#)で説明されているように STANDBY モードを終了します。[PWR ステータスレジスタ \(PWR_PMSR\)](#) の SBF ステータスフラグは MCU が STANDBY モードにあったことを示します。STANDBY からのウェイクアップ後、すべてのレジスタはリセットされますが、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) および [PWR I/O 保持レジスタ \(PWR_IOPRETR\)](#) を除きます (STANDBY モードの終了方法の詳細については、[表 96](#)を参照)。

STANDBY モードを終了するとき、STANDBY 中、IOPRETEN ビットによって保持されていた I/O 出力状態は、STANDBY モードの終了時から PWR_IOPRETR レジスタの IOPRETEN ビットがクリアされるまで、この設定を保ちます。IOPRETEN がクリアされると、I/O は、リセット値に設定されるか、GPIOx_PUPDR レジスタに従ってプルアップ/プルダウン状態に設定されます。

リセット後のプルアップまたはプルダウンが事前定義されている I/O の場合 (一部の JTAG/SWD I/O)、これらのプルアップまたはプルダウンが STANDBY 時に保持された値と異なる場合、IOPRETEN がクリアされ、保持された値が解除されるまで、プルダウンとプルアップの両方が適用されます。

STANDBY の終了後にプログラムされた GPIOx_PUPDR の値が STANDBY 時に保持された値と異なる場合、IOPRETEN がクリアされ、保持された値が解除されるまで、プルダウンとプルアップの両方が適用されます。

表 96. STANDBY モード

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) <ul style="list-style-type: none"> – Cortex-M33 システム制御レジスタの SLEEPDEEP ビットをセット – ペンディング中の割り込み (WFI) やイベント (WFE) なし – PWR_PMCR の LPMS = 1 – PWR_WUSR の WUFx ビットはクリアされている
	次の条件下での割り込みサービスルーチン (ISR) からの復帰時 <ul style="list-style-type: none"> – Cortex-M33 システム制御レジスタの SLEEPDEEP ビットをセット – SLEEPONEXIT = 1 – ペンディング中の割り込みなし – PWR_PMCR の LPMS = 1 – PWR_WUSR の WUFx ビットはクリアされている – 選択されたウェイクアップソースに対応する RTC/TAMP フラグ、クリア
モードの終了	WKUPx ピンのエッジ、RTC イベント、NRST ピンによる外部リセット、IWDG リセット、BOR リセット
ウェイクアップ 遅延時間	リセットフェーズ

10.8.7 電力モード出力ピン

デバッグを容易にするために、3つの信号をデバイスピンオルタネート機能として使用できます。

- **CSLEEP**
セットされたとき、CSLEEP はシステムが SLEEP モードであることを示します。WFI または WFE が実行されました。
セットされたとき、CSLEEP はシステムが RUN モードであることを示します。
- **CDSTOP**
セットされたとき、CDSTOP は、システムが CSTOP モードであることを示します。つまり、以下の条件が満たされています。
 - CPU SLEEPDEEP = 1 で WFI または WFE が実行されました。
 - AHB/APB クロックは動作していません。
 クリアされたとき、CDSTOP は、システムが CSTOP モードではないことを示します。AHB/APB クロックは動作しています。

次の表に、これらの信号状態に応じた MCU 電力モードを示します。

表 97. 電力モード出力状態対 MCU 電力モード

CSLEEP	CSTOP	MCU 電力モード ⁽¹⁾
0	0	RUN モード
1	0	SLEEP モードまたは STOP モードであり、AHB/APB クロックは動作している
1	1	STOP モード

1. CSLEEP と CDSTOP はコアドメインで生成されるため、STANDBY モードでは駆動されません。

10.9 PWR セキュリティおよび特権保護

10.9.1 PWR セキュリティ保護

TrustZone セキュリティが Flash メモリオプションバイト設定レジスタの TZEN オプションバイトで有効化されているとき、PWR レジスタのフィールドの一部を非セキュアアクセスから保護できます。

PWR TrustZone セキュリティにより、PWR_SECCFGR レジスタを通じて以下の機能を保護できます。

- 低電力モード
- ウェイクアップ (WKUP) ピン
- 電圧の検出および監視
- VBAT モード
- I/O 保持設定

その他の PWR 設定ビットは次の場合にセキュアです。

- RCC のシステムクロック選択はセキュアです。すなわち、電圧スケーリング (VOS) 設定はセキュアです。
- GTZC の UCPD1 はセキュアです。すなわち、PWR_UCPDR レジスタはセキュアです。

表 98 に、PWR_SECCFGR のセキュリティ設定ビットに対する PWR セキュアビットの要約を示します。

セキュリティ保護されたレジスタビットへの非セキュアアクセスは無視されます。

- セキュアビットは、非セキュア書込みアクセスで書き込まれません (WI)。
- セキュアビットは、非セキュア読出しアクセスによって 0 として読み出されます (RAZ)。

PWR_SECCFGR への非セキュア書込みアクセスは WI であり、GTZC で有効な場合、不正アクセスイベントと割込みを生成します。非セキュア読出しアクセスで読み出すことができます。

TrustZone セキュリティが無効なとき (TZEN = 0xC3)、PWR_SECCFGR は RAZ/WI であり、他のすべてのレジスタは非セキュアです。

表 98. PWR セキュリティ設定一覧

セキュア設定レジスタ	セキュリティ設定ビット	レジスタ名	セキュアビット	セキュアビットの非セキュアアクセス
PWR_SECCFGR	適用外 ⁽¹⁾	PWR_SECCFGR	すべてのビット	読出し OK。 WI および不正アクセスイベント
PWR_SECCFGR	最低でも 1 ビットをセット	PWR_PRIVCFGR	SPRIV	読出し OK。 WI
PWR_SECCFGR	LPMSEC	PWR_PMCRCR	すべてのビット	WI
PWR_SECCFGR	VUSBSEC	PWR_USBSCLR	すべてのビット	WI
PWR_SECCFGR	VBSEC	PWR_BDCR	すべてのビット	WI
		PWR_DBPCR	すべてのビット	WI
PWR_SECCFGR	RETSEC	PWR_IOPRETR	すべてのビット	WI
PWR_SECCFGR	WUPxSEC (x = 1~8)	PWR_WUCR	WUPENx	RAZ/WI
			WUPPx	RAZ/WI
			WUPPUPD	RAZ/WI
		PWR_WUSCR	CWUFx	WI
GTZC_TZSC_SECCFGR	UCPD1SEC	PWR_UCPDR	すべてのビット	RAZ/WI
RCC_SECCFGR	SYSCLKSEC	PWR_VOSCR	VOS[1:0]	RAZ/WI
RCC_SECCFGR	SCMSEC	PWR_SCCR	すべてのビット	WI
RCC_SECCFGR	SCMSEC	PWR_VMCR	すべてのビット	WI

1. PWR_SECCFGR は常にセキュアです。

10.9.2 PWR 特権保護

デフォルトでは、リセット後、すべてのレジスタは、特権アクセスと非特権アクセスのどちらでも読み書き可能ですが、PWR_PRIVCFGR への書込みは特権アクセスでのみ可能です。PWR_PRIVCFGR は、セキュアおよび非セキュア、特権および非特権アクセスによって読み出すことができます。PWR_PRIVCFGR の SPRIV ビットは、セキュア特権アクセスでのみ書き込むことができます。このビットは、(表 98 に示されているように、PWR_SECCFGR、GTZC、RCC、または GPIO によって定義される) すべての PWR セキュア機能の特権アクセスを設定します。

PWR_PRIVCFGR の SPRIV ビットがセットされているとき :

- PWR セキュアビットは、PWR_SECCFGR も含めて、特権アクセスでのみ書き込むことができます。
- PWR セキュアビットは、特権アクセスでのみ読み出すことができますが、PWR_SECCFGR と PWR_PRIVCFGR は、特権または非特権アクセスによって読み出すことができます。
- 特権 PWR ビットまたはレジスタへの非特権アクセスは破棄されます。すなわち、ビットは 0 として読み出され、これらのビットへの書込みは無視されます (RAZ/WI)。

PWR_PRIVCFGR の NSPRIV ビットは、セキュアまたは非セキュアの特権アクセスでのみ書き込むことができます。これは、非セキュアとして設定されたすべての PWR セキュリティ保護可能機能の特権アクセスを設定します (PWR_SECCFGR、GTZC、RCC、または GPIO によって定義。表 98 を参照)。

PWR_PRIVCFGR の NSPRIV ビットがセットされているとき :

- 非セキュアとして設定された PWR セキュリティ保護可能ビットは、特権アクセスでのみ書き込むことができます。
- 非セキュアとして設定された PWR セキュリティ保護可能ビットは、特権アクセスでのみ読み出すことができますが、PWR_PRIVCFGR は、特権または非特権アクセスによって読み出すことができます。
- PWR_VOSR、PWR_SR、PWR_SVMSR、PWR_BDSR、および PWR_WUSR の VOSRDY および BOOSTRDY ビットは、特権または非特権アクセスで読み出すことができます。
- 特権 PWR ビットまたはレジスタへの非特権アクセスは破棄されます。すなわち、ビットは 0 として読み出され、これらのビットへの書込みは無視されます (RAZ/WI)。

10.10 PWR 割込み

割込みソースの概要と、その制御方法については、表 99 を参照してください。

表 99. PWR 割込みリクエスト

割込みベクタ	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法	SLEEP、STOP モードの終了	STANDBY モードの終了
PVD/AVD 出力	EXTI ライン 16 を通じてプログラム可能な電圧検出器	PVDO/AVDO	EXTI ライン 16 有効	EXTI PIF16 = 1 を書き込む	はい	不可

10.11 PWR レジスタ

PWR レジスタには、特に指定がない限り、ワード、ハーフワード、およびバイトフォーマットでアクセスできます。

10.11.1 PWR 電力モード制御レジスタ (PWR_PMCR)

このレジスタは、PWR_SECCFGR レジスタの LPMSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの LPMSEC = 1 および SPRIV = 1 のとき、または LPMSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x000

リセット値 : 0x0000 : 000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SRAM1 SO	SRAM2 _48SO	SRAM2 _16SO	SRAM3 SO	Res.	Res.	Res.	Res.	Res.	Res.	ETHER NETSO
					rw	rw	rw	rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	AVD_ READY	BOOST E	Res.	Res.	FLPS	Res.	CSSF	Res.	Res.	Res.	SVOS[1:0]		Res.	LPMS
		rw	rw			rw		rw				rw	rw		rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **SRAM1SO** : STOP モードでの AHB SRAM1 シャットオフ

0 : AHB RAM1 の内容は STOP モードで保持されます。

1 : AHB RAM1 の内容は STOP モードで失われます。

ビット 25 **SRAM2_48SO** : STOP モードでの AHB SRAM2 48 KB シャットオフ。

0 : AHB RAM2 48 KB の内容は STOP モードで保持されます。

1 : AHB RAM2 48 KB の内容は STOP モードで失われます。

ビット 24 **SRAM2_16SO** : STOP モードでの AHB SRAM2 16 KB シャットオフ。

0 : AHB RAM2 16 KB の内容は STOP モードで保持されます。

1 : AHB RAM2 16 KB の内容は STOP モードで失われます。

ビット 23 **SRAM3SO** : STOP モードでの AHB SRAM3 シャットオフ。

0 : AHB RAM3 の内容は STOP モードで保持されます。

1 : AHB RAM3 の内容は STOP モードで失われます。

ビット 22:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **ETHERNETSO**: STOP モードでの ETHERNET RAM シャットオフ。

0 : ETHERNET RAM の内容は STOP モードで保持されます。

1 : ETHERNET RAM の内容は STOP モードで失われます。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **AVD_READY**: アナログ電圧レディ

このビットは、アナログスイッチブーストを有効化する必要がある場合のみ使用されます (BOOSTE ビットを参照)。

期待される V_{DDA} アナログ電源レベルが利用可能である場合に、ソフトウェアによってセットされる必要があります。

正しいアナログ電源レベルは、AVDEN ビット (PWR_VMCR レジスタ) をセットし、監視する電源レベルを選択 (ALS ビット) した後に、AVDO ビット (PWR_VMSR レジスタ) によって示されます。

0: ペリフェラルアナログ電圧 V_{DDA} はレディではありません (デフォルト)。

1: ペリフェラルアナログ電圧 V_{DDA} はレディです。

ビット 12 **BOOSTE**: アナログスイッチ V_{BOOST} 制御

このビットは、 V_{DD} 電源電圧が 2.7 V 未満のときのアナログスイッチ AC 性能を保証するためにブースタを有効にします (供給電圧範囲全体にわたって同じスイッチ性能を確保するための、全高調波ひずみの低減)。 V_{DD} 電源電圧は、PVD および PLS ビットを通じて監視できます。

0: ブースタ無効 (デフォルト)

1: アナログ電圧がレディ (AVD_READY = 1) の場合、ブースタは有効です。

ビット 11:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **FLPS**: STOP モードでの Flash メモリ低電力モード

このビットは、STOP モード終了時の消費電力の削減と再起動時間との間で最高のトレードオフを得るために使用されます。

セットされているとき、システムが STOP モードのときには、Flash メモリは低電力モードになります。

0: システムが STOP モードに移行するとき、Flash メモリは通常モードのままです (高速再起動時間)。

1: システムが STOP モードに移行するとき、Flash メモリは低電力モードに移行します (消費電力の削減)。

注: **SVOS5** が有効な状態でシステムが STOP モードに移行すると、Flash メモリは自動的に低電力モードになります。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CSSF**: STANDBY および STOP フラグのクリア (常に 0 として読み出される)

このビットはハードウェアによって 0 にクリアされます。

0: 影響なし。

1: STOPF および SBF フラグをクリアします。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:2 **SVOS[1:0]**: システム STOP モード電圧スケーリング選択

消費電力と性能の間に最高のトレードオフを実現するために、これらのビットは V_{CORE} 電圧レベルをシステム STOP モードで制御します。

00: 予約済み

01: SVOS5 スケール 5

10: SVOS4 スケール 4

11: SVOS3 スケール 3 (デフォルト)

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **LPMS**: 低電力モード選択

このビットは、DEEPSLEEP モードを定義します。

0: DEEPSLEEP に移行するとき、STOP モードを維持します。

1: DEEPSLEEP に移行するとき、STANDBY モードを許可します。

10.11.2 PWR ステータスレジスタ (PWR_PMSR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SBF	STOPF	Res.	Res.	Res.	Res.	Res.
									r	r					

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SBF** : システム STANDBY フラグ

このビットはハードウェアによってセットされ、POR または CSSF ビットをセットすることでのみクリアされます。

0 : システムは STANDBY モードではありません。

1 : システムは STANDBY モードです。

ビット 5 **STOPF** : STOP フラグ

このビットはハードウェアによってセットされ、すべてのリセットまたは CSSF ビットをセットすることでのみクリアされます。

0 : システムが STOP モードではありません。

1 : システムが STOP モードです。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

10.11.3 PWR 電圧スケーリング制御レジスタ (PWR_VOSCR)

一部のレジスタフィールドは、RCC_SECCFGR レジスタに応じて、非セキュアアクセスから保護されます。

これらのフィールドは、PWR_PRIVCFGR レジスタの設定に応じて、非特権アクセスから保護できます。

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOS[1:0]		Res.	Res.	Res.	Res.
										rw	rw				

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **VOS[1:0]** : 性能に応じた電圧スケーリング選択

これらのビットは、 V_{CORE} 電圧レベルを制御し、消費電力と性能の間で最高のトレードオフを実現します。

- バイパスモードでは、これらのビットは、外部から供給されるコア電圧レベルおよび関連する性能にも応じてセットする必要があります。
- 性能を高めるときには、システム周波数を増加する前に電圧スケーリングを変更する必要があります。
- 性能を低下させるときには、最初にシステム周波数を低減してから電圧スケーリングを変更する必要があります。

00 : スケール 3 (デフォルト)

01 : スケール 2

10 : スケール 1

11 : スケール 0

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

10.11.4 PWR 電圧スケーリングステータスレジスタ (PWR_VOSSR)

アドレスオフセット : 0x0014

リセット値 : 0x0000 2008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACTVOS[1:0]		ACTVOS RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOS RDY	Res.	Res.	Res.
r	r	r										r			

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:14 **ACTVOS[1:0]** : V_{CORE} に現在適用されている電圧出力スケーリング

このフィールドは、最後の VOS 値を提供します。

00 : VOS3 (最低電力)

01 : VOS2

10 : VOS1

11 : VOS0 (最高周波数)

ビット 13 **ACTVOSRDY** : 現在使用されている VOS に対して電圧レベルレディ

0 : V_{CORE} は、ACTVOS[1:0] によって提供される現在の電圧スケーリングを上回るか、下回っています。

1 : V_{CORE} は、ACTVOS[1:0] によって提供される現在の電圧スケーリングに等しいです。

ビット 12:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **VOSRDY** : V_{CORE} 電圧スケーリング出力選択のレディビット。

0 : レディではありません。電圧レベルは、選択された VOS のレベルを下回っています。

1 : レディです。電圧レベルは、選択された VOS のレベル以上です。

VOSRDY フラグは、低圧スケールから高圧スケールに切り替えるとき (VOS3 から VOS0 に切り替えるときなど) にのみ使用する必要があります。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

10.11.5 PWR バックアップドメイン制御レジスタ (PWR_BDCR)

このレジスタは、PWR_SECCFGR レジスタの VBSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの VBSEC = 1 および SPRIV = 1 のとき、または VBSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

このレジスタは、STANDBY モードからのウェイクアップ、RESET 信号、および VDD POR によってリセットされません。VSW POR および VSWRST リセットによってのみリセットされます。RCC_BDCR レジスタの VSWRST ビットによって VSW ドメインをリセットするときには、このレジスタにアクセスしないでください。

RCC_BDCR レジスタの VSWRST ビットによって VSW ドメインをリセットするときには、MONEN および BREN ビットにアクセスしないでください。リセット後、このレジスタの MONEN および BREN は書き込み保護されます。内容を変更する前に、PWR_DBPCR レジスタの DBP ビットをセットして、書き込み保護を無効にする必要があります。

アドレスオフセット : 0x20

パワーオンリセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	Res.	Res.	Res.	MONEN	BREN
						rw	rw							rw	rw

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **VBRS** : V_{BAT} 充電抵抗の選択

0 : 5 k Ω の抵抗で V_{BAT} を充電します。

1 : 1.5 k Ω の抵抗で V_{BAT} を充電します。

ビット 8 **VBE** : V_{BAT} 充電イネーブル

0 : V_{BAT} バッテリ充電を無効化します。

1 : V_{BAT} バッテリ充電を有効化します。

注 : **POR によってのみリセットされます。**

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MONEN** : バックアップドメイン電圧および温度監視イネーブル

0 : バックアップドメイン電圧および温度監視は無効です。

1 : バックアップドメイン電圧および温度監視は有効です。

ビット 0 **BREN** : STANDBY および V_{BAT} モードでのバックアップ RAM の保持

このビットがセットされると、バックアップレギュレータ (STANDBY モードおよび V_{BAT} モードでバックアップ RAM の内容を保持するために使用) が有効になります。

BREN がクリアされると、バックアップレギュレータはオフになります。RUN モードおよび STOP モードではバックアップ RAM をそのまま使用できます。ただし、STANDBY モードおよび V_{BAT} モードではその内容は失われることになります。

BREN をセットすると、アプリケーションは、バックアップレギュレータレディフラグ (BRRDY) がセットされ、SRAM に書き込まれたデータが STANDBY モードおよび V_{BAT} モードにおいて保持されることが示されるのを待つ必要があります。

0 : STANDBY および V_{BAT} モードでは、バックアップ RAM の内容は失われます。

1 : STANDBY および V_{BAT} モードでは、バックアップ RAM の内容は保持されます。

10.11.6 PWR 無効化バックアップドメイン制御レジスタ(PWR_DBPCR)

このレジスタは、PWR_SECCFGR レジスタの VBSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの VBSEC = 1 および SPRIV = 1 のとき、または VBSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBP
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DBP** : バックアップドメイン書き込み保護の無効化

リセット状態では、バックアップドメインのすべてのレジスタと SRAM は、寄生書き込みアクセスから保護されます。これらのレジスタへの書き込みアクセスを可能にするには、このビットをセットする必要があります。

0 : バックアップドメインへの書き込みアクセスは無効です。

1 : バックアップドメインへの書き込みアクセスは有効です。

10.11.7 PWR バックアップドメインステータスレジスタ (PWR_BDSR)

このレジスタは、STANDBY モードからのウェイクアップ、RESET 信号、および VDD POR によってリセットされません。VSW POR および VSWRST リセットによってのみリセットされます。

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRRDY
								r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TEMPH** : 温度レベルの監視と高閾値

0 : 高閾値レベルを下回る温度

1 : 高閾値レベル以上の温度

ビット 22 **TEMPL** : 温度レベルの監視と低閾値

0 : 低閾値レベルを上回る温度

1 : 低閾値レベル以下の温度

ビット 21 **VBATH** : V_{BAT} レベルの監視と高閾値

0 : V_{BAT} レベルは高閾値レベルを下回っています。

1 : V_{BAT} レベルは高閾値レベル以上です。

ビット 20 **VBATL** : V_{BAT} レベルの監視と低閾値

0 : V_{BAT} レベルは低閾値レベルを上回っています。

1 : V_{BAT} レベルは低閾値レベル以下です。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **BRRDY** : バックアップレギュレータレディ。

このビットは、バックアップレギュレータがレディ状態であることを示すために、ハードウェアによってセットされます。

0 : バックアップレギュレータはレディ状態ではありません。

1 : バックアップレギュレータはレディ状態です。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

10.11.8 PWR USB タイプ C パワーデリバリレジスタ (PWR_UCPDR)

このレジスタは、TZSC_SECCFGR レジスタの UCPD1SEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの UCPD1SEC = 1 および SPRIV = 1 のとき、または UCPD1SEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

STANDBY モードの終了による影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD_STBY	UCPD_DBDIS
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **UCPD_STBY** : USB タイプ C およびパワーデリバリの STANDBY モード

セットされると、このビットは、STANDBY モードで UCPD 設定を記憶するために使用されます。UCPD を使用しているときには、STANDBY モードに移行する直前に、このビットに 1 を書き込む必要があります。STANDBY モードの終了後は、UCPD レジスタに書き込む前に、0 を書き込む必要があります。

ビット 0 **UCPD_DBDIS** : USB タイプ C およびパワーデリバリデッドバッテリーディスエーブル

リセット終了後、USB タイプ C の「デッドバッテリー」動作が有効になり、CC1 および CC2 ピンがプルダウンの影響を受ける可能性があります。あらゆる場合にこのビットを無効にして、このプルダウンを停止するか、UCPD に制御を引き渡すことをお勧めします（そのため、無効にする前に UCPD を初期化する必要があります）。

0 : UCPD_x_CC1 および UCPD_x_CC2 ピンにおける USB タイプ C のデッドバッテリープルダウン動作を有効にします。

1 : UCPD_x_CC1 および UCPD_x_CC2 ピンにおける USB タイプ C のデッドバッテリープルダウン動作を無効にします。

10.11.9 PWR 電源設定制御レジスタ (PWR_SCCR)

このレジスタは、PWR_SECCFGR レジスタの SCMSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの SCMSEC = 1 および SPRIV = 1 のとき、または SCMSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x030

リセット値 : 0x0000 : 0X00

レジスタのリセット値は、パッケージに従って変わります。ビット 9 および 8 は、電力設定を示します。これらの値は排他的です。

POR によってのみリセットされ、STANDBY モードからのウェイクアップや RESET パッドによってリセットされません。このレジスタの BYPASS ビットは、POR 後に一度だけ書き込まれます。ライトワンスメカニズムにより、レジスタはロックされ、それ以後の書き込みアクセスは無視されます。新しい値を書き込む前に、システムを再起動する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SMPSE N	LDOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPAS S
						r	r								rwo

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **SMPSEN** : SMPS 有効化

値は、パッケージが SMPS レギュレータを使用するときにハードウェアによって設定されます。

ビット 8 **LDOEN** : LDO 有効化

値は、パッケージが LDO レギュレータを使用するときにハードウェアによって設定されます。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **BYPASS** : 電源管理ユニットバイパス

0 : 電源管理ユニットは通常動作内部レギュレータを使用します。

1 : 電源管理ユニットはバイパスされます。外部電源を使用します (電圧監視はアクティブです)。

10.11.10 PWR 電圧モニタ制御レジスタ (PWR_VMCR)

このレジスタは、PWR_SECCFGR レジスタの SCMSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの SCMSEC = 1 および SPRIV = 1 のとき、または SCMSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

PVDE および PLS ビットは、ロックメカニズムによって保護されます。ロック制御は、SBS モジュールにあり、SBS_CFGR2 レジスタの PVDL ビットによって制御されます。デフォルトでは、PVDE と PLS はロックされません。

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	ALS[1:0]		AVDEN	Res.	Res.	Res.	Res.	PLS[2:0]		PVDE	
					rw	rw	rw					rw	rw	rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:9 **ALS[1:0]** : アナログ電圧検出器 (AVD) レベル選択

これらのビットは、アナログ電圧検出器 (AVD) の閾値を選択します。

00 : AVD level0 ($V_{AVD0} \sim 1.7\text{ V}$)

01 : AVD level1 ($V_{AVD1} \sim 2.1\text{ V}$)

10 : AVD level2 ($V_{AVD2} \sim 2.5\text{ V}$)

11 : AVD level3 ($V_{AVD3} \sim 2.8\text{ V}$)

ビット 8 **AVDEN** : V_{DDA} ペリフェラル電圧モニタイネーブル

0 : V_{DDA} のペリフェラル電圧モニタを無効化します

1 : V_{DDA} のペリフェラル電圧モニタを有効化します。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:1 **PLS[2:0]** : プログラム可能な電圧検出器 (PVD) レベル選択

これらのビットは、プログラム可能な電圧検出器 (PVD) の閾値を選択します。

000 : PVD level0 ($V_{PVD0} \sim 1.95\text{ V}$)

001 : PVD level1 ($V_{PVD1} \sim 2.10\text{ V}$)

010 : PVD level2 ($V_{PVD2} \sim 2.25\text{ V}$)

011 : PVD level3 ($V_{PVD3} \sim 2.40\text{ V}$)

100 : PVD level4 ($V_{PVD4} \sim 2.55\text{ V}$)

101 : PVD level5 ($V_{PVD5} \sim 2.70\text{ V}$)

110 : PVD level6 ($V_{PVD6} \sim 2.85\text{ V}$)

111 : PVD_IN ピン

ビット 0 **PVDE** : PVD 有効化

0 : PVD は無効です。

1 : PVD は有効です。

10.11.11 PWR USB 電源制御レジスタ (PWR_USBSCR)

このレジスタは、PWR_SECCFGR レジスタの VUSBSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの VUSBSEC = 1 および SPRIV = 1 のとき、または VUSBSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	USB33 SV	USB33 DEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **USB33SV** : 独立 USB 電源有効

このビットは、電気的および論理的アイソレーションの目的で V_{DDUSB} 電源を検証するために使用されます。このビットをセットすると、USBFS ペリフェラルの使用が必須になります。 V_{DDUSB} がアプリケーションで常に存在するわけではない場合、 V_{DDUSB} 電圧モニタを使用して、この電源がレディかどうかを調べることができます。

0 : V_{DDUSB} が存在しません。論理的および電気的アイソレーションが適用され、この電源は無視されます。

1 : V_{DDUSB} は有効です。

ビット 24 **USB33DEN** : V_{DDUSB} 電圧レベル検出器イネーブル

0 : V_{DDUSB} 電圧レベル検出器は無効です。

1 : V_{DDUSB} 電圧レベル検出器は有効です。

ビット 23:0 予約済みであり、リセット値に保持する必要があります。

10.11.12 PWR 電圧モニタステータスレジスタ (PWR_VMSR)

アドレスオフセット : 0x03C

リセット値 : 0x00X0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USB33 RDY	Res.	PVDO	Res.	VDDIO 2RDY	AVDO	Res.	Res.	Res.
							r		r		r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **USB33RDY** : V_{DDUSB} レディ

0 : V_{DDUSB} は、 V_{DDUSB} 電圧モニタの閾値未満です。

1 : V_{DDUSB} は、 V_{DDUSB} 電圧モニタの閾値以上です。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PVDO** : プログラム可能な電圧検出器の出力

このビットは、ハードウェアによってセット／クリアされます。PVDE ビットによって PVD が有効化されている場合のみ有効です。

0 : V_{DD} が PLS[2:0] ビットで選択された PVD 閾値以上です

1 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より低いです。

注 : PVD は STANDBY モードでは無効になるため、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **VDDIO2RDY** : V_{DDIO2} の電圧検出器の出力

このビットは、ハードウェアによってセット／クリアされます。

0 : V_{DDIO2} は 1.2V 未満です。

1 : V_{DDIO2} は 1.2V 以上です。

ビット 19 **AVDO** : V_{DDA} のアナログ電圧検出器の出力

このビットは、ハードウェアによってセット／クリアされます。 V_{DDA} の AVD が AVDEN ビットによって有効化されている場合のみ有効です。

0 : V_{DDA} が ALS[2:0] ビットで選択された AVD 閾値以上です

1 : V_{DDA} が ALS[2:0] ビットで選択された AVD 閾値より低いです。

注 : AVD は STANDBY モードでは無効になるため、このビットは STANDBY またはリセット後、AVDEN ビットがセットされるまで 0 になります。

ビット 18:0 予約済みであり、リセット値に保持する必要があります。

10.11.13 PWR ウェイクアップステータスクリアレジスタ (PWR_WUSCR)

各レジスタビット CWUF x ($x = 1 \sim 8$) は、PWR_SECCFGR レジスタの WUPxSEC = 1 のとき、非セキュアアクセスから保護されます。

各ビット CWUF x は、PWR_SECCFGR の WUPxSEC = 1 および PWR_PRIVCFGR の SPRIV = 1 のとき、または WUPxSEC 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CWUF 8	CWUF 7	CWUF 6	CWUF 5	CWUF 4	CWUF 3	CWUF 2	CWUF 1
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **CWUF x** : WUF x ($x = 8 \sim 1$) のウェイクアップピンフラグをクリア

これらのビットは常に 0 として読み出されます。

0 : 影響なし。

1 : 1 を書き込むと、WUF x のウェイクアップピンフラグがクリアされます (ビットはハードウェアによって 0 にクリアされます)。

10.11.14 PWR ウェイクアップステータスレジスタ (PWR_WUSR)

アドレスオフセット : 0x044

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUF8	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **WUFx** : ウェイクアップピン WUFx フラグ (x = 8 ~ 1)

このビットはハードウェアによってセットされ、RESET ピンによって、または PWR_WUSCR レジスタの CWUFx ビットをセットすることによってのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : ウェイクアップイベントが WUFx ピンから受信されました。

10.11.15 PWR ウェイクアップ設定レジスタ (PWR_WUCR)

各レジスタビット WUPPUPDx (x = 1 ~ 8) および WUPPx (x = 1 ~ 8) および WUPENx (x = 1 ~ 8) は、PWR_SECCFGR レジスタの WUPxSEC = 1 (x = 1 ~ 8) のとき、非セキュアアクセスから保護されます。

各ビット WUPENx は、PWR_SECCFGR の WUPxSEC = 1 および PWR_PRIVCFGR, の SPRIV = 1 のとき、または WUPxSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x048

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUPPUPD8 [1:0]		WUPPUPD7 [1:0]		WUPPUPD6 [1:0]		WUPPUPD5 [1:0]		WUPPUPD4 [1:0]		WUPPUPD3 [1:0]		WUPPUPD2 [1:0]		WUPPUPD1 [1:0]	
rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUP P8	WUP P7	WUP P6	WUP P5	WUP P4	WUP P3	WUP P2	WUP P1	WUP EN8	WUP EN7	WUP EN6	WUP EN5	WUP EN4	WUP EN3	WUPE N2	WUP EN1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 **WUPPUPDx[1:0]** : WKUPx (x = 8 ~ 1) のウェイクアップピンのプル設定

これらのビットは、WUPENx = 1 のときに使用される I/O パッドのプル設定を定義します。関連する GPIO ポートのプル設定は、同じ値にセットするか、00 に設定する必要があります。ウェイクアップピンのプル設定は、STANDBY モードで保持されます。

00 : プルアップなし

01 : プルアップ

10 : プルダウン

11 : 予約済み

ビット 15:8 **WUPPx** : WUPx (x = 8 ~ 1) のウェイクアップピン極性ビット

これらのビットは、WUPx外部ウェイクアップピンのイベント検出に使用される極性を定義します。

0 : 高レベルで検出します (立上がりエッジ)。

1 : 低レベルで検出します (立下がりエッジ)。

ビット 7:0 **WUPENx** : ウェイクアップピン WUPx (x = 8 ~ 1) の有効化

これらのビットは、ソフトウェアによってセット/クリアされます。

0 : WUPx ピンでイベントが発生しても、システムは STANDBY モードからウェイクアップしません。

1 : WUPx ピンで立上がり/立下がりエッジが発生すると、システムは STANDBY モードからウェイクアップします。

注 : **WUPPx** で立上がりエッジが選択されているときに、WUPx ピンのレベルがすでに高い場合、または **WUPPx** で立下がりエッジが選択されているときに、このピンのレベルがすでに低い場合、WUPx ピンが有効化されると (WUPENx ビットをセットすることによって)、追加のウェイクアップイベントが検出されます。

10.11.16 PWR I/O 保持レジスタ (PWR_IOPRETR)

このレジスタは、PWR_SECCFGR レジスタの RETSEC = 1 のとき、非セキュアアクセスから保護されます。

このレジスタは、PWR_PRIVCFGR レジスタの RETBSEC = 1 および SPRIV = 1 のとき、または RETSEC = 0 および NSPRIV = 1 のとき、非特権アクセスから保護されます。

アドレスオフセット : 0x050

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JTAGIO RETEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IORET EN
															rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **JTAGIORETEN** : JTAG IO の IO 保持イネーブル

0 : IO 保持モードは無効です。

1 : PA13、PA14、PA15、および PB4 について、IO 保持モードは有効です。

STANDBY モードに移行するとき、出力がサンプリングされ、STANDBY 電力モード時に出力 IO に適用されます。

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **IORETEN** : IO 保持イネーブル :

0 : IO 保持モードは無効です。

1 : IO 保持モードは、スタンバイ機能をサポートする IO と PA13、PA14、PA15、および PB4 を除くすべての IO で有効です。

STANDBY モードに移行するとき、出力がサンプリングされ、STANDBY 電力モード時に出力 IO に適用されます。

注 : **DBGMCU_CR** レジスタの **DBG_STANDBY** ビットがセットされている場合、IO 状態は保持されません。

10.11.17 PWR セキュリティ設定レジスタ (PWR_SECCFGR)

このレジスタは、アクセスがセキュアなときのみ書き込むことができます。セキュアまたは非セキュアアクセスによって読み出すことができます。

このレジスタは、PWR_PRIVCFGR レジスタの SPRIV = 1 のとき、非特権書き込みアクセスから書き込み保護されます。

アドレスオフセット : 0x100

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VUSB SEC	VB SEC	SCM SEC	LPM SEC	RET SEC	Res.	Res.	Res.	WUP8 SEC	WUP7 SEC	WUP6 SEC	WUP5 SEC	WUP4 SEC	WUP3 SEC	WUP2 SEC	WUP1 SEC
rW	rW	rW	rW	rW				rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **VUSBSEC** : 電圧 USB セキュア保護

0 : PWR_USBSCR は、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_USBSCR は、セキュアアクセスでのみ読み書きが可能です。

ビット 14 **VBSEC** : バックアップドメインのセキュア保護

0 : PWR_BDCR、PWR_DBPCR は、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_BDCR、PWR_DBPCR は、セキュアアクセスでのみ読み書きが可能です。

ビット 13 **SCMSEC** : 電源設定と監視のセキュア保護。

0 : PWR_SCCR および PWR_VMCR は、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_SCCR および PWR_VMCR は、セキュアアクセスでのみ読み書きが可能です。

ビット 12 **LPMSEC** : 低電力モードのセキュア保護

0 : PWR_PMCR は、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_PMCR は、セキュアアクセスでのみ読み書きが可能です。

ビット 11 **RETSEC** : 保持のセキュア保護

0 : PWR_IORETR は、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_IORETR は、セキュアアクセスでのみ読み書きが可能です。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **WUPxSEC** : WUPx セキュア保護 (x = 8 ~ 1)

0 : PWR_WUSCR および PWR_WUCR レジスタの WKUPx ウェイクアップピンに関連するビットは、セキュアまたは非セキュアアクセスによる読み書きが可能です。

1 : PWR_WUSCR および PWR_WUCR の WKUPx ウェイクアップピンに関連するビットは、セキュアアクセスでのみ読み書きが可能です。

10.11.18 PWR 特権設定レジスタ (PWR_PRIVCFGR)

このレジスタは、アクセスが特権であるときのみ、書き込むことができます。特権または非特権アクセスによって読み出すことができます。

アドレスオフセット : 0x104

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSPRIV	SPRIV
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **NSPRIV** : PWR 非セキュア機能の特権設定

ソフトウェアでセット／クリアされます。このビットは、セキュアまたは非セキュアの特権アクセスによってのみ書き込むことができます。

0 : PWR 非セキュア機能の読み書きは、特権または非特権アクセスによって行うことができます。

1 : PWR 非セキュア機能の読み書きは、特権アクセスによってのみ行うことができます。

ビット 0 **SPRIV** : PWR セキュア機能の特権設定

ソフトウェアでセット／クリアされます。このビットは、セキュア特権アクセスによってのみ書き込むことができます。

0 : PWR セキュア機能の読み書きは、特権または非特権アクセスによって行うことができます。

1 : PWR セキュア機能の読み書きは、特権アクセスによってのみ行うことができます。

10.11.19 PWR レジスタマップ

表 100. PWR レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	PWR_PMCRR	Res.	Res.	Res.	Res.	Res.	SRAM1SO	SRAM2_48SO	SRAM2_16SO	SRAM3SO	Res.	Res.	Res.	Res.	Res.	Res.	ETHERNETSO	Res.	Res.	Res.	AVD_READY	BOOSTE	Res.	FLPS	Res.	CSSF	Res.	Res.	Res.	svos[1:0]	Res.	LPMS	
	リセット値						0	0	0	0							0			0	0		0		0		SBF	STOPF	1	1		0	
0x04	PWR_PMSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SBF	STOPF	Res.	Res.	Res.	Res.	
	リセット値																									0	0						
0x008 から 0x00F	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x10	PWR_VOSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOS [1:0]	Res.	Res.	Res.	Res.	
	リセット値																										0	0					
0x14	PWR_VOSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACTVOS[1:0]	Res.	Res.	ACTVOSRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOSRDY	Res.	Res.	Res.
	リセット値																0	0	1									1					
0x018 から 0x01F	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x20	PWR_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBRS	Res.	VBE	Res.	Res.	Res.	Res.	MONEN	BREN		
	リセット値																					0	0		0				0	0			
0x24	PWR_DBPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBP		
	リセット値																														0		
0x28	PWR_BDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値									0	0	0	0				0																
0x2C	PWR_UCPDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD_STBY	UCPD_DBDIS	
	リセット値																													0	0		
0x30	PWR_SCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x34	PWR_VMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALS [1:0]	Res.	AVDEN	Res.	Res.	Res.	Res.	PLS [2:0]	Res.	PVDE	
	リセット値																					0	0	0						0	0		
0x38	PWR_USBSCR	Res.	Res.	Res.	Res.	Res.	USB33SV	USB33DEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値						0	0																									
0x3C	PWR_VMSR	Res.	Res.	Res.	Res.	Res.	Res.	USB33RDY	Res.	PVDD	Res.	VDDIO2RDY	Res.	AVDD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値							0	0	0	0	0	0																				

表 100. PWR レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	PWR_WUSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x44	PWR_WUSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x48	PWR_WUCR	WUPPUPD8 [1:0]	WUPPUPD7 [1:0]	WUPPUPD6 [1:0]	WUPPUPD5 [1:0]	WUPPUPD4 [1:0]	WUPPUPD3 [1:0]	WUPPUPD2 [1:0]	WUPPUPD1 [1:0]	WUPPUPD0 [1:0]	WUPPUPD8 [1:0]	WUPPUPD7 [1:0]	WUPPUPD6 [1:0]	WUPPUPD5 [1:0]	WUPPUPD4 [1:0]	WUPPUPD3 [1:0]	WUPPUPD2 [1:0]	WUPPUPD1 [1:0]	WUPPUPD0 [1:0]	WUPPUPD8 [1:0]	WUPPUPD7 [1:0]	WUPPUPD6 [1:0]	WUPPUPD5 [1:0]	WUPPUPD4 [1:0]	WUPPUPD3 [1:0]	WUPPUPD2 [1:0]	WUPPUPD1 [1:0]	WUPPUPD0 [1:0]	WUPPUPD8 [1:0]	WUPPUPD7 [1:0]	WUPPUPD6 [1:0]	WUPPUPD5 [1:0]	WUPPUPD4 [1:0]
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x50	PWR_IJORETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x054 から 0x0FF	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	PWR_SECCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x104	PWR_PRIVCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

11 リセットおよびクロック制御 (RCC)

11.1 概要

リセットおよびクロック制御 (RCC) は、さまざまなリセットを管理し、バスおよびペリフェラルのためのクロックを生成します。

11.2 RCC ピンおよび内部信号

下の表に、パッケージのピンまたはボールに接続された RCC 入出力信号を示します。

表 101. パッケージのピンまたはボールに接続された RCC スレーブ入力／出力信号

信号名	信号タイプ	説明
NRST	I/O	システムリセット、外部デバイスにリセットを提供するために使用可能
OSC32_IN	I	32 kHz オシレータ入力
OSC32_OUT	O	32 kHz オシレータ出力
OSC_IN	I	システムオシレータ入力
OSC_OUT	O	システムオシレータ出力
MCO	O	外部デバイスの出力クロック
LSCO	O	外部デバイスの低速出力クロック
AUDIOCLK	I	SAI1、SAI2、I2S1、I2S2、および I2S3 の外部カーネルクロック入力

11.3 RCC リセット機能の説明

リセットには次の 3 種類があります。

- システムリセット
- 電源リセット
- バックアップドメインリセット

11.3.1 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

- ブラウンアウトリセット (BOR)
- STANDBY モードの終了時

パワーオンリセットまたはパワーダウマリセット (POR/PDR) を含むブラウンアウトリセットは、バックアップドメインを除くすべてのレジスタをリセット値に設定します。

STANDBY モードを終了すると、コアドメインのすべてのレジスタがリセット値にセットされます。コアドメインの外側のレジスタ (STANDBY 時および STANDBY モード終了時の RTC、WKUP、IWDG、および GPIO プルアップ/プルダウン設定) は影響を受けません。

11.3.2 システムリセット

システムリセットは、[RCC リセットステータスレジスタ \(RCC_RSR\)](#) のリセットフラグとバックアップドメインのレジスタを除き、すべてのレジスタをリセット値にセットします。

システムリセットは、次のイベントのいずれかの発生時に生成されます。

- NRST ピンのローレベル (外部リセット)
- ウィンドウ型ウォッチドッグイベント (WWDG リセット)
- 独立型ウォッチドッグイベント (IWDG リセット)
- ソフトウェアリセット (SW リセット) ([ソフトウェアリセット](#)を参照)
- 低電力モードセキュリティリセット ([低電力モードセキュリティリセット](#)を参照)
- ブラウンアウトリセット

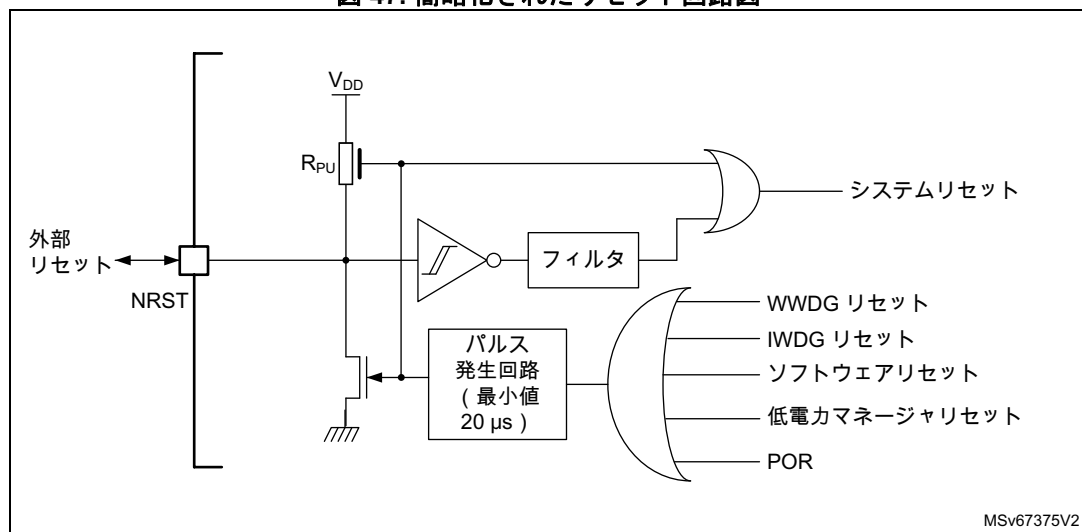
リセットソースは、[RCC リセットステータスレジスタ \(RCC_RSR\)](#) のリセットフラグを確認することによって識別できます。

これらのソースは NRST ピンに作用し、遅延フェーズの間中このピンをローに保ちます。リセットサービスルーチンベクタは、製品の状態、ブートオプションバイト、またはその両方に応じて選択されます。

デバイスに与えられるシステムリセット信号は、NRST ピンに出力されます。パルス発生回路は、各内部リセットソースについて 20 μ s の最小リセットパルス期間を保証します。外部リセットの場合、リセットパルスは NRST ピンがローにアサートされているときに生成されます。

内部リセットの場合、プルアップレジスタを通じて、消費電力を節減するために、内部プルアップ R_{PU} が無効化されます。

図 47. 簡略化されたリセット回路図



ソフトウェアリセット

デバイス上でソフトウェアリセットを実行するには、Cortex-M33 のアプリケーション割込みおよびリセット制御レジスタの SYSRESETREQ ビットをセットする必要があります。

低電力モードセキュリティリセット

誤って重要なアプリケーションが低電力モードに移行しないように、次の低電力モードセキュリティリセットを使用できます。オプションバイトで有効になっている場合、以下のいずれかの条件下でリセットを生成します。

- STANDBY モードへの移行：このリセットは、ユーザオプションバイトの nRST_STDBY ビットをリセットすることによって有効になります。この場合、STANDBY モードへの遷移シーケンスが正常に実行されるたびに、STANDBY モードに移行する代わりにデバイスがリセットされます。
- STOP モードへの移行：このリセットは、ユーザオプションバイトの nRST_STOP ビットをリセットすることによって有効になります。この場合、STOP モードへの遷移シーケンスが正常に実行されるたびに、STOP モードに移行する代わりにデバイスがリセットされます。

ユーザオプションバイトの詳細については、[セクション 7.4.1：オプション・バイトについて](#)を参照してください。

11.3.3 バックアップドメインリセット

バックアップドメインには 2 つの固有のリセットがあり、次のいずれかのイベントが発生したときに生成されます。

- ソフトウェアリセット。[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の VSWRST ビットをセットすることでトリガされます。リセットを行うには、VSWRST ビットをセットする前に、このドメインへの書込みアクセスを有効にする必要があります。
- V_{DD} または V_{BAT} パワーオン。ただし、両方の電源供給がともにオフ状態であった場合。

バックアップドメインリセットは、LSE オシレータ、RTC、バックアップレジスタ、バックアップ SRAM、および RCC_BDCR レジスタに影響します。

11.3.4 リセットソース識別

アプリケーションは、RCC_RSR レジスタのリセットフラグを確認することで、リセットソースを識別できます。

ソフトウェアは、RMVF ビットをセットすることで、フラグをリセットできます。

[表 102](#) に、RCC_RSR レジスタのステータスビットがリセット生成時の条件に応じてどのように挙動するかを示します。たとえば、IWDG タイムアウトが発生したとき、CPU がブートフェーズ中に RCC_RSR レジスタを読み出していた場合、PINRSTF と IWDGRSTF の両方のビットがセットされます。これは、IWDG もピンリセットを生成したことを示します。

表 102. リセットソース識別 (RCC_RSR)⁽¹⁾

リセット		LPWRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	BORRSTF	PINRSTF
1	パワーオンリセット	0	0	0	0	1	1
2	ピン/パッドリセット	0	0	0	0	0	1
3	ブラウンアウト（低または高）リセット	0	0	0	0	1	1
4	CPU によって生成されたシステムリセット	0	0	0	1	0	1
5	WWDG リセット	0	1	0	0	0	1
6	IWDG リセット	0	0	1	0	0	1
7	不正停止エントリリセット	1	0	0	0	0	1

1. セットされているレジスタビットのセルは、灰色でハイライトされています。

11.4 RCC クロック機能の説明

次の 4 つのクロックソースを使用して、システムクロック (SYSCLK) を駆動できます：

- HSI：最大 64 MHz の高速内部 RC オシレータクロック
- CSI：低速内部 RC オシレータクロック
- HSE：高速外部クリスタルまたはクロック (4 ~ 50 MHz)
- PLL1 クロック

HSI は、32 MHz で設定され、リセットからの起動後、システムクロックソースとして使用されます。

デバイスには、次の追加のクロックソースがあります。

- LSI：32 kHz の低速内部 RC。独立型ウォッチドッグを駆動し、オプションで、STOP/STANDBY モードからの自動ウェイクアップに使用される RTC を駆動します。
- LSE：32.768 kHz の低速外部クリスタルまたはクロック。オプションで、リアルタイムクロック (rtc_ck) を駆動します。
- HSI48：内部 48 MHz RC。潜在的に、USB FS と RNG を駆動します。
- PLL2 および PLL3 クロック

それぞれのクロックソースは、使用しないときに個別にオン／オフを切り替えて、電力消費を最適化可能です。

複数のプリスケアラを使用して AHB 周波数、APB1 および APB2 ドメインを設定できます。AHB および APB ドメインの最大周波数は、250 MHz です。

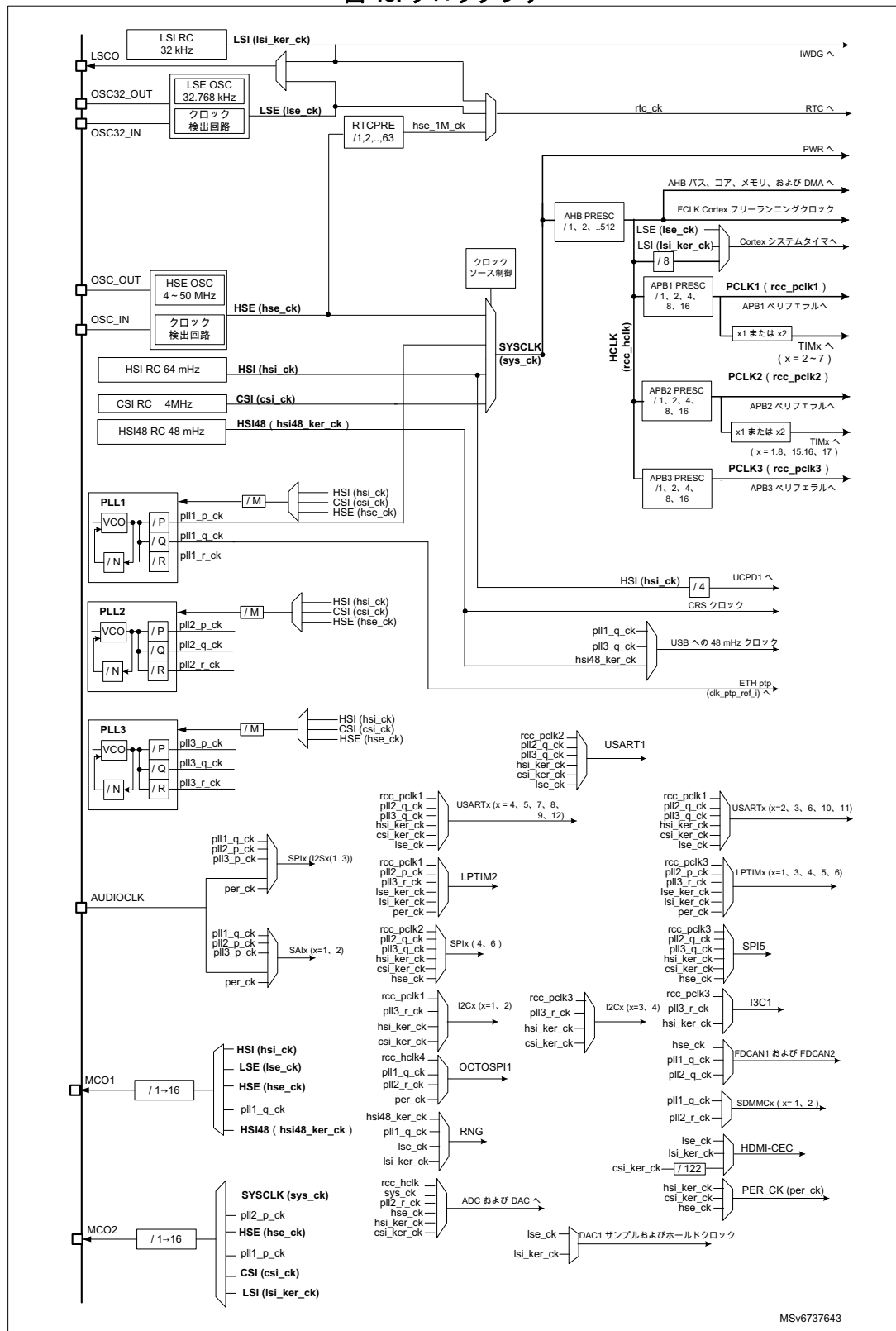
すべてのペリフェラルクロックは、バスクロック (HCLK、PCLK1、PCLK2、または PCLK3) から派生されますが、以下のものは、独立したカーネルクロックを受信します。ソフトウェアは、RCC_CCIPRx レジスタ (x = 1、2、3、4、5) を使用して、いくつかのソースからこのカーネルクロックを選択できます。

さらに、RTC カーネルクロックは、ソフトウェアによって RCC_BDCR で選択されます。IWDG クロックは、常に LSI 32 kHz クロックです。

RCC は、Cortex システムタイマ (SysTick) 外部クロックに、8 で除算された AHB クロック (HCLK)、LSE、または LSI を供給します。SysTick は、このクロックで、または Cortex クロック (HCLK) で直接動作でき、SysTick 制御およびステータスレジスタで設定可能です。

FCLK は、Cortex-M33 のフリーランニングクロックとして動作します。

図 48. クロックツリー



MSv6737643

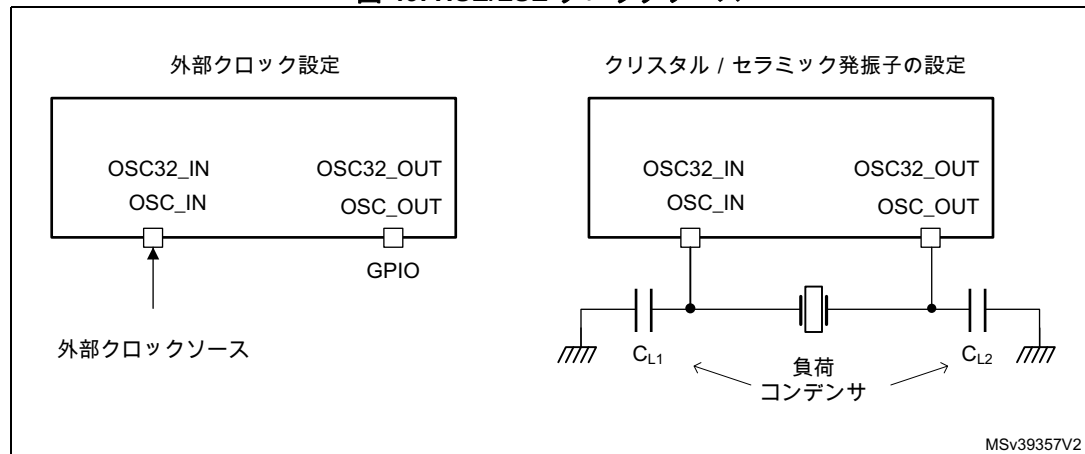
1. 内部/外部クロックソースの詳細については、データシートの電気的特性を参照してください。

11.4.1 HSE クロック

HSE ブロックでは、2 つの使用可能なソースからクロックを生成できます。

- 外部クリスタル/セラミック発振子
- 外部クロックソース

図 49. HSE/LSE クロックソース



外部クロックソース (HSE バイパス)

このモードでは、OSC_IN ピンに外部クロックソースを提供する必要があります。外部クロックは、小振幅（アナログ）とデジタルのいずれかです。このクロックがペリフェラルによって直接使用される場合、デューティサイクル要件は、ペリフェラルとアプリケーションによって定義されます（詳細については、データシートを参照）。

アナログクロック（小振幅）の場合、**RCC クロック制御レジスタ (RCC_CR)** の HSEBYP ビットと HSEON ビットを 1 にセットする必要があります。

デジタルクロックの場合、**RCC クロック制御レジスタ (RCC_CR)** の HSEBYP ビットと HSEEXT ビットを 1 にセットしてから、HSEON ビットを 1 にセットする必要があります。

外部クリスタル／セラミック発振子

オシレータは、HSEBYP ビットを 0 に、HSEON ビットを 1 にセットすることにより有効になります。

製品が非常に高精度のハイスピードクロックを必要とする場合は、HSE を使用できます。

関連するハードウェア設定を 図 49 に示します。出力ひずみと起動後の安定時間を少なくするために、発振子と負荷コンデンサは、可能な限りオシレータピンの近くに配置する必要があります。負荷コンデンサの値は、選択したクリスタルまたはセラミック発振子に応じて調整する必要があります。詳細については、データシートの電気的特性のセクションを参照してください。

RCC クロック制御レジスタ (RCC_CR) の HSERDY フラグは HSE オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、**hse_ck** クロック信号は出力されません。**RCC クロックソース割込み有効レジスタ (RCC_CIER)** で有効になっていれば、割込みを生成することができます。

HSE は、HSEON ビットを使用してオン／オフを切り替えることができます。次の 2 つの条件のうちのいずれかに該当する場合、HSE をオフにすることはできません。

- HSE がシステムクロックとして直接（ソフトウェアマルチプレクサを介して）使用されている。
- HSE が PLL1 のリファレンスクロックとして選択され、PLL1 が有効であり、（ソフトウェアマルチプレクサを介して）システムクロックを供給するために選択されている。

この場合、ハードウェアでは HSEON ビットを 0 にプログラミングすることはできません。

HSE は、システムが STOP または STANDBY モードに移行すると、ハードウェアによって自動的に無効化されます。

さらに、HSE クロックは MCO1 および MCO2 の出力に駆動することができ、他のアプリケーションコンポーネントのクロックソースとして使用できます。

11.4.2 HSI クロック

HSI ブロックは、製品にデフォルトクロックを供給します。

HSI は、ハイスピード内部 RC オシレータで、システムクロック、ペリフェラルクロック、または PLL 入力として直接使用できます。前置分周器では、アプリケーションで 8、16、32、または 64 MHz の HSI 出力周波数を選択することができます。この前置分周器は、HSIDIV で制御されます。

HSI の利点は以下のとおりです。

- 外部クリスタルが不要なため、低コストのクロックソース
- HSE よりも起動時間が短い（数ミリ秒）

HSI 周波数は、周波数の較正を実施しても、外部クリスタルオシレータやセラミック発振子よりも周波数の精度は劣ります。

HSI は、HSION ビットを使用してオン／オフを切り替えることができます。次の 2 つの条件のうちの 1 つが満たされると、HSI をオフにすることはできません。

- HSI がシステムクロックとして直接（ソフトウェアマルチプレクサを介して）使用されている。
- HSI が PLL1 のリファレンスクロックとして選択され、PLL1 が有効であり、（ソフトウェアマルチプレクサを介して）システムクロックを供給するために選択されている。

この場合、ハードウェアでは HSION ビットを 0 にプログラミングすることはできません。最低 1 つの有効な PLL（PLLxON ビットを 1 にセット）で HSI をリファレンスクロックとして選択している場合は、HSIDIV を変更することはできません。この場合、ハードウェアでは HSIDIV を新しい値で更新しません。ただし、HSI を直接システムクロックとして使用している場合は、HSIDIV を変更することができます。

HSIRDY フラグは、HSI が安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、HSI 出力クロックは出力されません。

HSI クロックは、HSE に障害がある場合のバックアップソース（補助クロック）としても使用できます（[セクション 11.4.10: クロックセキュリティシステム \(CSS\)](#) を参照）。HSI は、システムが STOP モードに移行する際に、無効にすることもできます。

さらに、HSI クロックは MCO1 出力に駆動することができ、他のアプリケーションコンポーネントのクロックソースとして使用できます。

HSI を通信ペリフェラルのカーネルクロックとして使用している場合は注意が必要です。アプリケーションでは以下のパラメータを考慮する必要があります。

- ペリフェラルがカーネルクロックリクエストを生成してから、クロックが実際に使用可能になるまでの時間間隔
- 周波数の精度

注： HSI は、システムが STOP モードのときに、有効のままにすることができます。

HSI 較正

RC オシレータの周波数は、製造工程でのばらつきのため、チップごとに異なります。このため、ACCHSI の精度に達するため、各デバイスは製造時に STMicroelectronics によって較正が行われています（詳細については、製品データシートを参照）。

パワーオンリセット後、工場較正值が HSICAL[11:0] ビットにロードされます。アプリケーションが電圧または温度の変動を受ける場合は、これが RC オシレータの周波数に影響を与えることがあります。ユーザアプリケーションでは、HSITRIM[6:0] ビットを使用して、HSI 周波数をトリミングできます。

注： HSICAL[11:0] および HSITRIM[6:0] ビットは、**RCC CSI 較正レジスタ (RCC_CSICFGR)** にあります。

11.4.3 CSI オシレータ

CSI は、低消費電力 RC オシレータで、システムクロック、ペリフェラルクロック、または PLL 入力として直接使用できます。

CSI の利点は以下のとおりです。

- 外部クリスタルが不要であるため、クロックソースのコストを抑えることが可能
- HSE よりも起動時間が短い（数ミリ秒）
- 非常に少ない消費電力

HSI では最大 64 MHz のクロック周波数を供給することができますが、CSI では約 4 MHz を供給します。

CSI 周波数は、周波数の較正を実施しても、外部クリスタルオシレータやセラミック発振子よりも周波数の精度は劣ります。

CSI は、CSION ビットを使用してオン／オフを切り替えることができます。CSIRDY フラグは、CSI が安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、CSI 出力クロックは出力されません。

次の 2 つの条件のうちの 1 つが満たされると、CSI をオフにすることはできません。

- CSI はシステムクロックとして直接（ソフトウェアマルチプレクサを介して）使用されます。
- CSI は PLL1 のリファレンスクロックとして選択されます。PLL1 はシステムクロックを供給するために有効化され、選択されます（ソフトウェアマルチプレクサを介して）。

この場合、ハードウェアでは CSION ビットを 0 にプログラミングすることはできません。

CSI は、システムが STOP モードに移行する際に、無効にすることもできます。

さらに、CSI クロックは MCO2 出力に駆動することができ、他のアプリケーションコンポーネントのクロックソースとして使用できます。

CSI の安定時間が HSI より速い場合でも、CSI を通信ペリフェラルのカーネルクロックとして使用している場合は注意が必要です。アプリケーションでは以下のパラメータを考慮する必要があります。

- ペリフェラルがカーネルクロックリクエストを生成する時点とクロックが実際に使用可能な時点とのあいだの時間間隔
- 周波数の精度

注： CSION および CSIRDY ビットは、**RCC クロック制御レジスタ (RCC_CR)** にあります。

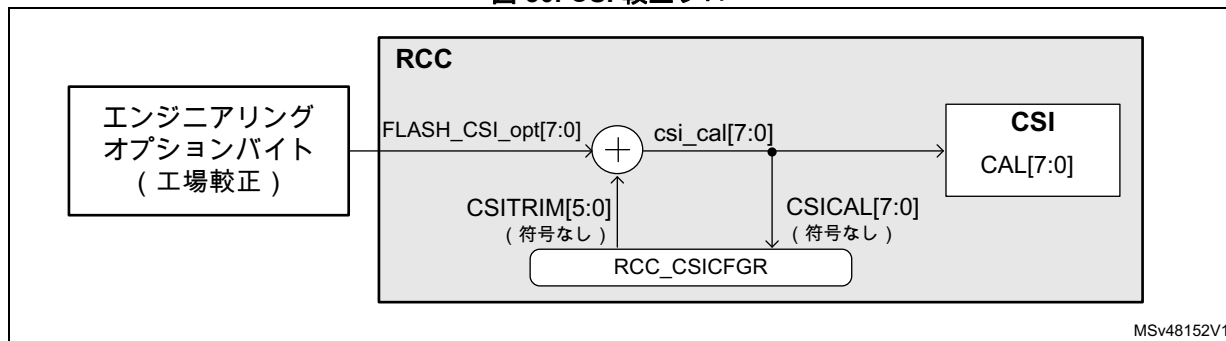
CSI 較正

RC オシレータの周波数は、製造工程でのばらつきのため、チップごとに異なります。このため、ACCcsi の精度を達成するため、各デバイスは製造時に較正されます（詳細については、製品データシートを参照）。リセット後、工場較正值が CSICAL[7:0] ビットにロードされます。

電圧や温度の変動は、RC オシレータの周波数に影響を与えます。ユーザアプリケーションでは、CSITRIM[5:0] ビットを使用して、CSI 周波数をトリミングできます。

注： CSICAL[7:0] および CSITRIM[5:0] ビットは、**RCC CSI 較正レジスタ (RCC_CSICFGR)** にあります。

図 50. CSI 較正フロー



11.4.4 HSI48 クロック

HSI48 クロック信号は、内部 48 mHz RC オシレータから生成され、直接 USB、または乱数生成器 (RNG) に使用できます。

内部 48 MHz RC オシレータを使用する主な目的は、特殊なクロックリカバリシステム (CRS) 回路によって USB ペリフェラルに高精度のクロックを提供することです。CRS は、USB SOF 信号、LSE または外部信号を使用して、オシレータの周波数をオンザフライで自動的かつ迅速に調整できます。これは、システムが STOP または STANDBY モードに移行した直後に無効化されます。CRS が使用されない場合、HSI48 RC オシレータはデフォルトの周波数で動作し、製造工程でのばらつきの影響を受けることがあります。

CRS ペリフェラルの設定および使用方法の詳細については、**セクション 12 : クロックリカバリシステム (CRS)** を参照してください。

RCC_CR レジスタの HSI48RDY フラグは、HSI48 RC オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、HSI48 RC オシレータ出力クロックはリリースされません。

HSI48 は、RCC_CR レジスタの HSI48ON ビットを使用してオン/オフできます。

11.4.5 PLL の説明

RCC には、次の 3 つの PLL が搭載されています。

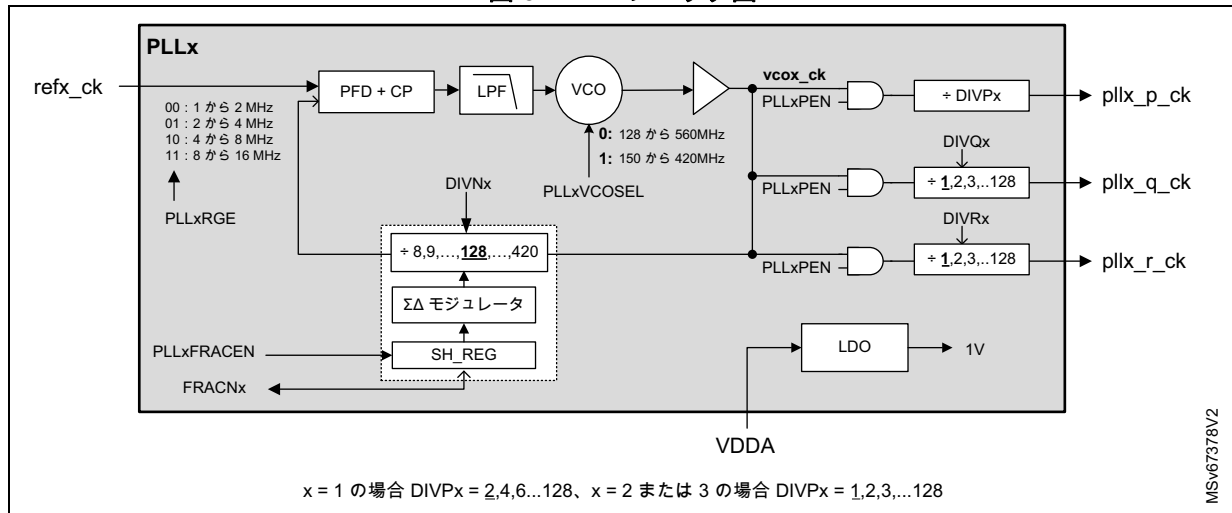
- メイン PLL (PLL1)。通常、CPU および一部のペリフェラルにクロックを供給するために使用されます。
- 2 つの専用 PLL (PLL2 と PLL3)。ペリフェラルのカーネルクロックを生成するために使用されます。

RCC に統合された PLL は、完全に独立しています。これらの PLL には以下の機能があります。

- 次の 2 つのモードをサポートする VCO
 - 広帯域
 - たとえば、オーディオアプリケーションでは低域が使用されます。
- 入力周波数範囲：
 - 広帯域 VCO では 2~16MHz
 - 低帯域 VCO では 1~2MHz

- 整数モードまたは分数モードで動作可能
- 11~0.3ppm のステップで VCO 周波数を微調整可能な 13 ビットのシグマデルタモジュレータ
- シグマデルタモジュレータは、PLL 出力上で周波数のオーバーシュートを生成することなく、オンザフライで更新できます。
- 各 PLL では、ポスト分周器で 3 つの出力を使用できます。

図 51. PLL ブロック図



PLL は、RCC_PLLxDIVR、RCC_PLLxFRACR、RCC_PLLCFGR、および RCC_CR レジスタによって制御されます。

PLL (refx_ck) に供給されるリファレンスクロックの周波数範囲は、1 から 16 MHz の間でなければなりません。RCC PLLx クロックソース選択レジスタ (RCC_PLLxCFGR) の PLLxM 分周器は、この条件に合わせて正しくプログラムする必要があります。さらに、パフォーマンスを最適化するには、RCC PLLx クロックソース選択レジスタ (RCC_PLLxCFGR) の PLLxRGE[1:0] フィールドは、リファレンス入力周波数に応じて設定する必要があります。

その後、ユーザアプリケーションは VCO を設定できます。リファレンスクロック周波数が 2 MHz 未満のときには、より狭い範囲 (150 ~ 420 mHz) を選択する必要があります。

消費電力を低減するには、VCO 出力を狭い方の範囲に設定することをお勧めします。

VCO 出力で期待される周波数を得るには、DIVNx ループ分周器をプログラムする必要があります。また、VCO 出力範囲を考慮する必要があります。

PLL は、FRACNx シャドウレジスタの SH_REG ビットの値を 0 に設定すると、整数モードで動作します。SH_REG ビットは、PLLxFRACEN ビットが 0 から 1 になると FRACNx 値とともに更新されます。シグマデルタモジュレータは、ごく小さな周波数ステップを可能にしながら、ジッタの影響を最小化するように設計されています。

PLL は、PLLxON を 1 にセットすることで有効化できます。PLLxRDY ビットは、PLL がレディ状態 (ロック状態) であることを示します。

注： PLL を有効にする前に、PLL に供給されるリファレンス周波数 (refx_ck) が安定していることを確認して、PLLx がオンのときにハードウェアによって PLLxM が変更されないようにしてください。これにより、PLL の 1 つがオンのときには、PLLSRC の変更が不可能になります。

ハードウェアは、PLL1 を使用してシステムクロックを供給している場合は、PLL1ON への 0 の書き込みを阻止します。クロックジェネレータには、その他のハードウェア保護もあります ([HSE クロック](#)、[HSI クロック](#)、および [CSI オシレータ](#) を参照)。

PLL が有効化された後は、次の PLL パラメータを変更することはできません。DIVNx、PLLxRGE、PLLxVCOSEL、PLLxP (DIVP)、PLLxQ (DIVQ)、および PLLxR (DIVR)。

ポスト分周器の 1 つ (DIVP、DIVQ、または DIVR) が使用されないときに PLL の最適な動作を得るには、アプリケーションは、イネーブルビット (DIVyEN) をセットするだけでなく、対応するポスト分周器ビット (DIVP、DIVQ、または DIVR) を 0 にセットする必要があります。

上記のルールに従わない場合、PLL 出力周波数の動作は保証されません。

出力周波数計算

PLL が整数モード (SH_REG = 0) で設定されている場合、VCO 周波数 (F_{VCO}) は以下の式で求めることができます。

$$F_{VCO} = F_{REF_CK} \times DIVN$$

$$F_{PLL_y_CK} = (F_{VCO} / (DIVy + 1)) \text{ with } y = P, Q \text{ or } R$$

PLL が分数モードで設定されている場合 (SH_REG が 0 以外の場合)、PLL を有効にする前に DIVN 分周器を初期化する必要があります。ただし、PLL 出力を損なうことなく FRACNx の値を動作中に変更することができます。

この機能は、精度の良い任意のクリスタル値から特定の周波数を生成するため、または動作中に周波数を微調整するために使用できます。

各 PLL の VCO 周波数は、次の式で与えられます。

$$F_{VCO} = F_{ref_ck} \times \left(DIVN + \frac{FRACN}{2^{(13)}} \right)$$

注： PLL1 の場合、DIVP は奇数値のみをとることができます。

PLL は、以下の場合にハードウェアによって無効にされます。

- システムが STOP または STANDBY モードに移行したとき
- HSE または HSE でクロックされた PLL をシステムクロックとして使用して、HSE で障害が発生したとき

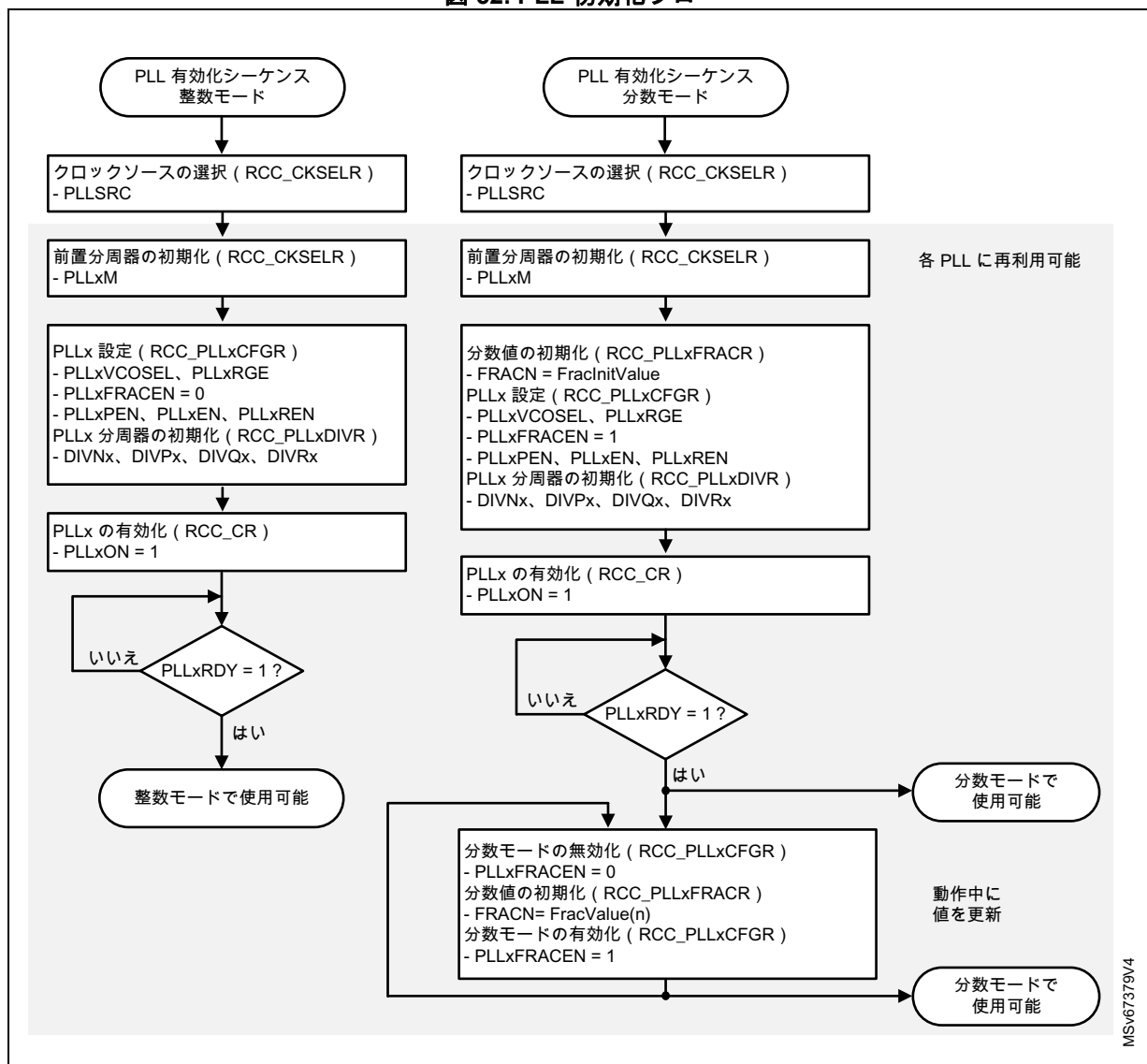
PLL 初期化フェーズ

図 52 に推奨する整数モードまたは分数モードの PLL 初期化シーケンスを示します。PLLx は、初期化シーケンスの始めに無効にします。

1. 必要な周波数に応じて PLL レジスタを初期化します。
 - 整数モードの場合は、RCC PLLx クロックソース選択レジスタ (RCC_PLLxCFGR) の PLLxFRACEN を 0 にセットします。
 - 分数モードの場合は、FRACN を必要な初期値 (FracInitValue) にセットし、その後 PLLxFRACEN を 1 にセットします。
2. PLLxON ビットを 1 にセットすると、ユーザアプリケーションは PLLxRDY ビットが 1 にセットされるまで待つ必要があります。PLLx が分数モードの場合、PLLxRDY = 0 である限り PLLxFRACEN ビットを 0 に設定し直すことはできません。
3. PLLxRDY ビットを 1 にセットすると、PLLx を使用できるようになります。
4. アプリケーションで PLLx 周波数を動作中に調整する場合には、以下の条件があります (分数モードのみ)。
 - a) PLLxFRACEN を 0 に設定する必要があります。PLLxFRACEN = 0 のとき、シグマデルタモジュレータは、値が SH_REG にラッチされても動作しています。アプリケーションは、refx_ck の 3 クロック周期の間、待つ必要があります (PLLxFRACEN ビットの伝播遅延)。
 - b) PLLxFRACR (FracValue(n)) に新しい値をロードする必要があります。
 - c) PLLxFRACR の内容をシャドーレジスタにラッチするには、PLLxFRACEN を 1 にセットする必要があります。新しい値は、refx_ck の 3 クロック周期後に考慮されます (PLLxFRACEN ビット伝播遅延)。

注 : PLLxRDY が 1 になると、PLLx 出力周波数とターゲット値の差は $\pm 2\%$ 未満になります。

図 52. PLL 初期化フロー



11.4.6 LSE クロック

LSE ブロックでは、2 つの使用可能なソースからクロックを生成できます。

- 外部クリスタル/セラミック発振子
- 外部ユーザクロック

外部クロックソース (LSE バイパス)

このモードでは、OSC32_IN ピンに外部クロックソースを提供する必要があります。入力クロックは、最大 1 mHz までの周波数を持ち、小振幅（アナログ）またはデジタルが可能です。50% に近いデューティサイクルが推奨されます。

この外部クロックは OSC32_IN ピンに供給され、OSC32_OUT ピンはハイインピーダンスに保たれる必要があります（図 48 を参照）。

アナログクロック（小振幅）の場合、LSEBYP ビットと LSEON ビットを 1 にセットする必要があります（[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#)）。

デジタルクロックの場合、LSEBYP ビットと LSEEXT ビットを 1 にセットしてから、LSEON ビットを 1 にセットする必要があります（[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#)）。RTC が使用される場合、LSE バイパスは、デジタルモードではなく、小振幅アナログモード（リセット後のデフォルト値）に設定する必要があります。

外部クリスタル／セラミック発振子（LSE クリスタル）

LSE クロックは、32.768 kHz のクリスタルまたはセラミック発振子から生成されます。時計/カレンダーまたはその他のタイミング機能のためのリアルタイムクロック（RTC）に、低消費電力の高精度なクロックソースを供給できるという利点があります。

[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の LSERDY フラグは、LSE クリスタルが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、LSE クリスタル出力クロック信号はリリースされません。[RCC クロックソース割込み有効レジスタ \(RCC_CIER\)](#) で有効になっていれば、割込みを生成することができます。

LSE オシレータは、LSEON ビットを使用してオン／オフの切り替えができます。LSE は、システムの STOP または STANDBY モードへの移行時にも有効なままです。

さらに、LSE クロックは MCO1 出力に駆動することができ、他のアプリケーションコンポーネントのクロックソースとして使用できます。

LSE は、アンプの駆動能力を調節するために使用できるプログラム可能な駆動能力（LSEDRV[1:0]）も提供します。この駆動能力は、安定した発振を確保するために、外部クリスタル/セラミック部品要件に従って選択されます。

駆動能力は、LSE オシレータを有効にする前にセットする必要があります。

11.4.7 LSI クロック

LSI は、独立型ウォッチドッグ（IWDG）や自動ウェイクアップユニット（AWU）で、システムが STOP モードまたは STANDBY モードの場合に動作状態を維持できる低電力のクロックソースとして動作します。

クロック周波数は約 32 kHz です。詳細については、データシートの電気的特性のセクションを参照してください。

LSI は、LSION ビットを使用してオン／オフを切り替えることができます。LSIRDY フラグは、LSI オシレータが安定しているかどうかを示します。独立型ウォッチドッグがハードウェアまたはソフトウェアとして開始された場合、LSI は強制的にオンになり、オフにすることはできません。

LSI は、システムの STOP または STANDBY モードへの移行時にも有効なままです。

LSI の起動時に、LSIRDY ビットがハードウェアによってセットされるまで、クロックは供給されません。[RCC クロックソース割込み有効レジスタ \(RCC_CIER\)](#) で有効になっていれば、割込みを生成することができます。

さらに、LSI クロックは MCO2 出力に供給され、他のアプリケーションコンポーネントのクロックソースとして使用できます。

注： LSION および LSIRDY ビットは、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) にあります。

11.4.8 システムクロック (SYSCLK) の選択

次の 4 つのクロックソースを使用して、システムクロック (SYSCLK) を駆動できます：

- HSI オシレータ
- CSI オシレータ
- HSE オシレータ
- PLL

システムクロックの最大周波数は 250 MHz です。システムリセット後（または STANDBY モードの終了後）、HSI オシレータが 32 MHz のシステムクロックとして選択されます。クロックソースが直接、または PLL を経由してシステムクロックとして使用されているときに、このクロック信号を停止することはできません。

あるクロックソースから別のクロックソースへの切り替えは、切り替え後に使用するクロックソースの準備ができていない場合（起動遅延時間を経てクロックが安定している状態、または、PLL がロックされている状態）にのみ行われます。まだレディではないクロックソースが選択された場合、クロックソースがレディになると、切り替えが行われます。[RCC クロック制御レジスタ \(RCC_CR\)](#) のステータスビットは、どのクロックがレディであるかと、どのクロックがシステムクロックとして使用されているかを示します。

11.4.9 STOP モードおよび STANDBY モードでのクロックジェネレータの処理

システム全体が STOP モードに移行すると、すべてのクロック（システムおよびカーネルクロック）と次のクロックソースが停止します。

- CSI、HSI (HSIKERON および CSIKERON ビットによる)
- HSE
- PLL1 および PLL2
- HSI48

RCC レジスタの内容は変更されませんが、PLL1ON、PLL2ON、HSEON、および HSI48ON は 0 にセットされます。

STOP モードの終了

システムがウェイクアップイベントによってこのモードを終了すると、アプリケーションは、再起動に使用するオシレータ (HSI および/または CSI) を選択できます。STOPWUCK ビットがシステムクロックとして使用されるオシレータを選択します。STOPKERWUCK ビットがペリフェラルのカーネルクロックとして使用されるオシレータを選択します。STOPKERWUCK ビットは、システムクロックとは異なるオシレータによって生成されたカーネルクロックを、ペリフェラルがシステム STOP 後に必要とする場合に有効です。

これらのビットは、[RCC クロック設定レジスタ 1 \(RCC_CFGR1\)](#) に属します

表 103. STOPWUCK と STOPKERWUCK の説明

STOPWUCK	STOPKERWUCK	システム STOP モード終了時の 有効化されたオシレータ	システムが STOP モードを終了する ときに分配されるクロック	
			システムクロック	カーネルクロック
0	0	HSI	HSI	HSI
	1	HSI および CSI		HSI および／ または CSI
1	0		CSI	CSI
	1			

STOP モード時

このモードで HSI または CSI を有効にできるケースが 2 つあります。

- 専用ペリフェラルがカーネルクロックをリクエストしたとき、このペリフェラルについて (CKPERSEL[1:0] を介して) 選択されたカーネルクロックソースに従って、ペリフェラルは HSI または CSI を受信します。
- RCC クロック制御レジスタ (RCC_CR)** の HSIKERON または CSIKERON ビットがセットされているとき、HSI および CSI は動作し続けますが、出力はゲーティングされます。システムが STOP モードを終了するか、ペリフェラルがカーネルクロックをリクエストすると、クロックは直ちに使用可能になります (詳細については、表 101 を参照)。

表 104. HSIKERON および CSIKERON の動作

HSIKERON (CSIKERON)	STOP モード時の HSI (CSI) 状態	HSI (CSI) 状態設定時間
0	OFF	$t_{su(HSI)} t_{su(CSI)}$ ⁽¹⁾
1	動作及びクロックゲート	即時

1. $t_{su(HSI)}$ と $t_{su(CSI)}$ は、それぞれ HSI オシレータと CSI オシレータの起動時間です (これらのパラメータの値については、製品データシートを参照)。

マイクロコントローラがシステム STANDBY モードを終了すると、HSI はシステムおよびカーネルクロックとして選択されます。RCC レジスタは初期値にリセットされますが、RCC_RSR および RCC_BDCR レジスタを除きます。

注： HSI および CSI 出力は、2つのクロックパスを提供します。

- 1 つはシステムクロック用 (hsi_ck または csi_ck) のパス
- 1 つはペリフェラルカーネルクロック用 (hsi_ker_ck または csi_ker_ck) のパス

ペリフェラルがシステム STOP モードでカーネルクロックをリクエストすると、hsi_ker_ck または csi_ker_ck を提供するパスのみがアクティブになります。

11.4.10 クロックセキュリティシステム (CSS)

HSE のクロックセキュリティシステム

クロックセキュリティシステムは、ソフトウェアによって HSECSSON ビットを介して有効化でき、HSEON が 0 にセットされているときでも有効化できます。

HSE の CSS は、HSE が有効かつレディ状態であり、HSECSSON が 1 にセットされている場合に、ハードウェアによって有効にすることができます。

HSE の CSS は、HSE が無効にされたときに無効化されます。結果として、この機能はシステムが STOP モードのときは機能しません。

HSECSSON ビットをソフトウェアによって直接クリアすることはできません。

HSECSSON ビットは、システムリセットが発生したとき、またはシステムが STANDBY モードに移行するときに、ハードウェアによってクリアされます。

HSE クロックで障害が検出された場合、**RCC クロック設定レジスタ 1 (RCC_CFGR1)** の STOPWUCK ビットの設定に応じて、システムは HSI または CSI に自動的に切り替えて、安全なクロックを提供します。その後、HSE は自動的に無効化され、クロック障害イベントが高機能制御タイマ (TIM1) のブレーク入力に送信され、NMI が自動的に生成されて、障害をアプリケーションに通知するため、MCU はレスキュー操作を実行できます。障害発生時に HSE 出力が PLL のクロックソースとして使用されている場合は、PLL も無効になります。

CSS を有効にしたときに HSE クロック障害が発生する場合は、CSS が 割込みを生成します。これにより、NMI の自動生成が実行されます。**RCC クロックソース割込みフラグレジスタ (RCC_CIFR)** の HSECSSF フラグは、アプリケーションで障害の原因を見つけるために 1 にセットされます。NMI ルーチンは、HSECSSF ビットがクリアされるまで、無限に実行されます。つまり、アプリケーションでは **RCC クロックソース割込みクリアレジスタ (RCC_CICR)** の HSECSSC ビットをセットして、NMI ISR で HSECSSF フラグをクリアする必要があります。

LSE のクロックセキュリティシステム

LSE のクロックセキュリティシステムは、**RCC バックアップドメイン制御レジスタ (RCC_BDCR)** の LSECSSON ビットを書き込むことによって、ソフトウェアで有効にできます。このビットは、ハードウェアリセットまたは RTC ソフトウェアリセット、または LSE で障害検出後にのみ無効にできます。LSE が有効化され (LSEON 有効)、レディ状態 (ハードウェアによって LSERDY がセット) になった後、また、RTCSEL によって RTC クロックが選択された後、LSECSSON に書き込む必要があります。

LSE の CSS は、VBAT を含め、すべてのモードで動作します。システムリセット (パワーオンリセットを除く) でも動作します。

LSE のクロックセキュリティシステムは、LSE が消えたことや過周波数を検出します。また、LSE のグリッチは、LSEGFON をセットすることでフィルタリングできます。LSEGFON は、LSE が無効 (LSEON = 0 かつ LSERDY = 0) なときに書き込む必要があります。

外部 32 kHz オシレータで障害が検出された場合、LSE クロックは RTC に供給しなくなりますが、レジスタに対するハードウェアアクションは行われません。

LSE 検出イベントの CSS は、TAMP ペリフェラルの内部タンパ 3 に接続されます。低電力モードからウェイクアップするには、内部タンパ 3 が有効 (TAMP_CR1 レジスタの ITAMP3E = 1) であり、関連する割込みが有効 (TAMP_IER の ITAMP3IE) である必要があります。これにより、TAMP バックアップレジスタとバックアップ SRAM も消去されますが、TAMP_CR3 の ITAMP3NOER = 1 の場合を除きます (詳細については、**セクション 47: タンパおよびバックアップレジスタ (TAMP)** を参照)。

LSE 検出イベントの CSS の場合 (RCC_BDCR の LSECSSD = 1)、ソフトウェアは、LSECSSON ビットを無効化し、欠陥のある 32 kHz オシレータを停止し (LSEON を無効化)、RTC クロックソースを変更するか (RTCSEL で、クロックなし、または LSI あるいは HSE)、アプリケーションをセキュリティ保護するアクションをとる必要があります。

LSE の CSS の電気特性については、データシートを参照してください。

11.4.11 クロック出力の生成（MCO1/MCO2）

2つのマイクロコントローラクロック出力ピン（MCO1 および MCO2）を使用できます。出力ごとにクロックソースを選択できます。選択したクロックは、設定可能なプリスケアラによって分周できます（信号選択の詳細については、図 48 を参照）。

MCO1 および MCO2 出力は、**RCC クロック設定レジスタ 1 (RCC_CFGR1)** にある MCO1PRE[3:0]、MCO1[2:0]、MCO2PRE[3:0]、および MCO2[2:0] を介して制御されます。

各 MCO ピンに対応する GPIO ポートを、オルタネート機能モードでプログラムする必要があります。

MCO 出力に提供されるクロックは、最大ピン速度を超えない必要があります（サポートされる速度については、製品データシートを参照）。

もう 1 つの出力（LSCO）により、低速クロック（LSI、LSE）の 1 つを外部 LSCO ピンに出力できます。この出力は、STOP モードで使用できますが、STANDBY および VBAT モードでは使用できません。選択は LSCOSEL ビットによって制御され、**RCC バックアップドメイン制御レジスタ (RCC_BDCR)** の LSCOEN ビットによって有効化されます。

MCO クロック出力には、MCO ピンで対応するオルタネート機能を選択する必要があります。LSCO ピンはデフォルトの POR 状態のままとしてください。

11.4.12 カーネルクロックの選択

一部のペリフェラルは、非同期的に動作する 2 つの異なるクロックドメインと動作するよう設計されています。

- レジスタおよびバスインタフェースと同期するクロックドメイン（**ckg_bus_perx** クロック）
- 通常ペリフェラルと同期するクロックドメイン（カーネルクロック）

これらの 2 つのクロックドメインをサポートするペリフェラルを持つ利点は、ユーザアプリケーションで CPU、バスマトリックス、およびペリフェラルのカーネル部分の最適化されたクロック周波数をより自由に選択できるという点です。このため、ユーザアプリケーションは、ペリフェラルを再プログラミングしなくても、バス周波数を変更できます。たとえば、APB クロックがオンザフライで変更されても、進行中の UART との転送は妨げられません。

表 105 に、RCC がペリフェラルに供給可能なカーネルクロックを示します。各行は、出力に接続されたマルチプレクサとペリフェラルを表します。

表 105. カーネルクロック分配の概要

ペリフェラル	クロック マルチプレクサ 制御ビット	pll1_q_ck	pll2_p_ck	pll2_q_ck	pll2_r_ck	pll3_p_ck	pll3_q_ck	pll3_r_ck	sys_ck	バスクロック ⁽¹⁾	hse_ck	hsi_ker_ck	csi_ker_ck	hsi48_ck	lse_ck	lsi_ck	per_ck ⁽²⁾	AUDIOCLK	無効
OCTOSPI	OSPISEL	1	-	-	2	-	-	-	-	0	-	-	-	-	-	-	3	-	-
SDMMC1 ⁽³⁾	SDMMC1SEL	0	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SDMMC2 ⁽³⁾	SDMMC2SEL	0	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FDCAN	FDCANSEL	1	-	2	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-
CEC	CECSEL	-	-	-	-	-	-	-	-	-	-	-	2 ⁽⁴⁾	-	0	1	-	-	-
I2C1	I2C1SEL	-	-	-	-	-	-	1	-	0	-	2	3	-	-	-	-	-	-
I2C2	I2C2SEL	-	-	-	-	-	-	1	-	0	-	2	3	-	-	-	-	-	-
I2C3	I2C3SEL	-	-	-	-	-	-	1	-	0	-	2	3	-	-	-	-	-	-

表 105. カーネルクロック分配の概要 (続き)

ペリフェラル	クロック マルチプレクサ 制御ビット	pll1_q_ck	pll2_p_ck	pll2_q_ck	pll2_r_ck	pll3_p_ck	pll3_q_ck	pll3_r_ck	sys_ck	バスクロック ⁽¹⁾	hse_ck	hsi_ker_ck	csi_ker_ck	hsi48_ck	lse_ck	lsi_ck	per_ck ⁽²⁾	AUDIOCLK	無効
I2C4	I2C4SEL	-	-	-	-	-	-	1	-	0	-	2	3	-	-	-	-	-	-
I3C1	I3C1SEL	-	-	-	-	-	-	1	-	0	-	2	-	-	-	-	-	-	-
LPTIM1	LPTIM1SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
LPTIM2	LPTIM2SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
LPTIM3	LPTIM3SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
LPTIM4	LPTIM4SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
LPTIM5	LPTIM5SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
LPTIM6	LPTIM6SEL	-	1	-	-	-	-	2	-	0	-	-	-	-	3	4	5	-	-
TIM[8:1],	-	-	-	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-	-
TIM[17:12]	-	-	-	-	-	-	-	-	-	x	-	-	-	-	-	-	-	-	-
TIM16/17	TIMICSEL	-	-	-	-	-	-	-	-	-	-	1	1	-	-	-	-	-	0
RNG	RNGSEL	1	-	-	-	-	-	-	-	-	-	-	-	0	2	3	-	-	-
SAI1	SAI1SEL	0	1	-	-	2	-	-	-	-	-	-	-	-	-	-	4	3	-
SAI2	SAI2SEL	0	1	-	-	2	-	-	-	-	-	-	-	-	-	-	4	3	-
SPI(I2S)1	SPI1SEL	0	1	-	-	2	-	-	-	-	-	-	-	-	-	-	4	3	-
SPI(I2S)2	SPI2SEL	0	1	-	-	2	-	-	-	-	-	-	-	-	-	-	4	3	-
SPI(I2S)3	SPI3SEL	0	1	-	-	2	-	-	-	-	-	-	-	-	-	-	4	3	-
SPI4	SPI4SEL	-	-	1	-	-	2	-	-	0	5	3	4	-	-	-	-	-	-
SPI5	SPI5SEL	-	-	1	-	-	2	-	-	0	5	3	4	-	-	-	-	-	-
SPI6	SPI6SEL	-	-	1	-	-	2	-	-	0	5	3	4	-	-	-	-	-	-
USARTx	USARTxSEL	-	-	1	-	-	2	-	-	0	-	3	4	-	5	-	-	-	-
UARTx	UARTxSEL	-	-	1	-	-	2	-	-	0	-	3	4	-	5	-	-	-	-
LPUART1	LPUART1SEL	-	-	1	-	-	2	-	-	0	-	3	4	-	5	-	-	-	-
USB	USBSEL	1	-	-	-	-	2	-	-	-	-	-	-	3	-	-	-	-	0
ADC DAC ⁽⁵⁾	ADC DACSEL	-	-	-	2	-	-	-	1	0	3	4	5	-	-	-	-	-	-
DAC	DACSEL	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	-	-	-
UCPD1	-	-	-	-	-	-	-	-	-	-	-	x ⁽⁶⁾	-	-	-	-	-	-	-
ETH (ptp)	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RTC/AWU	RTCSEL	-	-	-	-	-	-	-	-	-	3 ⁽⁷⁾	-	-	-	1	2	-	-	0
すべて	CKPERSEL	-	-	-	-	-	-	-	-	-	2	0	1	-	-	-	-	-	3

1. バス (AP または AHB) クロックは、ペリフェラルが接続されたバスインタフェースクロックです。
2. per_ck クロックは、CKPERSEL の選択に従って、hse_ck、hsi_ker_ck、または csi_ker_ck です。
3. デューティサイクルが約 50% の場合、DIV[P/Q/R]x 値は偶数である必要があります。SDMMCx では、DDR をサポートするデューティサイクルは 50% である必要があります。
4. クロック CSI は 122 分周されます。

5. デューティサイクルが約 50% の場合、 $DIV[P/Q/R] \times$ 値は偶数である必要があります。ADC については、DDR をサポートするとき、デューティサイクルは 50% である必要があります。
6. クロック HSI は 4 分周されます。
7. クロック HSE は RTCPRE で分周されます。

切り替え回数を減らすために、一部のペリフェラルは同じカーネルクロックソースを共有します。ただし、すべてのペリフェラルは専用のイネーブル信号を持っています。

オーディオアプリケーション専用のペリフェラル

オーディオペリフェラルは、一般に、特定の正確な周波数を必要とし、SAI または SPI(I2S) のカーネルクロックは、次によって生成できます。

- PLL1、アクティブな PLL の数を減らす必要がある場合 (SAI および SPI/I2S1~3)
- APB2 ペリフェラルクロック (SPI/I2S4 および 5)
- APB3 ペリフェラルクロック (SPI/I2S6)
- PLL2 または 3、周波数生成時に最適な柔軟性を得るため
- HSE、HSI、または CSI、消費電流が重要な使用例の場合
- AUDIOCLK、外部クロックリファレンスを使用する必要がある場合

制御およびデータ転送専用のペリフェラル

SPI、I2C、および UART では、正しいボーレートを生成可能な速度のクロック、またはシリアルインタフェースで必要なビットクロックを除き、特定のカーネルクロック周波数を必要としません。このため、次の中からソースを選択できます。

- PLL1、アクティブな PLL の数を減らす必要がある場合
- PLL2 または PLL3、より高い柔軟性が必要な場合。たとえば、この 2 つの方法では、一部のシリアルインタフェースの速度に影響を与えずに、PLL1 によって周波数バスを変更できます。
- HSI または CSI、低電力の使用例、またはペリフェラルを迅速に STOP モードからウェイクアップする必要がある場合 (UART、I2C など)

注： 高いボーレートが必要ではない場合、UART では LSE クロックが必要です。

OCTOSPI および SDMMC1/2 は、より高い柔軟性を得るために、バスインタフェースクロックとは異なるクロックを使用することもできます。

RTC/AWU クロック

rtc_ck クロックソースは次のいずれかです。

- hse_1M_ck (hse_ck をプログラム可能なプリスケアラで分周)
- lse_ck
- lsi_ck クロック

このソースクロックは、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1:0] ビットおよび [RCC クロック設定レジスタ 1 \(RCC_CFGR1\)](#) の RTCPRE[5:0] ビットをプログラムすることによって、選択できます。

この選択は、バックアップドメインをリセットしない限り変更できません。

RTC クロックとして LSE が選択されている場合、バックアップまたは V_{DD} 電源が無くなっても RTC は正常に動作します。

LSE クロックはバックアップドメインに属しますが、その他のオシレータはそうではありません。その結果、以下ようになります。

- LSE が RTC クロックとして選択されている場合、 V_{DD} 供給がオフになった場合でも、 V_{BAT} の供給が保たれている限り、RTC は動作を続けます。
- LSI が RTC クロックとして選択されている場合、 V_{DD} 供給がオフになると、AWU の状態は保証されません。
- HSE クロックが RTC クロックとして選択されている場合、 V_{DD} 供給または V_{CORE} 供給がオフになると、RTC の状態は保証されません。

rtc_ck クロックは [RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN ビットを使用して有効化されます。

RTC バスインタフェースクロック (APB クロック) は、RCC_APB3ENR/LPENR レジスタの RTCAPBEN および RTCAPBLPEN ビットを使用して有効化されます。

注： APB クロック周波数が RTC クロック周波数の 7 倍 未満である場合 ($F_{APB} < 7 \times F_{RTCLK}$) に RTC カレンダレジスタを読み込むには、ソフトウェアはカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR への 2 回目の読み出しアクセスの結果が最初の読み出しアクセスの結果と同じであれば、データは正しいということになります。同じでない場合は、3 回目の読み出しアクセスを実行する必要があります。

ウォッチドッグクロック

RCC では、回路で使用可能な 2 つのウォッチドッグブロックにクロックを供給します。独立型ウォッチドッグ (IWDG) は、LSI に接続されます。ウィンドウ型ウォッチドッグ (WWDG) は、APB クロックに接続されます。

独立型ウォッチドッグがハードウェアのオプションまたはソフトウェアアクセスによって開始された場合、LSI は強制的にオンになり、オフにすることはできません。LSI オシレータのセットアップ遅延後、このクロックが IWDG に供給されます。

TIMx を使用したクロック周波数の測定

クロックソースジェネレータのほとんどの周波数は、TIMx の入力キャプチャで測定できます。

- LSE による HSI または CSI の較正
LSE を TIMx の入力キャプチャに接続しておく主な目的は、HSI または CSI を正確に測定できるようにすることです。このとき、HSI または CSI を、システムクロックソースとして、直接または PLL1 を通じて使用する必要があります。LSE 信号の連続したエッジ間のシステムクロックカウンタ数により、内部クロック周期が測定されます。LSE クリスタルの高い精度 (通常は数十 ppm) を利用することにより、同じ分解能で内部クロック周波数を判断でき、製造工程、温度、電圧に関連した周波数偏差に対してクロックソースを調整して補正することができます。
この基本概念は、比較測定 (HSI/LSE 比など) ができることにあります。そのため、精度は 2 つのクロックソース間の比に密接にリンクしています。この比が大きければ大きいほど、測定精度は向上します。
HSI および CSI オシレータには、上記の目的のためにユーザがアクセスできる専用の較正ビットがあります ([RCC CSI 較正レジスタ \(RCC_CSICFGR\)](#) を参照)。PLLx によって HSI または CSI が使用されると、PLL の分数分周器を使用してシステムクロックを微調整することもできます。
- HSI による LSI の較正
LSI 周波数も測定できます。クリスタルを使用しないアプリケーションで有効です。超低消費電力 LSI オシレータには、大きな製造工程偏差があります。LSI クロック周波数は、より高精度な HSI クロックソースで測定できます。この測定を使用して、より正確な RTC 時間基準のタイムアウト (LSI が RTC クロックソースとして使用されているとき) や許容精度の IWDG タイムアウトを得ることができます。

11.4.13 RTC および TAMP クロック

RTCCLK クロックソースは、RTC および TAMP によって使用され、HSE/32、LSE、または LSI クロックのいずれかです。これは、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1:0] ビットをプログラムすることで選択できます。この選択は、バックアップドメインをリセットしない限り変更できません。RTC を正常に動作させるには、PCLK 周波数が常に RTCCLK 周波数以上になるようにシステムを設定する必要があります。バックアップレジスタのみが使用され、タンパがエッジ検出モードの場合、TAMP はカーネルクロックを必要としません。その他すべてのタンパ検出モードでは、カーネルクロックが必要です（詳細については、[セクション 47：タンパおよびバックアップレジスタ \(TAMP\)](#) を参照）。

LSE はバックアップドメインにありますが、HSE および LSI クロックはバックアップドメインにはありません。そのため、次のようになります。

- LSE が RTC および TAMP クロックとして選択された場合、これらのペリフェラルは V_{DD} 供給がオフになっても動作し続けますが、 V_{BAT} 供給が維持されている必要があります。
- プリスケアラで分周された HSE クロックが RTC または TAMP クロックとして使用されている場合、 V_{DD} 供給がオフになるか、内部電圧レギュレータがオフになった場合（コアドメインから電源が除去される）、RTC の状態は保証されません。TAMP 設定に応じて、カーネルクロックを必要としないモードで使用された場合、これは機能し続けることができます。
- LSI が RTC または TAMP クロックとして使用されている場合、 V_{DD} 供給がオフになった場合、RTC の状態は保証されません。TAMP 設定に応じて、カーネルクロックを必要としないモードで使用された場合、これは機能し続けることができます。

RTC および TAMP クロックが LSE または LSI の場合、RTC は、システムリセット中でもクロック供給され、機能を保ちます。

11.4.14 タイマクロック

タイマクロック周波数は、ハードウェアによって自動で設定されます。

これには、2 つのケースがあります。

- APB プリスケアラが 1 の場合、タイマクロック周波数は APB ドメインの周波数に設定されます。
- そうでない場合は、APB ドメインの周波数の 2 倍に設定されます。

11.4.15 ウォッチドッグクロック

独立型ウォッチドッグ (IWDG) がハードウェアオプションまたはソフトウェアアクセスによって開始された場合、LSI オシレータは強制的にオンになり、オフにすることはできません。LSI オシレータの過渡期間が終わると、LSI 32 kHz クロックが IWDG に供給されます。

11.4.16 ペリフェラルクロックゲーティングおよび自律モード

RUN モードでのペリフェラルクロックゲーティング

各ペリフェラルクロックは、RCC_AHBxENR および RCC_APBxENR レジスタの対応する EN ビットによって有効にできます。

ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタへの読み出し／書き込みアクセスはサポートされません。

有効化ビットには、グリッチのないペリフェラルのクロックを生成する同期メカニズムが含まれています。イネーブルビットがセットされた後、ペリフェラルバスクロックの 2 サイクル後にクロックはアクティブになります。

注意： ペリフェラルのクロックを有効にした直後、ソフトウェアは、ペリフェラルレジスタにアクセスする前に、この 2 クロックサイクルだけ待つ必要があります。

SLEEP モードでのペリフェラルクロックゲーティング

ペリフェラルが有効なとき、デバイスが SLEEP モードのときには、クロックは自動的にゲートオフされます。このためには、RCC_AHBxLPENR および RCC_APBxLPENR レジスタのペリフェラル LPEN ビットをクリアします。SLEEP モードでクロックをオンに保つには、ペリフェラルの EN ビットと LPEN ビットの両方をセットする必要があります。

11.5 RCC セキュリティおよび特権機能の説明

11.5.1 RCC TrustZone セキュリティ保護モード

Flash オプションバイト設定 レジスタの TZEN オプションバイトによって TrustZone セキュリティが有効なとき、RCC は RCC 設定およびステータスビットが非セキュアアクセスによって変更されないように保護することができます。

これは、[RCC セキュア設定レジスタ \(RCC_SECCFGR\)](#) を通じて設定され、非セキュアアクセスによって以下の機能が読み出されたり変更されたりするのを防ぎます。

- HSE、HSE-CSS、HSI、CSI、LSI、LSE、LSE-CSS、LSCO、HSI48 の設定およびステータスビット
- PLL1、PLL2、PLL3、AHB、および APB ブリスマーラ設定およびステータスビット
- システムクロック (SYSCLK) および ICLK ソースクロック選択およびステータスビット
- MCO クロック出力設定および STOPWUCK および STOPKERWUCK ビット
- リセットフラグ削除 RMVF 設定

[RCC 特権設定レジスタ \(RCC_PRIVCFGR\)](#) で SPRIV がセットされている場合、RCC_SECCFGR レジスタはセキュア特権アクセスによってのみ書き込むことができます。RCC_PRIVCFGR で SPRIV がクリアされた場合、RCC_SECCFGR は特権または非特権のセキュアアクセスによって書き込むことができます。

RCC_SECCFGR は、セキュア、非セキュア、特権、および非特権アクセスによって読み出すことができます。

ペリフェラルがセキュアとして設定されると、関連するクロック、リセット、クロックソース選択、および低電力モード時のクロックイネーブル制御ビットも、RCC_AHBxENR、RCC_APBxENR、RCC_CCIPRx、および RCC_BDCR レジスタでセキュアです。

以下の場合にペリフェラルはセキュアになります。

- TZSC によるセキュリティ保護可能なペリフェラルの場合 ([セクション 5.4.3: TrustZone セキュリティ・コントローラ \(TZSC\)](#) を参照)、このペリフェラルに対応する SEC セキュリティビットが GTZC TZSC セキュア設定レジスタでセットされます。
- TrustZone® 対応ペリフェラルの場合、このペリフェラルのセキュリティ機能は、専用ビットを通じて有効化されます。

[表 106](#) に、RCC_SECCFGR レジスタのセキュリティ設定ビットと RCC セキュアビットを示します。

1 つのセキュリティ設定ビットがセットされると、複数の設定ビットおよびステータスビットがセキュアになります。RCC レジスタに含まれるセキュアおよび非セキュアビットは以下のとおりです。

- セキュアビット：読み出しおよび書き込み操作は、セキュアアクセスによってのみ可能です。非セキュア読み出しは 0 を返し、書き込みアクセスは無視されます。不正アクセスイベントは生成されません。

- 非セキュアビット：制約はありません。読出しおよび書込み操作は、セキュアおよび非セキュアアクセスのどちらでも可能です。
- RCC_SECCFGR への非セキュア書込みアクセスは無視され、不正アクセスイベントが生成されます。GTZC TZIC レジスタで RCC 不正アクセス割込みが有効な場合、不正アクセス割込みが生成されます。RCC_SECCFGR は、セキュアまたは非セキュアアクセスによって読み出すことができます。

TrustZone セキュリティが無効なとき (TZEN = 0xC3)、すべてのレジスタは非セキュアです。RCC_SECCFGR 書込みアクセスは無視されます。

表 106. RCC セキュリティ設定一覧

RCC_SECCFGR の 設定ビット	セキュアビット	対応するレジスタ
HSISEC	HSION、HSIKERON、HSIRDY	RCC_CR
	HSICAL[11:0]、HSITRIM[6:0]	RCC_HSICFGR
	HSIRDYIE	RCC_CIER
	HSIRDYIF	RCC_CIFR
	HSIRDYC	RCC_CICR
HSESEC	HSEON、HSERDY、HSEBYP、HSECSSON、HSEEXT	RCC_CR
	HSERDYIE、HSECSSF	RCC_CIER
	HSERDYIF、HSECSSF	RCC_CIFR
	HSERDYC、HSECSSC	RCC_CICR
CSISEC	CSION、CSIKERON、CSISRDY	RCC_CR
	CSICAL[7:0]、CSITRIM[5:0]	RCC_CSICFGR
	CSISRDYIE	RCC_CIER
	CSISRDYIF	RCC_CIFR
	CSISRDYIC	RCC_CICR
LSISEC	LSION、LSIRDY、LSIPREDIV、LSCOSEL、LSCOEN	RCC_BDCR
	LSIRDYIE	RCC_CIER
	LSIRDYIF	RCC_CIFR
	LSIRDYC	RCC_CICR
LSESEC	LSECSSON、LSECSSD、LSEDRV[1:0]、LSEBYP、LSERDY、LSEON、LSEEXT、LSCOSEL、LSCOEN	RCC_BDCR
	LSERDYIE	RCC_CIER
	LSERDYF	RCC_CIFR
	LSERDYC	RCC_CICR
SYSCLKSEC	SW[1:0]、SWS[1:0]、STOPWUCK、STOPKERWUCK、MCO1SEL[3:0]、MCO1PRE[2:0]、MCO2SEL[3:0]、MCO2PRE[2:0]	RCC_CFGR1
	SYSTICKSEL[1:0]	RCC_CCIPR4
	VOS[1:0]	PWR_VOSR
	TIMPRE	RCC_CFGR1
PRESCSEC	HPRE[3:0]、PPRE1[2:0]、PPRE2[2:0]、PPRE3[2:0]	RCC_CFGR2

表 106. RCC セキュリティ設定一覧 (続き)

RCC_SECCFGR の 設定ビット	セキュアビット	対応するレジスタ
PLL1SEC	PLL1SRC[1:0]、PLL1RGE[1:0]、PLL1FRACEN、PLL1M[3:0]、PLL1VCOSEL、PLL1PEN、PLL1QEN、PLL1REN	RCC_PLL1CFGR
	PLL1N[8:0]、PLL1P[6:0]、PLL1Q[6:0]、PLL1R[6:0]	RCC_PLL1DIVR
	PLL1FRACN [12:0]	RCC_PLL1FRACR
	PLL1RDY、PLL1ON	RCC_CR
	PLL1RDYIE	RCC_CIER
	PLL1RDYF	RCC_CIFR
	PLL1RDYC	RCC_CICR
PLL2SEC	PLL2SRC[1:0]、PLL2RGE[1:0]、PLL2FRACEN、PLL2M[3:0]、PLL2PEN、PLL2QEN、PLL2REN、PLL2VCOSEL	RCC_PLL2CFGR
	PLL2N[8:0]、PLL2P[6:0]、PLL2Q[6:0]、PLL2R[6:0]	RCC_PLL2DIVR
	PLL2FRACN [12:0]	RCC_PLL2FRACR
	PLL2RDY、PLL2ON	RCC_CR
	PLL2RDYIE	RCC_CIER
	PLL2RDYF	RCC_CIFR
	PLL2RDYC	RCC_CICR
PLL3SEC	PLL3SRC[1:0]、PLL3RGE[1:0]、PLL3FRACEN、PLL3M[3:0]、PLL3PEN、PLL3QEN、PLL3REN、PLL3VCOSEL	RCC_PLL3CFGR
	PLL3N[8:0]、PLL3P[6:0]、PLL3Q[6:0]、PLL3R[6:0]	RCC_PLL3DIVR
	PLL3FRACN [12:0]	RCC_PLL3FRACR
	PLL3RDY、PLL3ON	RCC_CR
	PLL3RDYIE	RCC_CIER
	PLL3RDYF	RCC_CIFR
	PLL3RDYC	RCC_CICR
HSI48SEC ⁽¹⁾	HSI48ON、HSI48RDY	RCC_CR
	HSI48CAL [9:0]	RCC_CRRCR
	HSI48RDYIE	RCC_CIER
	HSI48RDYF	RCC_CIFR
	HSI48RDYC	RCC_CICR
IPKERSECCFG	CKERPSEL[1:0]	RCC_CCIPR5
RMVFSEC	RMVF	RCC_CSR

1. HSI48 の TRIM フィールドは、CRS ペリフェラルにあります。GTZC1 TZSC セキュア設定レジスタ 1 の CRSSEC ビットを使用して、セキュリティ保護してください。

11.5.2 RCC 特権保護モード

デフォルトでは、リセット後、すべての RCC レジスタは、特権アクセスと非特権アクセスのどちらでも読み書き可能ですが、[RCC 特権設定レジスタ \(RCC_PRIVCFGR\)](#) への書き込みは特権アクセスでのみ可能です。RCC_PRIVCFGR は、セキュアおよび非セキュア、特権および非特権アクセスによって読み出すことができます。

RCC_PRIVCFGR の SPRIV ビットは、セキュア特権アクセスでのみ書き込むことができます。このビットは、すべての RCC セキュア機能の特権アクセスを設定します (セキュア機能は、[RCC セキュア設定レジスタ \(RCC_SECCFGR\)](#) によって、またはセキュリティ保護可能なペリフェラルの場合は GTZC によって、または TrustZone 対応ペリフェラルの場合はペリフェラル自体によって定義されます)。

RCC_PRIVCFGR の SPRIV ビットがセットされているとき：

- RCC セキュアビットの書き込みは、RCC_SECCFGR も含め、特権アクセスでのみ可能です。
- RCC セキュアビットは、特権アクセスでのみ読み出すことができますが、RCC_SECCFGR および RCC_PRIVCFGR は、特権または非特権アクセスによって読み出すことができます。
- 特権 RCC ビットまたはレジスタへの非特権アクセスは破棄されます。すなわち、ビットはゼロとして読み出され、これらのビットへの書き込みは無視されます (RAZ/WI)。

RCC_PRIVCFGR の NSPRIV ビットは、セキュアまたは非セキュアの特権アクセスでのみ書き込むことができます。このビットは、すべての RCC 非セキュア機能の特権アクセスを設定します (非セキュア機能は、RCC_SECCFGR によって、またはセキュリティ保護可能なペリフェラルの場合は GTZC によって、または TrustZone 対応ペリフェラルの場合はペリフェラル自体によって定義されます)。

RCC_PRIVCFGR の NSPRIV ビットがセットされているとき：

- RCC 非セキュアビットの書き込みは、特権アクセスでのみ可能です。
- RCC 非セキュアビットは、特権アクセスでのみ読み出すことができますが、RCC_PRIVCFGR は、特権または非特権アクセスによって読み出すことができます。
- 特権 RCC ビットまたはレジスタへの非特権アクセスは破棄されます。すなわち、ビットはゼロとして読み出され、これらのビットへの書き込みは無視されます (RAZ/WI)。

11.6 RCC 低電力モード

- DMA クロックを含む AHB および APB ペリフェラルクロックは、ソフトウェアで無効にできます。
- SLEEP モードでは CPU クロックを停止します。メモリーインタフェースクロック (Flash メモリ、キャッシュ、およびすべての SRAM インタフェース) は、ソフトウェアによって SLEEP モード中に停止できます。AHB - APB ブリッジのクロックは、SLEEP モード中にこれらに接続されたペリフェラルのクロックがすべて無効になった場合に、ハードウェアによって無効化されます。
- STOP モードでは、コアドメインのすべてのクロックが停止し、PLL、HSI、HSI48、CSI、および HSE オシレータが無効になります。ただし、HSI または CSI をオンにして、ウェイクアップ割込みを生成することができます。LSI および LSE は、STOP モードでもアクティブなままです。
- STANDBY モードでは、コアドメインのすべてのクロックが停止し、PLL、HSI、HSI48、CSI、および HSE オシレータが無効になります。

CPU の DEEPSLEEP モードは、DBGMCU_CR レジスタの DBG_STOP または DBG_STANDBY ビットをセットすることによって、デバッグのためにオーバライドすることができます。

STOP モードを終了するとき、システムクロックは、[RCC CPU ドメインクロック設定レジスタ 2 \(RCC_CFGR2\)](#) の STOPWUCK のソフトウェア設定に応じて、HSI または CSI です。HSI の周波数 (レンジおよびユーザトリミング) は、STOP モードに移行する前に設定された周波数です。

システムクロックによって使用されるものに加えて、他の内部オシレータを自動的にウェイクアップできます。これにより、デバイスが RUN モードに復帰したときに、他のオシレータのウェイクアップ時間の間待つ必要がなくなります。これは、RCC_CFGR1 の STOPKERWUCK を使用して行われます。

STANDBY モードを終了したときのシステムクロックは、HSI (32 mHz) です。ユーザトリミングは失われます。

Flash メモリのプログラミング操作が実行中の場合、Flash メモリインタフェースアクセスが終了するまで、STOP または STANDBY モードへの移行は遅延されます。APB ドメインにアクセス中の場合、APB アクセスが終了するまで、STOP または STANDBY モードへの移行は遅延されます。

11.7 RCC 割込み

表 107 に、割込みソースの概要と、その制御方法について示します。

表 107. 割込みソースとその制御

割込み ベクタ	割込み イベント フラグ	説明	イネーブル 制御ビット	割込みのクリア方法	SLEEP モードの終了	STOP、 STANDBY モードの終了
RCC	LSIRDYF	LSI レディ	LSIRDYIE および LSISEC = 0	LSIRDYC を「1」にセット	はい	不可
	LSEIRDYF	LSE レディ	LSEIRDYIE および LSESEC = 0	LSEIRDYC を「1」にセット	はい	不可
	HSIDRYF	HSI レディ	HSIDRYIE および HSISEC = 0	HSIRDYC を「1」にセット	はい	不可
	HSEIRDYF	HSE レディ	HSEIRDYIE および HSESEC = 0	HSEIRDYC を「1」にセット	はい	不可
	CSISRDYF	CSIS レディ	CSISRDYIE および CSISEC = 0	CSISRDYC を 1 にセット	はい	不可
	HSI48RDYF	HSI48 レディ	HSI48RDYIE および HSI48SEC = 0	HSI48RDYC を「1」にセット	はい	不可
	PLL1RDYF	PLL1 レディ	PLL1RDYIE および PLL1SEC = 0	PLL1RDYC を「1」にセット	はい	不可
	PLL2RDYF	PLL2 レディ	PLL2RDYIE および PLL2SEC = 0	PLL2RDYC を「1」にセット	はい	不可
	PLL3RDYF	PLL3 レディ	PLL3RDYIE および PLL3SEC = 0	PLL3DYC を「1」にセット	はい	不可
RCC_S ⁽¹⁾	LSIRDYF	LSI レディ	LSIRDYIE および LSISEC = 1	LSIRDYC を「1」にセット	はい	不可
	LSEIRDYF	LSE レディ	LSEIRDYIE および LSESEC = 1	LSEIRDYC を「1」にセット	はい	不可
	HSIDRYF	HSI レディ	HSIDRYIE および HSISEC = 1	HSIRDYC を「1」にセット	はい	不可
	HSEIRDYF	HSE レディ	HSEIRDYIE および HSESEC = 1	HSEIRDYC を「1」にセット	はい	不可
	CSISRDYF	CSIS レディ	CSISRDYIE および CSISEC = 1	CSISRDYC を 1 にセット	はい	不可
	HSI48RDYF	HSI48 レディ	HSI48RDYIE および HSI48SEC = 1	HSI48RDYC を「1」にセット	はい	不可
	PLL1RDYF	PLL1 レディ	PLL1RDYIE および PLL1SEC = 1	PLL1RDYC を「1」にセット	はい	不可
	PLL2RDYF	PLL2 レディ	PLL2RDYIE および PLL2SEC = 1	PLL2RDYC を「1」にセット	はい	不可
	PLL3RDYF	PLL3 レディ	PLL3RDYIE および PLL3SEC = 1	PLL3RDYC を「1」にセット	はい	不可

表 107. 割込みソースとその制御 (続き)

割込み ベクタ	割込み イベント フラグ	説明	イネーブル 制御ビット	割込みのクリア方法	SLEEP モードの終了	STOP、 STANDBY モードの終了
TAMP	ITAMP3F ⁽²⁾	LSE の CSS 障害	LSECSSON および ITAMP3E ⁽²⁾ および ITAMP3IE ⁽²⁾	CITAMP3F ⁽²⁾ を 1 にセット	はい	はい
NMI	HSECSSF	HSE の CSS 障害	⁽³⁾	HSECSSC を「1」にセット	はい	不可

- 1. RCC セキュア割込みベクタは、TrustZone が有効なときのみ使用されます。
- 2. LSE CSS 障害イベント (LSECSSD) は、TAMP 内部タンパ 3 に接続されます。割込みをこのイベントに関連付けるには、内部タンパ 3 が有効でなければならず、内部タンパ 3 割込みが有効でなければなりません。ITAMP3F、ITAMP3E、ITAMP3IE、および CITAMP3F ビットは、TAMP ペリフェラルにあります。
- 3. セキュリティシステムの機能が有効である場合 (HSECSSON = 「1」)、この割込みをマスクすることはできません。

11.8 RCC レジスタ

11.8.1 RCC クロック制御レジスタ (RCC_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 002B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PLL3 RDY	PLL3 オン	PLL2 RDY	PLL2 オン	PLL1 RDY	PLL1 オン	Res.	Res.	Res.	HSE EXT	HSE CSSON	HSE BYP	HSE RDY	HSE オン
		r	rw	r	rw	r	rw				rw	rs	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	HSI48 RDY	HSI48 ON	Res.	CSI KERON	CSI RDY	CSI オン	Res.	Res.	HSI DIVF	HSIDIV[1:0]		HSI KERON	HSI RDY	HSI オン
		r	rw		rw	r	rw			r	rw	rw	rw	r	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **PLL3RDY** : PLL3 クロックレディフラグ

ハードウェアによってセットされ、PLL3 がロック状態であることを示します。

0 : PLL3 アンロック (リセット後のデフォルト)

1 : PLL3 ロック

ビット 28 **PLL3ON** : PLL3 有効化

PLL3 を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。

0 : PLL3 オフ (リセット後のデフォルト)

1 : PLL3 ON

ビット 27 **PLL2RDY** : PLL2 クロックレディフラグ

ハードウェアによってセットされ、PLL がロック状態であることを示します。

0 : PLL2 アンロック

1 : PLL2 ロック

ビット 26 **PLL2ON** : PLL2 有効化

PLL2 を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。

0 : PLL2 オフ (リセット後のデフォルト)

1 : PLL2 ON

ビット 25 **PLL1RDY** : PLL1 クロックレディフラグ

ハードウェアによってセットされ、PLL1 がロック状態であることを示します。

0 : PLL1 アンロック (リセット後のデフォルト)

1 : PLL1 ロック

ビット 24 **PLL1ON** : PLL1 有効化

PLL1 を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。ハードウェアはこのビットに 0 を書き込むのを防止することに注意してください (PLL1 出力がシステムクロックとして使用されている場合)。

0 : PLL1 オフ (リセット後のデフォルト)

1 : PLL1 ON

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 HSEEXT : バイパスモードでの外部高速クロックタイプ

外部クロックタイプ (アナログまたはデジタル) を選択するために、ソフトウェアによってセット/リセットされます。

デバイスで使用するには HSEON ビットで外部クロックを有効にする必要があります。HSEEXT ビットは、HSE オシレータが無効のときのみ、書込みができます。

0 : HSE はアナログモードです (リセット後のデフォルト)。

1 : HSE はデジタルモードです。

ビット 19 HSECSSON : HSE クロックセキュリティシステム有効化

HSE のクロックセキュリティシステムを有効にするために、ソフトウェアによってセットされます。

このビットはセット専用です (システムリセット、またはシステムが STANDBY モードに移行すると無効になります)。HSECSSON がセットされているとき、HSE がレディになるとクロック検出回路がハードウェアによって有効にされ、オシレータの障害が検出された場合ハードウェアによって無効にされます。

0 : HSE の CSS オフ (クロック検出回路オフ) (リセット後のデフォルト)

1 : HSE の CSS オン (HSE オシレータが安定していればクロック検出回路オン、そうでなければオフ)

ビット 18 HSEBYP : HSE クロックバイパス

外部クロックでオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。デバイスで使用するには HSEON ビットで外部クロックを有効にする必要があります。

HSEBYP ビットは、HSE オシレータが無効のときのみ、書込みができます。

0 : HSE オシレータはバイパスされません (リセット後のデフォルト)。

1 : HSE オシレータはバイパスされ、外部クロックが使用されます。

ビット 17 HSERDY : HSE クロックレディフラグ

HSE オシレータが安定していることを示すために、ハードウェアによってセットされます。

0 : HSE クロックはレディ状態ではありません (リセット後のデフォルト)

1 : HSE クロックはレディ状態です。

ビット 16 HSEON : HSE クロック有効化

ソフトウェアでセット/クリアされます。

STOP または STANDBY モードに入るときに、HSE を停止するためにハードウェアによってクリアされます。

HSE が直接 (ソフトウェアマルチプレクサを介して) システムクロックとして使用されている場合、または

HSE が PLL1 のリファレンスクロックとして選択され、PLL1 が有効なとき (PLL1ON ビットが 1 にセットされている)、このビットをクリアすることはできません。

0 : HSE がオフになります (リセット後のデフォルト)。

1 : HSE がオンになります。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 HSI48RDY : HSI48 クロックレディフラグ

HSI48 オシレータが安定していることを示すために、ハードウェアによってセットされます。

0 : HSI48 クロックはレディ状態ではありません (リセット後のデフォルト)

1 : HSI48 クロックはレディ状態です。

ビット 12 HSI48ON : HSI48 クロック有効化

ソフトウェアによってセットされ、システムが STOP または STANDBY モードに移行するとき、ソフトウェアまたはハードウェアによって

クリアされます。

0 : HSI48 はオフになります (リセット後のデフォルト)。

1 : HSI48 がオンになります。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CSIKERON** : STOP モードでの CSI クロック有効化

一部のペリフェラルのカーネルクロックとしてすばやく使用できるように、STOP モードでも強制的に CSI をオンにするために、ソフトウェアによってセット/リセットされます。このビットは、CSION の値への影響はありません。

0 : CSI への影響はありません (リセット後のデフォルト)。

1 : CSI は強制的にオンになります (STOP モードの場合も含む)。

ビット 9 **CSIRDY** : CSI クロックレディフラグ

CSI オシレータが安定していることを示すために、ハードウェアによってセットされます。このビットは、RC が CSION によって有効化された場合にのみ有効になります (CSI が CSIKERON またはペリフェラルリクエストで有効化された場合は有効になりません)。

0 : CSI クロックはレディ状態ではありません (リセット後のデフォルト)

1 : CSI クロックはレディ状態です。

ビット 8 **CSION** : CSI クロック有効化

システムおよび/またはペリフェラルの CSI クロックを有効化/無効化するために、ソフトウェアでセット/リセットされます。

システムが STOP モードを終了したときに、強制的に CSI をオンにするためにハードウェアによってセットされます (STOPWUCK = 1 または STOPKERWUCK = 1 の場合)。

CSI がシステムクロックとして直接 (ソフトウェアマルチプレクサを介して) 使用されている場合、または PLL1 が有効化された状態で CSI が PLL1 のリファレンスクロックとして選択されている場合 (PLL1ON ビットを 1 にセット) は、このビットをクリアできません。

0 : CSI がオフになります (リセット後のデフォルト)。

1 : CSI がオンになります。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **HSIDIVF** : HSI 分周器フラグ

ハードウェアでセット/リセットされます。

HSIDIVF への書込み操作は、周波数にすぐに影響を及ぼすことはないため、このフラグは

HSI 分周器の現在のステータスを示します。HSIDIVF は、HSIDIV 値が変更されるとすぐに 0 になり、出力周波数が HSIDIV にプログラムされた値と一致する場合に 1 に戻されます。

0 : 新しい分周比は、hsi_ck、hsi_ker_ck にまだ伝播されていません (リセット後のデフォルト)。

1 : hsi_ck、hsi_ker_ck クロック周波数は、新しい HSIDIV 値を反映しています (クロック設定が完了したときのデフォルトのレジスタ値)。

ビット 4:3 **HSIDIV[1:0]** : HSI クロック分周器

ソフトウェアでセット/クリアされます。

これらのビットにより、必要な HSI クロック周波数を設定する分周比を選択できます。最低 1 つの有効な PLL (PLLxON ビットを 1 にセット) で HSI をリファレンスクロックとして選択している場合は、HSIDIV を変更することはできません。この場合、新しい HSIDIV 値は無視されます。

00 : 分周比 1、hsi_ck、hsi_ker_ck = 64 mHz

01 : 分周比 2、hsi_ck、hsi_ker_ck = 32 mHz (リセット後のデフォルト)

10 : 分周比 4、hsi_ck、hsi_ker_ck = 16 mHz

11 : 分周比 8、hsi_ck、hsi_ker_ck = 8 mHz

ビット 2 **HSIKERON** : STOP モードでの HSI クロック有効化

ペリフェラルのカーネルクロックとしてすばやく使用できるように、STOP モードでも強制的に HSI をオンにするために、ソフトウェアによってセット/リセットされます。このビットは、HSION の値への影響はありません。

0 : HSI への影響はありません (リセット後のデフォルト)。

1 : HSI は強制的にオンになります (STOP モードの場合も含む)。

ビット 1 **HSIRDY** : HSI クロックレディフラグ

HSI オシレータが安定していることを示すために、ハードウェアによってセットされます。

0 : HSI クロックはレディ状態ではありません (リセット後のデフォルト)

1 : HSI クロックはレディ状態です。

ビット 0 **HSION** : HSI クロック有効化

ソフトウェアでセット/クリアされます。

製品が STOP モードを終了したときに、強制的に HSI をオンにするためにハードウェアによってセットされます (STOPWUCK = 1 または STOPKERWUCK = 1 の場合)。

製品が STANDBY モードを終了するとき、またはシステムクロックソースとして使用されている HSE に障害が発生した場合、HSI を強制的にオンにするために、ハードウェアによってセットされます。

HSI がシステムクロックとして直接 (ソフトウェアマルチプレクサを介して) 使用されている場合、または PLL1 が有効化された状態で HSI が PLL1 のリファレンスクロックとして選択されている場合 (PLL1ON ビットを 1 にセット) は、このビットをクリアできません。

0 : HSI がオフになります。

1 : HSI がオンになります (リセット後のデフォルト)。

11.8.2 RCC HSI 較正レジスタ (RCC_HSICFGR)

アドレスオフセット : 0x010

リセット値 : 0x0040 : 0XXX

リセット値は、Flash メモリオプションバイトの設定に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSITRIM[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSICAL[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **HSITRIM[6:0]** : HSI クロックトリミング

較正を調整するために、ソフトウェアでセットされます。

リセットフェーズ中にロードされたエンジニアリングオプションバイト (FLASH_HSI_OPT) に HSITRIM フィールドが追加されて、較正のトリミング値を形成します。

$HSICAL = HSITRIM + FLASH_HSI_OPT$ 。

HSITRIM の変更後、新しい HSITRIM 値が更新されるまで、1 システムクロックサイクルかかります。

注 : このフィールドのリセット値は 0x40 です。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **HSICAL[11:0]** : HSI クロック較正

ハードウェアによって、システムリセット nreset 中にオプションバイトをロードすることによってセットされます。ソフトウェアによって、トリミングビット HSITRIM を使用して調整されます。

このフィールドは、エンジニアリングオプションバイト較正值と HSITRIM ビット値の合計を表します。

11.8.3 RCC クロックリカバリ RC レジスタ (RCC_CRRCR)

アドレスオフセット : 0x014

リセット値 : 0x0000 : 0XXX

リセット値は、Flash オプションバイト設定に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL [9:0]									
						r	r	r	r	r	r	r	r	r	r

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **HSI48CAL[9:0]** : 内部 RC 48 MHz クロック較正

ハードウェアによって、システムリセット NRESET 中にオプションバイトをロードすることによってセットされます。読み出し専用。

11.8.4 RCC CSI 較正レジスタ (RCC_CSICFGR)

アドレスオフセット : 0x018

リセット値 : 0x0020 : 0XXX

リセット値は、Flash オプションバイト設定に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSITRIM[5:0]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSICAL[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:16 **CSITRIM[5:0]** : CSI クロックトリミング

較正を調整するために、ソフトウェアでセットされます。

リセットフェーズ中にロードされたエンジニアリングオプションバイト (FLASH_CSI_OPT) に CSITRIM フィールドが追加されて、較正のトリミング値を形成します。

CSICAL = CSITRIM + FLASH_CSI_OPT。

注 : このフィールドのリセット値は 0x20 です。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **CSICAL[7:0]** : CSI クロック較正

ハードウェアによって、システムリセット NRESET 中にオプションバイトをロードすることによってセットされます。ソフトウェアによって、トリミングビット CSITRIM を使用して調整されます。

このフィールドは、エンジニアリングオプションバイトの較正値と CSITRIM ビット値の合計を表します。

11.8.5 RCC クロック設定レジスタ 1 (RCC_CFGR1)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

アクセス : 0 ≤ ウェイトステート ≤ 2、ワード、ハーフワード、およびバイトアクセス

クロックソースの切替え中にアクセスが発生した場合に限り、1 または 2 個のウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2SEL [2:0]			MCO2PRE [3:0]				MCO1SEL [2:0]			MCO1PRE [3:0]				Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM PRE	Res.	RTCPRE[5:0]						STOP KER WUCK	STOP WUCK	Res.	SWS[1:0]		Res.	SW[1:0]	
rw		rw	rw	rw	rw	rw	rw	rw	rw		l	l		rw	rw

ビット 31:29 **MCO2SEL[2:0]** : マイクロコントローラクロック出力 2

ソフトウェアでセット／クリアされます。クロックソースの選択によって、MCO2 でグリッチが生成することがあります。

これらのビットは、リセット後に外部オシレータおよび PLL を有効にする前に設定することを強く推奨します。

000 : システムクロックの選択 (**sys_ck**) (リセット後のデフォルト)

001 : PLL2 オシレータクロックの選択 (**pll2_p_ck**)

010 : HSE クロックの選択 (**hse_ck**)

011 : PLL1 クロックの選択 (**pll1_p_ck**)

100 : CSI クロックの選択 (**csi_ck**)

101 : LSI クロックの選択 (**lsi_ck**)

その他 : 予約済み

ビット 28:25 **MCO2PRE[3:0]** : MCO2 プリスケアラ

MCO2 のプリスケアラを設定するために、ソフトウェアによってセット／クリアされます。このプリスケアラの変更によって、MCO2 でグリッチが生成することがあります。このプリスケアラは、リセット後に外部オシレータおよび PLL を有効にする前に変更することを強く推奨します。

0000 : プリスケアラは無効 (リセット後のデフォルト)

0001 : 分周比 1 (バイパス)

0010 : 分周比 2

0011 : 分周比 3

0100 : 分周比 4

.....

1111 : 分周比 15

ビット 24:22 **MCO1SEL[2:0]** : マイクロコントローラクロック出力 1

ソフトウェアでセット/クリアされます。クロックソースの選択によって、MCO1 でグリッチが生成することがあります。

これらのビットは、リセット後に外部オシレータおよび PLL を有効にする前に設定することを強く推奨します。

000 : HSI クロックの選択 (**hsi_ck**) (リセット後のデフォルト)

001 : LSE オシレータクロックの選択 (**lse_ck**)

010 : HSE クロックの選択 (**hse_ck**)

011 : PLL1 クロックの選択 (**pll1_q_ck**)

100 : HSI48 クロックの選択 (**hsi48_ck**)

その他 : 予約済み

ビット 21:18 **MCO1PRE[3:0]** : MCO1 プリスケーラ

MCO1 のプリスケーラを設定するために、ソフトウェアによってセット/クリアされます。このプリスケーラの変更によって、MCO1 でグリッチが生成することがあります。このプリスケーラは、リセット後に外部オシレータおよび PLL を有効にする前に変更することを強く推奨します。

0000 : プリスケーラは無効 (リセット後のデフォルト)

0001 : 分周比 1 (バイパス)

0010 : 分周比 2

0011 : 分周比 3

0100 : 分周比 4

.....

1111 : 分周比 15

ビット 17:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TIMPRE** : タイマクロックプリスケーラ選択

このビットは、APB1 ドメインおよび APB2 ドメインに接続されているすべてのタイマのクロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

0 : タイマカーネルクロックは、PPRE1 または PPRE2 が分周比 1 または分周比 2 に対応している場合は **rcc_hclk1** に等しくなり、それ以外の場合は $2 \times F_{\text{rcc_pclk1}}$ または $2 \times F_{\text{rcc_pclk2}}$ に等しくなります (リセット後のデフォルト)。

1 : タイマカーネルクロックは、PPRE1 または PPRE2 が分周比 1、2 または 4 に対応している場合は $2 \times F_{\text{rcc_pclk1}}$ または $2 \times F_{\text{rcc_pclk2}}$ に等しくなり、それ以外の場合は $4 \times F_{\text{rcc_pclk1}}$ または $4 \times F_{\text{rcc_pclk2}}$ に等しくなります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **RTCPRE[5:0]** : RTC クロックのための HSE 分周比

HSE を分周して RTC にクロックを生成するために、ソフトウェアによってセット/クリアされます。

注 : これらのビットは、RTC に供給されるクロックが 1MHz 未満になるように、ソフトウェアで正しく設定する必要があります。これらのビットは、必要に応じて RTC クロックソースが選択される前に設定する必要があります。

000000 : クロックなし (リセット後のデフォルト)

000001 : クロックなし

000010 : HSE/2

000011 : HSE/3

000100 : HSE/4

.....

111110 : HSE/62

111111 : HSE/63

ビット 7 **STOPKERWUCK** : システム STOP からのウェイクアップ後のカーネルクロック選択

システム STOP からカーネルウェイクアップクロックを選択するために、ソフトウェアによってセット／リセットされます。

0 : HSI が、システム STOP からのウェイクアップクロックとして選択されます (リセット後のデフォルト)。

1 : CSI が、システム STOP からのウェイクアップクロックとして選択されます。

ビット 6 **STOPWUCK** : システム STOP からのウェイクアップ後のシステムクロック選択

システム STOP からシステムウェイクアップクロックを選択するために、ソフトウェアによってセット／リセットされます。

選択されたクロックは、HSE のクロックセキュリティシステム (CSS) の非常時クロックにも使用されます。0 : HSI が、システム STOP からのウェイクアップクロックとして選択されます (リセット後のデフォルト)。

1 : CSI が、システム STOP からのウェイクアップクロックとして選択されます。

注意 : CSS が (HSECSSON ビットによって) 有効化され、システムクロックが HSE (SWS = 10) であるとき、または HSE のスイッチ (SW = 10) がリクエストされたときには、STOPWUCK を変更しないでください。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:3 **SWS[1:0]** : システムクロックスイッチステータス

どのクロックソースがシステムクロックとして使用されているかを示すために、ハードウェアによってセット／リセットされます。000 : HSI がシステムクロックとして使用されます (**hsi_ck**) (リセット後のデフォルト)。

01 : CSI がシステムクロックとして使用されます (**csi_ck**)。

10 : HSE がシステムクロックとして使用されます (**hse_ck**)。

11 : PLL1 がシステムクロックとして使用されます (**pll1_p_ck**)。

その他 : 予約済み

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **SW[1:0]** : システムクロックおよびトレースクロックスイッチ

システムクロックおよびトレースクロックソース (**sys_ck**) を選択するために、ソフトウェアによってセットおよびリセットされます。

以下を目的として、ハードウェアによってセットされます。

- システム STOP モード終了時に、HSI または CSI を強制的に選択するため (STOPWUCK 選択による)

- 直接的または間接的にシステムクロックとして使用されているとき、HSE に障害が発生した場合に HSI を強制的に選択するため

00 : HSI がシステムクロックとして選択されます (**hsi_ck**) (リセット後のデフォルト)。

01 : CSI がシステムクロックとして選択されます (**csi_ck**)。

10 : HSE がシステムクロックとして選択されます (**hse_ck**)。

11 : PLL1 がシステムクロックとして選択されます (**sys_ck** について **pll1_p_ck**)。

その他 : 予約済み

11.8.6 RCC CPU ドメインクロック設定レジスタ 2 (RCC_CFGR2)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

クロックソースの切替え中にアクセスが発生した場合に限り、1 または 2 個のウェイトステートが挿入されます。

APB または AHB プリスケアラ値の更新中にアクセスが発生した場合、0 ~ 15 個のウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB3 DIS	APB2 DIS	APB1 DIS	AHB4 DIS	Res.	AHB2 DIS	AHB1 DIS
									rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PPRE3 [2:0]			Res.	PPRE2 [2:0]			Res.	PPRE1 [2:0]			HPRE[3:0]			
	rw	rw	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **APB3DIS** : APB3 クロックディスエーブル値。ソフトウェアによってセットおよびクリアされます。

いずれの APB3 ペリフェラルも使用されておらず、それらのクロックが RCC_APB3ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、すべての APB3 ペリフェラルクロックがオフになります。

0 : APB3 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : APB3 クロック無効

ビット 21 **APB2DIS** : APB2 クロックディスエーブル値

いずれの APB2 ペリフェラルも使用されておらず、それらのクロックが RCC_APB2ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、すべての APB2 ペリフェラルクロックがオフになります。

0 : APB2 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : APB2 クロック無効

ビット 20 **APB1DIS** : APB1 クロックディスエーブル値

いずれの APB1 ペリフェラルも使用されておらず (IWDG を除く)、それらのクロックが RCC_APB1ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、IWDG を除くすべての APB1 ペリフェラルクロックがオフになります。

0 : APB1 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : APB1 クロック無効

ビット 19 **AHB4DIS** : AHB4 クロックは無効です。

RCC_AHB4ENR からの、いずれの AHB4 ペリフェラルも使用されておらず、それらのクロックが RCC_AHB4ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、RCC_AHB4ENR からのすべての AHB4 ペリフェラルクロックがオフになります。

0 : AHB4 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : AHB4 クロック無効

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **AHB2DIS** : AHB2 クロックは無効です。

RCC_AHB2ENR からの、いずれの AHB2 ペリフェラルも使用されておらず、それらのクロックが RCC_AHB2ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、SRAM2 および SRAM3 を除くすべての AHB2 ペリフェラルクロックがオフになります。

0 : AHB2 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : AHB2 クロック無効

ビット 16 **AHB1DIS** : AHB1 クロックは無効です。

RCC_AHB1ENR からの、いずれの AHB1 ペリフェラルも使用されておらず、それらのクロックが RCC_AHB1ENR で無効になっているとき、このビットをセットすることで、消費電力をさらに削減できます。このビットがセットされると、FLASH、BKPSRAM、ICACHE、DCACHE1、および SRAM1 を除くすべての AHB1 ペリフェラルクロックがオフになります。

0 : AHB1 クロックは有効であり、専用クロックイネーブル制御ビットに従ってペリフェラルに配分されます。

1 : AHB1 クロック無効

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **PPRE3[2:0]** : APB 低速プリスケアラ (APB3)

APB 低速クロックの分周比を制御するために、ソフトウェアによってセットおよびリセットされます。クロックは、PPRE3 に書き込んでから 1~16 APB の間の新しい分周比で分周されます。

0xx: rcc_pclk3 = rcc_hclk1

100: rcc_pclk3 = rcc_hclk1 / 2

101: rcc_pclk3 = rcc_hclk1 / 4

110: rcc_pclk3 = rcc_hclk1 / 8

111: rcc_pclk3 = rcc_hclk1 / 16

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **PPRE2[2:0]** : APB ハイスピードプリスケアラ (APB2)

APB 高速クロックの分周比を制御するために、ソフトウェアによってセットおよびリセットされます。クロックは、PPRE2 に書き込んでから 1~16 APB の間の新しい分周比で分周されます。

0xx: rcc_pclk2 = rcc_hclk1

100: rcc_pclk2 = rcc_hclk1 / 2

101: rcc_pclk2 = rcc_hclk1 / 4

110: rcc_pclk2 = rcc_hclk1 / 8

111: rcc_pclk2 = rcc_hclk1 / 16

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **PPRE1[2:0]** : APB 低速プリスケアラ (APB1)

rcc_pclk1 の分周比を制御するために、ソフトウェアによってセットおよびリセットされます。

クロックは、PPRE に書き込んでから、rcc_hclk の 1 ~ 16 サイクル後に新しいプリスケアラ分周比で分周されます。

0xx: rcc_pclk1 = rcc_hclk1 (リセット後のデフォルト)

100: rcc_pclk1 = rcc_hclk1 / 2

101: rcc_pclk1 = rcc_hclk1 / 4

110: rcc_pclk1 = rcc_hclk1 / 8

111: rcc_pclk1 = rcc_hclk1 / 16

ビット 3:0 **HPRE[3:0]** : AHB プリスケアラ

rcc_hclk の分周比を制御するために、ソフトウェアによってセットおよびリセットされます。この分周比を変更すると、すべてのバスマトリックスクロックの周波数に影響します

0xxx: rcc_hclk = sys_ck (リセット後のデフォルト)

1000: rcc_hclk = sys_ck / 2

1001: rcc_hclk = sys_ck / 4

1010: rcc_hclk = sys_ck / 8

1011: rcc_hclk = sys_ck / 16

1100: rcc_hclk = sys_ck / 64

1101: rcc_hclk = sys_ck / 128

1110: rcc_hclk = sys_ck / 256

1111: rcc_hclk = sys_ck / 512

注意： 電圧スケーリングを使用するときには注意が必要です。新しい分周比の伝搬遅延により、プリスケアラ分周比の変更後、V_{CORE} 電圧を下げる前に、このレジスタを読み出して、新しいプリスケアラ値が正しく反映されていることを確認する必要があります。

クロックソース周波数と電圧範囲に応じて、ソフトウェアアプリケーションは、HPRE に正しい値をプログラムして、システム周波数が最大周波数を超えないようにする必要があります。

11.8.7 RCC PLL1 クロックソース選択レジスタ (RCC_PLL1CFGR)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL1REN	PLL1QEN	PLL1PEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PLL1M [5:0]						Res.	Res.	PLL1VCOSEL	PLL1FRACEN	PLL1RGE [1:0]		PLL1SRC [1:0]	
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **PLL1REN** : PLL1 の DIVR 分周器出力有効化

PLL1 の pll1_r_ck 出力を有効化するために、ソフトウェアによってセット／リセットされます。

電力を削減するために、pll1_r_ck が使用されないときには、DIVR1EN および DIVR1 ビットを 0 にセットする必要があります。このビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。

0 : pll1_r_ck 出力は無効です (リセット後のデフォルト)。

1 : pll1_r_ck 出力は有効です。

ビット 17 **PLL1QEN** : PLL1 の DIVQ 分周器出力有効化

PLL1 の pll1_q_ck 出力を有効化するために、ソフトウェアによってセット／リセットされます。

電力を削減するために、PLL1 の pll1_q_ck 出力が使用されないときには、pll1_q_ck を無効にする必要があります。

このビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。

0 : pll1_q_ck 出力は無効です (リセット後のデフォルト)。

1 : pll1_q_ck 出力は有効です。

ビット 16 **PLL1PEN** : PLL1 の DIVP 分周器出力有効化

PLL1 の pll1_p_ck 出力を有効化するために、ソフトウェアによってセット／リセットされます。
このビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。
電力を削減するために、PLL1 の pll1_p_ck 出力が使用されないときには、pll1_p_ck を無効にする必要があります。
0 : pll1_p_ck 出力は無効です (リセット後のデフォルト)。
1 : pll1_p_ck 出力は有効です。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **PLL1M[5:0]** : PLL1 のプリスケアラ

PLL1 のプリスケアラを設定するために、ソフトウェアによってセット／クリアされます。
PLL1 が有効なとき (PLL1ON = 1 または PLL1RDY = 1)、ハードウェアはこのプリスケアラを変更できません。
電力を削減するために、PLL1 が使用されないときには、PLL1M の値を 0 にセットする必要があります。
000000 : プリスケアラは無効 (リセット後のデフォルト)
000001 : 分周比 1 (バイパス)
000010 : 分周比 2
000011 : 分周比 3
.....
100000 : 分周比 32
.....
111111 : 分周比 63

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PLL1VCOSEL** : PLL1 VCO 選択

PLL1 で使用する適切な VCO 周波数範囲を選択するために、ソフトウェアによってセット／リセットされます。このビットは、PLL1 を有効にする前に書き込む必要があります。
0 : 広帯域 VCO レンジ : 192 から 836 MHz (リセット後のデフォルト)
1 : 中帯域 VCO レンジ : 150 から 420 MHz

ビット 4 **PLL1FRACEN** : PLL1 分数ラッチイネーブル

FRACN1 の内容をシグマデルタモジュレータにラッチするために、ソフトウェアによってセット／リセットされます。
シグマデルタモジュレータに FRACN1 の値をラッチするには、PLL1FRACEN を 0 にセットしてから 1 にセットする必要があります。0 から 1 への遷移によって、FRACN1 の内容がモジュレータに転送されます。

ビット 3:2 **PLL1RGE[1:0]** : PLL1 入力周波数範囲

PLL1 で使用する適切なリファレンス周波数範囲を選択するために、ソフトウェアによってセット／リセットされます。このビットは、PLL1 を有効にする前に書き込む必要があります。
00 : PLL1 入力 (ref1_ck) クロック範囲の周波数は、1 から 2MHz です (リセット後のデフォルト)。
01 : PLL1 入力 (ref1_ck) クロック範囲の周波数は、2 から 4MHz です。
10 : PLL1 入力 (ref1_ck) クロック範囲の周波数は、4 から 8MHz です。
11 : PLL1 入力 (ref1_ck) クロック範囲の周波数は、8 から 16MHz です。

ビット 1:0 **PLL1SRC[1:0]** : PLL1M および PLL のクロックソース選択

PLL クロックソースを選択するために、ソフトウェアによってセット／リセットされます。これらのビットに書き込めるのは、PLL がすべて無効なときだけです。
電力を削減するために、PLL が使用されないときには、PLL1SRC の値を '00' にセットする必要があります。
00 : クロックは PLL1M 分周器および PLL に送信されません (リセット後のデフォルト)。
01 : HSI が PLL クロックとして選択されます (hsi_ck)。
10 : CSI が PLL クロックとして選択されます (csi_ck)。
11 : HSE が PLL クロックとして選択されます (hse_ck)。

11.8.8 RCC PLL2 クロックソース選択レジスタ (RCC_PLL2CFGR)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL2REN	PLL2QEN	PLL2PEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PLL2M [5:0]						Res.	Res.	PLL2VCOSEL	PLL2FRACEN	PLL2RGE [1:0]		PLL2SRC [1:0]	
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **PLL2REN** : PLL2 の DIVR 分周器出力有効化

PLL2 の pll2_r_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、pll2_r_ck が使用されないときには、DIVR2EN および DIVR2 ビットを 0 にセットする必要があります。

0 : pll2_r_ck 出力は無効です (リセット後のデフォルト)。

1 : pll2_r_ck 出力は有効です。

ビット 17 **PLL2QEN** : PLL2 の DIVQ 分周器出力有効化

PLL2 の pll2_q_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、PLL2 の pll2_q_ck 出力が使用されないときには、pll2_q_ck を無効にする必要があります。

0 : pll2_q_ck 出力は無効です (リセット後のデフォルト)。

1 : pll2_q_ck 出力は有効です。

ビット 16 **PLL2PEN** : PLL2 の DIVP 分周器出力有効化

PLL2 の pll2_p_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、PLL2 の pll2_p_ck 出力が使用されないときには、pll2_p_ck を無効にする必要があります。

0 : pll2_p_ck 出力は無効です (リセット後のデフォルト)。

1 : pll2_p_ck 出力は有効です。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **PLL2M[5:0]** : PLL2 のプリスケアラ

PLL2 のプリスケアラを設定するために、ソフトウェアによってセット/クリアされます。

PLL2 が有効なとき (PLL2ON = 1 または PLL2RDY = 1)、ハードウェアはこのプリスケアラを変更できません。

電力を削減するために、PLL2 が使用されないときには、PLL2M の値を 0 にセットする必要があります。

000000 : プリスケアラは無効 (リセット後のデフォルト)

000001 : 分周比 1 (バイパス)

000010 : 分周比 2

000011 : 分周比 3

.....

100000 : 分周比 32

.....

111111 : 分周比 63

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PLL2VCOSEL** : PLL2 VCO 選択

PLL2 で使用する適切な VCO 周波数範囲を選択するために、ソフトウェアによってセット/リセットされます。

このビットは、PLL2 を有効にする前に書き込む必要があります。

0 : 広帯域 VCO レンジ 192 から 836 mHz (リセット後のデフォルト)

1 : 中帯域 VCO レンジ 150 から 420 mHz

ビット 4 **PLL2FRACEN** : PLL2 分数ラッチイネーブル

PLL2 の pll2_p_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、PLL2 の pll2_p_ck 出力が使用されないときには、pll2_p_ck を無効にする必要があります。

0 : pll2_p_ck 出力は無効です (リセット後のデフォルト)。

1 : pll2_p_ck 出力は有効です。

ビット 3:2 **PLL2RGE[1:0]** : PLL2 入力周波数範囲

PLL2 で使用する適切なリファレンス周波数範囲を選択するために、ソフトウェアによってセット/リセットされます。これらのビットは、PLL2 を有効にする前に書き込む必要があります。

00 : PLL2 入力 (ref2_ck) クロック範囲の周波数は、1 から 2 mHz です (リセット後のデフォルト)。

01 : PLL2 入力 (ref2_ck) クロック範囲の周波数は、2 から 4MHz です。

10 : PLL2 入力 (ref2_ck) クロック範囲の周波数は、4 から 8MHz です。

11 : PLL2 入力 (ref2_ck) クロック範囲の周波数は、8 から 16MHz です。

ビット 1:0 **PLL2SRC[1:0]** : PLL2M および PLL のクロックソース選択

PLL クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

これらのビットに書き込めるのは、PLL がすべて無効なときだけです。

電力を削減するために、PLL が使用されないときには、PLL2SRC の値を '00' にセットする必要があります。

00 : クロックは PLL2M 分周器および PLL に送信されません (リセット後のデフォルト)

01 : HSI が PLL クロックとして選択されます (hsi_ck)。

10 : CSI が PLL クロックとして選択されます (csi_ck)。

11 : HSE が PLL クロックとして選択されます (hse_ck)。

11.8.9 RCC PLL3 クロックソース選択レジスタ (RCC_PLL3CFGR)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3R EN	PLL3Q EN	PLL3P EN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PLL3M [5:0]						Res.	Res.	PLL3 VCOSEL	PLL3 FRACEN	PLL3RGE [1:0]		PLL3SRC [1:0]	
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **PLL3REN** : PLL3 の DIVR 分周器出力有効化

PLL3 の pll3_r_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、pll3_r_ck が使用されないときには、DIVR2EN および DIVR2 ビットを 0 にセットする必要があります。

0 : pll3_r_ck 出力は無効です (リセット後のデフォルト)。

1 : pll3_r_ck 出力は有効です。

ビット 17 **PLL3QEN** : PLL3 の DIVQ 分周器出力有効化

PLL3 の pll3_q_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、PLL3 の pll3_q_ck 出力が使用されないときには、pll3_q_ck を無効にする必要があります。

0 : pll3_q_ck 出力は無効です (リセット後のデフォルト)。

1 : pll3_q_ck 出力は有効です。

ビット 16 **PLL3PEN** : PLL3 の DIVP 分周器出力有効化

PLL3 の pll3_p_ck 出力を有効化するために、ソフトウェアによってセット/リセットされます。

電力を削減するために、PLL3 の pll3_p_ck 出力が使用されないときには、pll3_p_ck を無効にする必要があります。

0 : pll3_p_ck 出力は無効です (リセット後のデフォルト)。

1 : pll3_p_ck 出力は有効です。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **PLL3M[5:0]** : PLL3 のプリスケアラ

PLL3 のプリスケアラを設定するために、ソフトウェアによってセット/クリアされます。

PLL3 が有効なとき (PLL3ON = 1 または PLL3RDY = 1)、ハードウェアはこのプリスケアラを変更できません。

電力を削減するために、PLL3 が使用されないときには、PLL3M の値を 0 にセットする必要があります。

000000 : プリスケアラは無効 (リセット後のデフォルト)

000001 : 分周比 1 (バイパス)

000010 : 分周比 2

000011 : 分周比 3

.....

100000 : 分周比 32

.....

111111 : 分周比 63

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PLL3VCOSEL** : PLL3 VCO 選択

PLL3 で使用する適切な VCO 周波数範囲を選択するために、ソフトウェアによってセット/リセットされます。

このビットは、PLL3 を有効にする前に書き込む必要があります。

0 : 広帯域 VCO レンジ 192 から 836 mHz (リセット後のデフォルト)

1 : 中帯域 VCO レンジ 150 から 420 mHz

ビット 4 **PLL3FRACEN** : PLL3 分数ラッチイネーブル

FRACN3 の内容をシグマデルタモジュレータにラッチするために、ソフトウェアによってセット/リセットされます

シグマデルタモジュレータに FRACN3 の値をラッチするには、PLL3FRACEN を 0 にセットしてから 1 にセットする必要があります。0 から 1 への遷移によって、FRACN3 の内容がモジュレータに転送されます。

ビット 3:2 **PLL3RGE[1:0]** : PLL3 入力周波数範囲

PLL3 で使用する適切なリファレンス周波数範囲を選択するために、ソフトウェアによってセット／リセットされます。

このビットは、PLL3 を有効にする前に書き込む必要があります。

00 : PLL3 入力 (ref3_ck) クロック範囲の周波数は、1 から 2MHz です (リセット後のデフォルト)。

01 : PLL3 入力 (ref3_ck) クロック範囲の周波数は、2 から 4MHz です。

10 : PLL3 入力 (ref3_ck) クロック範囲の周波数は、4 から 8MHz です。

11 : PLL3 入力 (ref3_ck) クロック範囲の周波数は、8 から 16MHz です。

ビット 1:0 **PLL3SRC[1:0]** : PLL3M および PLL のクロックソース選択

PLL クロックソースを選択するために、ソフトウェアによってセット／リセットされます。これらのビットは

すべての PLL が無効なときのみ、書き込むことができます。

電力を削減するために、PLL が使用されないときには、PLL3SRC の値を '00' にセットする必要があります。

00 : クロックは PLL3M 分周器および PLL に送信されません (リセット後のデフォルト)

01 : HSI が PLL クロックとして選択されます (hsi_ck)。

10 : CSI が PLL クロックとして選択されます (csi_ck)。

11 : HSE が PLL クロックとして選択されます (hse_ck)。

11.8.10 **RCC PLL1 分周器レジスタ (RCC_PLL1DIVR)**

アドレスオフセット : 0x034

リセット値 : 0x0101 0280

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLL1R [6:0]							Res.	PLL1Q [6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL1P [6:0]								PLL1N [8:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **PLL1R[6:0]** : PLL1 DIVR の分周比

pll1_r_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。これらのビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。

0000000: pll1_r_ck = vco1_ck / 1

0000001: pll1_r_ck = vco1_ck / 2 (リセット後のデフォルト)

0000010: pll1_r_ck = vco1_ck / 3

0000011: pll1_r_ck = vco1_ck / 4

.....

1111111: pll1_r_ck = vco1_ck / 128

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **PLL1Q[6:0]** : PLL1 DIVQ の分周比

pll1_q_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。

0000000: pll1_q_ck = vco1_ck

0000001: pll1_q_ck = vco1_ck / 2 (リセット後のデフォルト)

0000010: pll1_q_ck = vco1_ck / 3

0000011: pll1_q_ck = vco1_ck / 4

.....

1111111: pll1_q_ck = vco1_ck / 128

ビット 15:9 **PLL1P[6:0]** : PLL1 DIVP の分周比

pll1_p_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL1 が無効なときだけです (PLL1ON = 0 および PLL1RDY = 0)。

奇数の分周比は使用できません。

0000000 : 設定禁止

0000001: pll1_p_ck = vco1_ck / 2 (リセット後のデフォルト)

0000010 : 設定禁止

0000011: pll1_p_ck = vco1_ck / 4

.....

1111111: pll1_p_ck = vco1_ck / 128

ビット 8:0 **PLL1N[8:0]** : PLL1VCO の通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット／リセットされます。

これらのビットは、PLL が無効なとき (PLL1ON = 0 および PLL1RDY = 0) のみ、書き込むことができます。

0x003 : PLL1N = 4

0x004 : PLL1N = 5

0x005 : PLL1N = 6

.....

0x080 : PLL1N = 129 (リセット後のデフォルト)

.....

0x1FF : PLL1N = 512

その他 : 予約済み

11.8.11 RCC PLL1 分数分周器レジスタ (RCC_PLL1FRACR)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードおよびハーフワードアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL1FRACN [12:0]													Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW			

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:3 **PLL1FRACN[12:0]** : PLL1 VCO の通倍数の小数部

VCO の通倍数の小数部を制御するために、ソフトウェアによってセット／リセットされます。これらのビットはいつでも書き込めるため、PLL1 VCO を動的に微調整することができます。

注意 : ソフトウェアは、これらのビットを正しくセットして、VCO 出力周波数が次の有効な周波数範囲になるようにする必要があります。

*PLL1VCOSEL = 0 の場合、128 から 560MHz

* PLL1VCOSEL = 1 の場合、150 から 420 mHz

VCO 出力周波数 = $F_{\text{ref1_ck}} \times (\text{PLL1N} + (\text{PLL1FRACN} / 2^{13}))$ で、次の要件を満たすものとします。

* PLL1N は 8 から 420 の範囲

* PLL1FRACN は 0 から $2^{13} - 1$ の範囲

*入力周波数 $F_{\text{ref1_ck}}$ は 1 から 16 mHz の範囲

PLL が有効な場合でも、PLL1FRACN の値をオンザフライで変更するには、アプリケーションは次の手順に従う必要があります。

*PLL1FRACEN ビットを 0 にセットします。

* PLL1FRACN に新しい小数値を書き込みます。

* PLL1FRACEN ビットを 1 にセットします。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

11.8.12 RCC PLL1 分周器レジスタ (RCC_PLL2DIVR)

アドレスオフセット : 0x03C

リセット値 : 0x0101 0280

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLL2R [6:0]							Res.	PLL2Q [6:0]						
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL2P [6:0]								PLL2N [8:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **PLL2R[6:0]** : PLL2 DIVR の分周比

pll2_r_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。

これらのビットに書き込めるのは、PLL1 が無効なときだけです (PLL2ON = 0 および PLL2RDY = 0)。

0000000: pll2_r_ck = vco2_ck

0000001: pll2_r_ck = vco2_ck / 2 (リセット後のデフォルト)

0000010: pll2_r_ck = vco2_ck / 3

0000011: pll2_r_ck = vco2_ck / 4

.....

1111111: pll2_r_ck = vco2_ck / 128

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **PLL2Q[6:0]** : PLL2 DIVQ の分周比

pll2_q_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL2 が無効なときだけです (PLL2ON = 0 および PLL2RDY = 0)。

0000000: pll2_q_ck = vco2_ck

0000001: pll2_q_ck = vco2_ck / 2 (リセット後のデフォルト)

0000010: pll2_q_ck = vco2_ck / 3

0000011: pll2_q_ck = vco2_ck / 4

.....

1111111: pll2_q_ck = vco2_ck / 128

ビット 15:9 **PLL2P[6:0]** : PLL2 DIVP の分周比

pll2_p_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL2 が無効なときだけです (PLL2ON = 0 および PLL2RDY = 0)。

0000000: pll2_p_ck = vco2_ck

0000001: pll2_p_ck = vco2_ck / 2 (リセット後のデフォルト)

0000010: pll2_p_ck = vco2_ck / 3

0000011: pll2_p_ck = vco2_ck / 4

.....

1111111: pll2_p_ck = vco2_ck / 128

ビット 8:0 **PLL2N[8:0]** : PLL2VCO の通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット／リセットされます。

これらのビットは、PLL が無効なとき (PLL2ON = 0 および PLL2RDY = 0) のみ、書き込むことができます。

0x003 : PLL2N = 4

0x004 : PLL2N = 5

0x005 : PLL2N = 6

.....

0x080 : PLL2N = 129 (リセット後のデフォルト)

.....

0x1FF : PLL2N = 512

その他 : 予約済み

11.8.13 RCC PLL2 小数分周器レジスタ (RCC_PLL2FRACR)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードおよびハーフワードアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL2FRACN [12:0]													Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW			

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:3 **PLL2FRACN[12:0]** : PLL2 VCO の通倍数の小数部

VCO の通倍数の小数部を制御するために、ソフトウェアによってセット／リセットされます。これらのビットはいつでも書き込めるため、PLL2 VCO を動的に微調整することができます。

注意 : これらのビットは、VCO 出力周波数が次の有効な周波数範囲の間になるように、ソフトウェアで正しく設定する必要があります。

*PLL2VCOSEL = 0 の場合、128 から 560MHz

* PLL2VCOSEL = 1 の場合、150 から 420 mHz

VCO 出力周波数 = $\text{Fref2_ck} \times (\text{PLL2N} + (\text{PLL2FRACN} / 213))$ であり、次の要件を満たすものとします。

* PLL2N は 8 から 420 の範囲

* PLL2FRACN は 0 から $2^{13}-1$ の範囲

*入力周波数 Fref2_ck は 1 から 16 mHz の範囲

PLL が有効な場合でも、PLL2FRACN の値をオンザフライで変更するには、アプリケーションは次の手順に従う必要があります。

*PLL2FRACN ビットを 0 にセットします。

* PLL2FRACN に新しい小数值を書き込みます。

* PLL2FRACN ビットを 1 にセットします。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

11.8.14 RCC PLL3 分周器レジスタ (RCC_PLL3DIVR)

アドレスオフセット : 0x044

リセット値 : 0x0101 0280

アクセス : ノーウェイトステート、ワード、ハーフワード;およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLL3R [6:0]							Res.	PLL3Q [6:0]						
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL3P [6:0]								PLL3N [8:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **PLL3R[6:0]** : PLL3 DIVR の分周比

pll3_r_ck クロックの周波数を制御するために、ソフトウェアによってセット／リセットされます。

これらのビットに書き込めるのは、PLL1 が無効なときだけです (PLL3ON = 0 および PLL3RDY = 0)。

0000000 : **pll3_r_ck** = **vco3_ck**

0000001 : **pll3_r_ck** = **vco3_ck** / 2 (リセット後のデフォルト)

0000010 : **pll3_r_ck** = **vco3_ck** / 3

0000011 : **pll3_r_ck** = **vco3_ck** / 4

.....

1111111 : **pll3_r_ck** = **vco3_ck** / 128

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **PLL3Q[6:0]** : PLL3 DIVQ の分周比

pll3_q_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL3 が無効なときだけです (PLL3ON = 0 および PLL3RDY = 0)。

0000000: pll3_q_ck = vco3_ck

0000001: pll3_q_ck = vco3_ck / 2 (リセット後のデフォルト)

0000010: pll3_q_ck = vco3_ck / 3

0000011: pll3_q_ck = vco3_ck / 4

.....

1111111: pll3_q_ck = vco3_ck / 128

ビット 15:9 **PLL3P[6:0]** : PLL3 DIVP の分周比

pll3_p_ck クロックの周波数を制御するために、ソフトウェアによってセットおよびリセットされます。
これらのビットに書き込めるのは、PLL3 が無効なときだけです (PLL3ON = 0 および PLL3RDY = 0)。

0000000: pll3_p_ck = vco3_ck

0000001: pll3_p_ck = vco3_ck / 2 (リセット後のデフォルト)

0000010: pll3_p_ck = vco3_ck / 3

0000011: pll3_p_ck = vco3_ck / 4

.....

1111111: pll3_p_ck = vco3_ck / 128

ビット 8:0 **PLL3N[8:0]** : PLL3VCO の通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット/リセットされます。

これらのビットは、PLL が無効なとき (PLL3ON = 0 および PLL3RDY = 0) のみ、書き込むことができます。

0x003 : PLL3N = 4

0x004 : PLL3N = 5

0x005 : PLL3N = 6

.....

0x080 : PLL3N = 129 (リセット後のデフォルト)

.....

0x1FF : PLL3N = 512

その他 : 予約済み

11.8.15 RCC PLL3 小数分周器レジスタ (RCC_PLL3FRACR)

アドレスオフセット : 0x048

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードおよびハーフワードアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL3FRACN [12:0]													Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW			

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:3 **PLL3FRACN[12:0]** : PLL3 VCO の通倍数の小数部

VCO の通倍数の小数部を制御するために、ソフトウェアによってセット／リセットされます。これらのビットはいつでも書き込めるため、PLL3 VCO を動的に微調整することができます。

注意 : これらのビットは、VCO 出力周波数が次の有効な周波数範囲の間になるように、ソフトウェアで正しく設定する必要があります。

*PLL3VCOSEL = 0 の場合、128 から 560MHz

* PLL3VCOSEL = 1 の場合、150 から 420 mHz

VCO 出力周波数 = $Fref3_ck \times (PLL3N + (PLL3FRACN / 213))$ であり、次の要件を満たすものとします。

* PLL3N は 8 から 420 の範囲

* PLL3FRACN は 0 から $2^{13}-1$ の範囲

*入力周波数 Fref3_ck は 1 から 16 mHz の範囲

PLL が有効な場合でも、PLL3FRACN の値をオンザフライで変更するには、アプリケーションは次の手順に従う必要があります。

*PLL3FRACEN ビットを 0 にセットします。

* PLL3FRACN に新しい小数値を書き込みます。

* PLL3FRACEN ビットを 1 にセットします。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

11.8.16 RCC クロックソース割込み有効レジスタ (RCC_CIER)

アドレスオフセット : 0x050

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3R DYIE	PLL2R DYIE	PLL1R DYIE	HSI48R DYIE	HSERD YIE	HSIRD YIE	CSIRD YIE	LSERD YIE	LSIRD YIE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PLL3RDYIE** : PLL3 レディ割込み有効化

PLL3 ロックによって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : PLL3 ロック割込み無効 (リセット後のデフォルト)

1 : PLL3 ロック割込み有効

ビット 7 **PLL2RDYIE** : PLL2 レディ割込み有効化

PLL2 ロックによって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : PLL2 ロック割込み無効 (リセット後のデフォルト)

1 : PLL2 ロック割込み有効

ビット 6 **PLL1RDYIE** : PLL1 レディ割込み有効化

PLL1 ロックによって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : PLL1 ロック割込み無効 (リセット後のデフォルト)

1 : PLL1 ロック割込み有効

ビット 5 HSI48RDYIE : HSI48 レディ割込み有効化

HSI48 オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : HSI48 レディ割込みは無効です（リセット後のデフォルト）。

1 : HSI48 レディ割込みは有効です。

ビット 4 HSERDYIE : HSE レディ割込みイネーブル

HSE オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : HSE レディ割込みは無効です（リセット後のデフォルト）。

1 : HSE レディ割込みは有効です。

ビット 3 HSIRDYIE : HSI レディ割込みイネーブル

HSI オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : HSI レディ割込みは無効です（リセット後のデフォルト）。

1 : HSI レディ割込みは有効です。

ビット 2 CSIRDYIE : CSI レディ割込みイネーブル

CSI オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : CSI レディ割込みは無効です（リセット後のデフォルト）。

1 : CSI レディ割込みは有効です。

ビット 1 LSERDYIE : LSE レディ割込みイネーブル

LSE オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : LSE レディ割込みは無効です（リセット後のデフォルト）。

1 : LSE レディ割込みは有効です。

ビット 0 LSIRDYIE : LSI レディ割込みイネーブル

LSI オシレータの安定化によって発生する割込みを有効／無効にするために、ソフトウェアによってセット／リセットされます。

0 : LSI レディ割込みは無効です（リセット後のデフォルト）。

1 : LSI レディ割込みは有効です。

11.8.17 RCC クロックソース割込みフラグレジスタ（RCC_CIFR）

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HSECS SF	Res.	PLL3R DYF	PLL2R DYF	PLL1R DYF	HSI48R DYF	HSERD YF	HSIRD YF	CSIRD YF	LSERD YF	LSIRD YF
					r		r	r	r	r	r	r	r	r	r

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 HSECSSF : HSE クロックセキュリティシステム割込みフラグ

HSECSSC ビットに書き込むことによりソフトウェアによってリセットされます。

HSE クロックに障害が発生した場合に、ハードウェアによってセットされます。

0 : HSE クロック障害によるクロックセキュリティ割込みは発生していません (リセット後のデフォルト)。

1 : HSE クロック障害によってクロックセキュリティ割込みが発生しました。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 PLL3RDYF : PLL3 レディ割込みフラグ

PLL3RDYC ビットに書き込むことによりソフトウェアによってリセットされます。

PLL3 がロックされ、PLL3RDYIE がセットされているときに、ハードウェアによってセットされます。

0 : PLL3 ロックによるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : PLL3 ロックによるクロックレディ割込みが発生しました。

ビット 7 PLL2RDYF : PLL2 レディ割込みフラグ

PLL2RDYC ビットに書き込むことによりソフトウェアによってリセットされます。

PLL2 がロックされ、PLL2RDYIE がセットされているときに、ハードウェアによってセットされます。

0 : PLL2 ロックによるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : PLL2 ロックによるクロックレディ割込みが発生しました。

ビット 6 PLL1RDYF : PLL1 レディ割込みフラグ

PLL1RDYC ビットに書き込むことによりソフトウェアによってリセットされます。

PLL1 がロックされ、PLL1RDYIE がセットされているときに、ハードウェアによってセットされます。

0 : PLL1 ロックによるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : PLL1 ロックによるクロックレディ割込みが発生しました。

ビット 5 HSI48RDYF : HSI48 レディ割込みフラグ

HSI48RDYC ビットに書き込むことによりソフトウェアによってリセットされます。

HSI48 クロックが安定し、HSI48RDYIE がセットされているときに、ハードウェアによってセットされます。

0 : HSI48 オシレータによるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : HSI48 オシレータによるクロックレディ割込みが発生しました。

ビット 4 HSERDYF : HSE レディ割込みフラグ

HSERDYC ビットに書き込むことによりソフトウェアによってリセットされます。

HSE クロックが安定し、HSERDYIE がセットされているときに、ハードウェアによってセットされます。

0 : HSE によるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : HSE によるクロックレディ割込みが発生しました。

ビット 3 HSIRDYF : HSI レディ割込みフラグ

HSIRDYC ビットに書き込むことによりソフトウェアによってリセットされます。

HSI クロックが安定し、HSIRDYIE がセットされているときに、ハードウェアによってセットされます。

0 : HSI によるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : HSI によるクロックレディ割込みが発生しました。

ビット 2 CSIRDYF : CSI レディ割込みフラグ

CSIRDYC ビットに書き込むことによりソフトウェアによってリセットされます。

CSI クロックが安定し、CSIRDYIE がセットされているときに、ハードウェアによってセットされます。

0 : CSI によるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : CSI によるクロックレディ割込みが発生しました。

ビット 1 LSERDYF : LSE レディ割込みフラグ

LSERDYC ビットに書き込むことによりソフトウェアによってリセットされます。

LSE クロックが安定し、LSERDYIE がセットされているときに、ハードウェアによってセットされます。

0 : LSE によるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : LSE によるクロックレディ割込みが発生しました。

ビット 0 **LSIRDYF** : LSI レディ割込みフラグ

LSIRDYC ビットに書き込むことによりソフトウェアによってリセットされます。

LSI クロックが安定し、LSIRDYIE がセットされているときに、ハードウェアによってセットされます。

0 : LSI によるクロックレディ割込みは発生していません (リセット後のデフォルト)。

1 : LSI によるクロックレディ割込みが発生しました。

11.8.18 RCC クロックソース割込みクリアレジスタ (RCC_CICR)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HSECS SC	Res.	PLL3R DYC	PLL2R DYC	PLL1R DYC	HSI48R DYC	HSERD YC	HSIRD YC	CSIRD YC	LSERD YC	LSIRD YC
					rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **HSECSSC** : HSE クロックセキュリティシステム割込みクリア

HSECSSF をクリアするために、ソフトウェアによってセットされます。

クリア終了後にハードウェアによってリセットされます。

0 : HSECSSF への影響はありません (リセット後のデフォルト)。

1 : HSECSSF はクリアされます。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PLL3RDYC** : PLL3 レディ割込みクリア

PLL3RDYF をクリアするために、ソフトウェアによってセットされます。

クリア終了後にハードウェアによってリセットされます。

0 : PLL3RDYF への影響はありません (リセット後のデフォルト)。

1 : PLL3RDYF がクリアされます。

ビット 7 **PLL2RDYC** : PLL2 レディ割込みクリア

PLL2RDYF をクリアするために、ソフトウェアによってセットされます。

クリア終了後にハードウェアによってリセットされます。

0 : PLL2RDYF への影響はありません (リセット後のデフォルト)。

1 : PLL2RDYF がクリアされます。

ビット 6 **PLL1RDYC** : PLL1 レディ割込みクリア

PLL1RDYF をクリアするために、ソフトウェアによってセットされます。

クリア終了後にハードウェアによってリセットされます。

0 : PLL1RDYF への影響はありません (リセット後のデフォルト)。

1 : PLL1RDYF がクリアされます。

ビット 5 **HSI48RDYC** : HSI48 レディ割込みクリア

HSI48RDYF をクリアするために、ソフトウェアによってセットされます。

クリア終了後にハードウェアによってリセットされます。

0 : HSI48RDYF への影響はありません (リセット後のデフォルト)。

1 : HSI48RDYF がクリアされます。

ビット 4 **HSERDYF** : HSE レディ割込みクリア

HSERDYF をクリアするために、ソフトウェアによってセットされます。
 クリア終了後にハードウェアによってリセットされます。
 0 : HSERDYF への影響はありません (リセット後のデフォルト)。
 1 : HSERDYF がクリアされます。

ビット 3 **HSIRDYF** : HSI レディ割込みクリア

HSIRDYF をクリアするために、ソフトウェアによってセットされます。
 クリア終了後にハードウェアによってリセットされます。
 0 : HSIRDYF への影響はありません (リセット後のデフォルト)。
 1 : HSIRDYF がクリアされます。

ビット 2 **CSIRDYF** : HSI レディ割込みクリア

CSIRDYF をクリアするために、ソフトウェアによってセットされます。
 クリア終了後にハードウェアによってリセットされます。
 0 : CSIRDYF への影響はありません (リセット後のデフォルト)。
 1 : CSIRDYF がクリアされます。

ビット 1 **LSERDYF** : LSE レディ割込みクリア

LSERDYF をクリアするために、ソフトウェアによってセットされます。
 クリア終了後にハードウェアによってリセットされます。
 0 : LSERDYF への影響はありません (リセット後のデフォルト)。
 1 : LSERDYF がクリアされます。

ビット 0 **LSIRDYF** : LSI レディ割込みクリア

LSIRDYF をクリアするために、ソフトウェアによってセットされます。
 クリア終了後にハードウェアによってリセットされます。
 0 : LSIRDYF への影響はありません (リセット後のデフォルト)。
 1 : LSIRDYF がクリアされます。

11.8.19 **RCC AHB1 リセットレジスタ (RCC_AHB1RSTR)**

アドレスオフセット : 0x060

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH RST	Res.	RAMCFG RST	Res.
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMAC RST	CORDIC RST	Res.	CRC RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
rw	rw		rw												

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **ETHRST** : ETHRST ブロックリセット

ソフトウェアでセット/リセットされます。
 0 : ETHRST ブロックをリセットしません (リセット後のデフォルト)。
 1 : ETHRST ブロックをリセットします。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RAMCFGRST** : RAMCFG ブロックリセット

ソフトウェアでセット／クリアされます。

0 : RAMCFG ブロックをリセットしません (リセット後のデフォルト)。

1 : RAMCFG ブロックをリセットします。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **FMACRST** : FMAC ブロックリセット

ソフトウェアでセット／クリアされます。

0 : FMAC ブロックをリセットしません (リセット後のデフォルト)。

1 : FMAC ブロックをリセットします。

ビット 14 **CORDICRST** : CORDIC ブロックリセット

ソフトウェアでセット／クリアされます。

0 : CORDIC ブロックをリセットしません (リセット後のデフォルト)。

1 : CORDIC ブロックをリセットします。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCRST** : CRC ブロックリセット。ソフトウェアによってセットおよびリセットされます。

0 : CRC ブロックをリセットしません (リセット後のデフォルト)。

1 : CRC ブロックをリセットします。

ビット 11:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **GPDMA2RST** : GPDMA2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : GPDMA2 ブロックをリセットしません (リセット後のデフォルト)。

1 : GPDMA2 ブロックをリセットします。

ビット 0 **GPDMA1RST** : GPDMA1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : GPDMA1 ブロックをリセットしません (リセット後のデフォルト)。

1 : GPDMA1 ブロックをリセットします。

11.8.20 RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR)

アドレスオフセット : 0x064

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAES RST	PKA RST	RNG RST	HASH RST	AES RST
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DCMI_ PSSI RST	DAC1 RST	ADC RST	Res.	GPIOI RST	GPIOH RST	GPIOG RST	GPIOF RST	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST
			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **SAESRST** : SAES ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SAES ブロックをリセットしません (リセット後のデフォルト)。

1 : SAES ブロックをリセットします。

- ビット 19 **PKARST** : PKA ブロックリセット
ソフトウェアでセット／クリアされます。
0 : PKA ブロックをリセットしません (リセット後のデフォルト)。
1 : PKA ブロックをリセットします。
- ビット 18 **RNGRST** : RNG ブロックリセット
ソフトウェアでセット／クリアされます。
0 : RNG ブロックをリセットしません (リセット後のデフォルト)。
1 : RNG ブロックをリセットします。
- ビット 17 **HASHRST** : HASH ブロックリセット
ソフトウェアでセット／クリアされます。
0 : HASH ブロックをリセットしません (リセット後のデフォルト)。
1 : HASH ブロックをリセットします。
- ビット 16 **AESRST** : AES ブロックリセット
ソフトウェアでセット／クリアされます。
0 : AES ブロックをリセットしません (リセット後のデフォルト)。
1 : AES ブロックをリセットします。
- ビット 15:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **DCMI_PSSIRST** : デジタルカメラインタフェースブロックリセット (アクティブなインタフェースに応じて、DCMI または PSSI)
ソフトウェアでセット／クリアされます。
0 : DCMI/PSSI ブロックをリセットしません (リセット後のデフォルト)。
1 : DCMI/PSSI ブロックをリセットします。
- ビット 11 **DAC1RST** : DAC ブロックリセット
ソフトウェアでセット／クリアされます。
0 : DAC ブロックをリセットしません (リセット後のデフォルト)。
1 : DAC ブロックをリセットします。
- ビット 10 **ADCRST** : ADC1 および 2 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : ADC1 および 2 ブロックをリセットしません (リセット後のデフォルト)。
1 : ADC1 および 2 ブロックをリセットします。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **GPIOIRST** : GPIOI ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOI ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOI ブロックをリセットします。
- ビット 7 **GPIOHRST** : GPIOH ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOH ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOH ブロックをリセットします。
- ビット 6 **GPIOGRST** : GPIOG ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOG ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOG ブロックをリセットします。
- ビット 5 **GPIOFRST** : GPIOF ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOF ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOF ブロックをリセットします。

- ビット 4 **GPIOERST** : GPIOE ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOE ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOE ブロックをリセットします。
- ビット 3 **GPIODRST** : GPIOD ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOD ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOD ブロックをリセットします。
- ビット 2 **GPIOCRST** : GPIOC ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOC ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOC ブロックをリセットします。
- ビット 1 **GPIOBRST** : GPIOB ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOB ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOB ブロックをリセットします。
- ビット 0 **GPIOARST** : GPIOA ブロックリセット
ソフトウェアでセット／クリアされます。
0 : GPIOA ブロックをリセットしません (リセット後のデフォルト)。
1 : GPIOA ブロックをリセットします。

11.8.21 RCC AHB4 ペリフェラルリセットレジスタ (RCC_AHB4RSTR)

アドレスオフセット : 0x06C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSPI1 RST	Res.	Res.	Res.	FMC RST
											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SDMMC2 RST	SDMMC1 RST	Res.	Res.	Res.	OTFDEC1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw				rw							

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

- ビット 20 **OCTOSPI1RST** : OCTOSPI1 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : OCTOSPI1 ブロックをリセットしません (リセット後のデフォルト)。
1 : OCTOSPI1 ブロックをリセットします。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

- ビット 16 **FMC RST** : FMC ブロックリセット
ソフトウェアでセット／クリアされます。
0 : FMC ブロックをリセットしません (リセット後のデフォルト)。
1 : FMC ブロックをリセットします。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **SDMMC2RST** : SDMMC2 および SDMMC2 遅延ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SDMMC2 および SDMMC2 遅延ブロックをリセットしません (リセット後のデフォルト)。

1 : SDMMC2 および SDMMC2 遅延ブロックをリセットします。

ビット 11 **SDMMC1RST** : SDMMC1 および SDMMC1 遅延ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SDMMC1 および SDMMC1 遅延ブロックをリセットしません (リセット後のデフォルト)。

1 : SDMMC1 および SDMMC1 遅延ブロックをリセットします。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTFDEC1RST** : OTFDEC1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : OTFDEC1 ブロックをリセットしません (リセット後のデフォルト)。

1 : OTFDEC1 ブロックをリセットします。

ビット 6:0 予約済みであり、リセット値に保持する必要があります。

11.8.22 RCC APB1 ペリフェラル低リセットレジスタ (RCC_APB1LRSTR)

アドレスオフセット : 0x074

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8 RST	UART7 RST	Res.	CEC RST	USART11 RST	USART10 RST	USART6 RST	CRS RST	I3C1 RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	USART3 RST	USART2 RST	Res.
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Res.	Res.	Res.	Res.	Res.	TIM14 RST	TIM13 RST	TIM12 RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw						rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **UART8RST** : UART8 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : UART8 ブロックをリセットしません (リセット後のデフォルト)。

1 : UART8 ブロックをリセットします。

ビット 30 **UART7RST** : UART7 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : UART7 ブロックをリセットしません (リセット後のデフォルト)。

1 : UART7 ブロックをリセットします。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **CECRST** : HDMI-CEC ブロックリセット

ソフトウェアでセット／クリアされます。

0 : HDMI-CEC ブロックをリセットしません (リセット後のデフォルト)。

1 : HDMI-CEC ブロックをリセットします。

ビット 27 **USART11RST** : USART11 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : USART11 ブロックをリセットしません (リセット後のデフォルト)。

1 : USART11 ブロックをリセットします。

- ビット 26 **USART10RST** : USART10 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : USART10 ブロックをリセットしません (リセット後のデフォルト)。
1 : USART10 ブロックをリセットします。
- ビット 25 **USART6RST** : USART6 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : USART6 ブロックをリセットしません (リセット後のデフォルト)。
1 : USART6 ブロックをリセットします。
- ビット 24 **CRSRST** : CRS ブロックリセット
ソフトウェアでセット／クリアされます。
0 : CRS ブロックをリセットしません (リセット後のデフォルト)。
1 : CRS ブロックをリセットします。
- ビット 23 **I3C1RST** : I3C1 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : I3C1 ブロックをリセットしません (リセット後のデフォルト)。
1 : I3C1 ブロックをリセットします。
- ビット 22 **I2C2RST** : I2C2 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : I2C2 ブロックをリセットしません (リセット後のデフォルト)。
1 : I2C2 ブロックをリセットします。
- ビット 21 **I2C1RST** : I2C1 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : I2C1 ブロックをリセットしません (リセット後のデフォルト)。
1 : I2C1 ブロックをリセットします。
- ビット 20 **UART5RST** : UART5 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : UART5 ブロックをリセットしません (リセット後のデフォルト)。
1 : UART5 ブロックをリセットします。
- ビット 19 **UART4RST** : UART4 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : UART4 ブロックをリセットしません (リセット後のデフォルト)。
1 : UART4 ブロックをリセットします。
- ビット 18 **USART3RST** : USART3 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : USART3 ブロックをリセットしません (リセット後のデフォルト)。
1 : USART3 ブロックをリセットします。
- ビット 17 **USART2RST** : USART2 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : USART2 ブロックをリセットしません (リセット後のデフォルト)。
1 : USART2 ブロックをリセットします。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3RST** : SPI3 ブロックリセット
ソフトウェアでセット／クリアされます。
0 : SPI3 ブロックをリセットしません (リセット後のデフォルト)。
1 : SPI3 ブロックをリセットします。

ビット 14 **SPI2RST** : SPI2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SPI2 ブロックをリセットしません (リセット後のデフォルト)。

1 : SPI2 ブロックをリセットします。

ビット 13:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14RST** : TIM14 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM14 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM14 ブロックをリセットします。

ビット 7 **TIM13RST** : TIM13 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM13 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM13 ブロックをリセットします。

ビット 6 **TIM12RST** : TIM12 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM12 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM12 ブロックをリセットします。

ビット 5 **TIM7RST** : TIM7 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM7 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM7 ブロックをリセットします。

ビット 4 **TIM6RST** : TIM6 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM6 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM6 ブロックをリセットします。

ビット 3 **TIM5RST** : TIM5 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM5 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM5 ブロックをリセットします。

ビット 2 **TIM4RST** : TIM4 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM4 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM4 ブロックをリセットします。

ビット 1 **TIM3RST** : TIM3 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM3 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM3 ブロックをリセットします。

ビット 0 **TIM2RST** : TIM2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM2 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM2 ブロックをリセットします。

11.8.23 RCC APB1 ペリフェラル高リセットレジスタ (RCC_APB1HRSTR)

アドレスオフセット : 0x078

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FDCAN RST	Res.	Res.	Res.	LPTIM2 RST	Res.	DTS RST	Res.	UART12 RST	UART9 RST
						rw				rw		rw		rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **UCPD1RST** : UCPD1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : UCPD ブロックをリセットしません (リセット後のデフォルト)。

1 : UCPD ブロックをリセットします。

ビット 22:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **FDCANRST** : FDCAN1 および FDCAN2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : FDCAN1 および FDCAN2 ブロックをリセットしません (リセット後のデフォルト)。

1 : FDCAN1 および FDCAN2 ブロックをリセットします。

ビット 8:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **LPTIM2RST** : LPTIM2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM2 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM2 ブロックをリセットします。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **DTSRST** : DTS ブロックリセット

ソフトウェアでセット／クリアされます。

0 : DTS ブロックをリセットしません (リセット後のデフォルト)。

1 : DTS ブロックをリセットします。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **UART12RST** : UART12 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : UART12 ブロックをリセットしません (リセット後のデフォルト)。

1 : UART12 ブロックをリセットします。

ビット 0 **UART9RST** : UART9 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : UART9 ブロックをリセットしません (リセット後のデフォルト)。

1 : UART9 ブロックをリセットします。

11.8.24 RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)

アドレスオフセット : 0x07C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USB RST	Res.	SAI2 RST	SAI1 RST	SPI6 RST	SPI4 RST	TIM17 RST	TIM16 RST	TIM15 RST
							rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw											

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **USBRST** : USB ブロックリセット

ソフトウェアでセット／クリアされます。

0 : USB ブロックをリセットしません (リセット後のデフォルト)。

1 : USB ブロックをリセットします。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SAI2RST** : SAI2 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SAI2 ブロックをリセットしません (リセット後のデフォルト)。

1 : SAI2 ブロックをリセットします。

ビット 21 **SAI1RST** : SAI1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SAI1 をリセットしません (リセット後のデフォルト)。

1 : SAI1 をリセットします。

ビット 20 **SPI6RST** : SPI6 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SPI6 ブロックをリセットしません (リセット後のデフォルト)。

1 : SPI6 ブロックをリセットします。

ビット 19 **SPI4RST** : SPI4 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SPI4 ブロックをリセットしません (リセット後のデフォルト)。

1 : SPI4 ブロックをリセットします。

ビット 18 **TIM17RST** : TIM17 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM17 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM17 ブロックをリセットします。

ビット 17 **TIM16RST** : TIM16 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM16 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM16 ブロックをリセットします。

ビット 16 **TIM15RST** : TIM15 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM15 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM15 ブロックをリセットします。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **USART1RST** : USART1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : USART1 ブロックをリセットしません (リセット後のデフォルト)。

1 : USART1 ブロックをリセットします。

ビット 13 **TIM8RST** : TIM8 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM8 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM8 ブロックをリセットします。

ビット 12 **SPI1RST** : SPI1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SPI1 ブロックをリセットしません (リセット後のデフォルト)。

1 : SPI1 ブロックをリセットします。

ビット 11 **TIM1RST** : TIM1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : TIM1 ブロックをリセットしません (リセット後のデフォルト)。

1 : TIM1 ブロックをリセットします。

ビット 10:0 予約済みであり、リセット値に保持する必要があります。

11.8.25 RCC APB3 ペリフェラルリセットレジスタ (RCC_APB3RSTR)

アドレスオフセット : 0x080

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VREF RST	Res.	Res.	Res.	Res.
											rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPTIM6 RST	LPTIM5 RST	LPTIM4 RST	LPTIM3 RST	LPTIM1 RST	Res.	Res.	I2C4 RST	I2C3 RST	LPUART1 RST	SPI5 RST	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw			rw	rw	rw	rw					

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **VREFRST** : VREFBUF ブロックリセット

ソフトウェアでセット／クリアされます。

0 : VREFBUF ブロックをリセットしません (リセット後のデフォルト)。

1 : VREFBUF ブロックをリセットします。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **LPTIM6RST** : LPTIM6 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM6 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM6 ブロックをリセットします。

ビット 14 **LPTIM5RST** : LPTIM5 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM5 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM5 ブロックをリセットします。

ビット 13 **LPTIM4RST** : LPTIM4 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM4 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM4 ブロックをリセットします。

ビット 12 **LPTIM3RST** : LPTIM3 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM3 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM3 ブロックをリセットします。

ビット 11 **LPTIM1RST** : LPTIM1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPTIM1 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPTIM1 ブロックをリセットします。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **I2C4RST** : I2C4 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : I2C4 ブロックをリセットしません (リセット後のデフォルト)。

1 : I2C4 ブロックをリセットします。

ビット 7 **I2C3RST** : I2C3 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : I2C3 ブロックをリセットしません (リセット後のデフォルト)。

1 : I2C3 ブロックをリセットします。

ビット 6 **LPUART1RST** : LPUART1 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : LPUART1 ブロックをリセットしません (リセット後のデフォルト)。

1 : LPUART1 ブロックをリセットします。

ビット 5 **SPI5RST** : SPI5 ブロックリセット

ソフトウェアでセット／クリアされます。

0 : SPI5 ブロックをリセットしません (リセット後のデフォルト)。

1 : SPI5 ブロックをリセットします。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

11.8.26 RCC AHB1 ペリフェラルクロックレジスタ (RCC_AHB1ENR)

アドレスオフセット : 0x088

リセット値 : 0xD000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM1 EN	DCACHE HEEN	Res.	BKPRAM EN	Res.	Res.	Res.	TZSC1 EN	Res.	Res.	ETHRX EN	ETHTX EN	ETHEN	Res.	RAMC FGEN	Res.
rw	rw		rw				rw			rw	rw	rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMAC EN	CORDI CEN	Res.	CRCE N	Res.	Res.	Res.	FLITF EN	Res.	Res.	Res.	Res.	Res.	Res.	GPDM A2EN	GPDM A1EN
rw	rw		rw				rw							rw	rw

ビット 31 **SRAM1EN** : SRAM1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SRAM1 クロック無効

1 : SRAM1 クロック有効 (リセット後のデフォルト)。

ビット 30 **DCACHEEN** : DCACHE クロック有効化

ソフトウェアでセット/リセットされます。

0 : DCACHE ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : DCACHE ペリフェラルクロックは有効です。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **BKPRAMEN** : BKPRAM クロック有効化

ソフトウェアでセット/リセットされます。

0 : BKPRAM ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : BKPRAM ペリフェラルクロックは有効です。

ビット 27:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TZSC1EN** : TZSC1 クロック有効化

ソフトウェアでセット/リセットされます。

0 : TZSC1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TZSC1 ペリフェラルクロックは有効です。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **ETHRXEN** : ETHRX クロック有効化

ソフトウェアでセット/リセットされます。

0 : ETHRX クロックは無効です (リセット後のデフォルト)。

1 : ETHRX クロック有効

ビット 20 **ETHTXEN** : ETHTX クロック有効化

ソフトウェアでセット/リセットされます。

0 : ETHTX クロックは無効です (リセット後のデフォルト)。

1 : ETHTX クロック有効

ビット 19 **ETHEN** : ETH クロック有効化

ソフトウェアでセット/リセットされます。

0 : ETH ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : ETH ペリフェラルクロックは有効です。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RAMCFGEN** : RAMCFG クロック有効化

ソフトウェアでセット／クリアされます。

0 : RAMCFG ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : RAMCFG ペリフェラルクロックは有効です。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **FMACEN** : FMAC クロック有効化

ソフトウェアでセット／クリアされます。

0 : FMAC ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : FMAC ペリフェラルクロックは有効です。

ビット 14 **CORDICEN** : CORDIC クロック有効化

ソフトウェアでセット／クリアされます。

0 : CORDIC ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : CORDIC ペリフェラルクロックは有効です。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCEN** : CRC クロック有効化

ソフトウェアでセット／クリアされます。

0 : CRC ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : CRC ペリフェラルクロックは有効です。

ビット 11:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **FLITFEN** : Flash インタフェースクロックイネーブル

ソフトウェアでセット／クリアされます。

0 : FLASH インタフェースクロック無効

1 : FLASH インタフェースクロック有効 (リセット後のデフォルト)。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **GPDMA2EN** : GPDMA2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : GPDMA2 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : GPDMA2 ペリフェラルクロックは有効です。

ビット 0 **GPDMA1EN** : GPDMA1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : GPDMA1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : GPDMA1 ペリフェラルクロックは有効です。

11.8.27 RCC AHB2 ペリフェラルクロックレジスタ (RCC_AHB2ENR)

アドレスオフセット : 0x08C

リセット値 : 0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM3 EN	SRAM2 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAES EN	PKA EN	RNG EN	HASH EN	AES EN
rw	rw										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DCMI_P SSIEN	DAC1 EN	ADC EN	Res.	GPIOI EN	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **SRAM3EN** : SRAM3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SRAM3 クロック無効

1 : SRAM3 クロック有効 (リセット後のデフォルト)。

ビット 30 **SRAM2EN** : SRAM2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SRAM2 クロック無効

1 : SRAM2 クロック有効 (リセット後のデフォルト)。

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **SAESEN** : SAES クロック有効化

ソフトウェアでセット/クリアされます。

0 : SAES ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SAES ペリフェラルクロックは有効です。

ビット 19 **PKAEN** : PKA クロック有効化

ソフトウェアでセット/クリアされます。

0 : PKA ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : PKA ペリフェラルクロックは有効です。

ビット 18 **RNGEN** : RNG クロック有効化

ソフトウェアでセット/クリアされます。

0 : RNG ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : RNG ペリフェラルクロックは有効です。

ビット 17 **HASHEN** : HASH クロック有効化

ソフトウェアでセット/クリアされます。

0 : HASH ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : HASH ペリフェラルクロックは有効です。

ビット 16 **AESEN** : AES クロック有効化

ソフトウェアでセット/クリアされます。

0 : AES ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : AES ペリフェラルクロックは有効です。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DCMI_PSSIEN** : デジタルカメララインタフェースクロックイネーブル (アクティブなインタフェースに応じて、DCMI または PSSI)

ソフトウェアでセット/クリアされます。

0 : DCMI/PSSI ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : DCMI/PSSI ペリフェラルクロックは有効です。

- ビット 11 **DAC1EN** : DAC クロック有効化
ソフトウェアでセット/クリアされます。
0 : DAC ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : DAC ペリフェラルクロックは有効です。
- ビット 10 **ADCEN** : ADC1 および 2 ペリフェラルクロックは有効です。
ソフトウェアでセット/クリアされます。
0 : ADC1 および 2 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : ADC1 および 2 ペリフェラルクロックは有効です。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **GPIOIEN** : GPIOI クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOI ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOI ペリフェラルクロックは有効です。
- ビット 7 **GPIOHEN** : GPIOH クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOH ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOH ペリフェラルクロックは有効です。
- ビット 6 **GPIOGEN** : GPIOG クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOG ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOG ペリフェラルクロックは有効です。
- ビット 5 **GPIOFEN** : GPIOF クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOF ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOF ペリフェラルクロックは有効です。
- ビット 4 **GPIOEEN** : GPIOE クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOE ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOE ペリフェラルクロックは有効です。
- ビット 3 **GPIODEN** : GPIOD クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOD ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOD ペリフェラルクロックは有効です。
- ビット 2 **GPIOCEN** : GPIOC クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOC ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOC ペリフェラルクロックは有効です。
- ビット 1 **GPIOBEN** : GPIOB クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOB ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOB ペリフェラルクロックは有効です。
- ビット 0 **GPIOAEN** : GPIOA クロック有効化
ソフトウェアでセット/クリアされます。
0 : GPIOA ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : GPIOA ペリフェラルクロックは有効です。

11.8.28 RCC AHB4 ペリフェラルクロックレジスタ (RCC_AHB4ENR)

アドレスオフセット : 0x094

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSPI1 EN	Res.	Res.	Res.	FMC EN
											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SDMMC2 EN	SDMMC1 EN	Res.	Res.	Res.	OTFDEC1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw				rw							

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OCTOSPI1EN** : OCTOSPI1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : OCTOSPI1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : OCTOSPI1 ペリフェラルクロックは有効です。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **FMCEN** : FMC クロック有効化

ソフトウェアでセット／クリアされます。

0 : FMC ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : FMC ペリフェラルクロックは有効です。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **SDMMC2EN** : SDMMC2 および SDMMC2 遅延ペリフェラルクロックは有効です。

ソフトウェアでセット／クリアされます。

0 : SDMMC2 および SDMMC2 遅延ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SDMMC2 および SDMMC2 遅延ペリフェラルクロックは有効です。

ビット 11 **SDMMC1EN** : SDMMC1 および SDMMC1 遅延ペリフェラルクロックイネーブルリセット

0 : SDMMC1 および SDMMC1 遅延ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SDMMC1 および SDMMC1 遅延ペリフェラルクロックは有効です。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTFDEC1EN** : OTFDEC1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : OTFDEC1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : OTFDEC1 ペリフェラルクロックは有効です。

ビット 6:0 予約済みであり、リセット値に保持する必要があります。

11.8.29 RCC APB1 ペリフェラルクロックレジスタ (RCC_APB1LENR)

アドレスオフセット : 0x09C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8 EN	UART7 EN	Res.	CEC EN	USART11 EN	USART10 EN	USART6 EN	CRS EN	I3C1 EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Res.
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Res.	Res.	WWDG EN	Res.	Res.	TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **UART8EN** : UART8 クロック有効化

ソフトウェアでセット／クリアされます。

0 : UART8 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : UART8 ペリフェラルクロックは有効です。

ビット 30 **UART7EN** : UART7 クロック有効化

ソフトウェアでセット／クリアされます。

0 : UART7 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : UART7 ペリフェラルクロックは有効です。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **CECEN** : HDMI-CEC クロック有効化

ソフトウェアでセット／クリアされます。

0 : HDMI-CEC ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : HDMI-CEC ペリフェラルクロックは有効です。

ビット 27 **USART11EN** : USART11 クロック有効化

0 : USART11 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : USART11 ペリフェラルクロックは有効です。

ビット 26 **USART10EN** : USART10 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART10 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : USART10 ペリフェラルクロックは有効です。

ビット 25 **USART6EN** : USART6 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART6 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : USART6 ペリフェラルクロックは有効です。

ビット 24 **CRSEN** : CRS クロック有効化

ソフトウェアでセット／クリアされます。

0 : CRS ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : CRS ペリフェラルクロックは有効です。

ビット 23 **I3C1EN** : I3C1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : I3C1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : I3C1 ペリフェラルクロックは有効です。

- ビット 22 **I2C2EN** : I2C2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C2 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : I2C2 ペリフェラルクロックは有効です。
- ビット 21 **I2C1EN** : I2C1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C1 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : I2C1 ペリフェラルクロックは有効です。
- ビット 20 **UART5EN** : UART5 クロック有効化
ソフトウェアでセット／クリアされます。
0 : UART5 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : UART5 ペリフェラルクロックは有効です。
- ビット 19 **UART4EN** : UART4 クロック有効化
ソフトウェアでセット／クリアされます。
0 : UART4 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : UART4 ペリフェラルクロックは有効です。
- ビット 18 **USART3EN** : USART3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : USART3 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : USART3 ペリフェラルクロックは有効です。
- ビット 17 **USART2EN** : USART2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : USART2 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : USART2 ペリフェラルクロックは有効です。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3EN** : SPI3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI3 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : SPI3 ペリフェラルクロックは有効です。
- ビット 14 **SPI2EN** : SPI2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI2 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : SPI2 ペリフェラルクロックは有効です。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGEN** : WWDG クロック有効化
ソフトウェアでセット／クリアされます。
0 : WWDG ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : WWDG ペリフェラルクロックは有効です。
- ビット 10:9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **TIM14EN** : TIM14 クロック有効化
ソフトウェアでセット／クリアされます。
0 : TIM14 ペリフェラルクロックは無効です (リセット後のデフォルト)。
1 : TIM14 ペリフェラルクロックは有効です。

ビット 7 **TIM13EN** : TIM13 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM13 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM13 ペリフェラルクロックは有効です。

ビット 6 **TIM12EN** : TIM12 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM12 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM12 ペリフェラルクロックは有効です。

ビット 5 **TIM7EN** : TIM7 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM7 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM7 ペリフェラルクロックは有効です。

ビット 4 **TIM6EN** : TIM6 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM6 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM6 ペリフェラルクロックは有効です。

ビット 3 **TIM5EN** : TIM5 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM5 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM5 ペリフェラルクロックは有効です。

ビット 2 **TIM4EN** : TIM4 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM4 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM4 ペリフェラルクロックは有効です。

ビット 1 **TIM3EN** : TIM3 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM3 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM3 ペリフェラルクロックは有効です。

ビット 0 **TIM2EN** : TIM2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM2 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM2 ペリフェラルクロックは有効です。

11.8.30 RCC APB1 ペリフェラルクロックレジスタ (RCC_APB1HENR)

アドレスオフセット : 0x0A0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FDCAN EN	Res.	Res.	Res.	LPTIM2 EN	Res.	DTS EN	Res.	UART12 EN	UART9 EN
						rw				rw		rw		rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **UCPD1EN** : UCPD1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : UCPD ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : UCPD ペリフェラルクロックは有効です。

ビット 22:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **FDCANEN** : FDCAN1 および FDCAN2 ペリフェラルクロックイネーブル

ソフトウェアでセット／クリアされます。

0 : FDCAN1 および FDCAN2 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : FDCAN1 および FDCAN2 ペリフェラルクロックは有効です。

ビット 8:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **LPTIM2EN** : LPTIM2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : LPTIM2 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM2 ペリフェラルクロックは有効です。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **DTS**EN : DTS クロック有効化

ソフトウェアでセット／クリアされます。

0 : DTS ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : DTS ペリフェラルクロックは有効です。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **UART12EN** : UART12 クロック有効化

ソフトウェアでセット／クリアされます。

0 : UART12 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : UART12 ペリフェラルクロックは有効です。

ビット 0 **UART9EN** : UART9 クロック有効化

ソフトウェアでセット／クリアされます。

0 : UART9 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : UART9 ペリフェラルクロックリセットは有効です。

11.8.31 RCC APB2 ペリフェラルクロックレジスタ (RCC_APB2ENR)

アドレスオフセット : 0x0A4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBEN	Res.	SAI2EN	SAI1EN	SPI6EN	SPI4EN	TIM17EN	TIM16EN	TIM15EN
							rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1EN	TIM8EN	SPI1EN	TIM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw											

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **USBEN** : USB クロック有効化

ソフトウェアでセット／クリアされます。

0 : USB ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : USB ペリフェラルクロックは有効です。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SAI2EN** : SAI2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SAI2 クロック無効

1 : SAI2 クロック有効

ビット 21 **SAI1EN** : SAI1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SAI1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SAI1 ペリフェラルクロックは有効です。

ビット 20 **SPI6EN** : SPI6 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI6 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SPI6 ペリフェラルクロックは有効です。

ビット 19 **SPI4EN** : SPI4 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI4 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SPI4 ペリフェラルクロックは有効です。

ビット 18 **TIM17EN** : TIM17 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM17 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM17 ペリフェラルクロックは有効です。

ビット 17 **TIM16EN** : TIM16 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM16 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM16 ペリフェラルクロックは有効です。

ビット 16 **TIM15EN** : TIM15 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM15 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM15 ペリフェラルクロックは有効です。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **USART1EN** : USART1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : USART1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : USART1 ペリフェラルクロックは有効です。

ビット 13 **TIM8EN** : TIM8 クロック有効化

ソフトウェアでセット/クリアされます。

0 : TIM8 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM8 ペリフェラルクロックは有効です。

ビット 12 **SPI1EN** : SPI1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SPI1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SPI1 ペリフェラルクロックは有効です。

ビット 11 **TIM1EN** : TIM1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : TIM1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : TIM1 ペリフェラルクロックは有効です。

ビット 10:0 予約済みであり、リセット値に保持する必要があります。

11.8.32 RCC APB3 ペリフェラルクロックレジスタ (RCC_APB3ENR)

アドレスオフセット : 0x0A8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCAPB EN	VREF BUFEN	Res.	Res.	Res.	Res.
										rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPTIM6 EN	LPTIM5 EN	LPTIM4 EN	LPTIM3 EN	LPTIM1 EN	Res.	Res.	I2C4 EN	I2C3 EN	LPUART1 EN	SPI5 EN	Res.	Res.	Res.	SBS EN	Res.
rw	rw	rw	rw	rw			rw	rw	rw	rw				rw	

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **RTCAPBEN** : RTC APB インタフェースクロックイネーブル

ソフトウェアでセット/クリアされます。

0 : RTC APB インタフェースクロックは無効です (リセット後のデフォルト)。

1 : RTC APB インタフェースクロックは有効です。

ビット 20 **VREFBUFEN** : VREFBUF クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : VREFBUF ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : VREFBUF ペリフェラルクロックは有効です。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **LPTIM6EN** : LPTIM6 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPTIM6 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM6 ペリフェラルクロックは有効です。

ビット 14 **LPTIM5EN** : LPTIM5 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPTIM5 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM5 ペリフェラルクロックは有効です。

ビット 13 **LPTIM4EN** : LPTIM4 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPTIM4 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM4 ペリフェラルクロックは有効です。

ビット 12 **LPTIM3EN** : LPTIM3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPTIM3 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM3 ペリフェラルクロックは有効です。

ビット 11 **LPTIM1EN** : LPTIM1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPTIM1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPTIM1 ペリフェラルクロックは有効です。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **I2C4EN** : I2C4 クロック有効化

ソフトウェアでセット/クリアされます。

0 : I2C4 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : I2C4 ペリフェラルクロックは有効です。

ビット 7 **I2C3EN** : I2C3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : I2C3 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : I2C3 ペリフェラルクロックは有効です。

ビット 6 **LPUART1EN** : LPUART1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : LPUART1 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : LPUART1 ペリフェラルクロックは有効です。

ビット 5 **SPI5EN** : SPI5 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SPI5 ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SPI5 ペリフェラルクロックは有効です。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SBSSEN** : SBS クロック有効化

ソフトウェアでセット/クリアされます。

0 : SBS ペリフェラルクロックは無効です (リセット後のデフォルト)。

1 : SBS ペリフェラルクロックは有効です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

11.8.33 RCC AHB1 スリープクロックレジスタ (RCC_AHB1LPENR)

アドレスオフセット : 0x0B0

リセット値 : 0xF13A D103

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM1 LPEN	DCAC HELPE N	ICACH ELPEN	BKPRA MLPEN	Res.	Res.	Res.	TZSC1 LPEN	Res.	Res.	ETHRX LPEN	ETHTX LPEN	ETHLP EN	Res.	RAMC FGLPE N	Res.
rw	rw	rw	rw				rw			rw	rw	rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMACL PEN	CORDI CLPEN	Res.	CRCLP EN	Res.	Res.	Res.	FLITFL PEN	Res.	Res.	Res.	Res.	Res.	Res.	GPDM A2LPE N	GPDM A1LPE N
rw	rw		rw				rw							rw	rw

ビット 31 **SRAM1LPEN** : SLEEP モード時の SRAM1 クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、SRAM1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SRAM1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 30 **DCACHELPEN** : SLEEP モード時の DCACHE クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、DCACHE ペリフェラルクロックは無効です。

1 : SLEEP モード時、DCACHE ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 29 **ICACHELPEN** : SLEEP モード時の ICACHE クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、ICACHE ペリフェラルクロックは無効です。

1 : SLEEP モード時、ICACHE ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 28 **BKPRAML PEN** : SLEEP モード時の BKPRAM クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、BKPRAM ペリフェラルクロックは無効です。

1 : SLEEP モード時、BKPRAM ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 27:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TZSC1LPEN** : SLEEP モード時の TZSC1 クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、TZSC1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TZSC1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **ETHRXLPEN** : SLEEP モード時の ETHRX クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時 ETHRX クロック無効

1 : SLEEP モード時、ETHRX クロックは有効です (リセット後のデフォルト)。

ビット 20 **ETHTXLPEN** : SLEEP モード時の ETHTX クロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時 ETHTX クロック無効

1 : SLEEP モード時、ETHTX クロックは有効です (リセット後のデフォルト)。

- ビット 19 **ETHLPEN** : SLEEP モード時の ETH クロックイネーブル
ソフトウェアでセット/リセットされます。
0 : SLEEP モード時、ETH ペリフェラルクロックは無効です。
1 : SLEEP モード時、ETH ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 18 予約済みであり、リセット値に保持する必要があります。
- ビット 17 **RAMCFGLPEN** : SLEEP モード時の RAMCFG クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、RAMCFG ペリフェラルクロックは無効です。
1 : SLEEP モード時、RAMCFG ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **FMACLPEN** : SLEEP モード時の FMAC クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、FMAC ペリフェラルクロックは無効です。
1 : SLEEP モード時、FMAC ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 14 **CORDICLPEN** : SLEEP モード時の CORDIC クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、CORDIC ペリフェラルクロックは無効です。
1 : SLEEP モード時、CORDIC ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **CRCLPEN** : SLEEP モード時 CRC クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、CRC ペリフェラルクロックは無効です。
1 : SLEEP モード時、CRC ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 11:9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **FLITFLPEN** : SLEEP モード時の Flash インタフェース (FLITF) クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、FLITF ペリフェラルクロックは無効です。
1 : SLEEP モード時、FLITF ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 7:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **GPDMA2LPEN** : SLEEP モード時の GPDMA2 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPDMA2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPDMA2 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 0 **GPDMA1LPEN** : SLEEP モード時の GPDMA1 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPDMA1 ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPDMA1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

11.8.34 RCC AHB2 スリープクロックレジスタ (RCC_AHB2LPENR)

アドレスオフセット : 0x0B4

リセット値 : 0xC01F 1DFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM3 LPEN	SRAM2 LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAES LPEN	PKA LPEN	RNG LPEN	HASH LPEN	AES LPEN
rw	rw										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DCMI_PSSI LPEN	DAC1 LPEN	ADC LPEN	Res.	GPIOI LPEN	GPIOH LPEN	GPIOG LPEN	GPIOF LPEN	GPIOE LPEN	GPIOD LPEN	GPIOC LPEN	GPIOB LPEN	GPIOA LPEN
			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **SRAM3LPEN** : SLEEP モード時の SRAM3 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SRAM3 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SRAM3 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 30 **SRAM2LPEN** : SLEEP モード時の SRAM2 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SRAM2 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SRAM2 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **SAESLPEN** : SLEEP モード時の SAES クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SAES ペリフェラルクロックは無効です。

1 : SLEEP モード時、SAES ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 19 **PKALPEN** : SLEEP モード時の PKA クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、PKA ペリフェラルクロックは無効です。

1 : SLEEP モード時、PKA ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 18 **RNGLPEN** : SLEEP モード時の RNG クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、RNG ペリフェラルクロックは無効です。

1 : SLEEP モード時、RNG ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 17 **HASHLPEN** : SLEEP モード時の HASH クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、HASH ペリフェラルクロックは無効です。

1 : SLEEP モード時、HASH ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 16 **AESLPEN** : SLEEP モード時の AES クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、AES ペリフェラルクロックは無効です。

1 : SLEEP モード時、AES ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

- ビット 12 **DCMI_PSSILPEN** : SLEEP モード時のデジタルカメラインタフェースクロックイネーブル (アクティブなインタフェースに応じて、DCMI または PSSI)
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、DCMI/PSSI ペリフェラルクロックは無効です。
1 : SLEEP モード時、DCMI/PSSI ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 11 **DAC1LPEN** : SLEEP モード時の DAC クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、DAC ペリフェラルクロックは無効です。
1 : SLEEP モード時、DAC ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 10 **ADCLPEN** : SLEEP モード時の ADC1 および 2 ペリフェラルクロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、ADC1 および 2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、ADC1 および 2 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **GPIOLPEN** : SLEEP モード時の GPIOI クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOI ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOI ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 7 **GPIOHPEN** : SLEEP モード時の GPIOH クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOH ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOH ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 6 **GPIOGLPEN** : SLEEP モード時の GPIOG クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOG ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOG ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 5 **GPIOFLPEN** : SLEEP モード時の GPIOF クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOF ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOF ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 4 **GPIOELPEN** : SLEEP モード時の GPIOE クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOE ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOE ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 3 **GIODLPEN** : SLEEP モード時の GPIOD クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOD ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOD ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 2 **GPIOCLPEN** : SLEEP モード時の GPIOC クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOC ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOC ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 1 **GPIOBLPEN** : SLEEP モード時の GPIOB クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、GPIOB ペリフェラルクロックは無効です。
1 : SLEEP モード時、GPIOB ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 0 **GPIOALPEN** : SLEEP モード時の GPIOA クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、GPIOA ペリフェラルクロックは無効です。

1 : SLEEP モード時、GPIOA ペリフェラルクロックは有効です (リセット後のデフォルト)。

11.8.35 RCC AHB4 スリープクロックレジスタ (RCC_AHB4LPENR)

アドレスオフセット : 0x0BC

リセット値 : 0x0011 1880

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSPI1 LPEN	Res.	Res.	Res.	FMC LPEN
											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SDMMC2 LPEN	SDMMC1 LPEN	Res.	Res.	Res.	OTFDEC1 LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw				rw							

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OCTOSPI1LPEN** : SLEEP モード時の OCTOSPI1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、OCTOSPI1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、OCTOSPI1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **FMCLPEN** : SLEEP モード時の FMC クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、FMC ペリフェラルクロックは無効です。

1 : SLEEP モード時、FMC ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **SDMMC2LPEN** : SLEEP モード時の SDMMC2 および SDMMC2 遅延ペリフェラルクロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、SDMMC2 および SDMMC2 遅延ペリフェラルクロックは無効です。

1 : SLEEP モード時、SDMMC2 および SDMMC2 遅延ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 11 **SDMMC1LPEN** : SLEEP モード時の SDMMC1 および SDMMC1 遅延ペリフェラルクロックイネーブル

ソフトウェアでセット/リセットされます。

0 : SLEEP モード時、SDMMC1 および SDMMC1 遅延ペリフェラルクロックは無効です。

1 : SLEEP モード時、SDMMC1 および SDMMC1 遅延ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTFDEC1LPEN** : SLEEP モード時の OTFDEC1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、OTFDEC1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、OTFDEC1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 6:0 予約済みであり、リセット値に保持する必要があります。

11.8.36 RCC APB1 スリープクロックレジスタ (RCC_APB1LLPENR)

アドレスオフセット : 0x0C4

リセット値 : 0xDFFE C9FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8 LPEN	UART7 LPEN	Res.	CECLP EN	USART 11LPE N	USART 10LPE N	USART 6LPEN	CRSLP EN	I3C1LP EN	I2C2LP EN	I2C1LP EN	UART5 LPEN	UART4 LPEN	USART 3LPEN	USART 2LPEN	Res.
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3LP EN	SPI2LP EN	Res.	Res.	WWDG LPEN	Res.	Res.	TIM14L PEN	TIM13L PEN	TIM12L PEN	TIM7LP EN	TIM6LP EN	TIM5LP EN	TIM4LP EN	TIM3LP EN	TIM2LP EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **UART8LPEN** : SLEEP モード時の UART8 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、UART8 ペリフェラルクロックは無効です。

1 : SLEEP モード時、UART8 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 30 **UART7LPEN** : SLEEP モード時の UART7 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、UART7 ペリフェラルクロックは無効です。

1 : SLEEP モード時、UART7 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **CECLPEN** : SLEEP モード時の HDMI -CEC クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、HDMI-CEC ペリフェラルクロックは無効です。

1 : SLEEP モード時、HDMI-CEC ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 27 **USART11LPEN** : SLEEP モード時の USART11 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、USART11 ペリフェラルクロックは無効です。

1 : SLEEP モード時、USART11 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 26 **USART10LPEN** : SLEEP モード時の USART10 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、USART10 ペリフェラルクロックは無効です。

1 : SLEEP モード時、USART10 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 25 **USART6LPEN** : SLEEP モード時の USART6 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、USART6 ペリフェラルクロックは無効です。

1 : SLEEP モード時、USART6 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 24 **CRSLPEN** : SLEEP モード時の CRS クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、CRS ペリフェラルクロックは無効です。

1 : SLEEP モード時、CRS ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 23 **I3C1LPEN** : SLEEP モード時の I3C1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、I3C1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、I3C1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

- ビット 22 **I2C2LPEN** : SLEEP モード時の I2C2 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、I2C2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、I2C2 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 21 **I2C1LPEN** : SLEEP モード時の I2C1 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、I2C1 ペリフェラルクロックは無効です。
1 : SLEEP モード時、I2C1 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 20 **UART5LPEN** : SLEEP モード時の UART5 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、UART5 ペリフェラルクロックは無効です。
1 : SLEEP モード時、UART5 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 19 **UART4LPEN** : SLEEP モード時の UART4 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、UART4 ペリフェラルクロックは無効です。
1 : SLEEP モード時、UART4 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 18 **USART3LPEN** : SLEEP モード時の USART3 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、USART3 ペリフェラルクロックは無効です。
1 : SLEEP モード時、USART3 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 17 **USART2LPEN** : SLEEP モード時の USART2 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、USART2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、USART2 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3LPEN** : SLEEP モード時の SPI3 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、SPI3 ペリフェラルクロックは無効です。
1 : SLEEP モード時、SPI3 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 14 **SPI2LPEN** : SLEEP モード時の SPI2 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、SPI2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、SPI2 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGLPEN** : SLEEP モード時の WWDG クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、WWDG ペリフェラルクロックは無効です。
1 : SLEEP モード時、WWDG ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 10:9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **TIM14LPEN** : SLEEP モード時の TIM14 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM14 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM14 ペリフェラルクロックは有効です (リセット後のデフォルト)。

- ビット 7 **TIM13LPEN** : SLEEP モード時の TIM13 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM13 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM13 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 6 **TIM12LPEN** : SLEEP モード時の TIM12 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM12 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM12 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 5 **TIM7LPEN** : SLEEP モード時の TIM7 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM7 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM7 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 4 **TIM6LPEN** : SLEEP モード時の TIM6 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM6 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM6 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 3 **TIM5LPEN** : SLEEP モード時の TIM5 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM5 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM5 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 2 **TIM4LPEN** : SLEEP モード時の TIM4 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM4 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM4 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 1 **TIM3LPEN** : SLEEP モード時の TIM3 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM3 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM3 ペリフェラルクロックは有効です (リセット後のデフォルト)。
- ビット 0 **TIM2LPEN** : SLEEP モード時の TIM2 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、TIM2 ペリフェラルクロックは無効です。
1 : SLEEP モード時、TIM2 ペリフェラルクロックは有効です (リセット後のデフォルト)。

11.8.37 RCC APB1 スリープクロックレジスタ (RCC_APB1HLPENR)

アドレスオフセット : 0x0C8

リセット値 : 0x4080 022B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1 LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FDCAN LPEN	Res.	Res.	Res.	LPTIM2 LPEN	Res.	DTS LPEN	Res.	UART12 LPEN	UART9 LPEN
						rw				rw		rw		rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **UCPD1LPEN** : SLEEP モード時の UCPD1 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、UCPD1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、UCPD1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 22:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **FDCANLPEN** : SLEEP モード時の FDCAN1 および FDCAN2 ペリフェラルクロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、FDCAN1 および FDCAN2 ペリフェラルクロックは無効です。

1 : SLEEP モード時、FDCAN1 および FDCAN2 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 8:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **LPTIM2LPEN** : SLEEP モード時の LPTIM2 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、LPTIM2 ペリフェラルクロックは無効です。

1 : SLEEP モード時、LPTIM2 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **DTS LPEN** : SLEEP モード時の DTS クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、DTS ペリフェラルクロックは無効です。

1 : SLEEP モード時、DTS ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **UART12LPEN** : SLEEP モード時の UART12 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、UART12 ペリフェラルクロックは無効です。

1 : SLEEP モード時、UART12 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 0 **UART9LPEN** : SLEEP モード時の UART9 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、UART9 ペリフェラルクロックは無効です。

1 : SLEEP モード時、UART9 ペリフェラルクロックリセットは有効です (リセット後のデフォルト)。

11.8.38 RCC APB2 スリープクロックレジスタ (RCC_APB2LPENR)

アドレスオフセット : 0x0CC

リセット値 : 0x017F 7800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	USB LPEN	Res.	SAI2 LPEN	SAI1 LPEN	SPI6 LPEN	SPI4 LPEN	TIM17 LPEN	TIM16 LPEN	TIM15 LPEN
							rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 LPEN	TIM8 LPEN	SPI1 LPEN	TIM1 LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw											

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **USBLPEN** : SLEEP モード時の USB クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、USB ペリフェラルクロックは無効です。

1 : SLEEP モード時、USB ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SAI2LPEN** : SLEEP モード時の SAI2 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SAI2 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SAI2 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 21 **SAI1LPEN** : SLEEP モード時の SAI1 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SAI1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SAI1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 20 **SPI6LPEN** : SLEEP モード時の SPI6 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SPI6 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SPI6 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 19 **SPI4LPEN** : SLEEP モード時の SPI4 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、SPI4 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SPI4 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 18 **TIM17LPEN** : SLEEP モード時の TIM17 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、TIM17 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TIM17 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 17 **TIM16LPEN** : SLEEP モード時の TIM16 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、TIM16 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TIM16 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 16 **TIM15LPEN** : SLEEP モード時の TIM15 クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時、TIM15 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TIM15 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **USART1LPEN** : SLEEP モード時の USART1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、USART1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、USART1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 13 **TIM8LPEN** : SLEEP モード時の TIM8 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、TIM8 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TIM8 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 12 **SPI1LPEN** : SLEEP モード時の SPI1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、SPI1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、SPI1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 11 **TIM1LPEN** : SLEEP モード時の TIM1 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、TIM1 ペリフェラルクロックは無効です。

1 : SLEEP モード時、TIM1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 10:0 予約済みであり、リセット値に保持する必要があります。

11.8.39 RCC APB3 スリープクロックレジスタ (RCC_APB3LPENR)

アドレスオフセット : 0x0D0

リセット値 : 0x0030 : F9E2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCAPBLPEN	VREFLPEN	Res.	Res.	Res.	Res.
										rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPTIM6LPEN	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM1LPEN	Res.	Res.	I2C4LPEN	I2C3LPEN	LPUART1LPEN	SPI5LPEN	Res.	Res.	Res.	SBSLPEN	Res.
rw	rw	rw	rw	rw			rw	rw	rw	rw				rw	

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **RTCAPBLPEN** : SLEEP モード時 RTC APB インタフェースクロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 RTC APB インタフェースクロック無効

1 : SLEEP モード時、RTC APB インタフェースクロックは有効です (リセット後のデフォルト)。

ビット 20 **VREFLPEN** : SLEEP モード時の VREFBUF クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、VREFBUF ペリフェラルクロックは無効です。

1 : SLEEP モード時、VREFBUF ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **LPTIM6LPEN** : SLEEP モード時の LPTIM6 クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時、LPTIM6 ペリフェラルクロックは無効です。

1 : SLEEP モード時、LPTIM6 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 14 **LPTIM5LPEN** : SLEEP モード時の LPTIM5 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、LPTIM5 ペリフェラルクロックは無効です。
1 : SLEEP モード時、LPTIM5 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 13 **LPTIM4LPEN** : SLEEP モード時の LPTIM4 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、LPTIM4 ペリフェラルクロックは無効です。
1 : SLEEP モード時、LPTIM4 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 12 **LPTIM3LPEN** : SLEEP モード時の LPTIM3 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、LPTIM3 ペリフェラルクロックは無効です。
1 : SLEEP モード時、LPTIM3 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 11 **LPTIM1LPEN** : SLEEP モード時の LPTIM1 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、LPTIM1 ペリフェラルクロックは無効です。
1 : SLEEP モード時、LPTIM1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **I2C4LPEN** : SLEEP モード時の I2C4 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、I2C4 ペリフェラルクロックは無効です。
1 : SLEEP モード時、I2C4 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 7 **I2C3LPEN** : SLEEP モード時の I2C3 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、I2C3 ペリフェラルクロックは無効です。
1 : SLEEP モード時、I2C3 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 6 **LPUART1LPEN** : SLEEP モード時の LPUART1 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、LPUART1 ペリフェラルクロックは無効です。
1 : SLEEP モード時、LPUART1 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 5 **SPI5LPEN** : SLEEP モード時の SPI5 クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、SPI5 ペリフェラルクロックは無効です。
1 : SLEEP モード時、SPI5 ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SBSLPEN** : SLEEP モード時の SBS クロックイネーブル
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時、SBS ペリフェラルクロックは無効です。
1 : SLEEP モード時、SBS ペリフェラルクロックは有効です (リセット後のデフォルト)。

ビット 0 予約済みであり、リセット値に保持する必要があります。

11.8.40 RCC カーネルクロック設定レジスタ1 (RCC_CCIPR1)

アドレスオフセット : 0x0D8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMICSEL	Res.	USART10SEL [2:0]			UART9SEL [2:0]			UART8SEL [2:0]			UART7SEL [2:0]			USART6SEL [2:1]	
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART6SEL[0]	UART5SEL [2:0]			UART4SEL [2:0]			USART3SEL [2:0]			USART2SEL [2:0]			USART1SEL [2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **TIMICSEL** : TIM12、TIM15、および LPTIM2 入力キャプチャソース選択

ソフトウェアでセット/クリアされます。

0 : タイマ入力キャプチャに使用可能な内部クロックはありません (リセット後のデフォルト)。

1 : hsi_ker_ck/1024、hsi_ker_ck/8、および csi_ker_ck/128 がタイマ入力キャプチャ用に選択されます。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:27 **USART10SEL[2:0]** : USART10 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 26:24 **UART9SEL[2:0]** : UART9 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 23:21 **UART8SEL[2:0]** : UART8 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 20:18 UART7SEL[2:0] : UART7 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 17:15 USART6SEL[2:0] : USART6 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 14:12 UART5SEL[2:0] : UART5 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 11:9 UART4SEL[2:0] : UART4 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 8:6 USART3SEL[2:0] : USART3 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 5:3 USART2SEL[2:0] : USART2 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 2:0 **USART1SEL[2:0]** : USART1 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。

000 : rcc_pclk2 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

11.8.41 RCC カーネルクロック設定レジスタ2 (RCC_CCIPR2)

アドレスオフセット : 0x0DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LPTIM6SEL [2:0]			Res.	LPTIM5SEL [2:0]			Res.	LPTIM4SEL [2:0]			Res.	LPTIM3SEL [2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LPTIM2SEL [2:0]			Res.	LPTIM1SEL [2:0]			Res.	UART12SEL [2:0]			Res.	USART11SEL [2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **LPTIM6SEL[2:0]** : LPTIM6 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_p_ck がカーネルクロックとして選択されます。

010 : pll3_r_ck がカーネルクロックとして選択されます。

011 : lse_ker_ck がカーネルクロックとして選択されます。

100 : lsi_ker_ck がカーネルクロックとして選択されます。

101 : per_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **LPTIM5SEL[2:0]** : LPTIM5 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_p_ck がカーネルクロックとして選択されます。

010 : pll3_r_ck がカーネルクロックとして選択されます。

011 : lse_ker_ck がカーネルクロックとして選択されます。

100 : lsi_ker_ck がカーネルクロックとして選択されます。

101 : per_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **LPTIM4SEL[2:0]** : LPTIM4 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_r_ck がカーネルクロックとして選択されます。
011 : lse_ker_ck がカーネルクロックとして選択されます。
100 : lsi_ker_ck がカーネルクロックとして選択されます。
101 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **LPTIM3SEL[2:0]** : LPTIM3 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_r_ck がカーネルクロックとして選択されます。
011 : lse_ker_ck がカーネルクロックとして選択されます。
100 : lsi_ker_ck がカーネルクロックとして選択されます。
101 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **LPTIM2SEL[2:0]** : LPTIM2 カーネルクロックソースの選択

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_r_ck がカーネルクロックとして選択されます。
011 : lse_ker_ck がカーネルクロックとして選択されます。
100 : lsi_ker_ck がカーネルクロックとして選択されます。
101 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **LPTIM1SEL[2:0]** : LPTIM1 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_r_ck がカーネルクロックとして選択されます。
011 : lse_ker_ck がカーネルクロックとして選択されます。
100 : lsi_ker_ck がカーネルクロックとして選択されます。
101 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **UART12SEL[2:0]** : UART12 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : lse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **USART11SEL[2:0]** : USART11 カーネルクロックソースの選択

ソフトウェアでセット／クリアされます。

000 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

11.8.42 RCC カーネルクロック設定レジスタ3 (RCC_CCIPR3)

アドレスオフセット : 0x0E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	LPUART1SEL [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	SPI6SEL [2:1]	
					rw	rw	rw							rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI6SEL[0]	SPI5SEL [2:0]			SPI4SEL [2:0]			SPI3SEL [2:0]			SPI2SEL [2:0]			SPI1SEL [2:0]		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **LPUART1SEL[2:0]** : LPUART1 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : lse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:15 **SPI6SEL[2:0]** : SPI6 カーネルクロックソースの選択

000 : rcc_pclk2 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : hse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 14:12 **SPI5SEL[2:0]** : SPI5 カーネルクロックソースの選択

000 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。

001 : pll2_q_ck がカーネルクロックとして選択されます。

010 : pll3_q_ck がカーネルクロックとして選択されます。

011 : hsi_ker_ck がカーネルクロックとして選択されます。

100 : csi_ker_ck がカーネルクロックとして選択されます。

101 : hse_ck がカーネルクロックとして選択されます。

その他 : 予約済み、カーネルクロックは無効です。

ビット 11:9 **SPI4SEL[2:0]** : SPI4 カーネルクロックソースの選択

000 : rcc_pclk2 がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_q_ck がカーネルクロックとして選択されます。
010 : pll3_q_ck がカーネルクロックとして選択されます。
011 : hsi_ker_ck がカーネルクロックとして選択されます。
100 : csi_ker_ck がカーネルクロックとして選択されます。
101 : hse_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 8:6 **SPI3SEL[2:0]** : SPI3 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_p_ck がカーネルクロックとして選択されます。
011 : AUDIOCLK がカーネルクロックとして選択されます。
100 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 5:3 **SPI2SEL[2:0]** : SPI2 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_p_ck がカーネルクロックとして選択されます。
011 : AUDIOCLK がカーネルクロックとして選択されます。
100 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

ビット 2:0 **SPI1SEL[2:0]** : SPI1 カーネルクロックソースの選択

ソフトウェアでセット/クリアされます。
000 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
001 : pll2_p_ck がカーネルクロックとして選択されます。
010 : pll3_p_ck がカーネルクロックとして選択されます。
011 : AUDIOCLK がカーネルクロックとして選択されます。
100 : per_ck がカーネルクロックとして選択されます。
その他 : 予約済み、カーネルクロックは無効です。

11.8.43 RCC カーネルクロック設定レジスタ4 (RCC_CCIPR4)

アドレスオフセット : 0x0E4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	I3C1SEL [1:0]		I2C4SEL [1:0]		I2C3SEL [1:0]		I2C2SEL [1:0]		I2C1SEL [1:0]	
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMM C2SEL	SDMM C1SEL	USBSEL[1:0]		SYSTICKSEL[1:0]		OCTOSPI1SEL [1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **I3C1SEL[1:0]** : I3C1 カーネルクロックソースの選択

- 00 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll3_r_ck がカーネルクロックとして選択されます。
- 10 : hsi_ker_ck がカーネルクロックとして選択されます。
- 11 : 選択されたクロックがありません。

ビット 23:22 **I2C4SEL[1:0]** : I2C4 カーネルクロックソースの選択

- 00 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll3_r_ck がカーネルクロックとして選択されます。
- 10 : hsi_ker_ck がカーネルクロックとして選択されます。
- 11 : csi_ker_ck がカーネルクロックとして選択されます。

ビット 21:20 **I2C3SEL[1:0]** : I2C3 カーネルクロックソースの選択

- 00 : rcc_pclk3 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll3_r_ck がカーネルクロックとして選択されます。
- 10 : hsi_ker_ck がカーネルクロックとして選択されます。
- 11 : csi_ker_ck がカーネルクロックとして選択されます。

ビット 19:18 **I2C2SEL[1:0]** : I2C2 カーネルクロックソースの選択

- 00 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll3_r_ck がカーネルクロックとして選択されます。
- 10 : hsi_ker_ck がカーネルクロックとして選択されます。
- 11 : csi_ker_ck がカーネルクロックとして選択されます。

ビット 17:16 **I2C1SEL[1:0]** : I2C1 カーネルクロックソースの選択

- 00 : rcc_pclk1 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll3_r_ck がカーネルクロックとして選択されます。
- 10 : hsi_ker_ck がカーネルクロックとして選択されます。
- 11 : csi_ker_ck がカーネルクロックとして選択されます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **SDMMC2SEL** : SDMMC2 カーネルクロックソースの選択

- 0 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 1 : pll2_r_ck がカーネルクロックとして選択されます。

ビット 6 **SDMMC1SEL** : SDMMC1 カーネルクロックソースの選択

- 0 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 1 : pll2_r_ck がカーネルクロックとして選択されます。

ビット 5:4 **USBSEL[1:0]** : USB カーネルクロックソースの選択

- 00 : クロックはカーネルクロックとして選択されません (リセット後のデフォルト)。
- 01 : pll1_q_ck がカーネルクロックとして選択されます。
- 10 : pll3_q_ck がカーネルクロックとして選択されます。
- 11 : hsi48_ker_ck がカーネルクロックとして選択されます。

ビット 3:2 **SYSTICKSEL[1:0]** : SYSTICK クロックソース選択

- 00 : rcc_hclk/8 がクロックソースとして選択されます (リセット後のデフォルト)。
- 01 : lsi_ker_ck[1] がクロックソースとして選択されます。
- 10 : lse_ck[1] がクロックソースとして選択されます。
- 11 : 予約済み、カーネルクロックは無効です。

注 : rcc_hclk の周波数は、lsi_ker_ck/lse_ck の 4 倍以上高い必要があります (周期 (LSI/LSE) $\geq 4 * 周期 (HCLK)$)。

ビット 1:0 **OCTOSPI1SEL[1:0]** : OCTOSPI1 カーネルクロックソースの選択

- ソフトウェアでセット/クリアされます。
- 00 : rcc_hclk4 がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll1_q_ck がカーネルクロックとして選択されます。
- 10 : pll2_r_ck がカーネルクロックとして選択されます。
- 11 : per_ck がカーネルクロックとして選択されます。

11.8.44 RCC カーネルクロック設定レジスタ5 (RCC_CCIPR5)

アドレスオフセット : 0x0E8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKPERSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI2SEL [2:0]			SAI1SEL [2:0]		
rw	rw									rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FDCAN SEL[1:0]		CECSEL[1:0]		RNGSEL[1:0]		DACSEL	ADCDACSEL[2:0]		
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 **CKPERSEL[1:0]** : per_ck クロックソース選択

- 00 : hsi_ker_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : csi_ker_ck がカーネルクロックとして選択されます。
- 10 : hse_ck がカーネルクロックとして選択されます。
- 11 : 予約済み、per_ck クロックは無効です。

ビット 29:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:19 **SAI2SEL[2:0]** : SAI2 カーネルクロックソースの選択

- 000 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 001 : pll2_p_ck がカーネルクロックとして選択されます。
- 010 : pll3_p_ck がカーネルクロックとして選択されます。
- 011 : AUDIOCLK がカーネルクロックとして選択されます。
- 100 : per_ck がカーネルクロックとして選択されます。
- その他 : 予約済み、カーネルクロックは無効です。

ビット 18:16 **SAI1SEL[2:0]** : SAI1 カーネルクロックソースの選択

- 000 : pll1_q_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 001 : pll2_p_ck がカーネルクロックとして選択されます。
- 010 : pll3_p_ck がカーネルクロックとして選択されます。
- 011 : AUDIOCLK がカーネルクロックとして選択されます。
- 100 : per_ck がカーネルクロックとして選択されます。
- その他 : 予約済み、カーネルクロックは無効です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **FDCANSEL[1:0]** : FDCAN1 および FDCAN2 カーネルクロックソースの選択

- 00 : hse_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll1_q_ck がカーネルクロックとして選択されます。
- 10 : pll2_q_ck がカーネルクロックとして選択されます。
- 11 : 予約済み、カーネルクロックは無効です。

ビット 7:6 **CECSEL[1:0]** : HSMI-CEC カーネルクロックソースの選択

- 00 : lse_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : lsi_ker_ck がカーネルクロックとして選択されます。
- 10 : csi_ker_ck/122 がカーネルクロックとして選択されます。
- 11 : 予約済み、カーネルクロックは無効です。

ビット 5:4 **RNGSEL[1:0]** : RNG カーネルクロックソースの選択

- 00 : hsi48_ker_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 01 : pll1_q_ck がカーネルクロックとして選択されます。
- 10 : lse_ck がカーネルクロックとして選択されます。
- 11 : lsi_ker_ck がカーネルクロックとして選択されます。

ビット 3 **DACSEL** : DAC サンプルおよび保持クロック

- 0 : dac_hold_ck がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 1 : dac_hold_ck がカーネルクロックとして選択されます。

ビット 2:0 **ADCDACSEL[2:0]** : ADC および DAC カーネルクロックソース選択

- 000 : rcc_hclk がカーネルクロックとして選択されます (リセット後のデフォルト)。
- 001 : sys_ck がカーネルクロックとして選択されます。
- 010 : pll2_r_ck がカーネルクロックとして選択されます。
- 011 : hse_ck がカーネルクロックとして選択されます。
- 100 : hsi_ker_ck がカーネルクロックとして選択されます。
- 101 : csi_ker_ck がカーネルクロックとして選択されます。
- その他 : 予約済み、カーネルクロックは無効です。

11.8.45 RCC バックアップドメイン制御レジスタ (RCC_BDCR)

アドレスオフセット : 0x0F0

リセット値 : 0x0000 0000

バックアップドメインリセットによってリセットされます。

アクセス : $0 \leq \text{ウェイトステート} \leq 3$ 、ワード、ハーフワード、およびバイトアクセス。

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

注 : システムリセット後、RCC_BDCR レジスタは書き込み保護されます (ビット 27:24 およびビット 16 を除く)。バックアップドメインのビットを変更するには、[PWR backup domain control register 1 \(PWR_BDCR1\)](#) の DBP ビットを 1 にセットする必要があります。RCC_BDCR ビット (ビット 27:24 およびビット 16 を除く) は、バックアップドメインリセットにのみリセットされます ([セクション 11.3.3 : バックアップドメインリセット](#) を参照)。その他リセットは、これらのビットに影響しません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	LSI RDY	LSI オン	LSCO SEL	LSCO EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VSW RST
				rw	rw	rw	rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		LSE EXT	LSE CSSD	LSE CSSON	LSEDRV[1:0]		LSE BYP	LSE RDY	LSE オン
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **LSIRDY** : LSI オシレータはレディ状態です。

LSI オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSION ビットがクリアされた後、LSIRDY は、内部低速オシレータの 3 クロック後にローになります。このビットは、LSION = 0 の場合でも、LSI が IWDG または RTC によって使用されたときにセットされます。

0 : LSI オシレータはレディ状態ではありません。

1 : LSI オシレータはレディ状態です。

ビット 26 **LSION** : LSI オシレータイネーブル

ソフトウェアでセット/クリアされます。

0 : LSI オシレータはオフです。

1 : LSI オシレータはオンです。

ビット 25 **LSCOSEL** : ロースピードクロック出力選択

ソフトウェアでセット/クリアされます。

0 : LSI クロックの選択

1 : LSE クロックの選択

ビット 24 **LSCOEN** : ロースピードクロック出力 (LSCO) 選択

ソフトウェアでセット/クリアされます。

0 : LSCO 出力は無効です。

1 : LSCO 出力は有効です。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 VSWRST : VSwitch ドメインソフトウェアリセット

ソフトウェアでセット/クリアされます。

0 : リセットは有効ではありません (バックアップドメインリセット後のデフォルト)。

1 : VSW ドメイン全体をリセットします。

ビット 15 RTCEN : RTC クロック有効化

ソフトウェアでセット/クリアされます。

0 : rtc_ck は無効です (バックアップドメインリセット後のデフォルト)。

1 : rtc_ck は有効です。

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 RTCSEL[1:0] : RTC クロックソース選択

RTC のクロックソースを選択するために、ソフトウェアによってセットされます。

これらのビットは 1 度だけ書き込めます (LSE で障害が検出された場合を除く)。

これらのビットは、LSECSSON を有効にする前に書き込む必要があります。

これらのビットをリセットし、また 1 度だけ書き込めるようにするために、VSWRST ビットを使用できます。

HSE が RTC クロックとして選択されている場合 : このクロックは、システムが STOP モードの場合、またはピンリセット (NRST) 時に失われます。

00 : クロックなし (バックアップドメインリセット後のデフォルト)。

01 : LSE が RTC クロックとして選択されます。

10 : LSI が RTC クロックとして選択されます。

11 : RTCPRE の値で分周された HSE が RTC クロックとして選択されます。

ビット 7 LSEEXT : バイパスモードでの低速外部クロックタイプ

外部クロックタイプ (アナログまたはデジタル) を選択するために、ソフトウェアによってセット/リセットされます。

デバイスで使用するには、LSEON ビットで外部クロックを有効にする必要があります。

LSEEXT ビットは、LSE オシレータが無効のときのみ、書込みができます。

0 : LSE はアナログモードです (バックアップドメインリセット後のデフォルト)。

1 : LSE はデジタルモードです (RTC がアクティブな場合は使用しないでください)。

ビット 6 LSECSSD : LSE クロックセキュリティシステム障害の検出

外部 32kHz オシレータのクロックセキュリティシステムによって障害が検出されたことを示すために、ハードウェアによってセットされます。

0 : 32kHz オシレータに障害が検出されません (バックアップドメインリセット後のデフォルト)。

1 : 32kHz オシレータに障害が検出されました。

ビット 5 LSECSSON : LSE クロックセキュリティシステム有効化

32kHz オシレータのクロックセキュリティシステムを有効にするために、ソフトウェアによってセットされます。

LSECSSON は、LSE を有効 (LSEON を有効化) かつレディ状態 (ハードウェアで LSERDY をセット) にし、RTCSEL を選択した後で、有効にする必要があります。

このビットは、一度有効にすると、LSE の障害検出 (LSECSSD = 1) の後を除き、無効にすることはできません。この場合、ソフトウェアは LSECSSON を無効にする必要があります。

0 : 32kHz オシレータの CSS はオフです (バックアップドメインリセット後のデフォルト)。

1 : 32kHz オシレータの CSS はオンです。

ビット 4:3 **LSEDRV[1:0]** : LSE オシレータの駆動能力

LSE オシレータの駆動能力を選択するために、ソフトウェアによってセットされます。

これらのビットは、LSE オシレータが無効 (LSEON = 0 および LSERDY = 0) の場合のみ、書き込むことができます。

00 : 最低駆動 (バックアップドメインリセット後のデフォルト)

01 : 中低駆動

10 : 中高駆動

11 : 最高駆動

ビット 2 **LSEBYP** : LSE オシレータバイパス

デバッグモードのオシレータをバイパスするために、ソフトウェアによってセット/リセットされます。LSE が有効 (LSEON) またはレディ状態 (LSERDY = 1) のときは、このビットの書き込みを行わないでください。

0 : LSE オシレータはバイパスされません (バックアップドメインリセット後のデフォルト)。

1 : LSE オシレータはバイパスされます。

ビット 1 **LSERDY** : LSE オシレータはレディ状態です。

LSE が安定していることを示すために、ハードウェアによってセット/リセットされます。

このビットは、LSEON が 0 にセットされてから立ち下がるまで、lse_ck クロックが 6 サイクル必要です。

0 : LSE オシレータはレディ状態ではありません (バックアップドメインリセット後のデフォルト)。

1 : LSE オシレータはレディ状態です。

ビット 0 **LSEON** : LSE オシレータ有効

ソフトウェアでセット/クリアされます。

0 : LSE オシレータはオフです (バックアップドメインリセット後のデフォルト)。

1 : LSE オシレータオン

11.8.46 **RCC リセットステータスレジスタ (RCC_RSR)**

アドレスオフセット : 0x0F4

リセット値 : 0x0C00 0000

パワーオンリセットによってのみリセットされます。

アクセス : 0 ≤ ウェイトステート ≤ 3、ワード、ハーフワード、およびバイトアクセス。

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	BOR RSTF	PIN RSTF	Res.	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw			rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31 **LPWRRSTF** : 低電力リセットフラグ

STOP または STANDBY モードへの移行によりリセットが発生したときに、ハードウェアによってセットされ、対応する nRST_STOP、nRST_STBY オプションビットはクリアされます。

RMVF ビットに書き込むことによってクリアされます。

0 : 不正な低電力モードリセットは発生していません。

1 : 不正な低電力モードリセットが発生しました。

ビット 30 **WWDGRSTF** : ウィンドウウォッチドッグリセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってリセットされます。

ウィンドウ型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

0 : WWDG からウィンドウ型ウォッチドッグリセットは発生していません (パワーオンリセット後のデフォルト)。

1 : WWDG からウィンドウ型ウォッチドッグリセットが発生しました。

ビット 29 **IWDGRSTF** : 独立型ウォッチドッグリセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってリセットされます。

独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

0 : 独立型ウォッチドッグリセットは発生していません (パワーオンリセット後のデフォルト)。

1 : 独立型ウォッチドッグリセットが発生しました。

ビット 28 **SFTRSTF** : CPU からのシステムリセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってリセットされます。

CPU によってシステムがリセットされた場合に、ハードウェアによってセットされます。CPU ではコア M33 の AIRCR レジスタの SYSRESETREQ ビットを書き込むことで、システムリセットを生成できます。

0 : CPU のソフトウェアリセットは発生していません (パワーオンリセット後のデフォルト)。

1 : CPU によってシステムリセットが生成されます。

ビット 27 **BORRSTF** : BOR リセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってリセットされます。

BOR リセットが発生したときに、ハードウェアによってセットされます (pwr_bor_rst)。

0 : BOR リセットは発生していません。

1 : BOR リセットが発生しました (パワーオンリセット後のデフォルト)。

ビット 26 **PINRSTF** : ピンリセットフラグ (NRST)

RMVF ビットに書き込むことによりソフトウェアによってリセットされます。

ピンからリセットが発生したときに、ハードウェアによってセットされます。

0 : ピンからのリセットは発生していません。

1 : ピンからのリセットが発生しました (パワーオンリセット後のデフォルト)。

ビット 25:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **RMVF** : 削除リセットフラグ

リセットフラグの値をリセットするために、ソフトウェアによってセット/リセットされます。

0 : リセットフラグのリセットは有効になりません (パワーオンリセット後のデフォルト)。

1 : リセットフラグの値をリセットします。

ビット 22:0 予約済みであり、リセット値に保持する必要があります。

11.8.47 RCC セキュア設定レジスタ (RCC_SECCFGR)

アドレスオフセット : 0x110

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワードおよびバイトアクセス

システムがセキュアなとき (TZEN = 0xB4)、このレジスタは、SPRIV = 1 の場合はセキュア特権アクセスによってのみ書き込むことができ、SPRIV = 0 の場合はセキュア特権または非特権アクセスによって書き込むことができます。非セキュア書き込みアクセスは、不正アクセスイベントを生成し、データは書き込まれません。このレジスタは、セキュアまたは非セキュア、特権または非特権アクセスによって読み出すことができます。システムがセキュアでないとき (TZEN = 0xC3)、このレジスタは 0 として読み出され、レジスタ書き込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CKPERSELSEC	RMVFSEC	HSI48SEC	Res.	PLL3SEC	PLL2SEC	PLL1SEC	PRESEC	SYSCLKSEC	LSESEC	LSISEC	CSISEC	HSESEC	HSISEC
		rW	rW	rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CKPERSELSEC** : per_ck 選択セキュリティ

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

ビット 12 **RMVFSEC** : リセットフラグのセキュリティ解除

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

ビット 11 **HSI48SEC** : HSI48 クロック設定ビットとステータスビットのセキュリティ

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **PLL3SEC** : PLL3 クロック設定ビットとステータスビットのセキュリティ

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

ビット 8 **PLL2SEC** : PLL2 クロック設定ビットとステータスビットのセキュリティ

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

ビット 7 **PLL1SEC** : PLL1 クロック設定ビットとステータスビットのセキュリティ

ソフトウェアでセット/クリアされます。

0 : 非セキュア

1 : セキュア

- ビット 6 **PRESCSEC** : AHBx/APBx プリスケール設定ビットのセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 5 **SYSCLKSEC** : SYSCLK クロック選択、STOPWUCK ビット、MCO クロック出力設定のセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 4 **LSESEC** : LSE クロック設定ビットとステータスビットのセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 3 **LSISEC** : LSI クロック設定ビットとステータスビットのセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 2 **CSISEC** : CSI クロック設定ビットとステータスビットのセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 1 **HSESEC** : HSE クロック設定ビットとステータスビット、および HSE_CSS のセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア
- ビット 0 **HSISEC** : HSI クロック設定ビットとステータスビットのセキュリティ
ソフトウェアでセット/クリアされます。
0 : 非セキュア
1 : セキュア

11.8.48 RCC 特権設定レジスタ (RCC_PRIVCFGR)

アドレスオフセット : 0x114

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード;およびバイトアクセス

このレジスタは、特権アクセスによってのみ書き込むことができます。特権または非特権アクセスによって読み出すことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSPRIV	SPRIV
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **NSPRIV** : RCC 非セキュア機能の特権設定

ソフトウェアでセット/クリアされます。このビットは、セキュアまたは非セキュアの特権アクセスによってのみ書き込むことができます。

0 : RCC 非セキュア機能の読み書きは、特権または非特権アクセスによって行うことができます。

1 : RCC 非セキュア機能の読み書きは、特権アクセスによってのみ行うことができます。

ビット 0 **SPRIV** : RCC セキュア機能の特権設定

ソフトウェアでセット/クリアされます。このビットは、セキュア特権アクセスによってのみ書き込むことができます。

0 : RCC セキュア機能の読み書きは、特権または非特権アクセスによって行うことができます。

1 : RCC セキュア機能の読み書きは、特権アクセスによってのみ行うことができます。

11.9 RCC レジスタマップ

表 108. RCC レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	RCC_CR	Res.	Res.	PLL3RDY	PLL3ON	PLL2RDY	PLL2ON	PLL1RDY	PLL1ON	Res.	Res.	Res.	HSEEXT	HSECSSON	HSEBYP	HSERDY	HSEON	Res.	Res.	HSI48RDY	HSI48ON	Res.	Res.	CSIKERON	CSIRDY	CSION	Res.	Res.	Res.	HSIDIVF	HSIDIV[1:0]	HSIKERON	HSIRDY	HSION
	リセット値			0	0	0	0	0	0				0	0	0	0	0			0	0		0	0	0			1	0	1	0	1	1	
0x010	RCC_HSICTFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		HSITRIM[6:0]						Res.	Res.	Res.	Res.	HSICAL[11:0]												
	リセット値										1	0	0	0	0	0	0						X	X	X	X	X	X	X	X	X	X	X	
0x014	RCC_CRRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL [9:0]												
	リセット値																								X	X	X	X	X	X	X	X	X	
0x018	RCC_CSICFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSITRIM[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSICAL[7:0]						
	リセット値											1	0	0	0	0	0											X	X	X	X	X	X	
0x01C	RCC_CFGR1	MCO2SEL [2:0]			MCO2PRE [3:0]				MCO1SEL [2:0]			MCO1PRE [3:0]				Res.	Res.	TIMPRE			RTCPRE[5:0]				STOPKERWUCK	STOPWUCK	Res.	SWS[1:0]			Res.	SW[1:0]		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0	0	0	0	0	0		0	0	0	0	
0x020	RCC_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APB3DIS	APB2DIS	APB1DIS	AHB4DIS	Res.	AHB2DIS	AHB1DIS	Res.	PPRE3 [2:0]	PPRE2 [2:0]	PPRE1 [2:0]	PPRE1 [2:0]	PPRE1 [2:0]	PPRE1 [2:0]	PPRE1 [2:0]	Res.	PPRE1 [2:0]	PPRE1 [2:0]	HPRE[3:0]					
	リセット値										0	0	0	0		0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x028	RCC_PLL1CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL1REN	PLL1QEN	PLL1PEN	Res.	Res.	PLL1M [5:0]				Res.	Res.	Res.	Res.	PLL1VCOSEL	PLL1FRACEN	PLL1RGE [1:0]	PLL1SRC [1:0]			
	リセット値														0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x02C	RCC_PLL2CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL2REN	PLL2QEN	PLL2PEN	Res.	Res.	PLL2M [5:0]				Res.	Res.	Res.	Res.	PLL2VCOSEL	PLL2FRACEN	PLL2RGE [1:0]	PLL2SRC [1:0]			
	リセット値														0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x030	RCC_PLL3CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3REN	PLL3QEN	PLL3PEN	Res.	Res.	PLL3M [5:0]				Res.	Res.	Res.	Res.	PLL3VCOSEL	PLL3FRACEN	PLL3RGE [1:0]	PLL3SRC [1:0]			
	リセット値														0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x034	RCC_PLL1DIVR	Res.	PLL1R [6:0]							Res.	PLL1Q [6:0]							PLL1P [6:0]				PLL1N [8:0]												
	リセット値		0	0	0	0	0	0	1		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

表 108. RCC レジスタのマッピングとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x038	RCC_PLL1FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL1FRACN [12:0]												Res.	Res.	Res.		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x03C	RCC_PLL2DIVR	Res.	PLL2R [6:0]						Res.	PLL2Q [6:0]						PLL2P [6:0]						PLL2N [8:0]												
	リセット値		0	0	0	0	0	0	1		0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
0x040	RCC_PLL2FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL2FRACN [12:0]												Res.	Res.	Res.		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x044	RCC_PLL3DIVR	Res.	PLL3R [6:0]						Res.	PLL3Q [6:0]						PLL3P [6:0]						PLL3N [8:0]												
	リセット値		0	0	0	0	0	1		0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
0x048	RCC_PLL3FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLL3FRACN [12:0]												Res.	Res.	Res.		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x050	RCC_CIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x054	RCC_CIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECSSF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																						0											
0x058	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECSSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																						0											
0x060	RCC_AHB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x064	RCC_AHB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	

表 108. RCC レジスタのマッピングとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x06C	RCC_AHB4RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSPI1RST	Res.	Res.	Res.	FMCRST	Res.	Res.	Res.	SDMMC2RST	SDMMC1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値												0				0				0	0	Res.	Res.	Res.	Res.	0								
0x074	RCC_APB1LRSTR	UART8RST	UART7RST	Res.	CECRST	USART11	USART10RST	USART9RST	CRSRST	I3C1RST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	Res.	SPI3RST	SPI2RST	Res.	Res.	Res.	Res.	Res.	Res.	TIM14RST	TIM13RST	TIM12RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST	
	リセット値	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0	0	0		Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	
0x078	RCC_APB1HRSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	Res.	Res.	LPTIM2RST	DTSRST	Res.	UART12RST	UART9RST		
	リセット値									0													0					0				0	0	0	
0x07C	RCC_APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBRST	Res.	SAI2RST	SAI1RST	SPI6RST	SPI4RST	TIM17RST	TIM16RST	TIM15RST	Res.	USART1RST	TIM8RST	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値								0		0	0	0	0	0	0	0			0	0	0	0	Res.	Res.	Res.	Res.						0	0	
0x080	RCC_APB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VREFRST	Res.	Res.	Res.	Res.	LPTIM6RST	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM1RST	Res.	Res.	Res.	Res.	I2C4RST	I2C3RST	LPUART1RST	SPI9RST	Res.	Res.	Res.		
	リセット値												0					0	0	0	0	0	Res.	Res.	Res.	0	0	0							
0x088	RCC_AHB1ENR	SRAM1EN	DCACHEEN	Res.	BKPRAMEN	Res.	Res.	Res.	TZSC1EN	Res.	Res.	ETHRXEN	ETHTXEN	ETHEN	Res.	RAMCFGEN	Res.	FMACEN	CORDICEN	Res.	CRCEEN	Res.	Res.	Res.	Res.	FLITFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	1	1		1				0			0	0	0		0		0	0		0	0				1						0	0	0	
0x08C	RCC_AHB2ENR	SRAM3EN	SRAM2EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAESEN	PKAEN	RNGEN	HASHEN	AESEN	Res.	Res.	Res.	DCMI_PSSIEN	DAC1EN	ADCEN	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN	
	リセット値	1	1										0	0	0	0	0				0	0	0			0	0	0	0	0	0	0	0	0	
0x094	RCC_AHB4ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSPI1EN	Res.	Res.	Res.	FMCEEN	Res.	Res.	Res.	SDMMC2EN	SDMMC1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値												0				0				0	0					0								
0x09C	RCC_APB1LENR	UART8EN	UART7EN	Res.	CECEN	USART11	USART10EN	USART6EN	CRSEN	I3C1EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	Res.	SPI3EN	SPI2EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN	
	リセット値	0	0		0	0	0	0	0	0	0	0	0	0	0	0		0	0			0	0			0	0	0	0	0	0	0	0	0	
0x0A0	RCC_APB1HENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	Res.	Res.	LPTIM2EN	DTSEN	Res.	UART12EN	UART9EN	
	リセット値									0														0					0				0	0	0
0x0A4	RCC_APB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBRST	Res.	SAI2EN	SAI1EN	SPI6EN	SPI4EN	TIM17EN	TIM16EN	TIM15EN	Res.	USART1EN	TIM8EN	SPI1EN	TIM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																		

表 108. RCC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	リセット値								0		0	0	0	0	0	0	0			0	0	0	0									0			
0x0A8	RCC_APB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCAPBEN	VREFEN	Res.	Res.	Res.	Res.	LPTIM6EN	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM1EN	Res.	Res.	Res.	I2C4EN	I2C3EN	LPUART1EN	SPI5EN	Res.	Res.	Res.	SBSSEN	Res.	
	リセット値											0	0					0	0	0	0	0				0	0	0	0				0		
0x0B0	RCC_AHB1LPENR	SRAM1LPEN	DCACHELPEN	ICACHELPEN	BKPRAMLPEN	Res.	Res.	Res.	TZSC1LPEN	Res.	Res.	ETHRXLPEN	ETHTXLPEN	ETHLPEN	Res.	Res.	Res.	FMACLPEN	CORDICLPEN	LPTIM4EN	CRCLPEN	Res.	Res.	Res.	Res.	FLITFLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	1	1	1	1				1			1	1	1		1		1	1		1					1							1	1	
0x0B4	RCC_AHB2LPENR	SRAM3LPEN	SRAM2LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAESLPEN	PKALPEN	RNGLPEN	HASHLPEN	AESLPEN	Res.	Res.	Res.	Res.	DCMI_PSSILPEN	DAC1LPEN	ADCLPEN	Res.	Res.	GPIOLPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCPLPEN	GPIOBLPEN	GPIOALPEN	Res.
	リセット値	1	1										1	1	1	1	1				1	1	1			1	1	1	1	1	1	1	1	1	1
0x0BC	RCC_AHB4LPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCTOSP11LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2LPEN	SDMMC1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											1									1	1					1								
0x0C4	RCC_APB1LLPENR	UART8LPEN	UART7LPEN	Res.	CECLPEN	USART11LPEN	USART10LPEN	USART6LPEN	CRSLPEN	I3C1LPEN	I2C2LPEN	I2C1LPEN	UART5LPEN	UART4LPEN	USART3LPEN	USART2LPEN	Res.	SPI3LPEN	SPI2LPEN	Res.	Res.	Res.	Res.	Res.	Res.	TIM14LPEN	TIM13LPEN	TIM12LPEN	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN	Res.
	リセット値	1	1		1	1	1	1	1	1	1	1	1	1	1	1		1	1							1	1	1	1	1	1	1	1	1	1
0x0C8	RCC_APB1HLPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCPD1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値									1															1										
0x0CC	RCC_APB2LPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USBLPEN	Res.	SAI2LPEN	SAI1LPEN	SPI6LPEN	SPI4LPEN	TIM17LPEN	TIM16LPEN	TIM15LPEN	Res.	USART1LPEN	TIM8LPEN	SPI1LPEN	TIM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値								1		1	1	1	1	1	1	1		1	1	1	1													
0x0D0	RCC_APB3LPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTCAPBLPEN	VREFLPEN	Res.	Res.	Res.	Res.	LPTIM6LPEN	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM1LPEN	Res.	Res.	Res.	I2C4LPEN	I2C3LPEN	LPUART1LPEN	SPI6LPEN	Res.	Res.	Res.	SBSLPEN	Res.	
	リセット値											1	1					1	1	1	1	1				1	1	1	1				1		
0x0D8	RCC_CCIPR1	TIMICSEL	Res.	USART10SEL [2:0]		USART9SEL [2:0]		USART8SEL [2:0]		USART7SEL [2:0]		USART6SEL [2:0]		USART5SEL [2:0]		USART4SEL [2:0]		USART3SEL [2:0]		USART2SEL [2:0]		USART1SEL [2:0]													
	リセット値	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 108. RCC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x0DC	RCC_CCIPR2	Res	LPTIM6SEL [2:0]			Res	LPTIM5SEL [2:0]			Res	LPTIM4SEL [2:0]			Res	LPTIM3SEL [2:0]			Res	LPTIM2SEL [2:0]			Res	LPTIM1SEL [2:0]			Res	UART12SEL [2:0]			Res	USART11SEL [2:0]				
	リセット値		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		
0x0E0	RCC_CCIPR3	Res	Res	Res	Res	Res	LPUART1SEL [2:0]			Res	Res	Res	Res	Res	SPI6SEL [2:0]			Res	SPI5SEL [2:0]			Res	SPI4SEL [2:0]			Res	SPI3SEL [2:0]			Res	SPI2SEL [2:0]				
	リセット値						0	0	0						0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		
0x0E4	RCC_CCIPR4	Res	Res	Res	Res	Res	Res	I3C1SEL [1:0]			I2C4SEL [1:0]			I2C3SEL [1:0]			I2C2SEL [1:0]			I2C1SEL [1:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		
	リセット値							0	0		0	0	0	0	0	0	0	0		0	0	0					0	0	0	0	0	0	0	0	
0x0E8	RCC_CCIPR5	CKPERSEL [1:0]			Res	Res	Res	Res	Res	Res	Res	SAI2SEL [2:0]			SAI1SEL [2:0]			Res	Res	Res	Res	Res	Res	FDCANSEL [1:0]			CECSEL [1:0]	Res	RNGSEL [1:0]			DACSEL			
	リセット値	0	0									0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	
0x0F0	RCC_BDCR	Res	Res	Res	Res	LSIRDY	LSION	LSCOESEL	LSCOEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RTCSEL[1:0]			LSEEXT	LSECSSD	LSECSSON	LSEDRV[1:0]			LSEBYP	LSESDY	LSEON
	リセット値					0	0	0	0								0	0						0	0	0	0	0	0	0	0	0	0	0	
0x0F4	RCC_RSR	LPWRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	BORRSTF	PINRSTF	Res	Res	RMVF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値	0	0	0	0	1	1			0																									
0x110	RCC_SECCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																		0	0	0			0	0	0	0	0	0	0	0	0	0	0	
0x114	RCC_PRIVCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																			0	0	0		0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

12 クロックリカバリシステム (CRS)

12.1 概要

クロックリカバリシステム (CRS) は、高精度トリミング可能な内部 RC オシレータ HSI48 に作用する、高度なデジタルコントローラです。CRS は、選択可能な同期信号との比較に基づいてオシレータ出力周波数を評価するための強力な手段を提供しています。CRS は、測定した周波数の誤差値に基づいて自動トリミング調整をおこなうことができ、手動トリミングも可能です。

CRS は、USB ペリフェラルに高精度なクロックを提供する場合に最適です。この場合、USB ホストによって 1 ms 間隔で送信される USB バス上のフレーム開始 (SOF) パケット信号から同期信号を取得できます。

同期信号は、LSE オシレータ出力から派生することができ、またはユーザソフトウェアによって生成することもできます。

12.2 CRS の主な機能

- 次のプログラム可能なプリスケールと極性を持つ、選択可能な同期ソース :
 - LSE オシレータ出力
 - USB SOF パケット受信
- ソフトウェアによって同期パルスを生成する可能性
- CPU 不要の自動オシレータトリミング機能
- より迅速なスタートアップ収束のための手動制御オプション
- 自動誤差値のキャプチャと再ロードが可能な、16 ビットの周波数誤差カウンタ
- 自動周波数誤差値の評価とステータスレポートの、プログラム可能な制限値
- マスク可能な割込み/イベント :
 - 期待される同期 (ESYNC)
 - 同期 OK (SYNCOK)
 - 同期警告 (SYNCWARN)
 - 同期またはトリミングエラー (ERR)

12.3 CRS の実装

表 109. CRS の機能

機能	CRS1
TRIM 幅	6 ビット

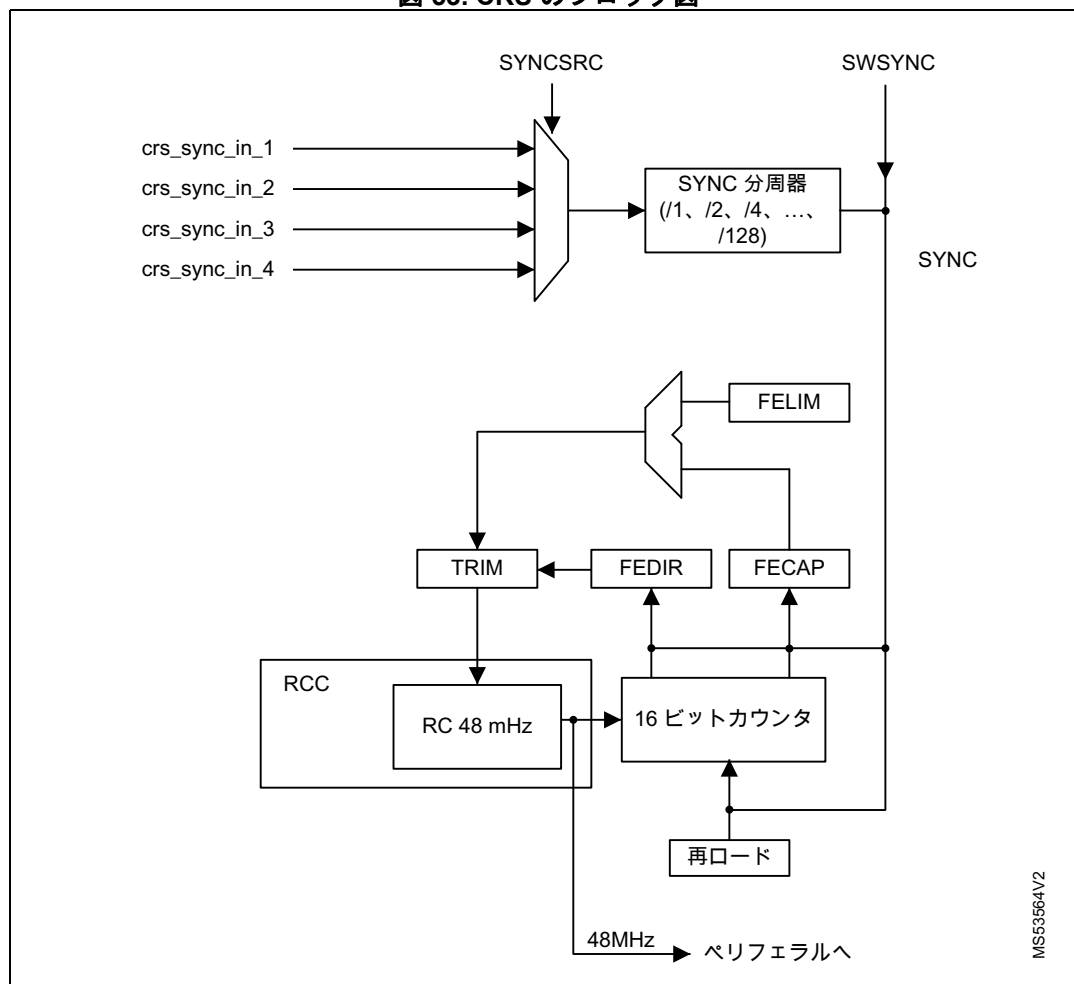
表 110. CRS 内部入力/出力信号

内部信号名	信号タイプ	説明
crs_sync_in_1	入力	00 : GPIO AF が SYNC 信号ソースとして選択されます。
crs_sync_in_2	入力	01 : SYNC 信号ソースとして LSE を選択します。
crs_sync_in_3	入力	10 : USB SOF が SYNC 信号ソースとして選択されます (デフォルト)。
crs_sync_in_4	入力	11 : 予約済み

12.4 CRS の機能説明

12.4.1 CRS ブロック図

図 53. CRS のブロック図



12.4.2 同期入力

CRS_CFGR レジスタから選択可能な CRS 同期 (SYNC) ソースは、LSE クロックからの信号または USB SOF 信号です。より安定した SYNC 入力を得るために、グリッチをフィルタするために、シンプルなデジタルフィルタ (3 対 2 の多数決方式、HSI48 クロックによるサンプリング方法) が実装されています。このソース信号には設定可能な極性があり、プログラム可能なバイナリプリスケアラによって分周して、適切な周波数範囲 (通常 1 kHz 前後) の同期信号を得ることができます。

CRS 同期ソース設定の詳細については、[セクション 12.7.2](#) を参照してください。

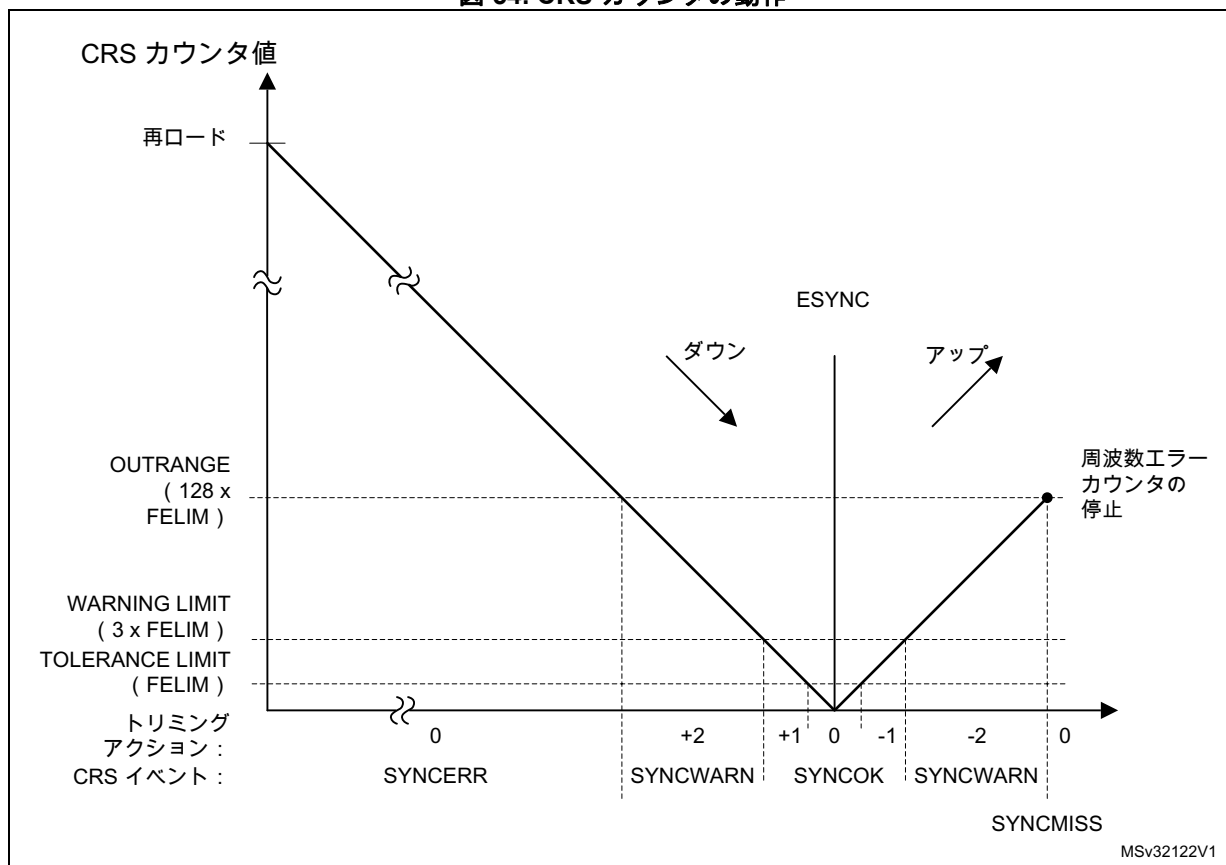
CRS_CR レジスタの SWSYNC ビットをセットして、ソフトウェアで同期イベントを生成することもできます。

12.4.3 周波数誤差測定

この周波数誤差カウンタは、SYNC イベントごとに RELOAD 値が再ロードされる 16 ビットのダウン/アップカウンタです。0 値に達するまでカウントダウンし、0 値に達すると、ESYNC（期待される同期）イベントが生成されます。次に、OUTRANGE 制限までカウントアップして、最終的に停止し（SYNC イベントが受信されなかった場合）、SYNCMISS イベントを生成します。OUTRANGE 制限は、周波数誤差制限（CRS_CFGR レジスタの FELIM フィールド）に 128 を乗算した値として定義されます。

SYNC イベントが検出されると、周波数誤差カウンタの実際の値とそのカウント方向が FECAP（周波数誤差キャプチャ）フィールドと CRS_ISR レジスタの FEDIR（周波数誤差方向）ビットに保管されます。カウントダウンフェーズ中に（0 値に達する前に）SYNC イベントが検出されると、実際の周波数がターゲットより低いことを意味します（TRIM 値を増やす必要があります）。カウントアップフェーズ中に検出されると、実際の周波数が高いことを意味します（TRIM 値を減らす必要があります）。

図 54. CRS カウンタの動作



MSv32122V1

12.4.4 周波数誤差評価および自動トリミング

測定した周波数誤差は、その値といくつかの制限を比較して評価されます。

- CRS_CFGR レジスタの FELIM フィールドに直接与えられる TOLERANCE LIMIT
- WARNING LIMIT。3 × FELIM 値として定義されます。
- OUTRANGE (エラー制限)。128 × FELIM 値として定義されます。

この比較の結果は、ステータスの表示を生成し、CRS_CR レジスタの AUTOTRIMEN ビットをセットすることで有効化される自動トリミングを制御するために使用されます。

- 周波数誤差が許容制限を下回る場合は、TRIM フィールドの実際のトリミング値が適切であり、トリミング動作が不要であることを意味します。
 - SYNCOK ステータスが示されます。
 - TRIM 値は、AUTOTRIM モードでは変更されません。
- 周波数誤差が警告制限を下回り、許容制限を上回るか同じである場合は、なんらかのトリミング動作が必要であることを意味します。ただし、1 回のトリミングステップによる調整で、十分適切な TRIM 値に達することができます。
 - SYNCOK ステータスが示されます。
 - AUTOTRIM モードで、1 回のトリミングステップによって TRIM 値が調整されます。
- 周波数誤差が警告制限を上回るか同じであり、エラー制限を下回る場合は、より強力なトリミング動作が必要であり、次の周期までに適切な TRIM 値に達しないリスクがあります。
 - SYNCWARN ステータスが示されます。
 - AUTOTRIM モードで、2 回のトリミングステップによって TRIM 値が調整されます。
- 周波数誤差が誤差制限を上回るか同じである場合、周波数はトリミング範囲外です。これは、SYNC 入力がクリーンでないとき、または、一部の SYNC パルスが欠落しているときにも発生する可能性があります (たとえば、1 つの USB SOF が破損しているとき)。
 - SYNCERR または SYNCMISS ステータスが示されます。
 - TRIM 値は、AUTOTRIM モードでは変更されません。

注 : TRIM フィールドの実際の値が制限値に近く、自動トリミングによってオーバーフローまたはアンダーフローが強制される場合、TRIM 値は制限に設定され、TRIMOVF ステータスが示されます。

AUTOTRIM モード (CRS_CR レジスタの AUTOTRIMEN ビットがセット) では、CRS_CR の TRIM フィールドがハードウェアによって調整され、読み出し専用になります。

12.4.5 CRS の初期化と設定

RELOAD 値

再ロード値は、ターゲット周波数とプリスケール後の同期ソースの周波数間の比率に従って選択する必要があります。この値は 1 だけ減分され、0 値で期待される同期に達します。計算式は、次のとおりです。

$$\text{RELOAD} = (f_{\text{TARGET}} / f_{\text{SYNC}}) - 1$$

RELOAD フィールドのリセット値は、48 MHz のターゲット周波数と 1 kHz の同期信号周波数 (USB からの SOF 信号) に対応します。

FELIM 値

FELIM 値の選択は、HSI48 オシレータ特性とその標準的なトリミングステップサイズに密接に関係しています。最適値は、オシレータクロック数を表すトリミングステップサイズの半分の値に対応します。次の計算式を使用できます。

$$\text{FELIM} = (f_{\text{TARGET}} / f_{\text{SYNC}}) * \text{STEP}[\%] / 100\% / 2$$

最高のトリム応答を取得するために、結果の小数点以下の値は常に切り上げられる必要があります。アプリケーション内でトリミング動作を頻繁に行う必要がない場合は、FELIM 値を少し上げることでヒステリシスを増やすことができます。

FELIM フィールドのリセット値は、 $(f_{\text{TARGET}} / f_{\text{SYNC}}) = 48000$ と、標準的なトリミングステップサイズ (0.14%) に対応しています。

注： トリミングステップサイズは、製品によって異なります。正確な設定については、データシートを確認してください。

注意： RELOAD および FELIM フィールドの誤った設定に対するハードウェアの保護はなく、不規則なトリム応答を引き起こす可能性があります。期待される動作モードでは、再ロード値を（同期ソース周波数に対して）適切に設定することを求めます。この値は、 $128 * \text{FELIM}$ 値よりも大きい値です（OUTRANGE 制限）。

12.5 CRS 低電力モード

表 111. 低電力モードが CRS に与える影響

モード	説明
SLEEP	影響はありません。CRS 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	CRS レジスタを停止します。CRS は、STOP モードが終了し、HSI48 オシレータがリスタートするまで、動作を停止します。
STANDBY	CRS ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

12.6 CRS 割込み

表 112. 割込み制御ビット

割込みイベント	イベントフラグ	有効制御ビット	フラグビットのクリア
期待される同期	ESYNCF	ESYNCE	ESYNCC
同期 OK	SYNCOKF	SYNCOKIE	SYNCOKC
同期警告	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同期またはトリミングエラー (TRIMOVF、SYNCMISS、SYNCERR)	ERRF	ERRIE	ERRC

12.7 CRS レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ワード（32 ビット）単位のみでアクセスすることができます。

12.7.1 CRS 制御レジスタ（CRS_CR）

アドレスオフセット：0x00

リセット値：0x0000 2000

リセット値：0x0000 4000（7 ビット TRIM 幅をサポートする製品）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRIM[5:0]						SW SYNC	AUTO TRIMEN	CEN	Res.	ESYNCI E	ERRIE	SYNC WARNIE	SYNC OKIE
		rw	rw	rw	rw	rw	rw	rt_w1	rw	rw		rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **TRIM[5:0]**：HSI48 オシレータ高精度トリミング

これらのビットは、HSI48 オシレータにユーザプログラム可能なトリミング値を提供します。これらをプログラムして、オシレータの周波数に影響する電圧や温度の変動に対応できるように調整できます。デフォルト値は 32（トリミング間隔の中間に相当）です。トリミングステップは製品データシートに明記されています。より高い TRIM 値は、より高い出力周波数に対応します。

AUTOTRIMEN ビットがセットされると、このフィールドはハードウェアによって調整され、読み専用になります。

ビット 7 **SWSYNC**：ソフトウェア SYNC イベントの生成

このビットは、ソフトウェア SYNC イベントを生成するためにソフトウェアによってセットされます。ハードウェアによって自動的にクリアされます。

0：影響なし。

1：ソフトウェア SYNC イベントが生成されます。

ビット 6 **AUTOTRIMEN**：自動トリミング有効

このビットは、2 つの SYNC イベント間で測定した周波数誤差に応じて、TRIM ビットの自動ハードウェア調整を有効にします。このビットがセットされると、TRIM ビットが読み専用になります。TRIM 値は、測定した周波数誤差値に応じて、ハードウェアで 1 度に 1 または 2 ステップずつ調整できます。詳細については、[セクション 12.4.4](#)を参照してください。

0：自動トリミングは無効です。ユーザは、TRIM ビットを調整できます。

1：自動トリミングは有効です。TRIM ビットは読み専用で、ハードウェアによって制御されます。

ビット 5 **CEN**：周波数誤差カウンタ有効

このビットは、周波数誤差カウンタのオシレータクロックを有効にします。

0：周波数誤差カウンタは無効です。

1：周波数誤差カウンタは有効です。

このビットをセットすると、CRS_CFGR レジスタは書き込み保護され、変更できません。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ESYNCE**：期待される SYNC 割込み有効

0：期待される SYNC（ESYNCF）割込みは無効です。

1：期待される SYNC（ESYNCF）割込みは有効です。

ビット 2 **ERRIE** : 同期またはトリミングエラー割込み有効

- 0 : 同期またはトリミングエラー (ERRF) 割込みは無効です。
- 1 : 同期またはトリミングエラー (ERRF) 割込みは有効です。

ビット 1 **SYNCWARNIE** : SYNC 警告割込み有効

- 0 : SYNC 警告 (SYNCWARNF) 割込みは無効です。
- 1 : SYNC 警告 (SYNCWARNF) 割込みは有効です。

ビット 0 **SYNCOKIE** : SYNC イベント OK 割込み有効

- 0 : SYNC イベント OK (SYNCOKF) 割込みは無効です。
- 1 : SYNC イベント OK (SYNCOKF) 割込みは有効です。

12.7.2 CRS 設定レジスタ (CRS_CFGR)

このレジスタに書き込めるのは、周波数誤差カウンタが無効なときだけです (CEN ビットは CRS_CR でクリアされる)。このカウンタを有効にすると、このレジスタは書き込み保護されます。

アドレスオフセット : 0x04

リセット値 : 0x2022 : BB7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNCPOL	Res	SYNCSRC[1:0]		Res	SYNCDIV[2:0]			FELIM[7:0]							
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **SYNCPOL** : SYNC 極性選択

- このビットは、SYNC 信号ソースの入力の極性を選択するために、ソフトウェアによってセット／クリアされます。
- 0 : SYNC は立上がりエッジでアクティブです (デフォルト)。
 - 1 : SYNC は立下がりエッジでアクティブです。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **SYNCSRC[1:0]** : SYNC 信号ソース選択

これらのビットは、SYNC 信号ソースを選択するために、ソフトウェアによってセットおよびクリアされます (表 110 を参照)。

- 00 : :crs_sync_in_1 が SYNC 信号ソースとして選択されます。
- 01 : :crs_sync_in_2 が SYNC 信号ソースとして選択されます。
- 10 : :crs_sync_in_3 が SYNC 信号ソースとして選択されます。
- 11 : :crs_sync_in_4 が SYNC 信号ソースとして選択されます。

注 : **USB LPM (Link Power Management、リンク電源管理)** の使用時にデバイスが **SLEEP モード** である場合、周期的 USB SOF はホストから生成されません。そのため、実行中に HSI48 オシレータを較正するために SYNC 信号が CRS に供給されることはありません。SLEEP モードからのウェイクアップ後に必要なクロック精度を確保するために、GPIO の LSE または リファレンスクロックを SYNC 信号として使用する必要があります。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **SYNCDIV[2:0]** : SYNC 分周器

これらのビットは、SYNC 信号の分周比を制御するために、ソフトウェアによってセット／クリアされます。

000 : SYNC は分周されません (デフォルト)。

001 : SYNC は 2 分周されます。

010 : SYNC は 4 分周されます。

011 : SYNC は 8 分周されます。

100 : SYNC は 16 分周されます。

101 : SYNC は 32 分周されます。

110 : SYNC は 64 分周されます。

111 : SYNC は 128 分周されます。

ビット 23:16 **FELIM[7:0]** : 周波数誤差制限

FELIM には、CRS_ISR レジスタの FECAP[15:0] ビットにラッチされた、キャプチャされた周波数誤差値を評価するために使用する値が含まれます。FECAP 評価の詳細については、[セクション 12.4.4](#) を参照してください。

ビット 15:0 **RELOAD[15:0]** : カウンタ再ロード値

RELOAD は、SYNC イベントごとに周波数誤差カウンタにロードされる値です。

カウンタの動作の詳細については、[セクション 12.4.3](#) を参照してください。

12.7.3 CRS 割込みおよびステータスレジスタ (CRS_ISR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FECAP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.	Res.	Res.	Res.	TRIM OVF	SYNC MISS	SYNC ERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNC WARNF	SYNC OKF
r					r	r	r					r	r	r	r

ビット 31:16 **FECAP[15:0]** : 周波数誤差キャプチャ

FECAP は、最後の SYNC イベント時にラッチされた、周波数誤差カウンタ値です。

FECAP の使用の詳細については、[セクション 12.4.4](#) を参照してください。

ビット 15 **FEDIR** : 周波数誤差方向

FEDIR は、最後の SYNC イベント時にラッチされた、周波数誤差カウンタのカウント方向です。実際の周波数が、ターゲットを下回るか上回るかを示します。

0 : アップカウント方向。実際の周波数はターゲットを上回ります。

1 : ダウンカウント方向。実際の周波数はターゲットを下回ります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **TRIMOVF** : トリミングのオーバーフローまたはアンダーフロー

このフラグは、自動トリミングによって TRIM 値のオーバーフローまたはアンダーフローが試みられた場合に、ハードウェアによってセットされます。CRS_CR レジスタの ERRIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : トリミングエラーは通知されていません。

1 : トリミングエラーが通知されました。

ビット 9 SYNCMISS : SYNC の欠落

このフラグは、周波数誤差カウンタが値 FELIM * 128 に達し、SYNC が検出されない場合にハードウェアによってセットされます。これは、SYNC パルスが欠落しているか、周波数誤差が大きすぎて（内部周波数が高すぎる）、TRIM 値を調整しても補正できないため、他の操作を行う必要があることを意味します。この時点で、周波数誤差カウンタは停止し（次の SYNC を待つ）、CRS_CR レジスタの ERRIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC 欠落エラーは通知されていません。

1 : SYNC 欠落エラーが通知されました。

ビット 8 SYNCERR : SYNC エラー

このフラグは、SYNC パルスが ESYNC イベントより前に発生し、測定した周波数誤差が FELIM * 128 以上である場合に、ハードウェアによってセットされます。これは、TRIM 値を調整して補正するには周波数誤差が大きすぎ（内部周波数が低すぎる）、他の操作を行う必要があるということを意味します。CRS_CR レジスタの ERRIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC エラーは通知されていません。

1 : SYNC エラーが通知されました。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 ESYNCF : 期待される SYNC フラグ

このフラグは、周波数誤差カウンタがゼロ値に達したときに、ハードウェアによってセットされます。CRS_CR レジスタの ESYNCIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの ESYNCC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : 期待された SYNC が通知されていません。

1 : 期待された SYNC が通知されました。

ビット 2 ERRF : エラーフラグ

このフラグは、同期またはトリミングエラー時に、ハードウェアによってセットされます。TRIMOVF、SYNCMISS、および SYNCERR ビットの論理和です。CRS_CR レジスタの ERRIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることへの反応として、ソフトウェアによってクリアされます。TRIMOVF、SYNCMISS、および SYNCERR ビットをクリアします。

0 : 同期またはトリミングエラーは通知されていません。

1 : 同期またはトリミングエラーが通知されました。

ビット 1 SYNCWARNF : SYNC 警告フラグ

このフラグは、測定した周波数誤差が FELIM * 3 以上で FELIM * 128 未満である場合に、ハードウェアによってセットされます。周波数誤差を補正するには、TRIM 値を複数のステップで調整する必要があります。CRS_CR レジスタの SYNCWARNIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの SYNCWARNC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC 警告は通知されていません。

1 : SYNC 警告が通知されました。

ビット 0 SYNCOKF : SYNC イベント OK フラグ

このフラグは、測定した周波数誤差が FELIM * 3 未満である場合に、ハードウェアによってセットされます。これは、TRIM 値の補正が必要であること、または周波数誤差の補正は 1 つのトリミングステップで十分であることを意味します。CRS_CR レジスタの SYNCOKIE ビットがセットされている場合、割込みが生成されます。CRS_ICR レジスタの SYNCOKC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC イベント OK は通知されていません。

1 : SYNC イベント OK が通知されました。

12.7.4 CRS 割込みフラグクリアレジスタ (CRS_ICR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNC WARNC	SYNC OKC
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ESYNCC** : 期待 SYNC クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの ESYNCF フラグをクリアします。

ビット 2 **ERRC** : エラークリアフラグ

このビットに 1 を書き込むと、TRIMOVF、SYNCMISS、および SYNCERR ビットがクリアされ、結果として CRS_ISR レジスタの ERRF フラグもクリアされます。

ビット 1 **SYNCWARNC** : SYNC 警告クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの SYNCWARNF フラグをクリアします。

ビット 0 **SYNCOKC** : SYNC イベント OK クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの SYNCOKF フラグをクリアします。

12.7.5 CRS レジスタマップ

表 113. CRS レジスタのマップとリセット値

オフ	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	CRS_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					SWSYNC		AUTOTRIMEN	CEN	Res.	ESYNCE	ERRIE	SYNCWARNIE	SYNCOKIE		
	リセット値																			1	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	CRS_CFGR	SYNCPOL	Res	SYNC SRC [1:0]		Res	SYNC DIV [2:0]		FELIM[7:0]							RELOAD[15:0]																			
	リセット値	0		1	0		0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	1		
0x08	CRS_ISR	FECAP[15:0]																FEDIR	Res.	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0					0	0	0	0	
0x0C	CRS_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC		
	リセット値																													0	0	0	0		

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

13 汎用 I/O (GPIO)

13.1 概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx_IDR、GPIOx_ODR)、1 つの 16 ビットリセットレジスタ (GPIOx_BRR)、および 1 つの 32 ビットセット/リセットレジスタ (GPIOx_BSRR) があります。

さらに、すべての GPIO には、それぞれ 32 ビットロックレジスタ (GPIOx_LCKR)、2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx_AFRH、GPIOx_AFRL)、1 つのセキュア設定レジスタ (GPIOx_SECCFGR)、および 1 つの低電圧のハイスピードレジスタ (GPIOx_HSLVR) があります。

13.2 GPIO の主な機能

- 出力状態：プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態：フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx_ODR へのビット単位の書込みアクセス用のビットセット/リセットレジスタ (GPIOx_BSRR)
- I/O ポート設定を固定するロック機構 (GPIOx_LCKR)
- アナログ機能
- オルタネート機能選択レジスタ
- 2 クロックサイクルで変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能
- TrustZone セキュリティをサポート
- STANDBY モード中の I/O 状態を維持

13.3 GPIO の機能説明

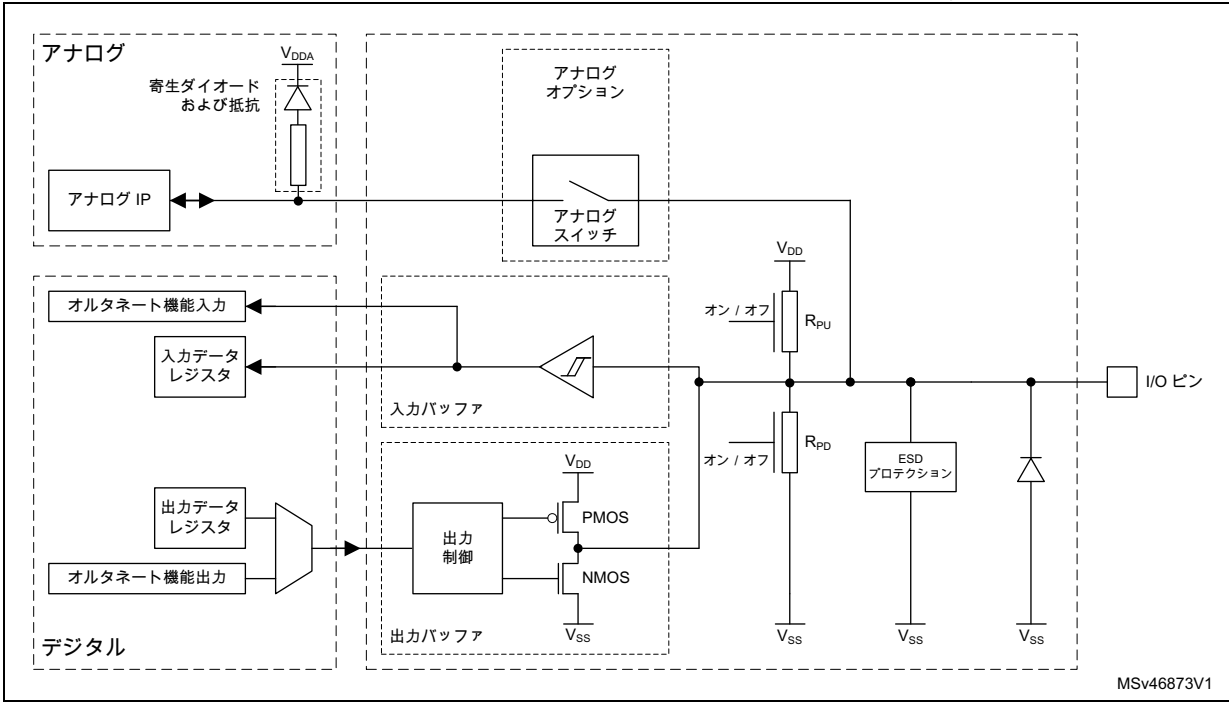
各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx_BSRR および GPIOx_BRR レジスタでは、任意の GPIOx_ODR レジスタに不可分な読出し／変更アクセスを行うことができます。これにより、読出しと変更アクセスの間に IRQ が発生するリスクを回避できます。

次の図に、3 V または 5 V トレラント GPIO (TT または FT) の基本構造を示します。表 114 には、実行可能なポートビットの設定を示します。

図 55. 3 V または 5 V トレラント GPIO (TT または FT) の構造



注： TT GPIO にはアナログスイッチは存在せず、直接接続に置き換えられます。アナログブロックの寄生回路では、5 V は許容されません。

表 114. ポートビット設定⁽¹⁾

MODE(i) [1:0]	OTYPE(i)	OSPEED(i) [1:0]	PUPD(i) [1:0]		I/O 設定	
01	0	SPEED [1:0]	0	0	GP 出力	PP
	0		0	1	GP 出力	PP + PU
	0		1	0	GP 出力	PP + PD
	0		1	1	予約済み	
	1		0	0	GP 出力	OD
	1		0	1	GP 出力	OD + PU
	1		1	0	GP 出力	OD + PD
	1		1	1	予約済み (GP 出力 OD)	

表 114. ポートビット設定⁽¹⁾ (続き)

MODE(i) [1:0]	OTYPE(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 設定	
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	予約済み	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	予約済み	
00	x	x	x	0	0	入力	フローティング
	x	x	x	0	1	入力	PU
	x	x	x	1	0	入力	PD
	x	x	x	1	1	予約済み (入力フローティング)	
11	x	x	x	0	0	入力/出力	アナログ
	x	x	x	0	1	予約済み	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 汎用、PP = プッシュプル、PU = プルアップ、PD = プルダウン、OD = オープンドレイン、AF = オルタネート機能

13.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、ほとんどの I/O ポートはアナログモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA15 : JTDI プルアップ
- PA14 : JTCK/SWCLK プルダウン
- PA13 : JTMS/SWDIO プルアップ
- PB4 : NJTRST プルアップ
- PB3 : JTDO/TRACESWO フローティング状態、プルアップ/プルダウンなし

BOOT0 は、少なくともオプションバイトのローディングが終了するまでリセット中は、入力モードになります ([セクション 13.3.15 : STANDBY モード中の I/O 状態を維持](#)を参照)。

ピンが出力として設定されている場合、出力データレジスタ (GPIOx_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモード (ローレベルのみが駆動され、ハイレベルはハイインピーダンス) で使用することができます。

入力データレジスタ (GPIOx_IDR) は、AHB クロックサイクルごとに、I/O ピン上のデータをキャプチャします。

すべての GPIO ピンに、内部ウィークプルアップ抵抗とウィークプルダウン抵抗があり、GPIOx_PUPDR レジスタの値によってこれらを有効化/無効化できます。

13.3.2 I/O ピンオルタネート機能マルチプレクサと配置

デバイスの I/O ピンは、マルチプレクサを介してオンボードのペリフェラル／モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間の競合はなくなります。

各 I/O ピンは、最大 16 のオルタネート機能入力 (AF0~AF15) を持つマルチプレクサを内蔵しており、これらの入力、GPIOx_AFRL (ピン 0~7) と GPIOx_AFRH (ピン 8~15) レジスタを介して設定することができます。

- リセット後、マルチプレクサの選択はオルタネート機能 0 (AF0) です。I/O は、GPIOx_MODER レジスタを通してオルタネート機能モードで設定されます。
- 各ピンに固有のオルタネート機能割り当てについての詳細は、デバイスデータシートに記載されています。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで利用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従う必要があります。

- デバッグ機能:** 各デバイスのリセット後、これらのピンはデバッグホストによってすぐに使用可能なオルタネート機能ピンとして割り当てられます。
- GPIO:** 必要とする I/O を、GPIOx_MODER レジスタで出力、入力、またはアナログとして設定します。
- ペリフェラルオルタネート機能:**
 - I/O を GPIOx_AFRL または GPIOx_AFRH レジスタのどちらかで必要とする AFx に接続します。
 - タイプ、プルアップ／プルダウン、出力スピードをそれぞれ GPIOx_OTYPER、GPIOx_PUPDR、GPIOx_OSPEEDR レジスタで選択します。
 - 必要とする I/O を、GPIOx_MODER レジスタでオルタネート機能として設定します。
- 追加機能:**
 - ADC および DAC については、GPIOx_MODER レジスタのアナログモードで必要とする I/O を設定し、ADC および DAC レジスタで必要な機能を設定します。
 - RTC、WKUPx、オシレータなどの追加機能については、関連する RTC、PWR、および RCC レジスタで必要な機能を設定します。これらの機能は、標準の GPIO レジスタの設定よりも優先されます。

オルタネート機能 I/O ピンの配置に関する詳細は、デバイスデータシートの「オルタネート機能配置」表を参照してください。

13.3.3 I/O ポート制御レジスタ

各 GPIO ポートには 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR) あり、最大 16 個の I/O を設定します。GPIOx_MODER レジスタは I/O モード (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx_OTYPER および GPIOx_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) およびスピードを選択するために使用されます。I/O の方向がどちらであっても、GPIOx_PUPDR レジスタは、プルアップ／プルダウンを選択するために使用されます。

13.3.4 I/O ポートデータレジスタ

各 GPIO には、16 ビットメモリマップドデータレジスタが 2 つあります。入力データレジスタ (GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A~I)) と出力データレジスタ (GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A~I)) です。

GPIOx_ODR は出力されるデータを格納し、読出し/書込みアクセスが可能です。I/O から入力されるデータは読出し専用の入力データレジスタ (GPIOx_IDR) に格納されます。

13.3.5 I/O データのビット単位の操作

ビットセット/リセットレジスタ (GPIOx_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx_ODR) のビット単位のセット/リセットを可能にします。ビットセット/リセットレジスタは GPIOx_ODR の 2 倍のサイズです。

GPIOx_ODR の各ビットには GPIOx_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。BS(i) に 1 を書き込むと、対応する ODR(i) ビットがセットされます。BR(i) に 1 を書き込むと、対応する ODR(i) ビットがリセットされます。

GPIOx_BSRR のいかなるビットに 0 を書き込んでも GPIOx_ODR の対応するビットには影響しません。仮に、GPIOx_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx_BSRR レジスタを使用した GPIOx_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx_ODR ビットを固定するものではありません。GPIOx_ODR のビットは常に直接アクセスすることができます。GPIOx_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx_ODR をビットレベルでプログラムする場合は、ソフトウェアで割込みを無効にする必要はありません。1 回の不可分な AHB 書込みアクセスで 1 ビットまたは複数ビットを変更できます。

13.3.6 GPIO ロック機構

GPIOx_LCKR レジスタへの特定の書込みシーケンスを行うことにより、GPIO 制御レジスタを停止できます。停止されるレジスタは、GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH、および GPIOx_HSLVR です。

GPIOx_LCKR レジスタに書き込むには、特定の書込み/読出しシーケンスを行う必要があります。このレジスタのビット 16 に適切な LOCK シーケンスを行う場合、LCKR[15:0] の値を使用して I/O の設定を固定します (この書込みシーケンス中、LCKR[15:0] の値は同じである必要があります)。あるポートビットに LOCK シーケンスが行われると、次の MCU リセットまたはペリフェラルリセットまで、そのポートビットの値を変更できなくなります。GPIOx_LCKR の各ビットによって、制御レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH) の対応するビットが停止されます。

GPIOx_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンスは、GPIOx_LCKR レジスタへのワード (32 ビット長) アクセスを使用してのみ実行できます。

13.3.7 I/O オルタネート機能の入力/出力

各 I/O が使用できるオルタネート機能入力/出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx_AFRL および GPIOx_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることになります。こうして各 I/O に使用可能な機

能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、任意の I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

どの機能が各 GPIO ピンに多重化されているかについてはデバイスデータシートを参照してください。

13.3.8 外部割込み／ウェイクアップライン

すべてのポートに外部割込み機能があります。外部割込みラインを使用するには、ポートを入力、出力、またはオルタネート機能モードに設定することができます（アナログモードには設定しないでください）。[セクション 23 : Octo-SPI インタフェース \(OCTOSPI\)](#) を参照してください。

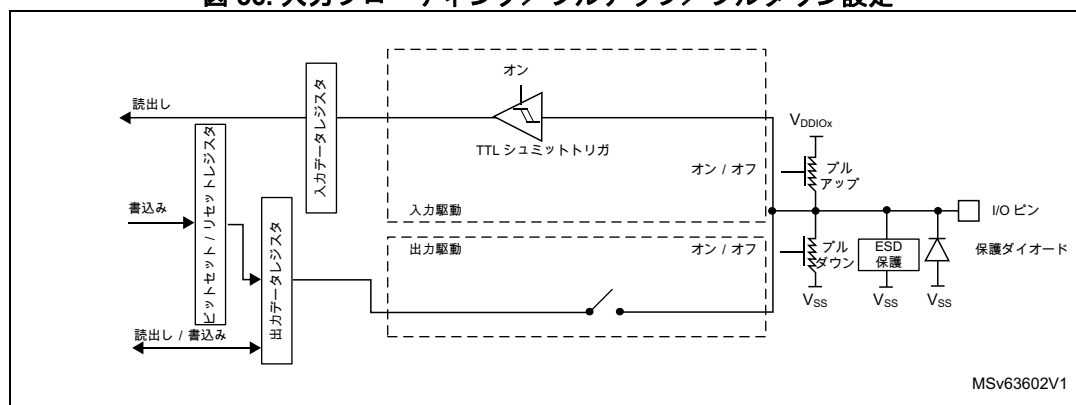
13.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力が有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読出しアクセスによって、I/O 状態が得られます。

次の図は、I/O ポートビットの入力設定を示しています。

図 56. 入力フローティング／プルアップ／プルダウン設定



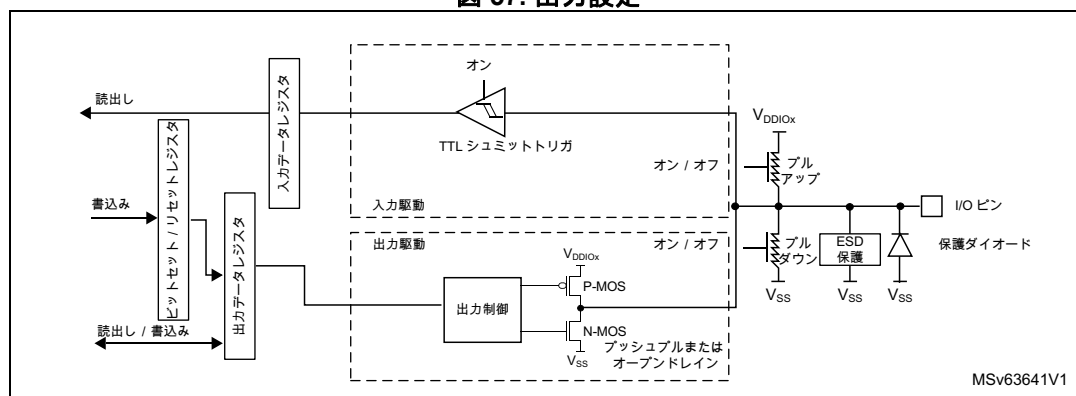
13.3.10 出力設定

I/O ポートが出力としてプログラムされた場合、

- 出力バッファが有効になります。
 - オープンドレインモード：出力レジスタが 0 のときには N-MOS が有効になり、1 のときにはポートはハイインピーダンス状態のままです (P-MOS が有効になることはありません)。
 - プッシュプルモード：出力レジスタが 0 のとき N-MOS が有効になり、1 のときには PMOS が有効になります。
- シュミットトリガ入力 that 有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読出しアクセスによって、I/O 状態が得られます。
- 出力データレジスタの読出しアクセスによって、最後に書き込まれたデータが得られます。

次の図は、I/O ポートビットの出力設定を示しています。

図 57. 出力設定



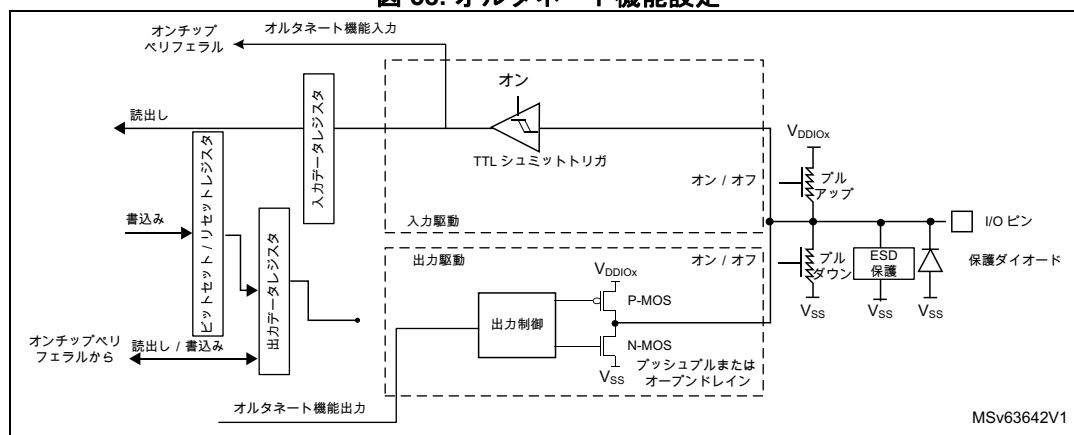
13.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルモードに設定することができます。
- 出力バッファが、ペリフェラル（トランスマITTERイネーブルおよびデータ）からの信号で駆動されます。
- シュミットトリガ入力 that 有効になります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読出しアクセスによって、I/O 状態が得られます。

次の図は、I/O ポートビットのオルタネート機能設定を示しています。

図 58. オルタネート機能設定



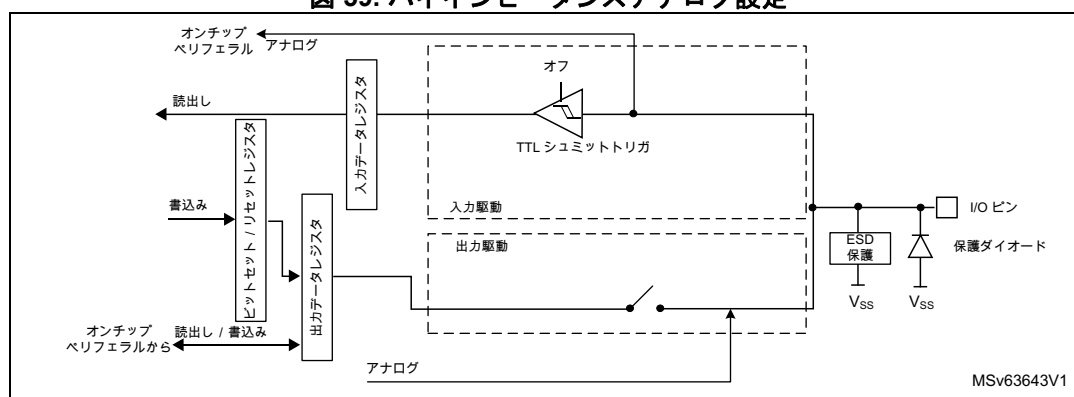
13.3.12 アナログ設定

I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗はハードウェアによって無効にされます。
- 入力データレジスタの読出しアクセスを行うと、値 0 が得られます。

次の図は、I/O ポートビットのハイインピーダンスアナログ入力設定を示しています。

図 59. ハイインピーダンスアナログ設定



13.3.13 HSE または LSE オシレータのピンを GPIO として使用

HSE または LSE オシレータがスイッチオフされた場合（リセット後のデフォルト状態）、関連のオシレータピンを通常の GPIO として使用することができます。

HSE または LSE オシレータがスイッチオンされた場合（RCC_CSR レジスタの HSEON または LSEON ビットを設定することで）、オシレータは関連ピンを制御しますが、これらのピンの GPIO 設定は無効です。

オシレータがユーザ外部クロックモードに設定されている場合、ピンのみがクロック入力のために確保されますが、OSC_OUT または OSC32_OUT ピンは通常の GPIO として使用することができます。

13.3.14 GPIO ピンを RTC 供給ドメインで使用

コア供給ドメインの電源がオフになったとき（デバイスが STANDBY モードに移行したとき）、PC13/PC14/PC15/PI8 の GPIO が機能しなくなります。この場合、それらの GPIO 設定が RTC 設定によってバイパスされなければ、これらのピンはアナログ入力モードに設定されます。

RTC による I/O 制御の詳細は、[セクション 46.3 : RTC の機能説明](#)を参照してください。

13.3.15 STANDBY モード中の I/O 状態を維持

STANDBY モードでは、I/O はデフォルトでフローティング状態にあります。

PWR_IOPSTR レジスタの IOPSTEN ビットがセットされている場合、I/O 状態は STANDBY への移行中にサンプリングされます。I/O 状態は、プルアップ抵抗とプルダウン抵抗を介してピンに適用されます。プルアップ抵抗とプルダウン抵抗は、PWR_IOPSTR レジスタの IOPSTEN ビットがソフトウェアによってクリアされるまで、STANDBY ウェイクアップ後も適用されたままです。

13.3.16 TrustZone セキュリティ

TrustZone セキュリティは Flash オプションバイトレジスタの TZEN オプションバイトで有効化されます。TrustZone が有効な場合（TZEN = 0xB4）、GPIO ポートの各 I/O ピンを GPIOx_SECCFGR レジスタによりそれぞれセキュアとして設定することができます。

選択した I/O ピンがセキュアとして設定されている場合、その対応するオルタネート機能、モード選択、I/O データの設定ビットは、非セキュアアクセスに対してセキュアです。非セキュアアクセスの場合、これらのビットは RAZ/WI です。

ペリフェラル機能を持つ I/O は、ペリフェラル・セキュリティ設定によっても調整されます（詳細については、[セクション 5 : グローバル TrustZone® コントローラ \(GTZC\)](#)を参照)。

- I/O ピン選択がオルタネート機能レジスタで行われるペリフェラル：ペリフェラルがセキュアに設定されている場合、非セキュア I/O ピンへの接続はできません。これが順守されない場合、セキュアなペリフェラルへの入力データは強制的に 0 になり（I/O 入力ピンの値は無視される）、出力ピンの値は強制的に 0 になります。これにより、非セキュア I/O からのセキュアな情報の漏洩を回避することができます。
- アナログスイッチが搭載された I/O は、ペリフェラル（ADC など）で直接制御されます。I/O がセキュアな場合、I/O のアナログスイッチは非セキュアなペリフェラルでは制御できません。これが順守されない場合、スイッチはオープンのままになります。これにより、アナログパスを経由した非セキュアなペリフェラル、または I/O へのセキュアデータのリダイレクトを防ぎます。詳細については、[セクション 3 : システムセキュリティ](#)を参照してください。
- I/O の「追加機能」とペリフェラルの間のパスの一部は、I/O がセキュアでペリフェラルが非セキュアである場合はブロックされません。そのため、アプリケーションで使用しない場合でも、これらのペリフェラルをセキュアに設定することが推奨されます。当該ペリフェラルの一覧については、[セクション 3 : システムセキュリティ](#)を参照してください。パスにセキュリティ制御がかかっている場合は、オルタネート機能による I/O 選択と同じ規則に従います。

ペリフェラルオルタネート機能や追加機能の配置の詳細については、データシートのデバイス・ピンの定義の表を参照してください。

リセット後、すべての GPIO ポートはセキュアです。

次の表は、GPIO_SECCFGR レジスタのセキュリティ設定ビットに対する I/O ポートのセキュアビットの一覧です。I/O ビットポートがセキュアとして設定される場合

- セキュアビット：読出しおよび書込み操作は、セキュアアクセスによってのみ可能です。セキュアビットへの非セキュア読出し／書込みアクセスは RAZ/WI です。不正アクセスイベントは生成されません。
- 非セキュアビット：制約はありません。読出しおよび書込み操作は、セキュアおよび非セキュアアクセスのどちらでも可能です。

TrustZone セキュリティが無効な場合（FLASH_OPTSR2 レジスタの TZEN = 0xC3）、すべてのレジスタビットは非セキュアです。GPIOx_SECCFGR レジスタは RAZ/WI です。

表 115. GPIO セキュアビット

セキュア設定ビット	セキュアビット	レジスタ名	セキュアビットの 非セキュアアクセス
GPIOx_SECCFGR において SECy = 1 ⁽¹⁾	MODEy[1:0]	GPIOx_MODER	RAZ/WI
	OTy	GPIOx_OTYPER	
	OSPEEDy[1:0]	GPIOx_OSPEEDR	
	PUPDy[1:0]	GPIOx_PUPDR	
	IDy	GPIOx_IDR	
	ODy	GPIOx_ODR	
	BSy および BRy	GPIOx_BSRR	
	LCKy	GPIOx_LCKR	
	BRy	GPIOx_BRR	
	AFSELy[3:0]	GPIOx_AFRH	
		GPIOx_AFRL	
	HSLVy	GPIOx_HSLVR	

1. GPIOx、x = A~I。x = A~H の場合、y = 0~15。x = I の場合、y = 0~11。

13.3.17 特権および非特権モード

すべての GPIO レジスタは、セキュリティ状態（セキュアまたは非セキュア）に関係なく、特権および非特権アクセスで読み書きできます。

13.3.18 ハイスピード低電圧モード (HSLV)

一部の I/O は、HSLV モードに設定することで、低電圧で最大速度を上げる機能があります。I/O HSLV ビットは、I/O の出力スピードが 3.3 V（デフォルト設定）または 1.8 V（HSLV = 1）で動作するように最適化されるかどうかを制御します。

注意： I/O の HSLV 設定ビットは、I/O 供給電圧（V_{DD} または V_{DDIO2}）が 2.7 V を超える場合はセットできません。電圧が 2.7 V よりも高い場合にセットすると、デバイスを傷つける恐れがあります。I/O の HSLV ビットは、対応するオプションビットが有効な場合にのみセットできます（I/O 供給電圧に応じて IO_VDD_HSLV または IO_VDDIO2_HSLV、[セクション 7.4: FLASH のオプション・バイト](#)を参照）。この機能に関連付けられたハードウェア保護はありません。固定された I/O 供給電圧向けのスタティック・コンフィギュレーションとしてのみ使用することを推奨します。

13.3.19 I/O 補正セル

I/O 転流スルーレート (tfall/trise) は、電源の I/O ノイズを低減するために、プロセス、電圧、および温度の条件に応じて、ソフトウェアで対応可能です。詳細については、[セクション 14 : システム設定、ブート、およびセキュリティ \(SBS\)](#) を参照してください。

13.4 GPIO レジスタ

このセクションには、GPIO レジスタの詳細な説明が記載されています。

ペリフェラルレジスタはワード、ハーフワード、バイトのいずれかのモードで書き込むことができます。

13.4.1 GPIO ポートモードレジスタ (GPIOx_MODER) (x = A~I)

アドレスオフセット : 0x00

リセット値 : 0xABFF FFFF (ポート A)

リセット値 : 0xFFFF FEBF (ポート B)

リセット値 : 0xFFFF FFFF (ポート C~H)

リセット値 : 0x00FF FFFF (ポート I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15 [1:0]		MODE14 [1:0]		MODE13 [1:0]		MODE12 [1:0]		MODE11 [1:0]		MODE10 [1:0]		MODE9 [1:0]		MODE8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7 [1:0]		MODE6 [1:0]		MODE5 [1:0]		MODE4 [1:0]		MODE3 [1:0]		MODE2 [1:0]		MODE1 [1:0]		MODE0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **MODEy[1:0]** : ポート x 設定用 I/O ピン y (y = 15~0)

これらのビットは、I/O モードを設定するために、ソフトウェアによって書き込まれます。

00 : 入力モード

01 : 汎用出力モード

10 : オルタネート機能モード

11 : アナログモード (リセット状態)

注 : このビットフィールドは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.2 GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A~I)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **OTy** : ポート x 設定用 I/O ピン y (y = 15~0)

これらのビットは、I/O 出力タイプを設定するために、ソフトウェアによって書き込まれます。

0 : 出力プッシュプル (リセット状態)

1 : 出力オープンドレイン

注 : このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.3 GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A~I)

アドレスオフセット : 0x08

リセット値 : 0x0C00 0000 (ポート A)

リセット値 : 0x0000 00C0 (ポート B)

リセット値 : 0x0000 0000 (他のポート)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **OSPEEDy[1:0]** : ポート x 設定用 I/O ピン y (y = 15~0)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

00 : ロースピード

01 : ミディアムスピード

10 : ハイスピード

11 : 超ハイスピード

注 : 周波数仕様、およびスピード別の電源や負荷条件については、デバイスデータシートを参照してください。

このビットフィールドは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.4 GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A~I)

アドレスオフセット : 0x0C

リセット値 : 0x6400 0000 (ポート A)

リセット値 : 0x0000 0100 (ポート B)

リセット値 : 0x0000 0000 (他のポート)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15 [1:0]		PUPD14 [1:0]		PUPD13 [1:0]		PUPD12 [1:0]		PUPD11 [1:0]		PUPD10 [1:0]		PUPD9 [1:0]		PUPD8 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7 [1:0]		PUPD6 [1:0]		PUPD5 [1:0]		PUPD4 [1:0]		PUPD3 [1:0]		PUPD2 [1:0]		PUPD1 [1:0]		PUPD0 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **PUPDy[1:0]** : ポート x 設定用 I/O ピン y (y = 15~0)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

00 : プルアップ／プルダウンなし

01 : プルアップ

10 : プルダウン

11 : 予約済み

注 : このビットフィールドは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.5 GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A~I)

アドレスオフセット : 0x10

リセット値 : 0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **IDy** : ポート x 入力データ用 I/O ピン y (y = 15~0)

これらのビットは読み出し専用です。これらのビットには、対応する I/O ポートの入力値が格納されています。

注 : このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.6 GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A~I)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ODy** : ポート x 出力データ用 I/O ピン y (y = 15~0)

これらのビットは、ソフトウェアによって読出し／書込みができます。

注 : 不可分なビット単位のセット／リセットのために、OD ビットを、GPIOx_BSRR または GPIOx_BRR レジスタ (x = A~I) への書込みによって、個々にセット／リセットすることができます。

このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.7 GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR) (x = A~I)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 **BRy** : ポート x リセット用 I/O ピン y (y = 15~0)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODy ビットの値は変化しません。

1 : 対応する ODy ビットをリセットします。

注 : BSy ビットと BRy ビットの両方がセットされた場合、BSy ビットが優先されます。

このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

ビット 15:0 **BSy** : ポート x セット用 I/O ピン y (y = 15~0)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODy ビットの値は変化しません。

1 : 対応する ODy ビットをセットします。

注 : このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.8 GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A~I)

このレジスタは、ビット 16 (LCKK) に正しい書込みシーケンスが行われたときに、ポートビットの設定をロックするために使用されます。ビット [15:0] の値は、GPIO の設定をロックするために使用されます。書込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU リセットまたはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

注： GPIOx_LCKR レジスタへの書込みには特定の書込みシーケンスが使われます。このロックシーケンスではワードアクセス (32 ビット長) のみ可能です。

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **LCKK** : ロックキー

このビットは常時読出しができます。ロックキー書込みシーケンスを使用しなければ変更できません。

0 : ポート設定ロックキーは無効です。

1 : ポート設定ロックキーは有効です。GPIOx_LCKR レジスタは、次の MCU リセットまたはペリフェラルリセットまでロックされます。

- LOCK キー書込みシーケンス :

書込み LCKR[16] = 1 + LCKR[15:0]

書込み LCKR[16] = 0 + LCKR[15:0]

書込み LCKR[16] = 1 + LCKR[15:0]

- LOCK キー読出し

読出し LCKR[16] = 1 (この読出し操作は任意だが、ロックが有効であることを確認できる)

注： ロックキー書込みシーケンスの間は、LCK[15:0] の値を変更することはできません。

ロックシーケンス中にエラーが発生すると、LOCK は中止されます。

ポートの任意のビットの最初の LOCK シーケンスの後、次の MCU リセットまたはペリフェラルリセットまでは、LCKK ビットのいかなる読出しアクセスに対しても、1 が返されます。

ビット 15:0 **LCKy** : ポート x ロック用 I/O ピン y (y = 15~0)

これらのビットは読出し／書込みができますが、書き込めるのは LCKK ビットが 0 のときだけです。

0 : ポート設定はロックされません。

1 : ポート設定はロックされます。

注： このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.9 GPIO オルタネート機能下位レジスタ (GPIOx_AFRL) (x = A~I)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7 [3:0]				AFSEL6 [3:0]				AFSEL5 [3:0]				AFSEL4 [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 AFSELY[3:0] : ポート x オルタネート機能選択用 I/O ピン y (y = 7~0)

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

0000 : AF0

0001 : AF1

0010 : AF2

0011 : AF3

0100 : AF4

0101 : AF5

0110 : AF6

0111 : AF7

1000 : AF8

1001 : AF9

1010 : AF10

1011 : AF11

1100 : AF12

1101 : AF13

1110 : AF14

1111 : AF15

注 : このビットフィールドは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.10 GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A~H)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15 [3:0]				AFSEL14 [3:0]				AFSEL13 [3:0]				AFSEL12 [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11 [3:0]				AFSEL10 [3:0]				AFSEL9 [3:0]				AFSEL8 [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 AFSELy[3:0] : ポート x オルタネート機能選択用 I/O ピン y (y = 15~8)

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

0000 : AF0

0001 : AF1

0010 : AF2

0011 : AF3

0100 : AF4

0101 : AF5

0110 : AF6

0111 : AF7

1000 : AF8

1001 : AF9

1010 : AF10

1011 : AF11

1100 : AF12

1101 : AF13

1110 : AF14

1111 : AF15

注 : このビットフィールドは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.11 GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A~I)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **BRy** : ポート x リセット用 I/O ピン y (y = 15~0)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODy ビットの値は変化しません。

1 : 対応する ODy ビットをリセットします。

注 : このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.12 GPIO ハイスピード低電圧レジスタ (GPIOx_HSLVR) (x = A~I)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSLV15	HSLV14	HSLV13	HSLV12	HSLV11	HSLV10	HSLV9	HSLV8	HSLV7	HSLV6	HSLV5	HSLV4	HSLV3	HSLV2	HSLV1	HSLV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **HSLVy** : ポート x ハイスピード低電圧設定 (y = 15~0)

これらのビットは、I/O 供給電圧が低い場合に、I/O スピードを最適化するために、ソフトウェアによって書き込まれます。

各ビットは、対応する IO_VDD_HSLV/IO_VDDIO2_HSLV ユーザオプションビットがセットされたときのみ有効です。これは、I/O 供給電圧が 2.7 V 未満の場合にのみ使用する必要があります。

I/O 供給電圧 (VDD または VDDIO2) が 2.7 V よりも高い場合にこれらのビットをセットすると、デバイスにダメージを与えます。

0 : I/O スピードの最適化は無効です。

1 : I/O スピードの最適化は有効です。

注 : すべての I/O が HSLV モードをサポートしているわけではありません。この機能をサポートする I/O のリストについては、対応するデータシートの I/O 構造を参照してください。他の I/O の HSLV 設定は、リセット値に保持する必要があります。

このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.13 GPIO セキュア設定レジスタ (GPIOx_SECCFGR) (x = A~I)

システムがセキュアな場合 (TZEN = 0xB4)、このレジスタは書込みアクセスのセキュリティを提供し、セキュアアクセスによってのみ書込みできます。選択した I/O ピンをセキュアとして設定するために使用します。このレジスタへの非セキュア書込みアクセスは破棄されます。

システムが非セキュアな場合 (TZEN = 0xC3)、このレジスタは RAZ/WI です。

アドレスオフセット : 0x30

リセット値 : 0x0000 FFFF (ポート A~H)

リセット値 : 0x0000 0FFF (ポート I)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **SECy** : ポート x セキュアビット有効化用 I/O ピン y (y = 15~0)

これらのビットは、I/O ポートピンのセキュリティを有効/無効にするために、ソフトウェアによって書き込まれます。

0 : I/O ピンは非セキュアです。

1 : I/O ピンはセキュアです。対応するすべてのセキュアビットについては、[表 115](#) を参照してください。

注 : このビットは予約済みであり、選択したパッケージで対応する I/O を使用できない場合にはリセット値に保持する必要があります。

13.4.14 GPIO レジスタマップ

表 116. GPIO レジスタのマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOx_MODER (x = A~I)	MODE15 [1:0]		MODE14 [1:0]		MODE13 [1:0]		MODE12 [1:0]		MODE11 [1:0]		MODE10 [1:0]		MODE9 [1:0]		MODE8 [1:0]		MODE7 [1:0]		MODE6 [1:0]		MODE5 [1:0]		MODE4 [1:0]		MODE3 [1:0]		MODE2 [1:0]		MODE1 [1:0]		MODE0 [1:0]	
	ポート A の リセット値	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	ポート B の リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	
	ポート C~H の リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
	ポート I のリセット値	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x04	GPIOx_OTYPER (x = A~I)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	GPIOx_OSPEEDR (x = A~I)	OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [1:0]		OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
	ポート A の リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	ポート B の リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	
	ポート C~I の リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	GPIOx_PUPDR (x = A~I)	PUPD15 [1:0]		PUPD14 [1:0]		PUPD13 [1:0]		PUPD12 [1:0]		PUPD11 [1:0]		PUPD10 [1:0]		PUPD9 [1:0]		PUPD8 [1:0]		PUPD7 [1:0]		PUPD6 [1:0]		PUPD5 [1:0]		PUPD4 [1:0]		PUPD3 [1:0]		PUPD2 [1:0]		PUPD1 [1:0]		PUPD0 [1:0]	
	ポート A の リセット値	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	ポート B の リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
	ポート C~I の リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	GPIOx_IDR (x = A~I)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	リセット値																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x14	GPIOx_ODR (x = A~I)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (x = A~I)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (x = A~I)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A~I)	AFSEL7 [3:0]			AFSEL6 [3:0]			AFSEL5 [3:0]			AFSEL4 [3:0]			AFSEL3 [3:0]			AFSEL2 [3:0]			AFSEL1 [3:0]			AFSEL0 [3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 116. GPIO レジスタのマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	GPIOx_AFRH (x = A~H)	AFSEL15 [3:0]						AFSEL14 [3:0]			AFSEL13 [3:0]			AFSEL12 [3:0]			AFSEL11 [3:0]			AFSEL10 [3:0]			AFSEL9 [3:0]			AFSEL8 [3:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	GPIOx_BRR (x = A~I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	GPIOx_HSLVR (x = A~I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLV15	HSLV14	HSLV13	HSLV12	HSLV11	HSLV10	HSLV9	HSLV8	HSLV7	HSLV6	HSLV5	HSLV4	HSLV3	HSLV2	HSLV1	HSLV0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	GPIOx_SECCFGR (x = A~I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	A~H のリセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	ポート I のリセット値																	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

14 システム設定、ブート、およびセキュリティ (SBS)

14.1 SBS の概要

STM32H563/H573 および STM32H562 デバイスは、SBS に一連の設定レジスタを備えています。この SBS ペリフェラルは、さまざまなデバイス設定の上で、デバッグ制御やセキュアストレージ制御を含めた主要なブートおよびセキュリティ機能を制御します。

14.2 SBS の主な機能

- System configuration(システム設定)
 - 安全機能の管理
 - I/O アナログスイッチの一部の I/O および電圧ブースタの FMP (Fast-mode Plus: 高速モードプラス) 高駆動機能の有効化/無効化
 - I/O 補正セルの管理
 - レジスタセキュリティアクセスの設定
- ブート制御
 - システムリセット時、PRODUCT_STATE (sbs_product_state)、BOOT_UBE (sbs_irot_select)、TZEN (sbs_tzen) など、現在の設定に応じて、Cortex-M33 ブートアドレスと時間的分離 (temporal isolation) レベルを設定します。
 - 非表示保護レベル (HDPL) 単調カウンタによって実装される時間的分離機能を管理します。
- デバッグ制御
 - デバイスデバッグインタフェースの開始 (オープニング) を制御し、デバイスのセキュリティを保証するイベントの順序付けを確保します。
- ハードウェア・セキュアストレージ制御
 - セキュアストレージ選択を制御します。OBK-HDPL 選択 (Flash メモリのセキュアストレージの領域を選択し、SAES に対応する DHUK を選択)、EPOCH 選択

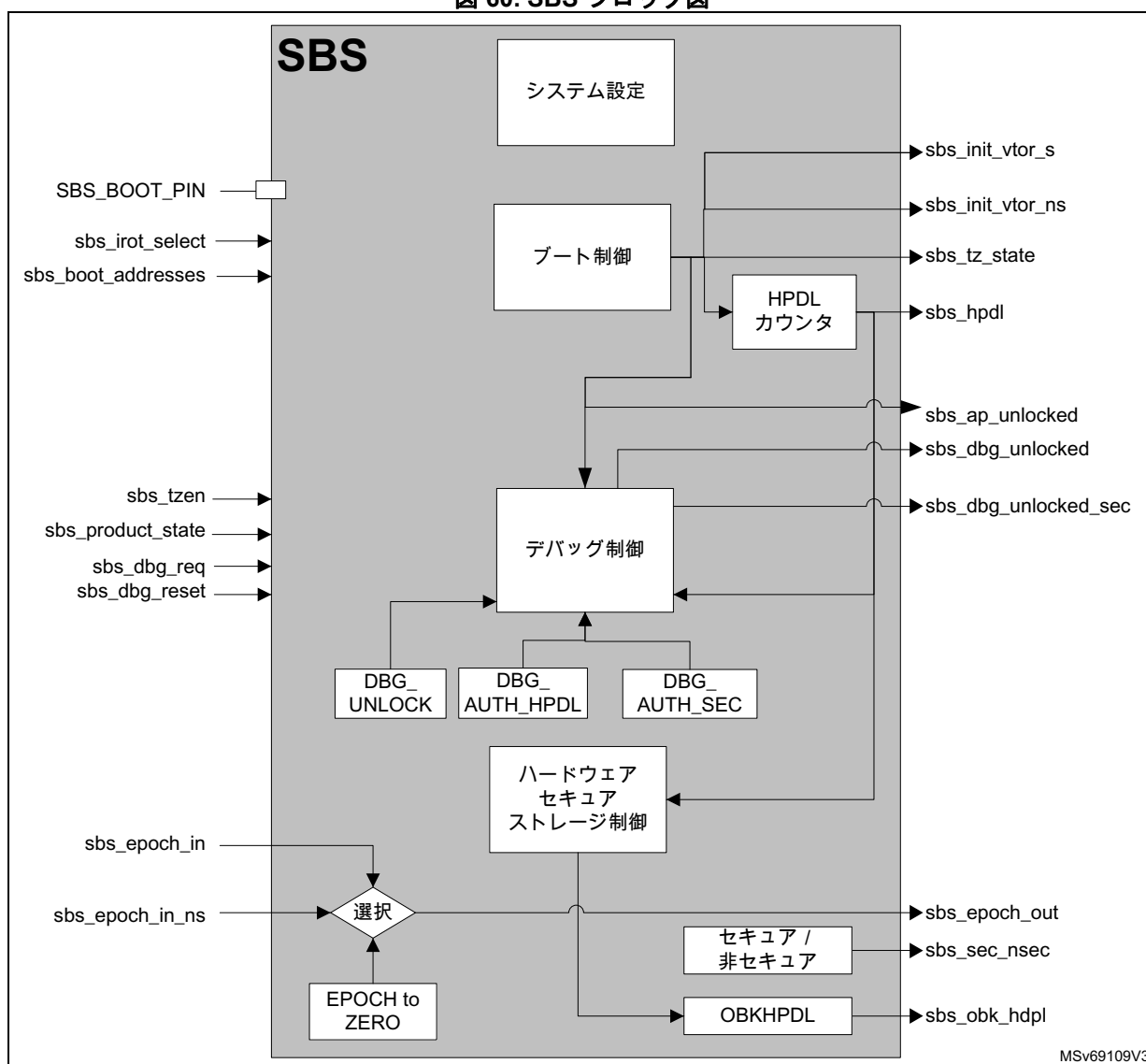
14.3 SBS の機能詳細

14.3.1 SBS ブロック図

次の図は、SBS ブロック図を示し、4 つの主な機能も示しています。

- System configuration(システム設定)
- ブート制御
- デバッグ制御
- ハードウェア・セキュアストレージ制御

図 60. SBS ブロック図



14.3.2 SBS 信号

次の表に、SBS とインタフェースするユーザ関連の内部信号を示します。

表 117. SBS 内部入力／出力信号

信号名	タイプ	説明
BOOT0	入力	TrustZone が無効なときには (TZEN = 0xC3)、ユーザ Flash メモリまたはブートローダでのブート、または TrustZone が有効なときには (TZEN = 0xB4)、RSS でのブートを選択します。
sbs_irot_select		BOOT_UBE オプションバイトに基づき、ST-iROT とユーザ Flash メモリ (OEM-iROT) のいずれかの iROT を選択する信号
sbs_tzen		TZEN オプションバイトに基づき、TrustZone を有効化/無効化する信号
sbs_boot_addresses		Flash メモリによって定義されたアドレスのリスト : – NSBOOTADD : 非セキュアブートアドレス – SECBOOTADD : セキュアブートアドレス
sbs_product_state		PRODUCT_STATE オプションバイトに基づいて、製品用途に応じてさまざまなセキュリティメカニズムを有効にする信号。期待値については、 セクション 7 : 内蔵 Flash メモリ (FLASH) で説明されています。
sbs_dbg_req		ブート時にデバッグ認証プロトコルを起動します。
sbs_init_vtor_s	出力	Cortex-M33 セキュアエントリポイントのベクタアドレス
sbs_init_vtor_ns		Cortex-M33 非セキュアエントリポイントのベクタアドレス
sbs_tz_state		コアのセキュア状態を Cortex-M33 に通知します。
sbs_hdpl		HDPL (時間的分離レベル、ブートレベルの ID、ブートレベルを分離するために使用) この信号は、単調カウンタを反映します。単調カウンタは、0 から 3 までインクリメントし、ソフトウェアリセットまたは POR 時にのみゼロにリセットされます。
sbs_ap_unlocked		Cortex-M33 アクセスポートを制御します。
sbs_dbg_unlocked		Cortex-M33 非セキュア部分のデバッグ (1 にセットされているとき) をアンロックします。
sbs_dbg_unlocked_sec		Cortex-M33 セキュア部分のデバッグ (1 にセットされているとき) をアンロックします。
sbs_dbg_reset	入力	この信号は、システムリセットまたはパワーオンリセットで行われるデバッグ認証設定のリセットを制御するために使用されます。設定は、DBGMCU_CR レジスタの DCRT ビットフィールドを使用して、DBGMCU を介して行われます。
sbs_obk_hdpl	出力	現在の HDPL のセキュアストレージドメイン (OBK-HDPL)、またはより大きいものを (プロビジョニングを可能にするために) 選択します。
sbs_epoch_out		EPOCH カウンタ (NS_EPOCH および SEC_EPOCH、Flash メモリからの入力) は、REPLAY 保護を管理するために使用されます。
sbs_sec_nsec		セキュアストレージのセキュア/非セキュア選択を反映します。

14.3.3 SBS のリセットおよびクロック

SBS 設定ポートは、AHB バスクロックによってクロックされます。一般リセットとデバッグ設定リセットが DBGMCU で制御されます。

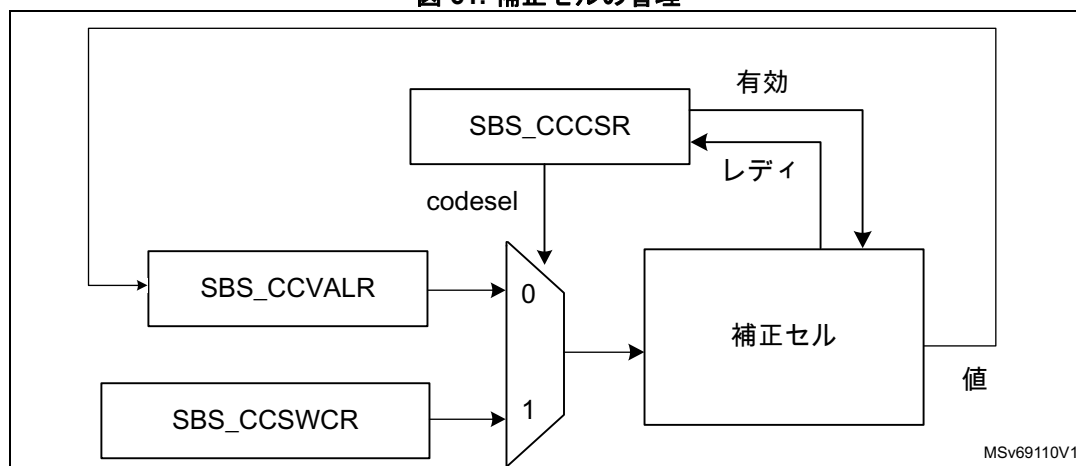
14.3.4 SBS システム設定

SBS I/O 補正セルの管理

I/O 補正セルは、I/O バッファ向けに 8 ビットの値を生成し（N-MOS 用の 4 ビットと P-MOS 用の 4 ビット）、PVT 動作条件（プロセス、電圧、温度）に依存します。これらのビットは、I/O バッファの出力インピーダンスと I/O 通信のスルーレート（ t_{fall} および t_{rise} 時間）を制御するために使用され、電源装置の I/O ノイズを軽減します。

次の図に示されているように、補正セルは2つのブロックに分割されています。1つは現在のPVTに対して最適なコードを提供するためのブロックであり、もう1つのブロックは、ソフトウェアによって制御されるブロックを駆動します。

図 61. 補正セルの管理



補正セルの値は、SBS_CCCSR で READY フラグがセットされているときに読み出すことができます。SBS_CCCSR の CODESEL により、アプリケーションは適用する値を 2 つのオプションから選択できます。つまり、セルからのコード、または SBS CCSWCR からのコードです。

2つの補正セルは、STM32H563/H573およびSTM32H562 デバイスに内蔵されています。

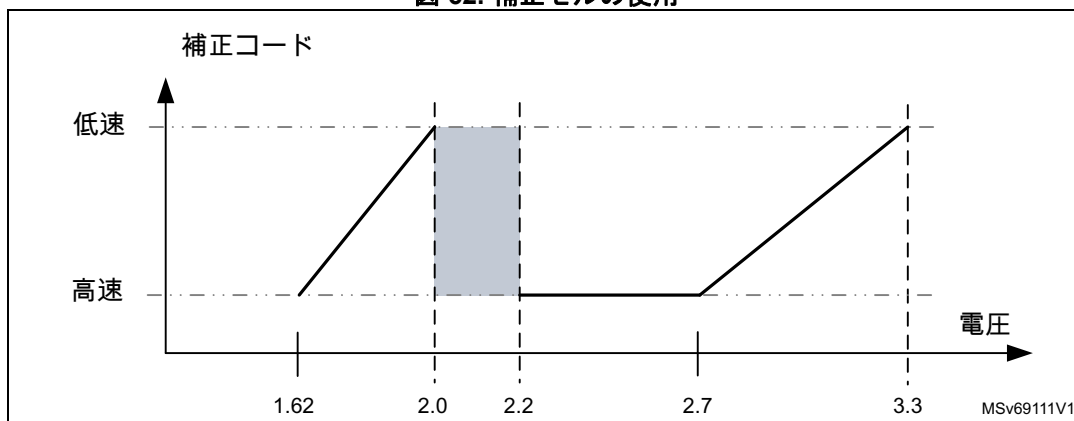
- 1 つは、V_{DDIO} 電源レールによって供給される I/O 用
- 1 つは、V_{DDIO2} 電源レールによって供給される I/O 用

デフォルトでは、補正セルは無効であり、すべての I/O に固定コードが適用されます。

注：補正セルは、 $2.7\text{ V} \leq V_{\text{DDIOx}} \leq 3.6\text{ V}$ もしくは $1.62\text{ V} \leq V_{\text{DDIOx}} \leq 2\text{ V}$ のときのみ使用できます (図 62 を参照)。

注： 補正セルは、CSI オシレータが有効なときのみ使用できます。CSI オシレータの詳細については、[セクション 11：リセットおよびクロック制御（RCC）](#)を参照してください。

図 62. 補正セルの使用



SBS TrustZone のセキュリティと特権

- SBS TrustZone のセキュリティ

TrustZone セキュリティが有効なとき、SBS はセキュアレジスタを非セキュアアクセスによる変更から保護できます。

TrustZone セキュリティは、FLASH_OPTSR2_PRG の TZEN オプションバイトによって有効化されます。

セキュアなレジスタへの非セキュア読み出し／書き込みアクセスは RAZ/WI (Read-As-Zero, Writes Ignored: 読み出しは0、書き込みは無効) で、不正アクセスイベントが生成されます。SBS 不正アクセスイベントが GTZC で有効な場合、不正アクセス割込みが生成されます。

- 特権／非特権モード

SBS レジスタは、特権および非特権アクセスによって読み書きできますが、CPU 設定の SBS レジスタである

- SBS_CSLCKR、SBS_FPUIMR、および SBS_CNSLCKR を除きます。
- SBS_SECCFGR の FPUSEC

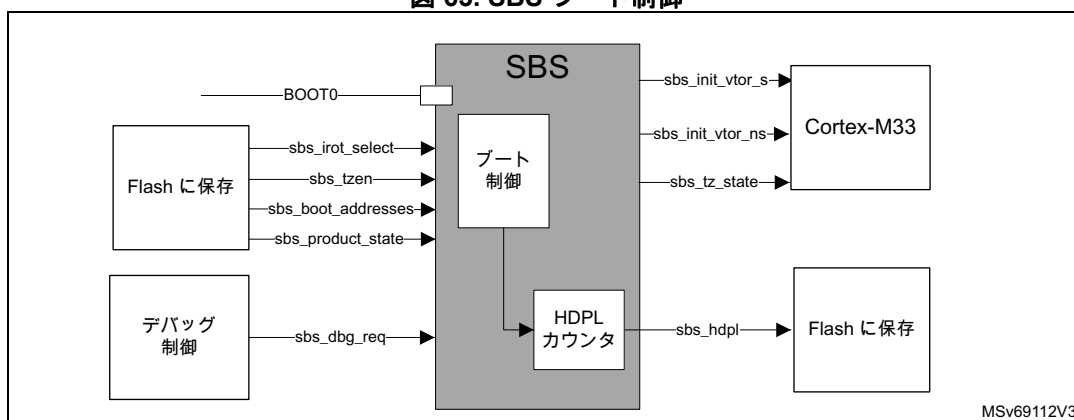
特権レジスタへの非特権アクセスは RAZ/WI です。

14.3.5 SBS ブート制御

SBS を使用すると、製品設定を考慮してブートエントリーポイントを制御できます。主なブート制御アクションを以下に示します。

- TrustZone が有効または無効な状態で製品を実行します。
- ST-iROT と OEM-iROT のいずれかを選択します。
- デバッグ認証シーケンスを起動するときにブートします。
- ブートローダかユーザ Flash メモリブートかを選択します。
- HDPL ブート値を初期化します。

図 63. SBS ブート制御



MSv69112V3

ブート設定は、製品設定を考慮して選択されます。

- **BOOT0**: ユーザ Flash メモリまたは RSS (ルートセキュアサービス) でのブートを選択します。
- **ST-iROT** か **OEM-iROT** のいずれかの iROT を選択するための **BOOT_UBE** オプションバイト
- **TrustZone** を有効化/無効化するための **TZEN** オプションバイト
- **sbs_boot_addresses**: Flash メモリによって定義されたアドレスのリスト:
 - **NSBOOTADD**: 非セキュアブートアドレス
 - **SECBOOTADD**: セキュアブートアドレス
- **PRODUCT_STATE**: 製品の用途に応じてさまざまなセキュリティメカニズムを有効にするためのオプションバイト。期待値については、[セクション 7: 内蔵 Flash メモリ \(FLASH\)](#) で説明されています。
- **sbs_dbg_req**: ブート時にデバッグ認証プロトコルを起動するために使用されます。

ブート制御ロジックは、以下のデータを設定します。

- **sbs_init_vtor_s**: Cortex-M33 セキュアエントリポイントのベクタアドレス
- **sbs_init_vtor_ns**: Cortex-M33 非セキュアエントリポイントのベクタアドレス
- **sbs_tz_state** (セキュア/非セキュア): コアのセキュア状態を Cortex-M33 に通知します。
- **HDPL** (下の説明を参照)

SBS HDPL (時間的分離レベル) の管理

HDPL は、ブートステージ中にインクリメントされる単調カウンタです。HDPL は、パワーオンリセットまたはシステムリセット後にのみデフォルト値にリセットされます。デフォルト値 (0 または 1) は、ブートロジックで定義されたデバイスのライフサイクルによって異なります。

STM32H563/H573 および STM32H562 デバイスは HDPL 情報を使用して、ブートプロセス時に、コードと関連するシークレット (キーなど) を自動的に分離します。HDPL をインクリメントすることで、あるブートステージのプライベートコードおよびデータが後のブートステージから直接アクセスできなくなります。

HDPL は、ユーザ Flash メモリによって使用されます。詳細については、[セクション 7: 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。HDPL は、0 から 3 の値を取ります。値 3 に達すると、HDPL は、リセットされるまでこの値のままです。現在の HDPL 値は、**SBS_HDPLSR** の HDPL ビットフィールドで読み出し可能です。

HDPL を 1 だけインクリメントするには、アプリケーションは 0x6A を SBS_HDPLCR の INCR_HDPL に書き込む必要があります。そのようなインクリメントの後、後続のアクションを実行する前に、ユーザは SBS_HDPLSR を読み出して、HDPL が実際にインクリメントされたことを確認する必要があります。

表 118. HDPL エンコード値

HDPL	エンコード
0	0xB4
1	0x51
2	0x8A
3	0x6F
	その他すべての値

表 119. SBS ブートロジック

入力						出力			
sbs_product_ 状態	sbs_dbg_req	sbs_tzen	sbs_irotselect	BOOT0	sbs_boot _addresses	sbs_init_vtor_s	sbs_init_vtor_ns	sbs_hdpl	sbs_tz_state
Locked 以外	1	x	x	x	BOOT_DBG_AUTH_ADD	BOOT_DBG_AUTH_ADD	x	1	セキュア
オープン	0	0	x	0	NSBOOT_ADD	x	NSBOOT_ADD	1	非セキュア
	0	0	x	1	BOOT_ST_RSS_ADD	BOOT_ST_RSS_ADD	x	0	セキュア
	0	1	x	0	SECBOOT_ADD	SECBOOT_ADD	x	1	セキュア
	0	1	x	1	BOOT_ST_RSS_ADD	BOOT_ST_RSS_ADD	x	0	セキュア
プロビジョニング	0	x	x	x	BOOT_ST_RSS_ADD	BOOT_ST_RSS_ADD	x	0	セキュア
iROT プロビジョン	0	0	x	x	NSBOOT_ADD	x	NSBOOT_ADD	1	非セキュア
	0	1	0	x	BOOT_ST_iROT_ADD	BOOT_ST_iROT_ADD	x	1	セキュア
	0	1	1	x	SECBOOT_ADD	x	SECBOOT_ADD	1	セキュア
TZ クローズド	-	1	0	x	BOOT_ST_iROT_ADD	BOOT_ST_iROT_ADD	x	1	セキュア
	-	1	1	x	SECBOOT_ADD	x	SECBOOT_ADD	1	セキュア

表 119. SBS ブートロジック (続き)

入力						出力			
sbs_product_ 状態	sbs_dbg_req	sbs_tzen	sbs_iroselect	BOOT0	sbs_boot _addresses	sbs_init_vtor_s	sbs_init_vtor_ns	sbs_hdpl	sbs_tz_state
クローズド	0	0	x	x	NSBOOT ADD	x	NSBOOT ADD	1	非セキュア
	0	1	0	x	BOOT_ST_ iROT_ADD	BOOT_ST_ iROT_ADD	x	1	セキュア
	0	1	1	x	SECBOOT ADD	x	SECBOOT ADD	1	セキュア
ロック	0	0	x	x	NSBOOT ADD	x	NSBOOT ADD	1	非セキュア
	0	1	0	x	BOOT_ST_ iROT_ADD	BOOT_ST_ iROT_ADD	x	1	セキュア
	0	1	1	x	SECBOOT ADD	x	SECBOOT ADD	1	セキュア
リグレッション	0	0	x	x	BOOT_ DBG_AUTH_ ADD	BOOT_ DBG_AUTH_ ADD	x	0	セキュア
	0	1	x	x	BOOT_ DBG_AUTH_ ADD	BOOT_ DBG_AUTH_ ADD	x	0	セキュア
NS-リグレッション	0	1	x	x	BOOT_ DBG_AUTH_ ADD	BOOT_ DBG_AUTH_ ADD	x	0	セキュア

14.3.6 SBS デバッグ制御

SBS デバッグ制御は、デバッグのオープンを管理するために使用され、レジスタ設定で、またはデバッグ認証制御を通じて、製品の状況 (PRODUCT_STATE、TZEN、HDPL) を処理します。

デバッグが禁止されているとき、メールボックスアクセスポート、Cortex-M33 アクセスポート、および CPU デバッグインタフェースはロックされます。この状況では、デバッグは CPU にアクセスできず、事実上、デバッグを行うことはできません。詳細については、[セクション 58: デバッグサポート \(DBG\)](#) を参照してください。

認証済みデバッグシーケンス

1. 外部ホストは、DBGMCU アクセスポートメールボックスを介して、デバッグ認証プロトコルの起動をリクエストします。デバイスの残りは、リセット状態のままです。
2. SBS は、STMicroelectronics RSS-DA (デバッグ認証ライブラリ) ブートアドレスを選択し、CPU にリセットからのリリースをリクエストします。
3. RSS-DA ライブラリを実行している CPU は、システム Flash メモリのデバッグ認証プロトコルを実行します。デバイスがクローズされた場合、RSS-DA が認証シーケンス開始リクエストに確認応答するまで、アクセスポートメールボックスはクローズされます。

4. 認証方法は、TrustZone の有効化によって異なります。
 - TrustZone が有効なとき (TZEN = 0xB4)、認証方法は証明書に基づきます。デバッグ証明書連鎖がデバイスによって完全に確認されるとすぐに、証明書がデバッグアクセス許可に関連していた場合、RSS-DA は Cortex-M33 のデバッグオープンをプログラムします。または、証明書は部分的または完全なリグレッションを承認して、リグレッションされた部分のデバッグを可能にすることもできます。
 - TrustZone が無効なとき (TZEN = 0xC3)、認証方法はパスワードに基づきます。この方法では、製品の完全なリグレッションのみを制御できます。
5. 上記の再オープンは、SBS_HDPLSR の HDPL が SBS_DBGCR の DBG_AUTH_HDPL でプログラムされた値以上のときのみ有効です。認証が失敗した場合、ホストインタフェースを通じてユーザに通知されます。

注： システム Flash メモリのデバッグ認証ライブラリは、SBS_HDPLSRレジスタのHDPL = 0 または 1 のときのみ使用可能です。このライブラリのみが、上記のステップ 3 および 4 を実行できます。

デバッグリセット

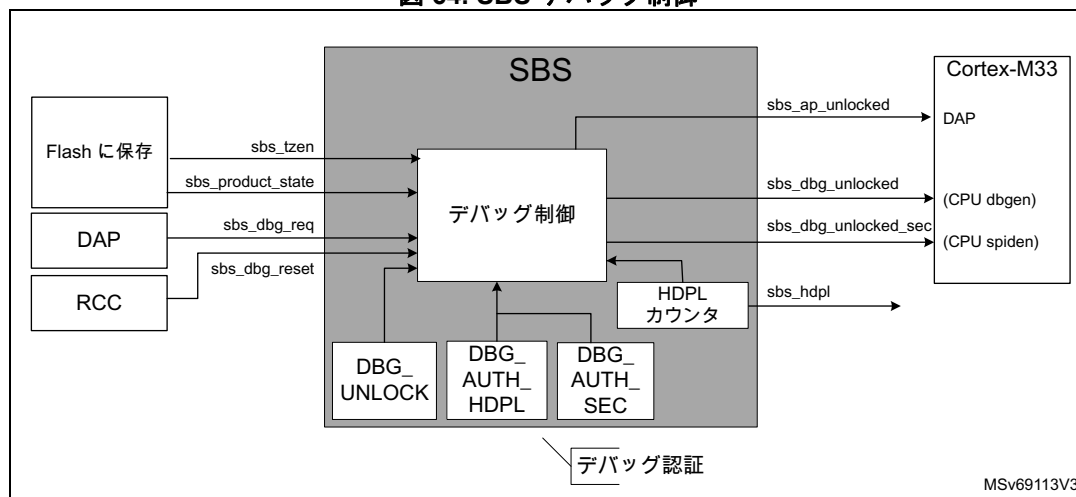
デバッグオープンは、DBGMCU レジスタフィールドに応じて、システムリセットまたはパワーオンリセットによってリセットされるように設定できます。

デバッグロック

デバッグ設定は、SBS_DBGLOCKR の DBGCFG_LOCK によってロックできます。その場合、SBS_DBGCR は書き込み不可になります。

DBGCFG_LOCK が 1 にセットされているときには、システムリセットまたはパワーオンリセットによってのみリセットできます。設定は、DBGMCU_CR の DCRT フィールドを使用して、DBGMCU を通じて行われます。

図 64. SBS デバッグ制御



デバッグオープンの制御に使用される入力：

- sbs_dbg_req：ホストがデバッグ認証プロトコルの起動をリクエストする入力信号。この信号がセットされると、ブートアドレスが設定されて、デバッグ認証ライブラリを起動します。
- sbs_tzen：TrustZone の有効化に関する、選択されたプラットフォームの設定を通知します。TZEN の値がゼロに設定されているとき (0xC3)、プラットフォームは TrustZone をサポートせず、単純化されたライフサイクルが提示されます。詳細は、[セクション 7: 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。

- sbs_product_state : 現在の PRODUCT_STATE (Flash メモリから) を提供します。詳細は、[セクション 7 : 内蔵 Flash メモリ \(FLASH\)](#) を参照してください。
- sbs_dbg_reset : RCC から着信した情報をリセットし、SBS_DBGCR をリセットします (このレジスタがリセットされるように設定されている場合)。

設定

- SBS_DBGCR の DBG_UNLOCK : DBG_AUTH_HDPL に達したときにデバッグアンロック。
- SBS_DBGCR の AP_UNLOCK : アクセスポートのアンロック。
- SBS_DBGCR の DBG_AUTH_HDPL : 認証済みデバッグの時間的分離レベル。デバッグをオープンできる HDPL 値を定義します。値は、1 から 3 までです。
- SBS_DBGCR の DBG_AUTH_SEC : デバッグの再オープンが非セキュアの場合のみか、両方 (セキュアと非セキュア) の場合かを指定します。
- SBS_DBGLOCKR の DBG_LOCK : 現在のデバッグ設定をロックします (リセットによってのみリリース)。

出力

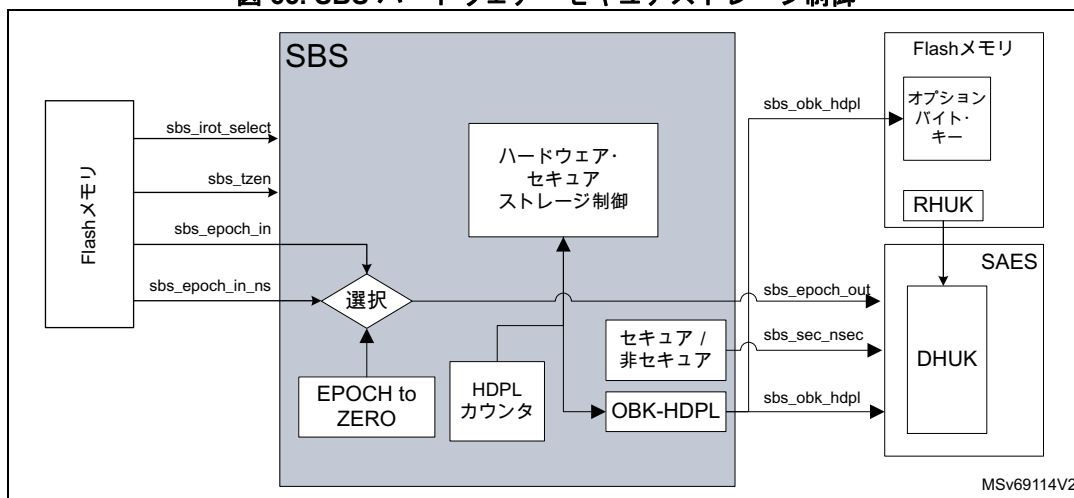
- sbs_ap_unlocked : Cortex-M33 アクセスポートを制御する信号。
 - 1 : 0xB4
 - 0 : その他すべてのコード
- sbs_dbg_unlocked : Cortex-M33 非セキュアのデバッグをアンロックします (1 に設定されているとき)。
 - 1 : 0xB4
 - 0 : その他すべてのコード
- sbs_dbg_unlocked_sec : Cortex-M33 セキュアのデバッグ (1 にセットされているとき) をアンロックします。
 - 1 : 0xB4
 - 0 : その他すべてのコード

14.3.7 SBS ハードウェア・セキュアストレージ制御

この機能は、デバッグを再オープンするとき (製品が現場にあるとき)、ROT (root-of-trust : 信頼の起点) に関するキーとデータの分離を確保します。

これには、Flash メモリ内の OB-Keys という名前の専用領域 (詳細については、[セクション 7 : 内蔵 Flash メモリ \(FLASH\)](#) を参照) と、識別されたドメインについて異なるハードウェアキーを使用してデータを保護するキー導出 (DHUK) メカニズムが含まれます。

図 65. SBS ハードウェア・セキュアストレージ制御



- **sbs_obk_hdpl** : 現在の HDPL のセキュアストレージコンテキストまたはそれ以上のものを選択します (プロビジョニングを可能にするため)。この信号は、SBS_NEXTHDPLCR レジスタによって設定されます。この情報は、Flash メモリによって正しい領域を選択するために使用され、また、SAES が導出キーを処理して HDPL コンテキストに関連するキー導出を生成するために使用されます。
- **sbs_epoch_out** : EPOCH カウンタは、REPLAY 保護を管理するために使用されます。これらのカウンタは、リグレーションが行われるとインクリメントされます。EPOCH を DHUK (SAES 内) に挿入すると、DHUK で暗号化されたすべての情報がリグレーション後に再生されなくなります。
- **sbs_sec_nsec** : セキュアおよび非セキュアアセットについて、異なる DHUK を適用します。

SAES/DHUK を使用して暗号化されたすべてのキーは、RHUK のプロパティ (デバイスごとに一意) を継承します。

SAES により DHUK を使用して暗号化されたすべてのデータは、
[sbs_obk_hdpl + sbs_epoch_out + sbs_sec_nsec] の組み合わせに固有です。

ハードウェア・セキュアストレージ制御に使用される入力

sbs_epoch_in and **sbs_epoch_in_ns** : Flash メモリからの 24 ビット値であり、セキュアと非セキュアそれぞれのリグレーション・カウンタを表します。

設定

- **SBS_NEXTHDPLCR** の **NEXTHDPL** : 次の HDPL セキュアストレージ領域にアクセスします。
- **SBS_EPOCHSELCR** の **EPOCH_SEL** : 制御下のセキュアストレージに関する EPOCH ソースを選択します (SEC_EPOCH、NS_EPOCH、または FORCED_To_Zero)。

表 120. OBK-HDPL ロジック

SBS_HDPLSR の HDPL[7:0]	NEXTHDPL[1:0]			
	0x0	0x1	0x2	0x3
0 (0xB4)	0xB4	0x51	0x8A	0x6F
1 (0x51)	0x51	0x8A	0x6F	
2 (0x8A)	0x8A	0x6F		
3 (0x6F)	0x6F			
その他				

14.4 SBS 割込み

SBS は割込みをサポートしません。

14.5 SBS レジスタ

14.5.1 SBS 時間的分離 (temporal isolation) 制御レジスタ (SBS_HDPLCR)

アドレスオフセット : 0x010

リセット値 : 0x0000 00B4

リセット : システムリセット

レジスタセキュリティ : 制限なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INCR_HDPL[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **INCR_HDPL[7:0]** : HDPL 値をインクリメント

0xB4 : インクリメントなし

0x6A : HDPL レベルを 1 だけインクリメントするための推奨値

その他 : その他すべての値では、HDPL レベルのインクリメントが可能です。

14.5.2 SBS 時間的分離ステータスレジスタ (SBS_HDPLSR)

アドレスオフセット : 0x014

リセット値 : 0xFFFF XXXX

リセット値は、ブートケースに依存 : ST コードの場合は HDPL0 でブート、その他すべてのケースでは HDPL1。詳細については、表 119 を参照してください。

リセット : システムリセット

レジスタセキュリティ : 制限なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDPL[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **HDPL[7:0]** : 時間的分離レベル

このビットフィールドは、現在の時間的分離レベルを返します。

0xB4 : HDPL0、RSS

0x51 : HDPL1、iRoT

0x8A : HDPL2、uRoT

0x6F : HDPL3、アプリケーション (セキュア/非セキュア)

14.5.3 SBS 次の HDPL 制御レジスタ (SBS_NEXTHDPLCR)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : 制限なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NEXTHDPL[1:0]	
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **NEXTHDPL[1:0]** : 現在より高い HDPL を指すインデックス

次のセキュアストレージ領域を (OBK-HDPL を通じて) 指すように、現在の HDPL に追加するインデックス (OBK-HDPL = HDPL + NEXTHDPL) .詳細は、表 120 : OBK-HDPL ロジックを参照してください。

14.5.4 SBS デバッグ制御レジスタ (SBS_DBGCR)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

リセット : デバッグリセット (システムリセットまたはパワーオンリセット)

レジスタセキュリティ : HDPL0/1、それ以外は RAZ/WI (Read-As-Zero, Writes Ignored : 読み出しは0、書き込みは無効)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_AUTH_SEC[7:0]								DBG_AUTH_HDPL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_UNLOCK[7:0]								AP_UNLOCK[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **DBG_AUTH_SEC[7:0]** : デバッグオープン (セキュア/非セキュア) を制御

セキュアおよび非セキュアについてデバッグをオープンするには、このビットフィールドに 0xB4 を書き込みます。

その他の値を書き込むと、非セキュアでオープンします。

ビット 23:16 **DBG_AUTH_HDPL[7:0]** : 認証済みデバッグ時間的分離レベル

このビットフィールドへの書き込みによって、認証済みデバッグがオープンする HDPL が定義されます。

0x51 : HDPL1

0x8A : HDPL2

0x6F : HDPL3

注 : 他の値を書き込むと、無視されます。他の値の読出しは、デバッグがオープンしないことを意味します。

ビット 15:8 **DBG_UNLOCK[7:0]** : DBG_AUTH_HDPL に達したときにデバッグアンロック

このビットフィールドに 0xB4 を書き込むと、SBS_HDPLSR の HDPL がこのレジスタの DBG_AUTH_HDPL に等しいときにデバッグをオープンします。

ビット 7:0 **AP_UNLOCK[7:0]** : アクセスポートのアンロック

このビットフィールドに 0xB4 を書き込むと、デバイスアクセスポートが開きます。

14.5.5 SBS デバッグロックレジスタ (SBS_DBGLOCKR)

アドレスオフセット : 0x024

リセット値 : 0x0000 00B4

リセット : デバッグリセット (システムリセットまたはパワーオンリセット)

レジスタセキュリティ : HDPL0/1、それ以外は RAZ/WI

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGCFG_LoCK[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DBGCFG_LOCK[7:0]** : デバッグ設定のロック

このビットフィールドの値が 0xB4 以外の場合、このビットフィールドを読み出すと、0x6A が返されます。

このビットフィールドを使用してデバッグ設定ロックするための推奨値は、0xC3 です。

0xB4 : SBS_DBGCR への書き込み可能 (デフォルト)

その他 : SBS_DBGCR への書き込みは無視される

14.5.6 SBS RSS コマンドレジスタ (SBS_RSSCMDR)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

リセット : パワーオンリセット

レジスタセキュリティ : 常にセキュア (非セキュアの場合は RAZ/WI)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSSCMD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RSSCMD[15:0]** : RSS コマンド

アプリケーションはこのビットフィールドを使用して、次のリセット時に実行されるコマンドを RSS に渡すことができます。

RSSCMD ≠ 0 かつ PRODUCT_STATE が Open のとき、ブートピンの値に関係なく、システムは常に RSS でブートします。

14.5.7 SBS EPOCH 選択制御レジスタ (SBS_EPOCHSELCR)

アドレスオフセット : 0x0A0

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : TZ_STATE = 1 のときセキュア (非セキュアアクセスでは RAZ、WI) 。 TZ_STATE = 0 のときには、非セキュア保護。このレジスタは、TZ_STATE に関係なく特権によって保護され、非特権アクセスの場合は RAZ/WI。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EPOCH_SEL [1:0]	
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **EPOCH_SEL[1:0]** : SAES に送信される EPOCH 値を選択します。

00 : NS_EPOCH (非セキュア) カウンタ入力を選択されます。

01 : SEC_EPOCH カウンタ入力を選択されます。

1x : EPOCH はゼロに強制されます (ブート時に PUF 参照値を取得するために使用される値)。

14.5.8 SBS セキュリティモード設定制御レジスタ (SBS_SECCFGR)

アドレスオフセット : 0x0C0

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : 常にセキュア。非セキュアトランザクション (非セキュア通信) かつ TZ_STATE = 1 の場合、RAZ/WI。TZ_STATE = 0 の場合、RAZ/WI。

このレジスタは、ユーザがシステムレジスタを通じて設定可能な機能をセキュアまたは非セキュアにしたい場合に、セキュアソフトウェアによってプログラムされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPUSEC	Res.	CLASSBSEC	SBSSEC
												rw		rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **FPUSEC** : FPU セキュリティ有効

0 : SBS_FPUIMP レジスタは、セキュアまたは非セキュアトランザクションを通じてアクセス可能

1 : SBS_FPUIMP レジスタは、セキュアトランザクションを通じてのみアクセス可能

注 : このビットは、特権トランザクションを通じてのみ書き込むことができます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CLASSBSEC** : クラス B セキュリティ有効

0 : SBS_CFGR2 レジスタは、セキュアまたは非セキュアトランザクションを通じてアクセス可能

1 : SBS_CFGR2 レジスタは、セキュアトランザクションを通じてのみアクセス可能

ビット 0 **SBSSEC**: SBS クロック制御、メモリ消去ステータスレジスタ、および補正セルレジスタセキュリティ有効

0 : SBS_MESR、SBS_CCCSR、SBS_CCVALR、SBS_CCWCWCR レジスタは、セキュアまたは非セキュアトランザクションを通じてアクセス可能

1 : SBS_MESR、SBS_CCCSR、SBS_CCVALR、SBS_CCWCWCR レジスタは、セキュアトランザクションを通じてのみアクセス可能

14.5.9 SBS プロダクト・モードおよび設定レジスタ (SBS_PMCR)

アドレスオフセット : 0x100

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ :

- TrustZone が有効なとき (TZEN = 0xB4) :
 - ANASWVDD の ADC1_2_SEC_EN セキュア入力と IO_ANA_BOOST_EN 設定ビットに依存
 - PBx_FMP 設定ビットの I/O PBx_SEC_EN セキュア入力に依存
 - ETH_SEL_PHY 設定ビットの Ethernet ETH_SEC_EN セキュア入力に依存
 - SMPS_DIV_CLOCK_EN の SBS_SECCFGR の SDCE_SEC_EN に依存
- TrustZone が無効なとき (TZEN = 0xC3)、アクセス制限はありません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH_SEL_PHY[2:0]			Res.	PB9_FMP	PB8_FMP	PB7_FMP	PB6_FMP
								rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:21 **ETH_SEL_PHY[2:0]** : Ethernet PHY インタフェース選択

- 000 : GMII または MII
- 001 : 予約済み (RGMII)
- 100 : RMII
- その他 : 予約済み

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **PB9_FMP** : PB9 の高速モードプラス駆動機能有効化

- このビットは、GPIOB で PB9 がセキュアの場合、セキュアアクセスでのみ読み書き可能です。このビットは、I2C ペリフェラルで PB9 が使用されないとき、PB9 の Fm+ 駆動モードを有効にします。たとえば、これを使用して LED を駆動できます。
- 0 : PB9 ピンは標準モードで動作します。
- 1 : Fm+ モードは PB9 ピンで有効にされ、速度制御はバイパスされます。

ビット 18 **PB8_FMP** : PB8 の高速モードプラス駆動機能有効化

- このビットは、GPIOB で PB8 がセキュアの場合、セキュアアクセスでのみ読み書き可能です。このビットは、I2C ペリフェラルで PB8 が使用されないとき、PB8 の Fm+ 駆動モードを有効にします。たとえば、これを使用して LED を駆動できます。
- 0 : PB8 ピンは標準モードで動作します。
- 1 : Fm+ モードは PB8 ピンで有効にされ、速度制御はバイパスされます。

ビット 17 **PB7_FMP** : PB7 の高速モードプラス駆動機能有効化

- このビットは、GPIOB で PB7 がセキュアの場合、セキュアアクセスでのみ読み書き可能です。このビットは、I2C ペリフェラルで PB7 が使用されないとき、PB7 の Fm+ 駆動モードを有効にします。たとえば、これを使用して LED を駆動できます。
- 0 : PB7 ピンは標準モードで動作します。
- 1 : Fm+ モードは PB7 ピンで有効にされ、速度制御はバイパスされます。

ビット 16 **PB6_FMP** : PB6の高速モードプラス駆動機能有効化

このビットは、GPIOB で PB6 がセキュアの場合、セキュアアクセスでのみ読み書き可能です。このビットは、I2C ペリフェラルで PB6 が使用されないとき、PB6 の Fm+ 駆動モードを有効にします。たとえば、これを使用して LED を駆動できます。

0 : PB6 ピンは標準モードで動作します。

1 : Fm+ モードは PB6 ピンで有効にされ、速度制御はバイパスされます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 予約済みであり、リセット値に保持する必要があります。

14.5.10 SBS FPU 割込みマスクレジスタ (SBS_FPUIMR)

アドレスオフセット : 0x104

リセット値 : 0x0000 001F

リセット : システムリセット

レジスタセキュリティ : SBS_SECCFGR の FPUSEC に依存

このレジスタは、特権トランザクションを通じてのみアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPU_IE[5:0]					
										rW	rW	rW	rW	rW	rW

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **FPU_IE[5:0]** : FPU 割込み有効

Cortex-M33 FPU 割込みを有効にするために、ソフトウェアによってセットおよびクリアされます。

FPU_IE[5] : 不正確な割込み有効 (リセット時割込み無効)

FPU_IE[4] : 入力異常割込み有効

FPU_IE[3] : オーバーフロー割込み有効

FPU_IE[2] : アンダーフロー割込みイネーブル

FPU_IE[1] : 0 による除算割込み有効

FPU_IE[0] : 無効操作割込み有効

14.5.11 SBS メモリ消去ステータスレジスタ (SBS_MESR)

アドレスオフセット : 0x108

リセット値 : 0x0000 000X (ビット 0 はシステムリセットの影響を受けません)。

レジスタセキュリティ : SBS_SECCFGR の SBSSEC に依存

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPMEE
															rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCLR
															rc_w1

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **IPMEE** : ICACHE 消去ステータス

このビットは、タンパ検出の可能性または製品状態のリグレーションがあった後、ICACHE および PKA RAM の消去が完了したときにハードウェアによってセットされます (詳細については、[セクション 47: タンパおよびバックアップレジスタ \(TAMP\)](#) を参照)。

このビットは、ソフトウェアによって 1 を書き込むことによってクリアされます。

0 : ICACHE および PKA RAM の消去中

1 : ICACHE および PKA SRAM の消去完了

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **MCLR** : デバイスメモリ消去ステータス

このビットは、パワーオンリセットまたはタンパ検出または製品状態のリグレーション後に SRAM2、BKPSRAM、ICACHE、DCACHE、および PKA RAM の消去が完了したときにハードウェアによってセットされます ([セクション 47: タンパおよびバックアップレジスタ \(TAMP\)](#) を参照)。

このビットは、システムリセットによってリセットされず、ソフトウェアによって 1 を書き込むことでクリアされます。

0 : ソフトウェアによってまだクリアされていない場合、メモリ消去中

1 : メモリ消去完了

14.5.12 SBS I/O 制御およびステータスレジスタの補正セル (SBS_CCCSR)

アドレスオフセット : 0x110

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : SBS_SECCFGR の SBSSEC に依存

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RDY2	RDY1	Res.	Res.	Res.	Res.	CS2	EN2	CS1	EN1
						r	r					rw	rw	rw	rw

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **RDY2** : VDDIO2 補正セルレディ・フラグ

このビットは、VDDIO2 補正セルのステータスを提供します。

0 : VDDIO2 補正セルはレディ状態ではありません。

1 : VDDIO2 補正セルはレディ状態です (セルによって提供されるコード値を使用できます)。

ビット 8 **RDY1** : VDDIO 補正セルレディフラグ

このビットは、補正セルのステータスを提供します。

0 : VDDIO 補正セルはレディ状態ではありません。

1 : VDDIO 補正セルはレディ状態です (セルによって提供されるコード値を使用できます)。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CS2** : VDDIO2 電源レールのコード選択 (リセット値は 1 にセット)

このビットは、I/O 補正セルに適用するコードを選択します。

0 : セルからのコード (SBS_CCVR で入手可能)

1 : SBS_CCCR からコード

ビット 2 **EN2** : VDDIO2 電源レールの補正セルを有効にします。

このビットは、I/O 補正セルを有効にします。

0 : I/O 補正セルは無効です。

1 : I/O 補正セルは有効です。

ビット 1 **CS1** : VDDIO 電源レールのコード選択 (リセット値は 1 にセット)

このビットは、I/O 補正セルに適用するコードを選択します。

0 : セルからのコード (SBS_CCVR で使用可能)

1 : SBS_CCCR からコード

ビット 0 **EN1** : VDDIO 電源レールの補正セルを有効にします。

このビットは、I/O 補正セルを有効にします。

0 : I/O 補正セルは無効です。

1 : I/O 補正セルは有効です。

14.5.13 SBS I/O 値レジスタの補正セル (SBS_CCVALR)

アドレスオフセット : 0x114

リセット値 : 0x0000 0088

リセット : システムリセット

レジスタセキュリティ : SBS_SECCFGR の SBSSEC に依存

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APSRC2 [3:0]				ANSRC2 [3:0]				APSRC1 [3:0]				ANSRC1 [3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **APSRC2[3:0]** : PMOS トランジスタの補正值

この値は、セルによって提供され、機能範囲のスルーレートを補正するためにプロセッサによって認識される必要があります。

ビット 11:8 **ANSRC2[3:0]** : NMOS トランジスタの補正值

この値は、セルによって提供され、機能範囲のスルーレートを補正するためにプロセッサによって認識される必要があります。

ビット 7:4 **APSRC1[3:0]** : PMOS トランジスタの補正值

この値は、セルによって提供され、機能範囲のスルーレートを補正するためにプロセッサによって認識される必要があります。

ビット 3:0 **ANSRC1[3:0]** : NMOS トランジスタの補正值

この値は、セルによって提供され、機能範囲のスルーレートを補正するためにプロセッサによって認識される必要があります。

14.5.14 SBS I/O ソフトウェアコードレジスタの補正セル (SBS_CCSWCR)

アドレスオフセット : 0x118

リセット値 : 0x0000 7878

リセット : システムリセット

レジスタセキュリティ : SBS_SECCFGR の SBSSEC に依存

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW_APSRC2 [3:0]				SW_ANSRC2 [3:0]				SW_APSRC1 [3:0]				SW_ANSRC1 [3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **SW_APSRC2[3:0]** : V_{DDIO} 電源レールの PMOS 補正コード

このビットフィールドは、V_{DDIO} 電源レールの PMOS トランジスタの I/O 補正セルコードを定義するために、ソフトウェアによって書き込まれます。このコードは、SBS_CCSR で CS2 がセットされているときに、I/O に適用されます。

ビット 11:8 **SW_ANSRC2[3:0]** : V_{DDIO} 電源レールの NMOS 補正コード

このビットフィールドは、V_{DD} 電源レールの NMOS トランジスタの I/O 補正セルコードを定義するために、ソフトウェアによって書き込まれます。このコードは、SBS_CCSR で CS2 がセットされているときに、I/O に適用されます。

ビット 7:4 **SW_APSRC1[3:0]** : V_{DD} 電源レールの PMOS 補正コード

このビットフィールドは、V_{DDIO} 電源レールの PMOS トランジスタの I/O 補正セルコードを定義するために、ソフトウェアによって書き込まれます。このコードは、SBS_CCSR で CS1 がセットされているときに、I/O に適用されます。

ビット 3:0 **SW_ANSRC1[3:0]** : V_{DD} 電源レールの NMOS 補正コード

このビットフィールドは、V_{DD} 電源レールの NMOS トランジスタの I/O 補正セルコードを定義するために、ソフトウェアによって書き込まれます。このコードは、SBS_CCSR で CS1 がセットされているときに、I/O に適用されます。

14.5.15 SBS クラス B レジスタ (SBS_CFGR2)

アドレスオフセット : 0x120

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : SBS_SECCFGR の CLASSBSEC に依存

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCL	PVDL	SEL	CLL
												rs	rs	rs	rs

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ECCL** : ECC ロック

このビットは、ソフトウェアによってセット／クリアされます。これを使用して、TIM1/8/15/6/17 のブレーク入力による Flash メモリダブル ECC エラーを有効化およびロックできます。

0 : ダブル ECC エラーフラグはタイマブレーク入力に接続していない

1 : ダブル ECC エラーフラグはタイマブレーク入力に接続している

ビット 2 **PVDL** : PVD ロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。これを使用して、TIM1/8/15/16/17 ブレーク入力との PVD 接続を有効化およびロックできます。

0 : PVD 割込みはタイマブレーク入力から切断PWR レジスタの PVD_EN および PVD_SEL[2:0] は読出し/書込み

1 : PVD 割込みはタイマブレーク入力に接続されています。PWR レジスタの PVD_EN および PVD_SEL[2:0] は読出し専用

ビット 1 **SEL** : SRAM ECC エラーロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。これを使用して、TIM1/8/15/16/17 のブレーク入力による SRAM ダブル ECC エラーシグナルを有効化およびロックできます。

0 : SRAM ダブル ECC エラーフラグはタイマブレーク入力に接続していない

1 : SRAM ダブル ECC エラーフラグはタイマブレーク入力に接続している

ビット 0 **CLL** : コアロックアップロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。これを使用して、TIM1/8/15/16/17 ブレーク入力による Cortex-M33 のロックアップ（ハードフォルト）出力を有効化およびロックできます。

0 : ロックアップ出力はタイマブレーク入力から切断されている

1 : ロックアップ出力はタイマブレーク入力に接続されている

14.5.16 SBS CPU 非セキュアロックレジスタ (SBS_CNSLCKR)

アドレスオフセット : 0x144

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : このレジスタは特権アクセスのみで読み書きできます。非特権アクセスは RAZ/WI です。

このレジスタは、Cortex-M33 の非セキュア MPU および VTOR_NS レジスタの設定をロックするために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKNSMPU	LOCKNSVTOR
														rs	rs

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **LOCKNSMPU** : 非セキュア MPU レジスタロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。セットされているとき、このビットは非セキュア MPU_CTRL_NS、MPU_RNR_NS、および MPU_RBAR_NS レジスタへの書き込みアクセスを無効にします。

0 : 非セキュア MPU レジスタへの書き込みは有効です。

1 : 非セキュア MPU レジスタへの書き込みは無効です。

ビット 0 **LOCKNSVTOR** : VTOR_NS レジスタロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。

0 : VTOR_NS レジスタへの書き込みは有効です。

1 : VTOR_NS レジスタへの書き込みは無効です。

14.5.17 SBS CPU セキュアロックレジスタ (SBS_CSLCKR)

アドレスオフセット : 0x148

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : このレジスタは、アクセスがセキュア/特権のときのみ書込み可能です。

非セキュア読み出し/書込みアクセスは RAZ/WI で、不正アクセスイベントが生成されます。システムが非セキュアなとき (TZ_STATE = 0)、このレジスタは RAZ/WI です。

このレジスタは、Cortex-M33 の AIRCR、SAU、セキュア MPU、および VTOR_S レジスタの PRIS および BFHFNMINS の設定をロックするために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKSAU	LOCKSMPU	LOCKSVTAIRCR
													rs	rs	rs

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 LOCKSAU : SAU レジスタロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。セットされているとき、このビットは、SAU_CTRL、SAU_RNR、SAU_RBAR、および SAU_RLAR レジスタへの書込みアクセスを無効にします。

0 : SAU レジスタへの書込みは有効です。

1 : SAU レジスタへの書込みは無効です。

ビット 1 LOCKSMPU : セキュア MPU レジスタロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。セットされているとき、このビットは、セキュア MPU_CTRL、MPU_RNR、および MPU_RBAR レジスタへの書込みアクセスを無効にします。

0 : セキュア MPU レジスタへの書込みは有効です。

1 : セキュア MPU レジスタへの書込みは無効です。

ビット 0 LOCKSVTAIRCR : VTOR_S および AIRCR レジスタロック

このビットは、ソフトウェアでセットされ、システムリセットでのみクリアされます。セットされているとき、このビットは、VTOR_S レジスタ、AIRCR レジスタの PRIS および BFHFNMINS ビットへの書込みアクセスを無効にします。

0 : VTOR_S レジスタ、AIRCR レジスタの PRIS および BFHFNMINS ビットへの書込みは有効です。

1 : VTOR_S レジスタ、AIRCR レジスタの PRIS および BFHFNMINS ビットへの書込みは無効です。

14.5.18 SBS flitf ECC NMI マスクレジスタ (SBS_ECCNMIR)

アドレスオフセット : 0x14C

リセット値 : 0x0000 0000

リセット : システムリセット

レジスタセキュリティ : TZ_STATE = 1 のときのみセキュアアクセス (非セキュアアクセスでは RAZ/WI)。TZ_STATE = 0 の場合、セキュリティ保護なし。

このレジスタは、特権トランザクションを通じてのみアクセス可能です。

このレジスタは、Flash メモリからのダブル ECC エラーに関する NMI の期待される動作をセットアップします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCNMI_MASK_EN
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **ECCNMI_MASK_EN** : flitf データ部分でダブル ECC エラーが発生したときの NMI 動作のセットアップ

- 0 : flitf データ部分でダブル ECC エラーが発生した場合、NMI が生成されます。
- 1 : flitf データ部分でダブル ECC エラーが発生した場合、NMI は生成されません。

表 121. SBS レジスタのマップとリセット値

表 121. SBS レジスタのマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x114	SBS_CCVALR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	APSRC2 [3:0]				ANSRC2 [3:0]				APSRC1 [3:0]				ANSRC1 [3:0]							
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0				
0x118	SBS_CCSWCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SW_APSRC2 [3:0]				SW_ANSRC2 [3:0]				SW_APSRC1 [3:0]				SW_ANSRC1 [3:0]							
	リセット値																	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0				
0x120	SBS_CFGR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ECCL	PVDL	SEL	CLL			
	リセット値																													0	0	0	0				
0x124 から 0x140	予約済み	予約済み																																			
0x144	SBS_CNSLCKR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LOCKNSMPU	LOCKNSVTR				
	リセット値																														0	0	0				
0x148	SBS_CSLCKR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LOCKSAU	LOCKSMPU	LOCKSVTAIRCR				
	リセット値																														0	0	0				
0x14C	SBS_ECCNMIR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ECCNMI_MASK_EN					
	リセット値																																				

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

15 ペリフェラル相互接続マトリックス

15.1 相互接続マトリックスの概要

いくつかのペリフェラルの間には直接接続があり、自律的な通信や動機が可能です。このアプローチにより、CPU リソースと消費電力が節減されます。さらに、これらのハードウェア接続によってソフトウェアの遅延がなくなり、予測可能なシステム設計が可能になります。

ペリフェラルに応じて、これらの相互接続は、RUN、SLEEP、および STOP モードで動作できます。

15.2 接続の一覧

表 122. ペリフェラル相互接続マトリックス^{(1) (2)}

転送元	転送先																			
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM12	TIM15	TIM16	TIM17	LPTIM1/2/3	LPTIM4	LPTIM5/6	ADC1/2	DAC1/2	GPDMA1/2	TAMP	RTC
TIM1	-	1	1	1	1	1	-	-	1	1	-	-	-	-	-	2	4	-	-	-
TIM8	1	-	1	1	1	1	-	-	1	1	-	-	-	-	-	2	4	-	-	-
TIM2	1	1	-	1	1	1	-	-	1	1	-	-	-	-	-	2	4	10	-	-
TIM3	1	1	1	-	1	1	-	-	1	1	-	-	-	-	-	2	-	-	-	-
TIM4	1	1	1	1	-	1	-	-	1	1	-	-	-	-	-	2	4	-	-	-
TIM5	1	1	1	1	1	-	-	-	1	1	-	-	-	-	-	-	4	-	-	-
TIM6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	4	-	-	-
TIM7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-
TIM12	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	4	10	-	-
TIM13	1	1	1	1	1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	-
TIM14	1	1	1	1	1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	-
TIM15	1	1	1	1	1	-	-	-	1	-	-	-	-	-	-	2	4	10	-	-
TIM16	1	1	1	1	1	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-
TIM17	1	1	1	1	1	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-
LPTIM1/2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	4	10	-	-
LPTIM3/4/5/6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	-	-
ADC1	3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ADC2	3	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13	-
GPDMA1/2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	-	-
EXTI	-	-	-	-	-	-	-	-	-	-	-	-	6	6	6	2	4	10	-	-
RTC ウェイクアップ	-	-	-	-	-	7	-	-	-	-	7	-	-	-	-	-	-	10	-	-
RTC アラーム	-	-	-	-	-	-	-	-	-	-	-	-	6	6	6	-	-	10	-	-
TAMP	-	-	-	-	-	-	-	-	-	-	-	-	6	6	6	-	-	10	-	14
HSE	-	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-
LSE	-	-	5	-	-	-	-	-	-	5	5	-	-	-	-	-	12	-	-	-
LSE の CSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13	-
CSI	-	-	-	-	-	-	-	-	5	5	-	-	-	-	-	-	-	-	-	-
HSI	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-

表 122. ペリフェラル相互接続マトリックス^{(1) (2)} (続き)

転送元	転送先																				
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM12	TIM15	TIM16	TIM17	LPTIM1/2/3	LPTIM4	LPTIM5/6	ADC1/2	DAC1/2	GPDMA1/2	TAMP	RTC	AES/SAES
LSI	-	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-	-
MCO1	-	-	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-
MCO2	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-
V _{CORE}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	-	-	-	-
V _{REFINT}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	-	-	-	-
V _{sensor}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	-	-	-	-
V _{BAT8}	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	-	-	-	-
V _{BAT} /温度モニタ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13	-	-
システムエラー	8	8	-	-	-	-	-	-	-	8	8	8	-	-	-	-	-	-	-	-	-
システム Flash メモリ	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15
AES/SAES	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15
Ethernet	-	-	9	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1. この表の数字は、[セクション 15.3](#) の対応するサブセクションにリンクしています。
2. 「-」は、相互接続がないことを意味します。

15.3 相互接続の詳細

15.3.1 タイマのマスタからスレーブへの相互接続

タイマ（TIM1/2/3/4/5/8/12/13/14/15/16/17）からタイマ（TIM1/2/3/4/5/8/12/15）へ。

目的

一部のタイマは、同期または連鎖のために、内部でリンクされています。

マスタモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

機能の説明については、[セクション 39.4.23 : タイマの同期](#)を参照してください。

同期モードの詳細については、以下を参照してください。

- [セクション 38.3.30](#)（高機能制御タイマ TIM1/8）
- [セクション 39.4.22](#)（汎用タイマ TIM2/3/4/5）
- [セクション 41](#)（汎用タイマ TIM12/13/14）
- [セクション 42.4.23](#)（汎用タイマ TIM15）

トリガ信号

出力（マスタから）は、設定可能なタイマイベントに応じた信号 TIMx_TRGO（および TIM1/8 では TIMx_TRGO2）で行われます。TIM16/17 の場合は、信号 tim16_oc1 および tim17_oc1 でも行われます。入力（スレーブへの）は、信号 TIMx_ITR0/1/2/3 で行われます。

使用可能なマスタ／スレーブ接続を以下に示します。

- [表 381](#)（高機能制御タイマ TIM1/8）
- [表 404](#)（汎用タイマ TIM2/3/4/5）
- [表 425](#)（汎用タイマ TIM12/13/14）
- [表 437](#)（汎用タイマ TIM15）

アクティブな電力モード

タイマは、オブションで RUN および SLEEP モードでアクティブです。低電力モードが TIMx に与える影響については、以下を参照してください。

- [表 393](#)：低消費電力モードが TIM1/TIM8 に与える影響
- [表 411](#)：低消費電力モードが TIM2/TIM3/TIM4/TIM5 に与える影響
- [表 427](#)：低消費電力モードが TIM12/TIM13/TIM14 に与える影響
- [表 443](#)：低消費電力モードが TIM15/TIM16/TIM17 に与える影響

15.3.2 ADC へのトリガ

EXTI、タイマ（TIM1/2/3/4/6/8/15）、および LP タイマ（LPTIM1/2）から ADC1/ADC2 へ。

目的

変換（または一連の変換）は、ソフトウェアによって、または外部イベント（タイマキャプチャまたは入力ピンなど）によってトリガされます。ADC12 の場合、EXTEN[1:0] ビット（レギュラ変換の場合）または JEXTEN[1:0] ビット（インジェクト変換の場合）が 0b00 以外の場合、外部イベントは選択された極性で変換をトリガできます。

詳細については、以下を参照してください。

- [セクション 26.4.18：外部トリガおよびトリガ極性での変換 \(EXTSEL、EXTEN、JEXTSEL、JEXTEN\)](#)
- EXTEN[1:0] は、[ADC 設定レジスタ \(ADC_CFGR\)](#) で定義されています。
- JEXTEN[1:0] は、[ADC インジェクトシーケンスレジスタ \(ADC_JSQR\)](#) で定義されています。

汎用タイマ (TIM2/3/4)、基本タイマ (TIM6)、高機能制御タイマ (TIM1/8)、および汎用タイマ (TIM15) を使用して、タイマ出力 tim_oc および tim_trgo を通じて ADC トリガイベントを生成できます。

低電力タイマ (LPTIM1/2) を使用して、チャンネル 11 および 15 の EXTI に加えて、LPTIM チャンネルを通じて、ADC トリガイベントを生成できます。

トリガ信号

ADC1/ADC2 の場合、入力トリガ信号と、ADC1/ADC2 とタイマとの相互接続の説明については、以下を参照してください。

- adc_ext_trg : [表 242：ADC 相互接続](#)
- adc_jext_trg : [表 242：ADC 相互接続](#)
- [セクション 26.4.18：外部トリガおよびトリガ極性での変換 \(EXTSEL、EXTEN、JEXTSEL、JEXTEN\)](#)
- [セクション 26.4.25：タイミング図の例 \(シングル/連続モード、ハードウェア/ソフトウェアトリガ\)](#)

アクティブな電力モード

この相互接続は、すべての ADC について、RUN および SLEEP モードでアクティブです。タイマは、RUN および SLEEP モードでのみアクティブです。低電力モードの影響については、以下を参照してください。

- [表 393：低消費電力モードが TIM1/TIM8 に与える影響](#)
- [表 411：低消費電力モードが TIM2/TIM3/TIM4/TIM5 に与える影響](#)
- [表 443：低消費電力モードが TIM15/TIM16/TIM17 に与える影響](#)
- [表 449：STM32H563/H573および STM32H562 LPTIM 機能](#)
- [表 464：低消費電力モードが LPTIM に与える影響](#)

15.3.3 タイマのトリガとしての ADC アナログウォッチドッグ

ADC1/ADC2 から TIM1/8 へ。

目的

ADC1/ADC2 から着信する内部アナログウォッチドッグ出力信号は、オンチップタイマに接続されません。ADC1/ADC2 は、アナログウォッチドッグ信号を通じて、高機能制御タイマ (TIM1/8) に入力イベントを供給し、カウンタのリセット、開始、停止、またはクロック供給を行うことができます。

ADC アナログウォッチドッグとタイマトリガの設定の説明については、以下を参照してください。

- [セクション 38.3.6：外部トリガ入力 \(TIM1/8\)](#)
- [表 382 \(TIM1/8 \(tim_etr\) 入力マルチプレクサに接続された内部 ADC1/ADC2 ソース\)](#)
- [セクション 26.4.28 \(ADC1/ADC2/ADC_AWDy_OUT 信号出力生成\)](#)

トリガ信号

出力（ADC からの）は、信号 ADCn_AWDx_OUT で行われます。n は ADC インスタンスであり、x = 1、2、3 です（1 つの ADC につき 3 つのウォッチドッグ）。入力（タイマへの）は、信号 TIMx_ETR（外部トリガ）で行われます。

アクティブな電力モード

ADC1/ADC2 は、RUN および SLEEP モードでアクティブです。

15.3.4 DAC へのトリガ

タイマ（TIM1/2/4/5/6/7/8/15）、低電力タイマ（LPTIM1/2）、および EXTI から DAC へ。

目的

汎用タイマ（TIM2/4/5/15）、基本タイマ（TIM6/7）、高機能制御タイマ（TIM1/8）、LP タイマ（LPTIM1/2）出力チャネル（lptim1_ch1 および lptim2_ch1）、および EXTI をトリガイメントとして使用して、DAC 変換を開始できます。

トリガ信号

TIMx_TRGO 信号での出力（タイマからの）と LP タイマからの出力は、対応する DAC 入力に直接接続されます。

DAC での入力トリガの選択については、以下を参照してください。

- [表 266 : DAC の相互接続](#)
- [セクション 28.4.8 : DAC トリガ選択](#)

アクティブな電力モード

相互接続は、RUN、SLEEP、および STOP モードでアクティブです。

15.3.5 タイマへのクロックソース

HSE、LSE、LSI、HSI、および MCO からタイマ（TIM2/12/15/16/17）および LP タイマ（LPTIM1/2）へ。

目的

タイマ入力またはカウンタは、異なるクロックソースから受信でき、たとえば、リファレンスクロックでの内部オシレータの較正に使用できます。

外部クロック（HSE、LSE）、内部クロック（LSI、CSI、HSI）、およびマイクロコントローラ出力クロック（MCO）は、タイマへの入力として使用できます。

- LSE、HSI、および CSI は、外部入力信号として汎用タイマ TIM2 に割り当てられます。LSE は、モード 1（tim_ti1_in）およびモード 2（外部トリガ入力 tim_etr_in）の外部クロックソースによって供給されるカウンタクロックとして選択できます。入力割り当てとクロック選択の説明については、以下を参照してください。
 - [セクション 39.4.5 : クロック選択（TIM2）](#)
 - 外部クロックモード 1 : [表 423 : tim_ti1 入力マルチプレクサへの相互接続](#)（TIM12、tim_ti1_in4（HSI）、および tim_ti1_in5（CSI））
 - 外部クロックモード 2 : [表 405 : tim_etr 入力マルチプレクサへの相互接続](#)（tim_etr3（LSE））
- LSE、LSI、CSI、および HSE は、外部入力信号として汎用タイマ TIM15/16/17 に割り当てられます。LSE/LSI/CSI/HSE は、モード 1 の外部クロックソース（tim_ti1 または tim_ti2 信号）に

よって供給されるカウンタクロックとして選択できます。入力割り当てとクロック選択の説明については、以下を参照してください。

- セクション 42.4.6 : クロック選択 (TIM15/16/17) .外部クロックモード 1 : 外部入力ピン (使用可能な場合、tim_ti1 または tim_ti2)
- 表 423 : tim_ti1 入力マルチプレクサへの相互接続、tim_ti1_in1 (LSI-TIM16)、tim_ti1_in2 (LSE-TIM16/HSE-TIM17)、tim_ti1_in4 (LSE- TIM15)、および tim_ti1_in5 (CSI-TIM15)
- マイクロコントローラ出力クロック (MCO) : MCO1 は外部入力として汎用タイマ TIM17 に接続され、MCO2 は外部入力として汎用タイマ TIM15 に接続されます。
 - 表 435 : tim_ti1 入力マルチプレクサへの相互接続 (TIM15/TIM16/TIM17)
- LSI および LSE は、表 458 : LPTIM1 入力キャプチャ 2 の接続で説明されているように、LPTIM1 への入力キャプチャ 2 として選択できます。
- HSI/1024、CSI/128、および HSI/8 は、表 459 : LPTIM2 入力キャプチャ 2 の接続で説明されているように、LPTIM2 への入力キャプチャ 2 として選択できます。

トリガ信号

lptim_ic2_mux1 LPTIM 入力キャプチャ選択は、LPTIM 設定レジスタ 2 (LPTIM_CFGR2) で設定できます。タイマの場合、内部クロック信号は、モード 1 (tim_ti1_in) およびモード 2 (外部トリガ入力 tim_etr_in) の外部クロックソースとして選択できます。 .

アクティブな電力モード

この機能は、RUN および SLEEP モードで使用できます。

15.3.6 低電力タイマへのトリガ

EXTI、TAMP、および RTC アラームから LP タイマ (LPTIM1/2/3/4/5/6) へ

目的

LPTIM1/2/3/4/5/6 カウンタは、ソフトウェアによって、または 8 つのトリガ入力の 1 つでのアクティブエッジの検出後に開始されます (セクション 43.4.7 : トリガマルチプレクサを参照)。

GPIO は、LPTIM 設定レジスタ 2 (LPTIM_CFGR2) に従って、LPTIM 入力キャプチャ選択または LPTIM 入力選択としても選択できます。

トリガ信号

このトリガ機能については、セクション 43.4.7 : トリガマルチプレクサ および以下のセクション) で説明されています。入力選択については、表 454 : LPTIM1/2/3/4/5/6 外部トリガ接続で説明されています。

アクティブな電力モード

この相互接続は、RUN、SLEEP、および STOP モードでアクティブです。

15.3.7 タイマへの入力としての RTC ウェイクアップ

RTC からタイマ (TIM16) へ。

目的

RTC ウェイクアップ割込みは、汎用タイマ (TIM16) チャンネル 1 への入力として使用できます。

トリガ信号

RTC ウェイクアップ信号は、TIM 16 の場合、表 435 : [tim_ti1 入力マルチプレクサへの相互接続](#)で説明されているように、tim_ti1_in3 信号に接続されます。

アクティブな電力モード

この相互接続は、STOP モードまでアクティブです。タイマはアクティブではありませんが、カウントはウェイクアップ時に実行されます。

15.3.8 タイマへのブレーク信号としてのシステムエラー

システムエラーからタイマ (TIM1/8/15/16/17) へ

目的

CSS、CPU ロックアップ、SRAM2/3 ECC ダブルエラー、SRAM1 パリティエラー、FLASH ECC ダブルエラー検出、および PVD は、タイマ (TIM1/8/15/16/17) へのタイマブレークの形式でシステムエラーを生成できます。

ブレーク機能の目的は、タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。

トリガ信号

可能なブレークのソースについては、以下を参照してください。

- [セクション 38.3.18 : ブレーク機能の使用](#) (TIM1/8)
- [セクション 42.4.15 : ブレーク機能の使用](#) (TIM15/16/17)
- [表 385 : システムブレーク相互接続](#) (TIM1/8)
- [表 439 : システムブレーク相互接続](#) (TIM15/16/17)

アクティブな電力モード

タイマは、オプションで RUN および SLEEP モードでアクティブです。低電力モードが TIMx に与える影響については、以下を参照してください。

- [表 393 : 低消費電力モードが TIM1/TIM8 に与える影響](#)
- [表 411 : 低消費電力モードが TIM2/TIM3/TIM4/TIM5 に与える影響](#)
- [表 443 : 低消費電力モードが TIM15/TIM16/TIM17 に与える影響](#)

15.3.9 通信ペリフェラルのためのトリガ

Ethernet からタイマ (TIM2/TIM3) へ。

目的

システムクロックをネットワークと同期するために、タイマと PTP の内部接続が使用可能であり、クロックドリフトを確認できます。

トリガ信号

- 出力 (タイマからの) は、Ethernet PTP トリガ入力に直接接続されます。詳細については、以下のセクションを参照してください。
- [セクション 57.3 : Ethernet ピンおよび内部信号](#)
- [表 400 : tim_ti1 入力マルチプレクサへの相互接続](#)
- [表 405 : tim_etr 入力マルチプレクサへの相互接続](#)

アクティブな電力モード

これらの相互接続は、RUN および SLEEP モードでもアクティブなままです。

15.3.10 GPDMA1/2 へのトリガ

EXTI、RTC (アラーム/ウェイクアップ)、TAMP、タイマ (TIM2/12/15)、LP タイマ (LPTIM1/2/3/4/5/6)、GPDMA1 転送完了 (gpdma1_chx_tc、gpdma2_chx_tcf) から GPDMA1/2 へ。

目的

GPDMA トリガは、GPDMA チャネル x に割り当てることができます。プログラムされた GPDMA 転送は、選択された入力トリガイベントの立上がり/立下がりエッジによってトリガできます。トリガモードも、LLI リンク転送に応じてプログラムできます。詳細については、以下のセクションを参照してください。

- [セクション 16.3.7 : GPDMA トリガ](#)
- [セクション 16.4.12 : GPDMA トリガ転送](#)
- 以下の詳細については、[GPDMA チャネル x 転送レジスタ 2 \(GPDMA_CxTR2\)](#) を参照してください。
 - トリガ選択 TRIGSEL[5:0] フィールド
 - TRIGM[1:0] によって定義されたトリガモード (LLI)
 - TRIGPOL[1:0] によって定義されたトリガ極性

トリガ信号

GPDMA トリガマッピングは、[表 129 : プログラムされた GPDMA1/2 トリガ](#) で指定され、GPDMA_CxTR2.TRIGSEL[5:0] に従います。

アクティブな電力モード

この相互接続は、SLEEP モードでも機能し続けます。

以下を参照してください。

- [セクション 16.6 : 低消費電力モードの GPDMA](#)

15.3.11 アナログペリフェラルへの内部アナログ信号

内部アナログソースから ADC (ADC1/2) へ。

目的

内部リファレンス電圧 (V_{REFINT})、内部温度センサ (V_{SENSE})、内部デジタルコア電圧 (V_{DDCORE})、および V_{BAT} 監視チャンネルは、ADC (ADC1/2) 入力チャンネルに接続されます。

これは、以下に従います。

- [セクション 26.2 : ADC の主な機能](#)
- [セクション 26.4.11 : チャンネルの選択 \(SQRx、JSQRx\)](#)

15.3.12 DAC サンプルおよびホールドモードのクロックソース

LSI/LSE から DAC1 へ。

目的

DAC1 は、STOP モードで動作できます。サンプルおよびホールドブロックとそれに関連するレジスタは、STOP モードで LSI または LSE クロックソース (dac_hold_ck) を使用します。

[表 265 : DAC 内部入力／出力信号](#) : サンプルおよびホールドモードでは、dac_hold_ck、入力、DAC 低電力クロックが使用されます。

アクティブな電力モード

この機能は、RUN、SLEEP、および STOP モードでも使用できます。

15.3.13 内部タンパソース

内部ペリフェラル、クロック、または監視からタンパへ。

目的

異常な活動やデバイスを一時的に破損する活動を検出するために、内蔵タンパは望ましくないイベントについてシステムにアラートを発します。それに応じて、さまざまなアクションを取ることができます。

タンパソースのリストは、[表 489 : TAMP 相互接続](#)にあります。

アクティブな電力モード

この相互接続は、タンパソースがアクティブな場合、すべての電力モードでアクティブです。

15.3.14 タンパから RTC への出力

TAMP から RTC へ

目的

RTC はタンパイVENTにタイムスタンプを付けて、検出履歴の取得を可能にします。また、RTC は、GPIO を制御し、MCU の外部のタンブまたはアラームステータスに基づいて信号を設定できます。

詳細については、セクション [セクション 46.3.3 : RTC および TAMP によって制御される GPIO](#) を参照してください。

アクティブな電力モード

この相互接続は、すべての電力モードでアクティブなままです。

15.3.15 AES/SAES への暗号キー

TAMP バックアップレジスタ、システム Flash メモリから SAES および AES および逆の方向

目的

暗号メカニズムにはハードウェアキーが必要であり、保護された不揮発性メモリに保存される必要があります。キーを読み出し不能な方法でロードするために、さまざまなアプローチが実装されています。タンパバックアップレジスタまたはシステム Flash メモリを使用して、それぞれ BHK または RHUK を保存し、SAES に渡す専用バスを実装できます。

詳細については、[セクション 34.4.14 : ラップキーを使った SAES の動作](#)を参照してください。

AES 暗号メカニズム (SAES より高速) は、SAES の共有キーのメリットを享受できます。詳細については、[セクション 34.4.15 : 共有キーを使った SAES の動作](#)を参照してください。

アクティブな電力モード

AES および SAES は、RUN および SLEEP モードで動作します。

16 汎用ダイレクト・メモリ・アクセス・コントローラ (GPDMA)

16.1 GPDMA の概要

汎用ダイレクト・メモリ・アクセス (GPDMA) コントローラは、バス・マスタとシステム・ペリフェラルで構成されます。

GPDMA は、CPU の制御のもと、リンクされたリスト (linked-list : リンクリスト) を介し CPU の負荷を低減した状態で、メモリマップ・ペリフェラル間やメモリ間でのプログラム可能なデータの転送を実行する際に使用します。

16.2 GPDMA の主な機能

- 2つの双方向 AHB マスタ
- メモリマップされたデータの転送元から転送先への転送。
 - ペリフェラルからメモリ
 - メモリからペリフェラル
 - メモリ間
 - ペリフェラル間
- 低消費電力モード時の自律 (autonomous) データ転送 ([セクション 16.3.3](#) 参照)
チャンネル・レベルでの 4 グレードのプログラムされた優先順位に基づく、転送のアービトレーション (調停)。
 - タイミング重視のチャンネルでの、優先順位が高いトラフィック・クラス (キュー 3)。
 - タイミングの制約が緩いチャンネルでの、重み付けを行ったラウンド・ロビン (持ち回り) の割り付けによる優先順位が低い3つのトラフィック・クラス (キュー 0、1、2)
- 次のいずれかのイベントに対する、チャンネルごとのイベント生成 : 転送完了、1/2 転送完了、データ転送エラー、ユーザ設定エラー、リンク転送エラー、中断完了、トリガ・オーバーラン
- イベントごとに別々にプログラムされた割り込みイネーブルでの、チャンネルごとの割り込み生成
- 8 個の並行 GPDMA チャンネル。
 - 転送元／転送先の転送のキューのためのチャンネルごとの FIFO ([セクション 16.3.2](#) 参照)
 - 2 つの実行モード、run-to-completion (ラン・トゥ・コンプリーション) とリンク・ステップ・モードをサポートしている、プログラム可能なリンクリストによる、メモリへのチャンネル内 GPDMA 転送チェーン
 - プログラム可能な GPDMA インプットを経由するチャンネル内およびチャンネル間 GPDMA 転送チェーンは、GPDMA タスク完了イベントへの接続をトリガします。
- チャンネル内の各リンクリスト項目ごとに次の操作を行います。
 - 独立してプログラムされる転送元／転送先の転送
 - 転送元と転送先の間でのプログラム可能なデータの操作 : バイト・ベースの記録、パッキングおよびアンパッキング、パディングまたは切捨て、符号拡張、右/左詰め
 - プログラム可能なブロックレベル転送でのソースからの転送バイト数
 - リニアな転送元および転送先のアドレス指定 : 固定または連続インクリメント・アドレス指定、ブロックレベルで、連続したバースト転送間でプログラム

- 2D の転送元および転送先のアドレス指定 : チャネル・セットを減らすための、2 番目の 2D / 繰り返しブロック・レベルでの、連続したブロック間のプログラム可能な符号付きアドレス・オフセットと組み合わせた、連続したバースト転送間のプログラム可能な符号付きオフセット (ブロック内での非連続アドレス指定) ([セクション 16.3.2](#) 参照)
- 2D アドレス指定による、スキッタ／ギャザー (マルチ・バッファ転送)、データ・インターリーブ、デインターリーブのサポート
- プログラム可能な GPDMA リクエストとトリガ選択
- プログラム可能な GPDMA 1/2 転送および転送完了イベントの生成
- GPDMA リンクリスト制御レジスタでの自動更新機能付き、メモリ上のデータ構造ポインタ
- DEBUG :
 - チャネル・サスペンドおよび再開のサポート
 - FIFO レベルやイベント・フラグを含むチャネル状態のレポート機能
- TrustZone サポート :
 - 最初のチャネルレベルや転送元／転送先、リンク・サブレベルに関係なく、セキュア／非セキュア GPDMA 転送をサポートします。
 - セキュア (非セキュア) ・チャネルからの結果による、セキュア (非セキュア) 割込みレポート機能
 - TrustZone を認識する AHB スレーブ・ポートが、すべての GPDMA セキュア・リソース (レジスタ、レジスタ・フィールド) を非セキュア・アクセスから保護します。
- 特権／非特権をサポート
 - チャネル・レベルで独立に、特権および非特権 GPDMA 転送をサポート
 - 特権を認識する AHB スレーブポート。

16.3 GPDMA の実装

16.3.1 GPDMA インスタンス

このデバイスには、GPDMA1 と GPDMA2 という名称の、2 つの GPDMA インスタンスが存在します。

各 GPDMA インスタンスは、以下のサブ・セクションで説明するように、同じようにチャネル・ベースで実装され、同じリクエストやトリガに接続されています。

16.3.2 GPDMA チャネル

特定の GPDMA チャネル x は、次の機能と用途で実装されています。GPDMA 性能を最大限に利用するため、ある実装 FIFO サイズやリクエストされた GPDMA 転送に対して、ユーザがチャネルを選択し割り当てられるよう、下の表にいくつかの一般的な推奨事項を列記します。

表 123. GPDMA1/2 チャンネル実装

チャンネル x	ハードウェア・パラメータ		機能
	dma_fifo_size[x]	dma_addressing[x]	
x = 0 から 3	2	0	チャンネル x (x = 0 から 3) を次のとおり実装します。 – 8 バイトの FIFO、2 ワード – 固定／連続インクリメント・アドレス指定 これらのチャンネルは通常、APB または AHB ペリフェラルと SRAM 間の GPDMA 転送に割り当てる必要があります。
x = 4、5	4	0	チャンネル x (x = 4、5) を次のとおり実装します。 – 32 バイトの FIFO、8 ワード – 固定／連続インクリメント・アドレス指定 これらのチャンネルは要求の厳しい AHB ペリフェラルと SRAM の間の GPDMA 転送、または外部メモリとの間の相互転送に使用することができます。
x = 6、7	4	1	チャンネル x (x = 6、7) を次のとおり実装します。 – 32 バイトの FIFO、8 ワード – 2D アドレス指定 これらのチャンネルは要求の厳しい AHB ペリフェラルと SRAM の間の GPDMA 転送、または外部メモリとの間の相互転送に使用することができます。

16.3.3 低消費電力モードにおける GPDMA 自律モード

GPDMA 自律モードとウェイクアップ機能が、下の表に示すようにデバイスの低消費電力モードで実装されています。

表 124. 低消費電力モードでの GPDMA1/2 自律モードとウェイクアップ

機能	低消費電力モード
自律モードとウェイクアップ	SLEEP モードでの GPDMA1/2

16.3.4 GPDMA リクエスト

ペリフェラルからの GPDMA リクエストは、SWREQ = 0 の場合、GPDMA_CxTR2 の REQSEL[7:0] で GPDMA チャンネル x に割り当てることが可能です。

GPDMA リクエスト・マッピングを下の表に示します。

表 125. プログラムされた GPDMA1/2 リクエスト

GPDMA_CxTR2.REQSEL[7:0]	選択される GPDMA リクエスト
0	adc1_dma
1	adc2_dma
2	dac1_ch1_dma
3	dac1_ch2_dma
4	tim6_upd_dma
5	tim7_upd_dma

表 125. プログラムされた GPDMA1/2 リクエスト (続き)

GPDMA_CxTR2.REQSEL[7:0]	選択される GPDMA リクエスト
6	spl1_rx_dma
7	spl1_tx_dma
8	spl2_rx_dma
9	spl2_tx_dma
10	spl3_rx_dma
11	spl3_tx_dma
12	i2c1_rx_dma
13	i2c1_tx_dma
14	予約済み
15	i2c2_rx_dma
16	i2c2_tx_dma
17	予約済み
18	i2c3_rx_dma
19	i2c3_tx_dma
20	予約済み
21	uart1_rx_dma
22	uart1_tx_dma
23	uart2_rx_dma
24	uart2_tx_dma
25	uart3_rx_dma
26	uart3_tx_dma
27	uart4_rx_dma
28	uart4_tx_dma
29	uart5_rx_dma
30	uart5_tx_dma
31	uart6_rx_dma
32	uart6_tx_dma
33	uart7_rx_dma
34	uart7_tx_dma
35	uart8_rx_dma
36	uart8_tx_dma
37	uart9_rx_dma
38	uart9_tx_dma
39	uart10_rx_dma
40	uart10_tx_dma
41	uart11_rx_dma
42	uart11_tx_dma

表 125. プログラムされた GPDMA1/2 リクエスト (続き)

GPDMA_CxTR2.REQSEL[7:0]	選択される GPDMA リクエスト
43	uart12_rx_dma
44	uart12_tx_dma
45	lpuart1_rx_dma
46	lpuart1_tx_dma
47	spi4_rx_dma
48	spi4_tx_dma
49	spi5_rx_dma
50	spi5_tx_dma
51	spi6_rx_dma
52	spi6_tx_dma
53	sai1_a_dma
54	sai1_b_dma
55	sai2_a_dma
56	sai2_b_dma
57	ospi1_dma
58	tim1_cc1_dma
59	tim1_cc2_dma
60	tim1_cc3_dma
61	tim1_cc4_dma
62	tim1_upd_dma
63	tim1_trg_dma
64	tim1_com_dma
65	tim8_cc1_dma
66	tim8_cc2_dma
67	tim8_cc3_dma
68	tim8_cc4_dma
69	tim8_upd_dma
70	tim8_tig_dma
71	tim8_com_dma
72	tim2_cc1_dma
73	tim2_cc2_dma
74	tim2_cc3_dma
75	tim2_cc4_dma
76	tim2_upd_dma
77	tim3_cc1_dma
78	tim3_cc2_dma
79	tim3_cc3_dma

表 125. プログラムされた GPDMA1/2 リクエスト (続き)

GPDMA_CxTR2.REQSEL[7:0]	選択される GPDMA リクエスト
80	tim3_cc4_dma
81	tim3_upd_dma
82	tim3_trg_dma
83	tim4_cc1_dma
84	tim4_cc2_dma
85	tim4_cc3_dma
86	tim4_cc4_dma
87	tim4_upd_dma
88	tim5_cc1_dma
89	tim5_cc2_dma
90	tim5_cc3_dma
91	tim5_cc4_dma
92	tim5_upd_dma
93	tim5_trg_dma
94	tim15_cc1_dma
95	tim15_upd_dma
96	tim15_trg_dma
97	tim15_com_dma
98	tim16_cc1_dma
99	tim16_upd_dma
100	tim17_cc1_dma
101	tim17_upd_dma
102	lptim1_ic1_dma
103	lptim1_ic2_dma
104	lptim1_ue_dma
105	lptim2_ic1_dma
106	lptim2_ic2_dma
107	lptim2_ue_dma
108	dcmi_dma または pssi_dma ⁽¹⁾
109	aes_out_dma
110	aes_in_dma
111	hash_in_dma
112	ucpd1_rx_dma
113	ucpd1_tx_dma
114	cordic_read_dma
115	cordic_write_dma
116	fmac_read_dma

表 125. プログラムされた GPDMA1/2 リクエスト (続き)

GPDMA_CxTR2.REQSEL[7:0]	選択される GPDMA リクエスト
117	fmac_write_dma
118	saes_out_dma
119	saes_in_dma
120	i3c1_rx_dma
121	i3c1_tx_dma
122	i3c1_tc_dma
123	i3c1_rs_dma
124	i2c4_rx_dma
125	i2c4_tx_dma
126	予約済み
127	lptim3_ic1_dma
128	lptim3_ic2_dma
129	lptim3_ue_dma
130	lptim5_ic1_dma
131	lptim5_ic2_dma
132	lptim5_ue_dma
133	lptim6_ic1_dma
134	lptim6_ic2_dma
135	lptim6_ue_dma

1. どの専用ファンクションを使用するかによる。

16.3.5 GPDMA ブロック・リクエスト

GPDMA リクエストによっては、バースト・リクエストではなくブロック・リクエストとしてプログラムする必要があるものがあります。その後、ハードウェア・レベルでの、リクエストされたペリフェラル転送の正しい GPDMA の実行のために、GPDMA_CxTR2 の BREQ を設定する必要があります。

表 126. ブロック・リクエストとしてプログラムされた GPDMA1/2 リクエスト

GPDMA ブロック・リクエスト
lptim1_ue_dma
lptim2_ue_dma
lptim3_ue_dma
lptim5_ue_dma
lptim6_ue_dma

16.3.6 ペリフェラルの早期終了を伴う GPDMA チャンネル

GPDMA チャンネルをこの機能と共に実装すると、この機能をサポートしたペリフェラルからのデータ転送の早期終了をサポートすることができます。

表 127. ペリフェラルの早期終了を伴う GPDMA1/2 チャンネル

ペリフェラルの早期終了を伴う GPDMA チャンネル x
x = 0 および x = 7

この GPDMA のサポートは、チャンネル x が GPDMA_CxTR2.PFREQ = 1 でプログラムされたときに有効になります。その後、GPDMA がブロック全体を転送してしまう前に、このペリフェラルそのものがデータ転送完了の始動とリクエストをすることが可能です（詳細は [セクション 16.4.14](#) 参照）。

表 128. ペリフェラルの早期終了を伴う、プログラムされた GPDMA1/2 リクエスト

ペリフェラルの早期終了を伴う、プログラムされた GPDMA チャンネル x リクエスト
i3c1_rx_dma

16.3.7 GPDMA トリガ

TRIGPOL[1:0] が選択されたトリガ（TRIGPOL[1:0] = 01 または TRIGPOL[1:0] = 10）の立上がり、または立下りのエッジを決定する場合、GPDMA トリガを GPDMA_CxTR2 の TRIGSEL[5:0] 経由で GPDMA チャンネル x に割り当てることが可能です。

表129.プログラムされた GPDMA1/2 トリガ

GPDMA_CxTR2.TRIGSEL[5:0]	選択される GPDMA トリガ
0	exti0
1	exti1
2	exti2
3	exti3
4	exti4
5	exti5
6	exti6
7	exti7
8	tamp_trg1
9	tamp_trg2
10	tamp_trg3
11	lptim1_ch1
12	lptim1_ch2
13	lptim2_ch1
14	lptim2_ch2
15	rtc_alra_trg
16	rtc_alrb_trg
17	rtc_wut_trg

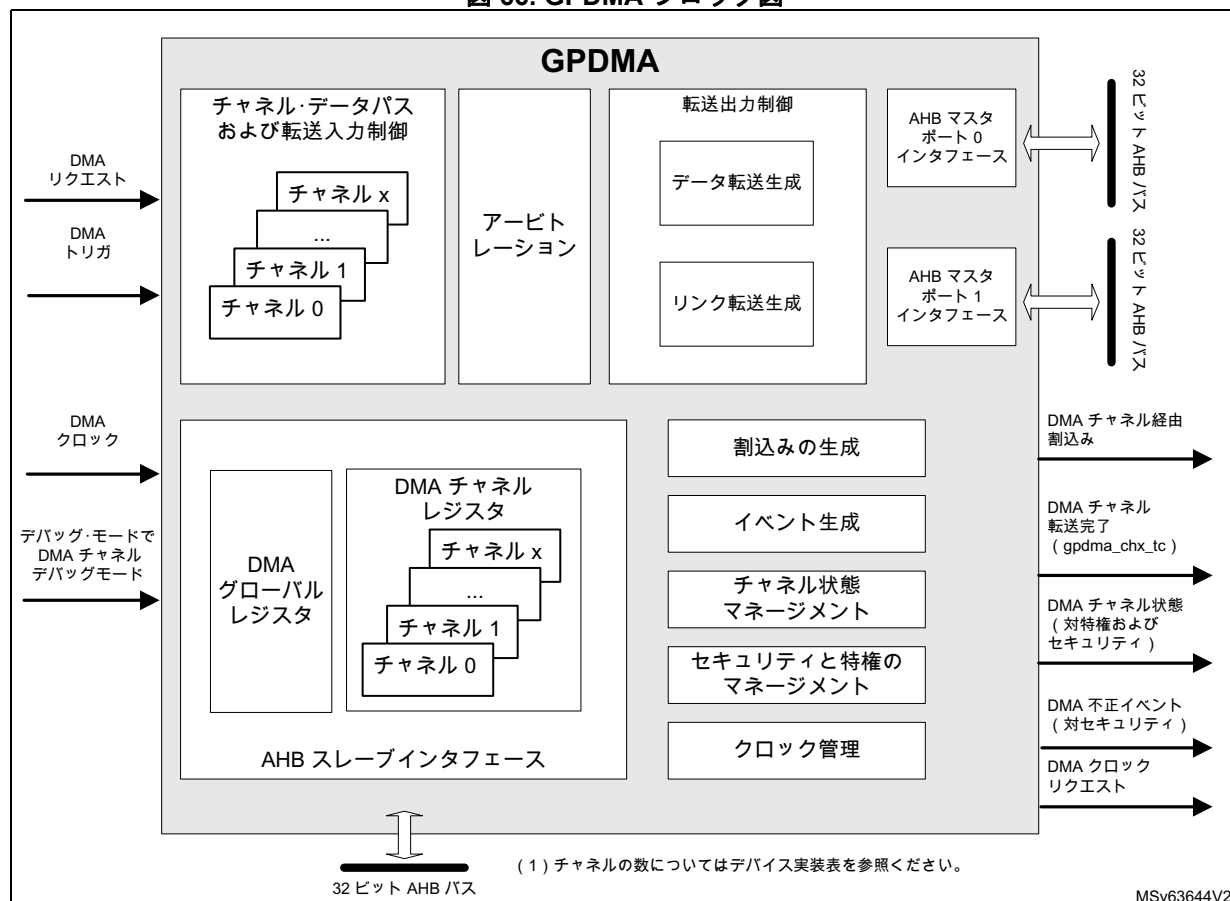
表129. プログラムされた GPDMA1/2 トリガ (続き)

GPDMA_CxTR2.TRIGSEL[5:0]	選択される GPDMA トリガ
18	gpdma1_ch0_tc
19	gpdma1_ch1_tc
20	gpdma1_ch2_tc
21	gpdma1_ch3_tc
22	gpdma1_ch4_tc
23	gpdma1_ch5_tc
24	gpdma1_ch6_tc
25	gpdma1_ch7_tc
26	gpdma2_ch0_tc
27	gpdma2_ch1_tc
28	gpdma2_ch2_tc
29	gpdma2_ch3_tc
30	gpdma2_ch4_tc
31	gpdma2_ch5_tc
32	gpdma2_ch6_tc
33	gpdma2_ch7_tc
34	tim2_trgo
35	tim15_trgo
36	tim12_trgo
37	lptim3_ch1
38	lptim3_ch2
39	lptim4_ait
40	lptim5_ch1
41	lptim5_ch2
42	lptim6_ch1
43	lptim6_ch2

16.4 GPDMA の機能説明

16.4.1 GPDMA ブロック図

図 66. GPDMA ブロック図



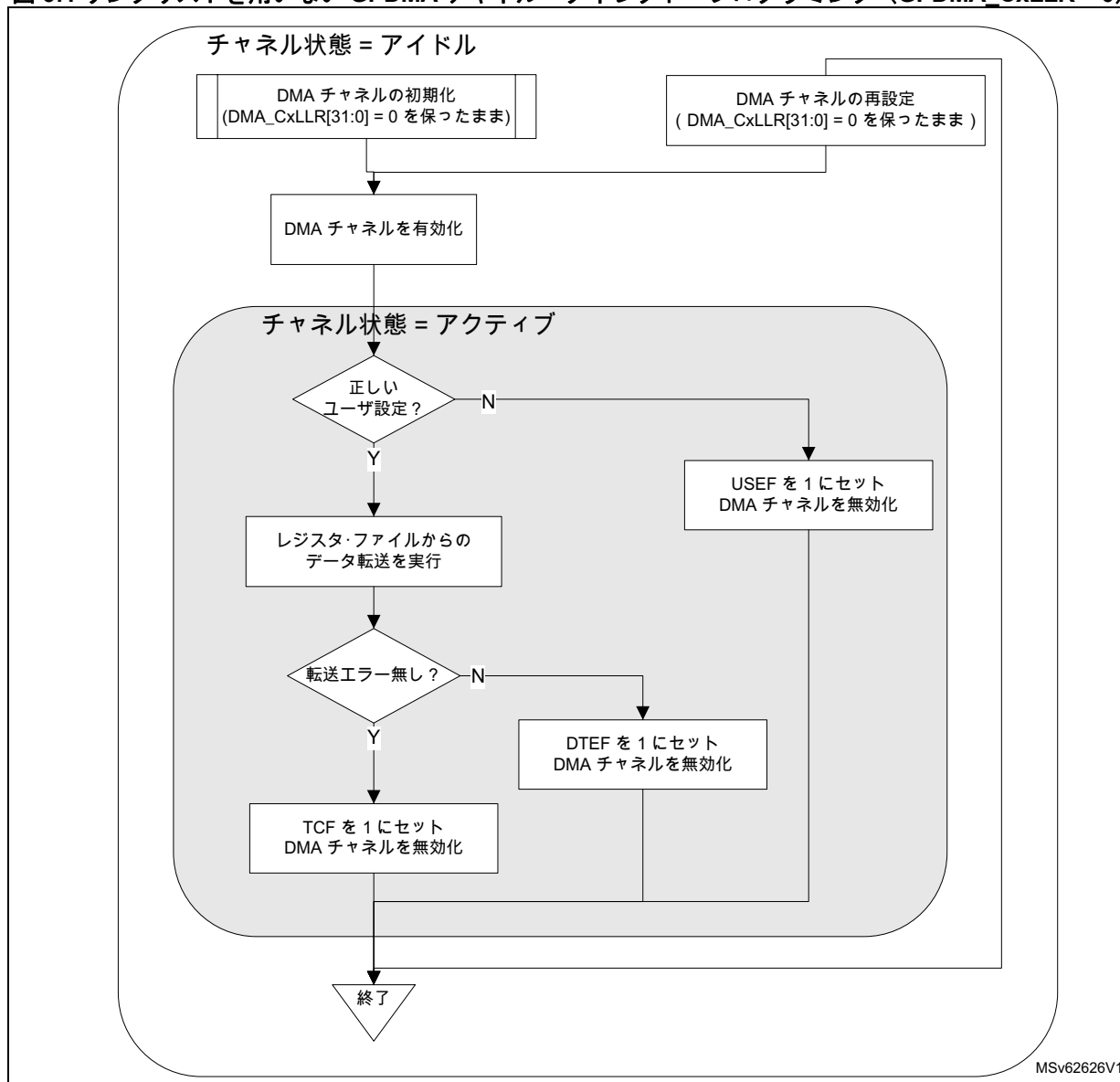
16.4.2 GPDMA チャンネル状態とリンクリストを使用しない直接プログラミング

GPDMA リセット後、GPDMA チャンネル x はアイドル状態となります。GPDMA_CxCR.EN にソフトウェアが 1 を書き込むと、そのチャンネルは他のチャンネルの設定レジスタ (GPDMA_CxXXX) の値を考慮して有効／非アイドル状態に切り替わり、対応するリクエストされたデータの転送の実行を開始します。

GPDMA_CxCR.EN に 1 を書き込むことで GPDMA チャンネル転送を有効化／開始した後、GPDMA チャンネルがアイドル状態に戻り (その後、EN がハードウェアによりデアサートされ)、再設定と再有効化の準備が整っていることを、転送完了時に GPDMA チャンネル割り込みがソフトウェアに通知します。

下の図は、リンクリストを使用しない、この GPDMA の直接プログラミング (GPDMA_CxLLR = 0) を図示したものです。

図 67. リンクリストを用いない GPDMA チャンネル・ダイレクト・プログラミング (GPDMA_CxLLR = 0)



16.4.3 GPDMA チャンネルのサスペンドとレジューム

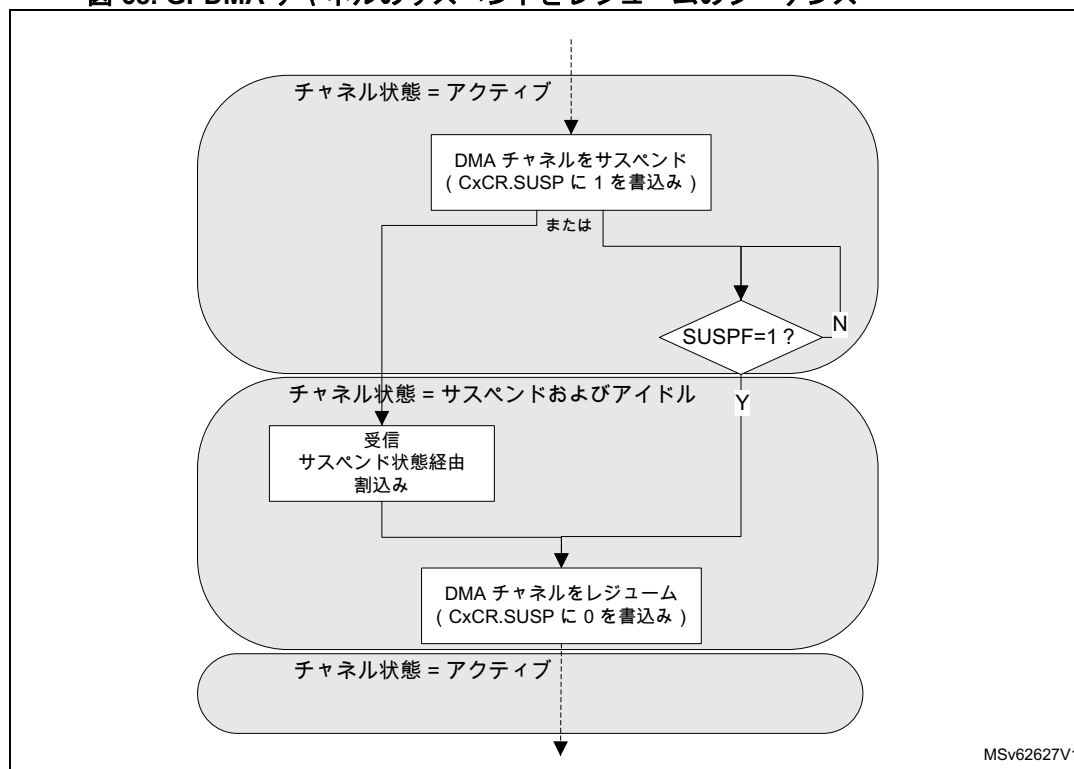
次の手順により、有効になっているチャンネルをソフトウェアのみでサスペンドすることが可能です。

1. ソフトウェアで GPDMA_CxCR.SUSP ビットに 1 を書き込みます。
2. サスペンドされた GPDMA_CxSR.SUSPF フラグをソフトウェアで SUSPF = 1 になるまでポーリングするか、GPDMA_CxCR.SUSPIE に 1 を書き込むことで既に有効になっている割り込みを待ちます。チャンネルが実質的にサスペンドした状態を待つということは、そのマスタ・ポートによる進行中のあらゆる GPDMA 転送の完了を待つということです。その後ソフトウェアは定常状態で、あらゆる読出しレジスタ、またはハードウェアによる変更が可能なレジスタ・フィールドを観察することが可能になります。

進行中の GPDMA 転送は、データ転送（転送元／転送先間のバースト転送）あるいは、次のリンクリスト項目からのリンクリスト・レジスタ・ファイルの内部更新のためのリンク転送の可能性があることに注意してください。

3. ソフトウェアは、GPDMA_CxCR.SUSP に 0 を書き込むことでサスペンドしたチャンネルを安全にレジュームします。

図 68. GPDMA チャンネルのサスペンドとレジュームのシーケンス



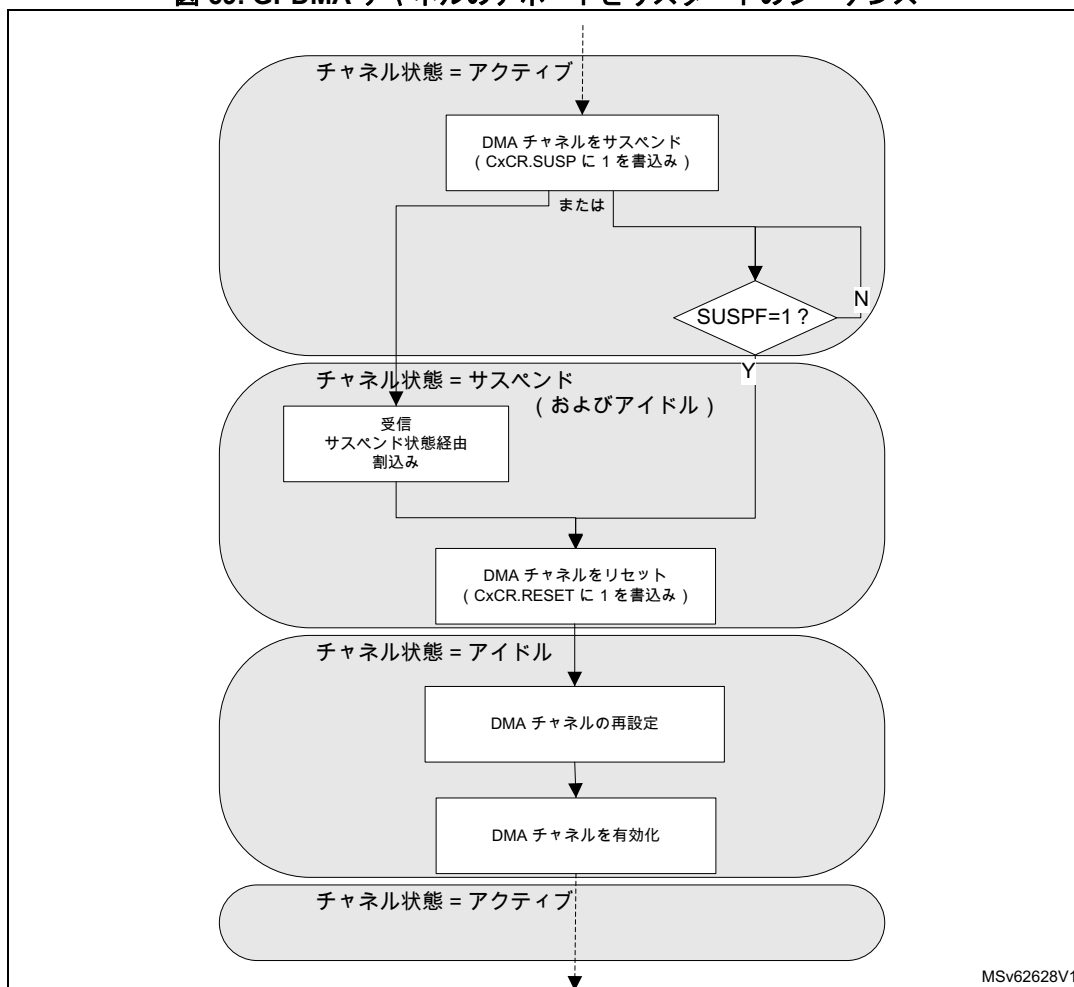
注： サスペンドとレジュームのシーケンスは GPDMA_CxCR.EN ビットには影響しません。チャンネル（転送）のサスペンドが、開始したトリガ検出をサスペンドすることはありません。

16.4.4 GPDMA チャンネルのアボートとリスタート

サーキュラ・バッファまたはダブル・バッファによる連続 GPDMA 転送のアボート同様、以下に示すシーケンスにより、いまだに有効なチャンネルをソフトウェアのみでアボートすることも可能です。

1. ソフトウェアで GPDMA_CxCR.SUSP ビットに 1 を書き込みます。
2. サスペンドされた GPDMA_CxCR.SUSPF フラグをソフトウェアで SUSPF = 1 になるまでポーリングするか、GPDMA_CxCR.SUSPIE に 1 を書き込むことで既に有効になっている割り込みを待ちます。チャンネルが実質的にサスペンドした状態を待つということは、マスタ・ポートによる進行中のあらゆる GPDMA 転送の完了を待つということです。
3. ソフトウェアは、GPDMA_CxCR.RESET に 1 を書き込むことでチャンネルをリセットします。これにより FIFO やチャンネルの内部状態、GPDMA_CxCR.EN ビット、GPDMA_CxCR.SUSP ビットがリセットされます。
4. ソフトウェアがチャンネルを安全に再設定します。ソフトウェアは、ハードウェアにより変更された GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR レジスタを再プログラムする必要があります。
5. ここでアボートしてから再プログラムされたチャンネルをリスタートするために、ソフトウェアが GPDMA_CxCR.EN ビットに 1 を書き込むことでこれを再び可能にします。

図 69. GPDMA チャンネルのアボートとリスタートのシーケンス



16.4.5 GPDMA のリンクリスト・データ構造

直接プログラミング・モードの他、チャンネルは、リンクリスト項目 (LLI) のリストとして知られている転送リストによりプログラムすることが可能です。各 LLI はそのデータ構造により定義されます。

チャンネル x の次の LLI_{n+1} のデータ構造を持つメモリのベース・アドレスは以下の合計です。

- (GPDMA_CxLBAR での) チャンネル x のリンク・ベース・アドレス
- リンク・アドレス・オフセット (GPDMA_CxLLR の LA[15:2] フィールド)。リンクリスト・レジスタ GPDMA_CxLLR は、チャンネル x の前の LLI_n のデータ構造から更新された結果です。

各 LLI のデータ構造は、それぞれ異なる場合があります。

リンクリストのデータ構造は、GPDMA_CxLLR の UT1、UT2、UB1、USA、UDA、ULL ビットの値に加え、UB2 と UT3 の値に従ってアドレス指定されます。

リンクリスト・モードでは、リンクリスト・レジスタ (GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR または GPDMA_CxLLR に加え GPDMA_CxTR3 または GPDMA_CxBR2) は、条件付きで自動的に次のリンクリスト・データのメモリ構造から、GPDMA_CxLLR レジスタの現在の値に従って更新されます。この GPDMA_CxLLR レジスタの現在の値は、前の LLI のリンクリスト・データ構造から条件付きで更新されたものです。

静的リンクリスト・データ構造

例えば、GPDMA_CxLLR の更新ビット (UT1、UT2、UB1、USA、UDA、ULL に加え UB2 と UT3) がすべてアサートされると、メモリにあるリンクリストのデータ構造は次のときに最大となります。

- GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR (図 70 参照)、とメモリの最初のリンクリスト・レジスタ・ファイル (LLI₀) および次の LLI (LLI₁、LLI₂ など) を含む、チャンネル x (x = 0 から 5) の連続 32 bit 位置。
- GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR に加え、GPDMA_CxTR3、GPDMA_CxBR2 (図 71 参照)、とメモリの最初のリンクリスト・レジスタ・ファイル (LLI₀) および次の LLI (LLI₁、LLI₂ など) を含む、チャンネル x (x = 6 から 7) の連続 32 bit 位置。

図 70. チャンネル x のリニア・アドレス指定の静的リンクリスト・データ構造 (すべて U_{xx} = 1)

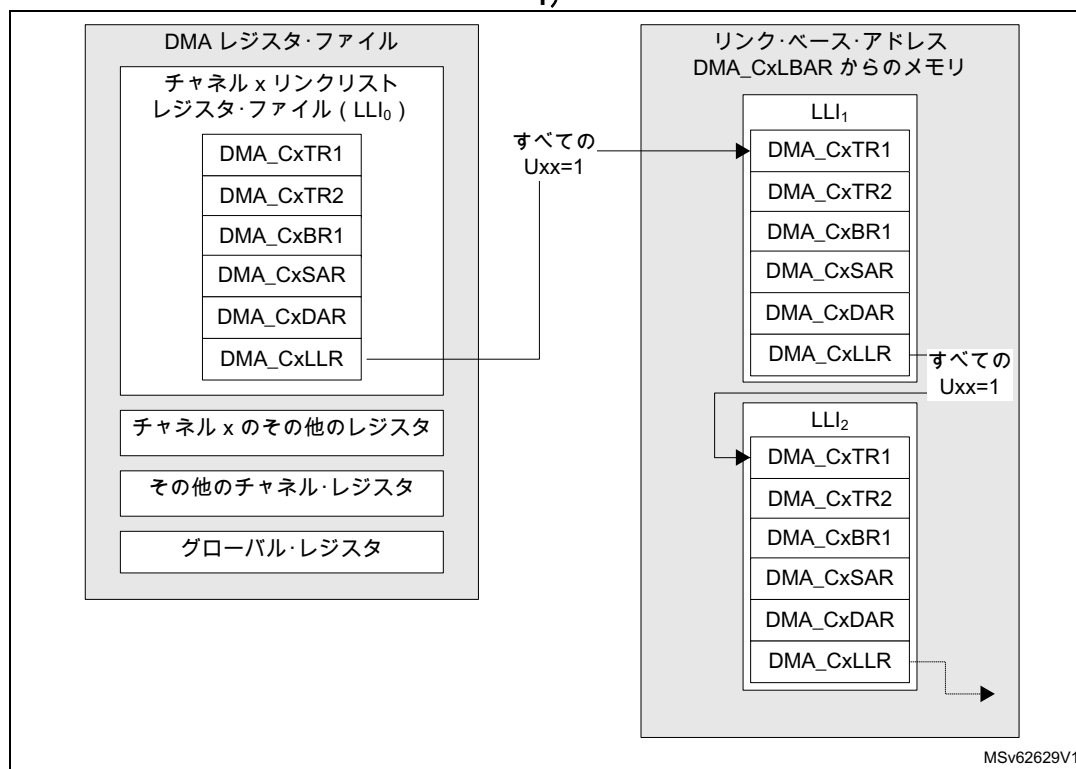
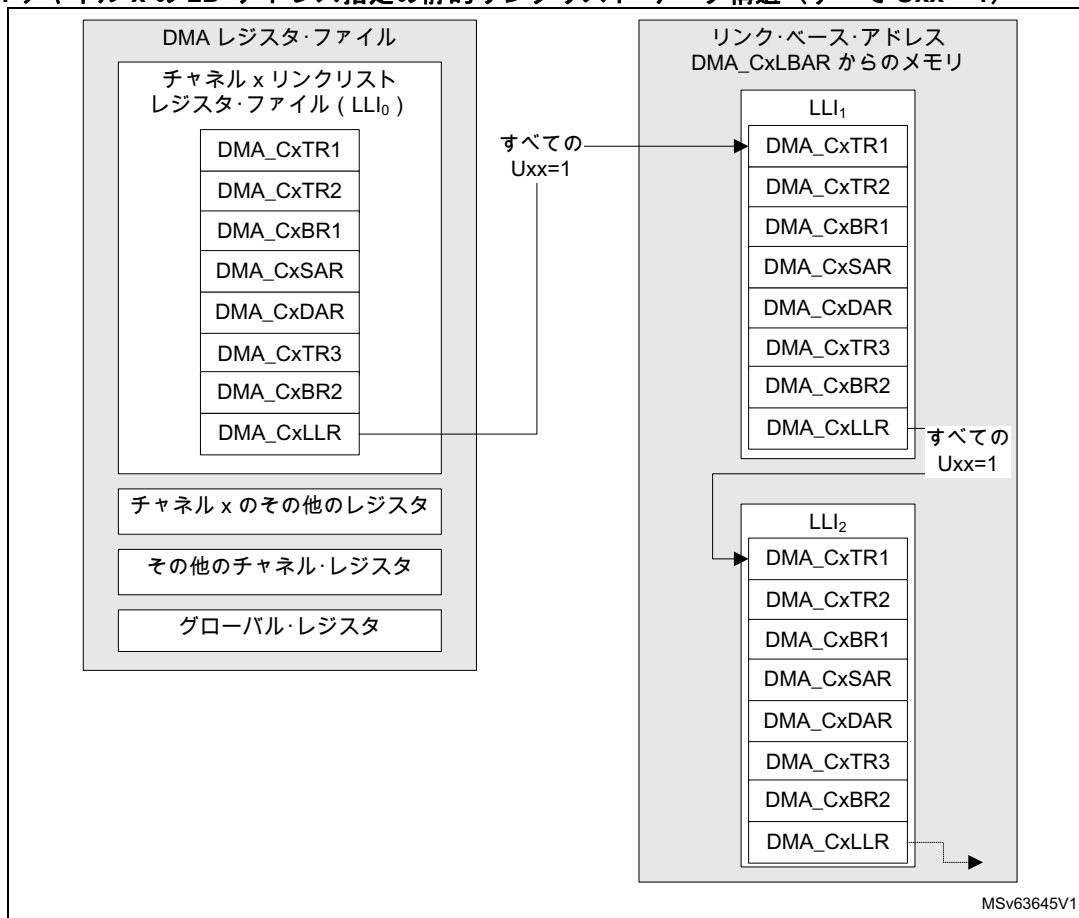


図 71. チャンネル x の 2D・アドレス指定の静的リンクリスト・データ構造 (すべて $U_{xx} = 1$)

動的リンクリスト・データ構造

LLI の全リストのメモリ構成は、各 LLI 特有のデータ構造でコンパクト化が可能です。

UT1 = 0 かつ UT2 = 1 の場合、レジスタ GPDMA_CxLLR のリンク・アドレス・オフセットは、変更しない GPDMA_CxTR1 の代わりに、GPDMA_CxTR2 の更新された値を指します (図 72 参照)。

例：チャンネル x が 2D アドレス指定され UT1 = UB1 = USA = 0 かつ UT3 = UB2 = 0 の場合、および UT2 = UDA = ULL = 1 の場合、次の LLI は、GPDMA_CxTR1、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxTR3、GPDMA_CxBR2 の (更新された) 値をいずれも含みません。図 73 に示すように、次の LLI は GPDMA_CxTR2、GPDMA_CxDAR、GPDMA_CxLLR の更新された値を含みます。

図 72. チャンネル x の リニア・アドレス指定の GPDMA 動的リンクリスト・データ構造

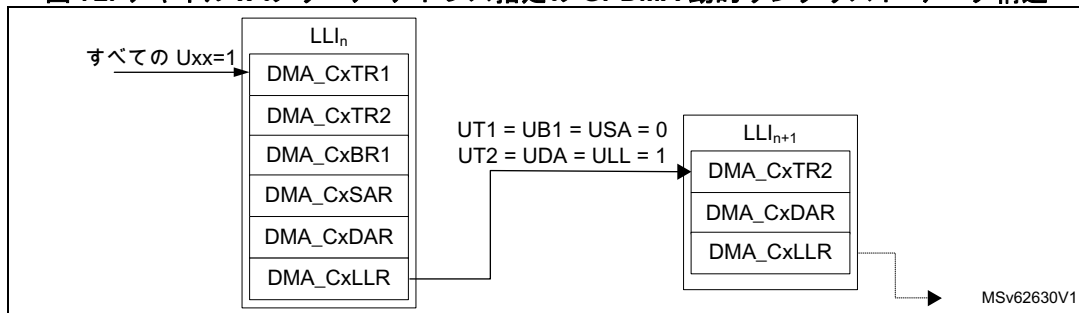
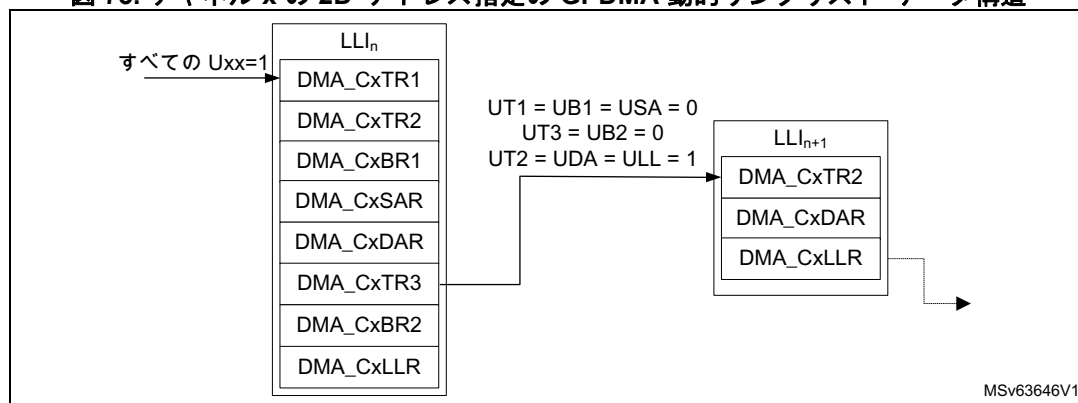


図 73. チャネル x の 2D・アドレス指定の GPDMA 動的リンクリスト・データ構造



ユーザは各 LLI_n の GPDMA_CxLLR が 32 bit に整列し、GPDMA_CxLBAR が指す 64 KB のアドレス空間を超えないようにプログラムする必要があります。

16.4.6 リンクリスト項目の転送実行

LLI_n 転送は次のシーケンスです。

1. データ転送 : GPDMA の内部レジスタ・ファイルに記載されているとおり、GPDMA がデータ転送を実行 (このデータ転送は LLI_0 にはボイド (void) /ヌル (null) とすることができます)
2. 条件付きリンク転送 : LLI_n の GPDMA_CxLLR 値で定義されるように、次の LLI_{n+1} のデータ構造により内部レジスタ・ファイルを GPDMA が条件付きで自動的に更新。

注 : 内部レジスタ・ファイル (LLI_0) により定義される最初のデータ転送は、GPDMA_CxLLR の条件付き更新ビット UB1 が設定されている (つまり、メモリの中の次の LLI_1 により記述される null ではないデータ転送が実行されることになっている) 場合、null (GPDMA_CxBR1.BNDT[15:0] = 0 かつ GPDMA_CxTR2.PFREQ = 0) にすることができます。

次のセクションで説明するとおり、意図する GPDMA の利用方法に従い、GPDMA チャネル x はフル・リンクリストの記述により実行する (run-to-completion モード、GPDMA_CxCR.LSM = 0) か、あるいは、LLI のシングル実行 (リンク・ステップ・モード、GPDMA_CxCR.LSM = 1) をプログラムすることが可能です。

16.4.7 run-to-completion モードでの GPDMA チャネル状態およびリンクリスト・プログラミング

GPDMA_CxCR.LSM = 0 (フル・リスト実行モードでは、LLI のフル・シーケンスの実行は run-to-completion モードと呼ばれます) の場合、GPDMA チャネル x は、GPDMA_CxCR.EN に 1 を書き込むことで開始し、チャネル・レベルで完了した後、最初にプログラムされます。チャネル転送は次のとおりです。

- 少なくとも次の設定をします。
 - 最初の LLI_0 、内部リンクリスト・レジスタ・ファイル : GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR に加え、GPDMA_CxTR3 および GPDMA_CxBR2
 - 最後の LLI_N 、最後の前の LLI_{N-1} を反映した GPDMA_CxLLR で定義された、メモリ内のリンクリスト データ構造によって記述されたもの
- 最後の LLI_{N-1} 転送の終わりに、GPDMA_CxLLR[31:0] = 0、GPDMA_CxBR1.BRC[10:0] = 0、GPDMA_CxBR1.BNDT[15:0] = 0 となったときに完了

GPDMA_CxLLR[31:0] = 0 は、リンクリスト・ベースのチャンネル完了の条件で、次のことを意味しています。

- 次のリンク・アドレスの下位 16 bit GPDMA_CxLLR.LA[15:0] が null。
- すべての更新されたビット GPDMA_CxLLR.Uxx が null (UT1、UT2、UB1、USA、UDA、ULL に加え、UB2 および UT3)。

GPDMA_CxLLR.LSM = 0 の場合、チャンネルは決して完了しません。

- 最後の LLI_N は周期的で次の LLI として自分自身を指しています。
 - GPDMA_CxLLR.ULL = 1 または GPDMA_CxLLR.LA[15:2] のどちらかが同じ値で更新されます。
 - あるいは GPDMA_CxLLR.ULL = 0 です。
- LLI_N が前の LLI を指す場合。

ブロック・レベルで完了する通常のデータ転送では、GPDMA_CxBR1.BNDT[15:0] = 0 および GPDMA_CxBR1.BRC[10:0] = 0 (存在する場合)。あるいは、ブロック転送がペリフェラルにより早期に完了する場合があります (例えば RX モードの I3C)、その場合 BNDT[15:0] は null ではありません (より詳細は [セクション 16.4.14](#) を参照してください)。

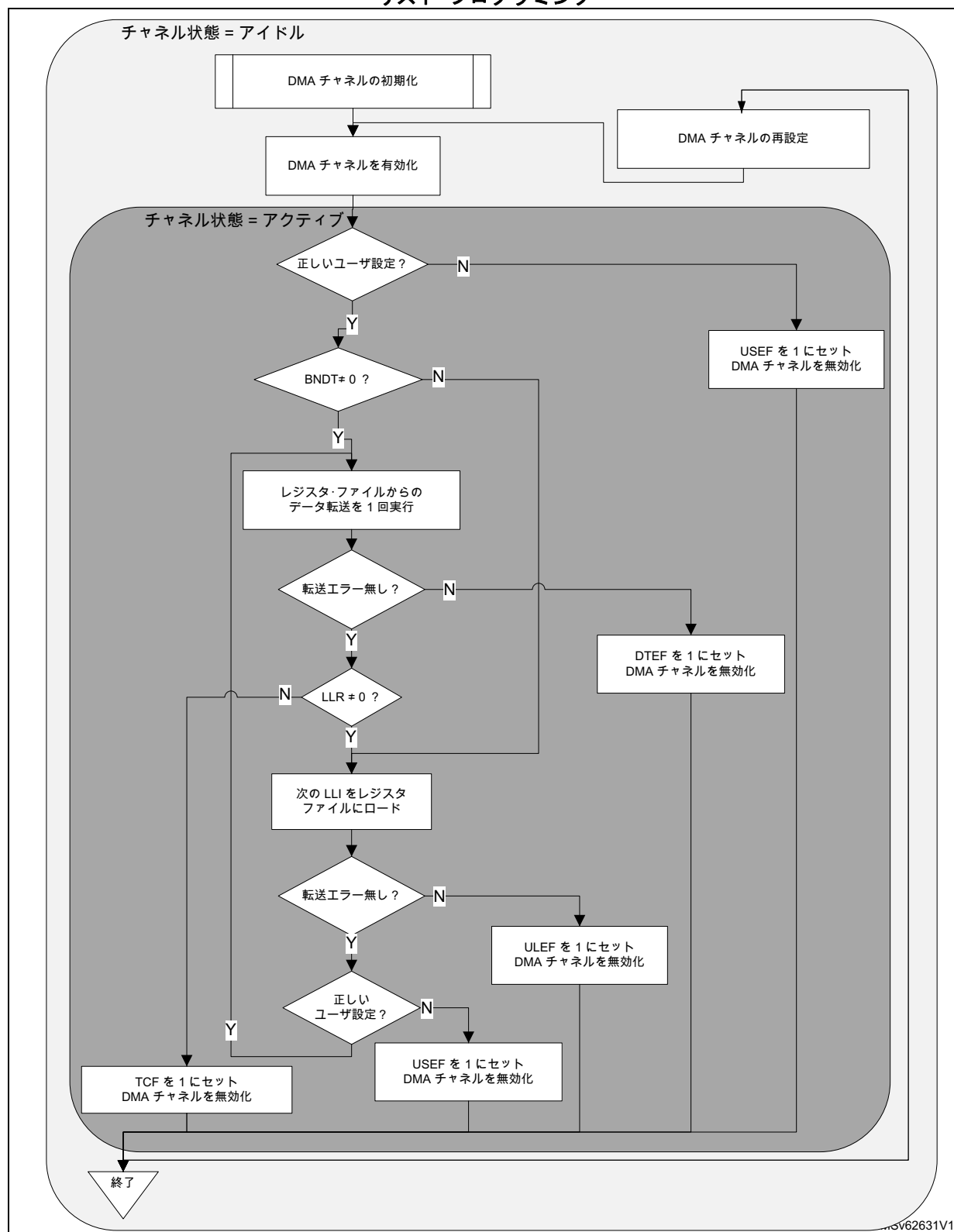
通常の run-to-completion モードでは、GPDMA チャンネルの割当は、詳細プログラミングを含め、GPDMA の初期化の際に 1 度行われます。予約済みデータの通信リンクと、ラン・タイム時 (実行時) の GPDMA サービスを行うため、および連続した繰り返し転送のため (ペリフェラルやメモリとの相互転送、あるいはメモリ間転送のため)。この予約されたデータ通信リンクはチャンネルで構成することも、チャンネルを共有して一連の LLI で反復転送を構成することもできます。

run-to-completion モードでの GPDMA チャンネル実行と、そのレジスタのプログラミングを [図 74](#) に示します。

注 : [図 74](#) は、GPDMA_CxTR2 の TCEM[1:0] のプログラムされた値により、TCEF をどれぐらい頻繁に発生させることができるかを示すことを意図したものではありません。これは (各) ブロックの完了時、(各) 2D ブロックの完了時、(各) LLI の完了時、あるいはチャンネルの完了時のみに発生させることができます。run-to-completion モードでは、チャンネル完了時の TCEM[1:0] の値に関わらず、ハードウェアは常に TCEF = 1 を設定し、そのチャンネルを無効にします。

[図 74](#) では BNDT ≠ 0 が、この図で最初のデータ転送開始の典型的な条件です。この条件は、ペリフェラルが早期に終了するデータ転送をリクエストすると (BNDT ≠ 0 および PFREQ = 1) となります ([セクション 16.3.6](#) 参照)。

図 74. run-to-completion モード(GPDMA_CxCR.LSM = 0) での GPDMA チャンネル実行およびリンクリスト・プログラミング



run-to-completion モード（実行時）での補助チャネル経由でLLI_nの挿入

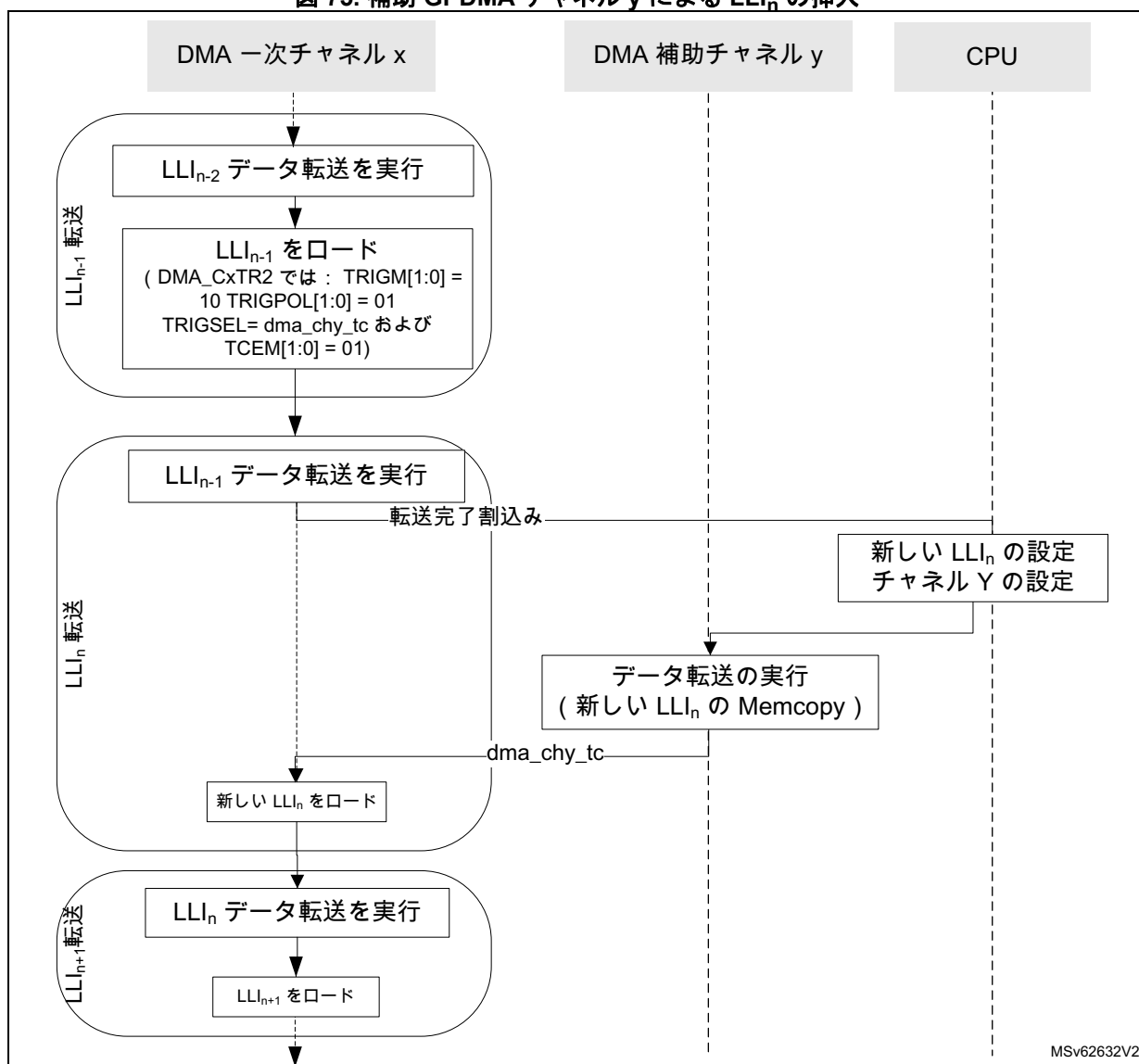
LLI_{n-1} のリンク転送の開始（LLI_n ロードの開始）は、LLI_{n-1} のデータ構造の GPDMA_CxTR2 の以下のフィールドをプログラミングする場合、トリガの発生を条件とすることができます。

- TRIGM[1:0] = 10（リンク転送トリガ・モード）
- TRIGPOL[1:0] = 01 または 10（立上がり、または立下リエッジ）
- TRIGSEL[5:0]（トリガ選択の詳細については [セクション 16.3.7](#) 参照）

別の補助チャネル y は、チャネル x LLI_n をメモリに格納するため、および転送完了イベント gpdma_chy_tc の生成に使用することができます。このイベントをチャネル x の LLI_{n-1} のリンク転送の入力トリガとして選択することで、ソフトウェアはその LLI_{n-1} データ転送後、実際に LLI_n が実際に書き込まれるまで、一次チャネル x を休止することが可能です。

下の図は、別の補助チャネル y を経由した、1次チャネル x のリンクリストの動的な詳細の説明です。

注意： この使用事例はトリガを必要としない LLI_{n-1} データ転送のアプリケーションに限定されています。この LLI_{n-1} のトリガ・モードは、次の LLI_n のロードに使用されます。

図 75. 補助 GPDMA チャンネル y による LLI_n の挿入

16.4.8 リンク・ステップ・モードでの GPDMA チャンネル状態およびリンクリスト・プログラミング

GPDMA_CxCR.LSM = 1 (リンク・ステップ実行モードでの 1 つの LLI のシングル実行) の場合、 LLI の各シングル実行後に、その (条件付き) データ転送と (条件付き) リンク転送を含む、チャンネル転送を実行して完了します。

GPDMA チャンネル転送は LLI レベルでプログラムし、GPDMA_CxCR.EN に 1 を書き込んで開始することが可能で、 LLI レベルで完了した後、次のようになります。

- 現在の LLI_n 転送は次のように記述されます。
 - GPDMA_CxTR1 が転送元／転送先の基本シングル／バースト転送を定義。
 - GPDMA_CxBR1 が、ブロック・レベルでのバイト数 (BNDT[15:0])、チャンネル x ($x = 6$ から 7) での、2D／繰り返しブロック・レベル (BRC[10:0]+1) のブロック数およびアドレス・オフセットのインクリメント／デクリメント・モードを規定します。

- GPDMA_CxTR2 が転送の入力制御（リクエスト、トリガ）および出力制御（転送完了イベント）を定義します。
- GPDMA_CxSAR/GPDMA_CxDAR が転送元／転送先の転送開始アドレスを定義します。
- チャネル x (x = 6 から 7) の GPDMA_CxTR3 が、バースト転送間の転送元／転送先の追加アドレス・オフセットを定義します。
- チャネル x (x = 6 から 7) の GPDMA_CxBR2 が、2D／繰り返しブロック・レベルでのブロック間の転送元／転送先の追加アドレスオフセットを定義します。
- GPDMA_CxLLR が、データ構造とメモリの中の次の LLI_{n+1} アドレス・オフセットを定義します。
- 現在の LLI_n のシングル実行後に現在の LLI_n 転送が完了します。
 - （条件付き）データ転送の完了後（GPDMA_CxBR1.BRC[10:0] = 0 および GPDMA_CxBR1.BNDT[15:0] = 0 の場合）
 - メモリの中の次の LLI_{n+1} のデータ構造からの GPDMA リンク・レジスタ・ファイルの（条件付き）更新後

注： LLI が周期的な（次の LLI として自分自身を指している、GPDMA_CxLLR.ULL = 1 または GPDMA_CxLLR.LA[15:2] のいずれかが同じ値で更新されている、あるいは GPDMA_CxLLR.ULL = 0）場合はリンクステップモードのチャンネルはLLIの各シングル実行毎に完了されます。

ブロック・レベルで完了する通常のデータ転送では、GPDMA_CxBR1.BNDT[15:0] = 0 および GPDMA_CxBR1.BRC[10:0] = 0。あるいは、ブロック転送がペリフェラルにより早期に完了する場合があります（例えば RX モードの I3C）、その場合 BNDT[15:0] は null ではありません（より詳細は [セクション 16.4.14](#) を参照してください）。

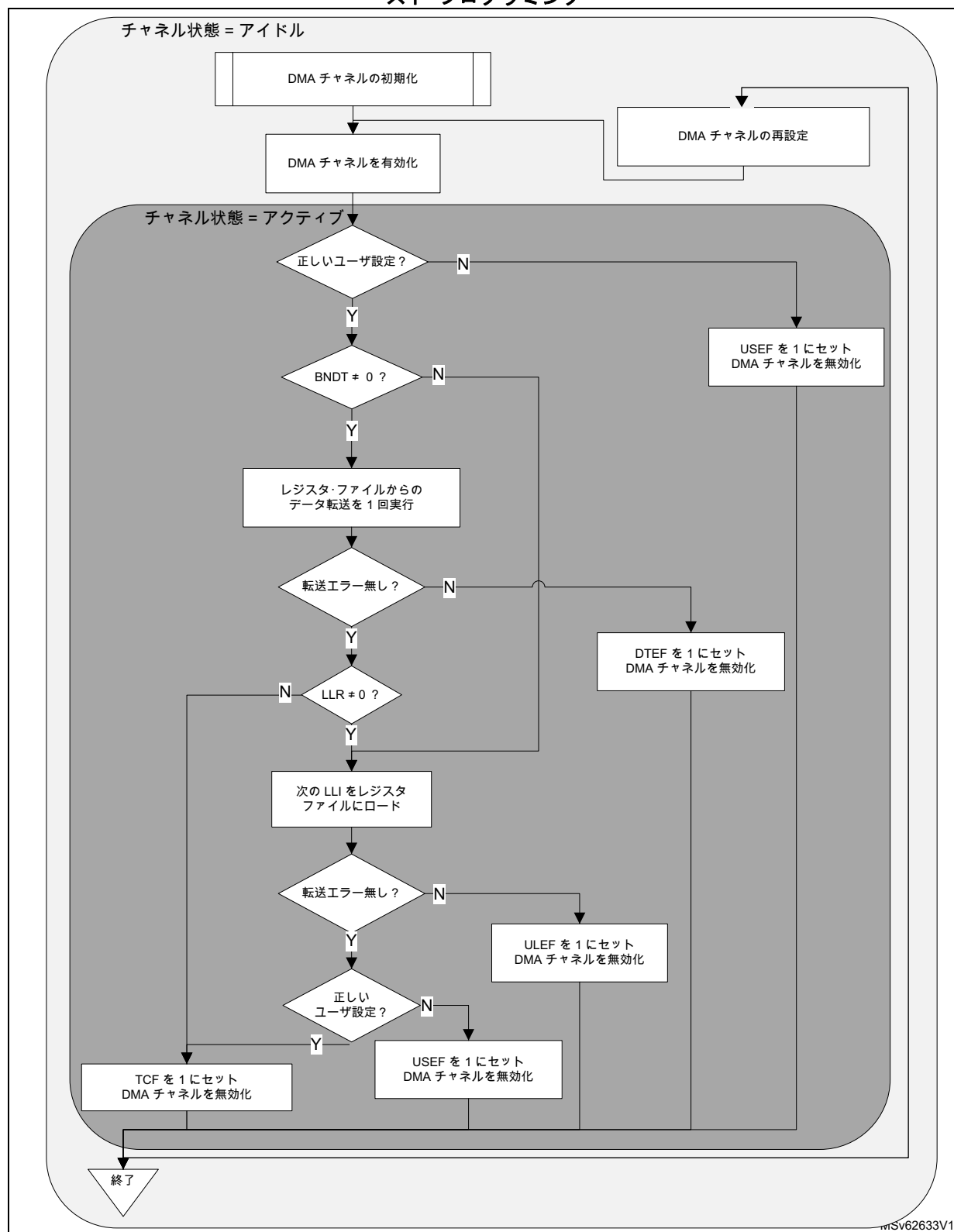
リンク ステップ モードを使用すると、実行時にメモリ内で動的に LLI を作成できます。あらゆる LLI_n に静的データ構造を使用することにより、ソフトウェアを支援することができます（GPDMA_CxLLR のすべての更新ビットが静的な値 $LLI_n.LLR.LA = LLI_{n-1}.LLR.LA + \text{定数}$ を持ちます）。

[図 76](#) に GPDMA チャンネル実行モードと、そのリンク・ステップ・モードでのプログラミングを示します。

注： [図 76](#) は、GPDMA_CxTR2 の TCEM[1:0] のプログラムされた値により、TCEF をどれぐらい頻繁に発生させることができるかを示すことを意図したものではありません。これは（各）ブロックの完了時、（各）2D ブロックの完了時、（各）LLI の完了時、あるいは最後の LLI データ転送の完了時のみに発生させることができます。リンク・ステップ・モードでは、LLI の各シングル実行後にチャンネルが無効になり、TCEM[1:0] の値により TCEF を立てるかどうかが変わります。

[図 76](#) では BNDT ≠ 0 が、最初のデータ転送開始の典型的な条件です。この条件は、ペリフェラルが早期に終了するデータ転送をリクエストすると（BNDT ≠ 0 および PFREQ = 1）となります（[セクション 16.3.6](#) 参照）。

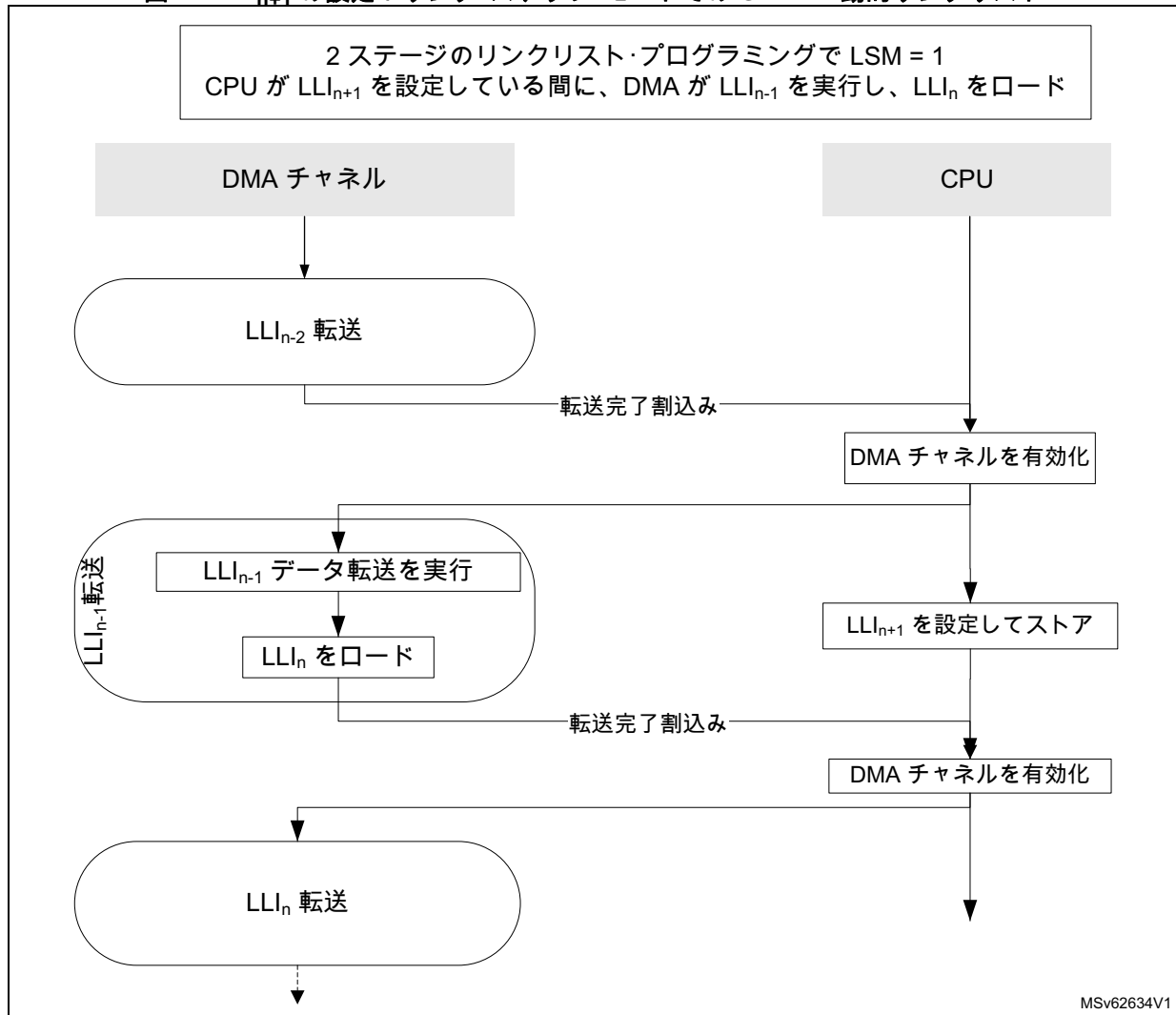
図 76. リンク・ステップ・モード(GPDMA_CxCR.LSM = 1) での GPDMA チャンネル実行およびリンクリスト・プログラミング



リンク・ステップ・モード（実行時）での LLI_{n+1} の追加

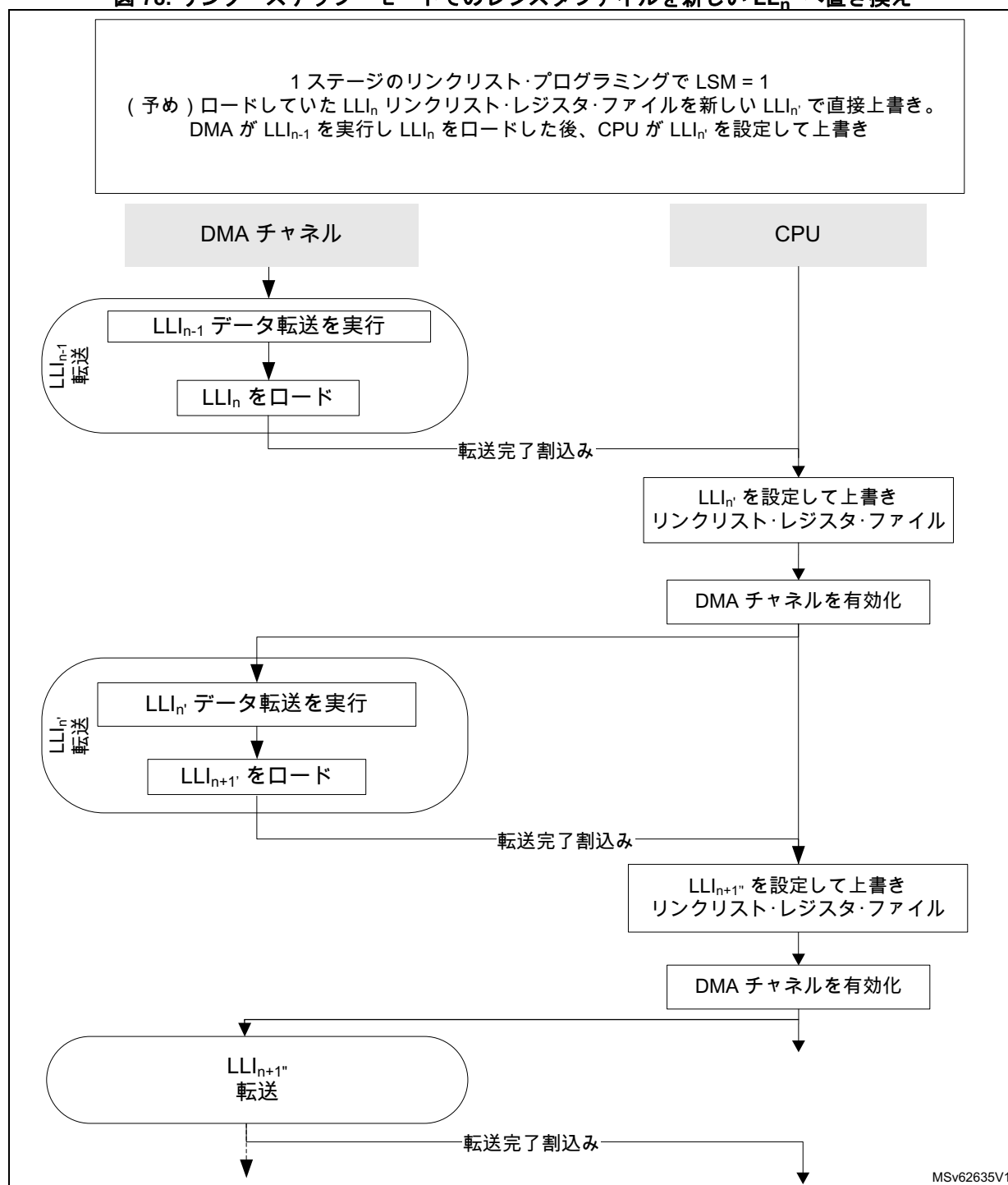
下の図に示したとおり、実行時、ソフトウェアは LLI_{n+1} （と次の LLI）の詳細を、GPDMA が LLI_{n-1} の転送を実行し、メモリからの LLI_n をロードするまで延期することができます。

図 77. LLI_{n+1} の設定：リンク・ステップ・モードでの GPDMA 動的リンクリスト



リンク・ステップ・モード（実行時）での LLI を新しい LLI_n へ置き換え (リンクリスト・レジスタ・ファイル)

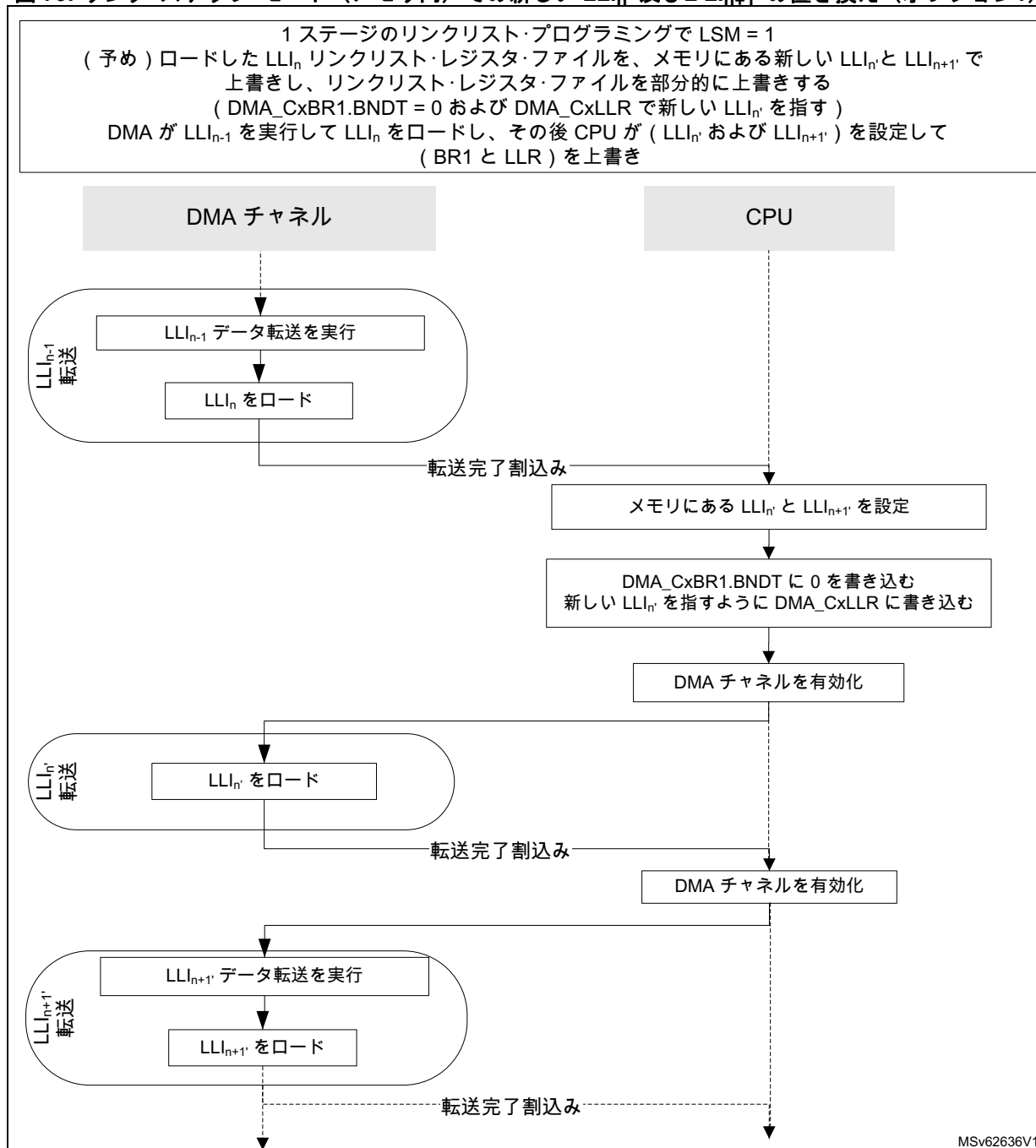
このリンク・ステップ・モードでは下の図に示すように、実行時、GPDMA が LLI_{n-1} からの転送の実行と、リンクリスト・リスト・レジスタ・ファイルを新しい LLI_n で直接上書きすることで、メモリから既に作成した LLI_n をロードした後、新しい LLI_n を設定し挿入することができます。

図 78. リンク・ステップ・モードでのレジスタファイルを新しい LLI_n' へ置き換え

リンク・ステップ・モード（実行時）でのLLIを新しいLLI_nへ置き換え（メモリ内）

ソフトウェアは、GPDMA が LLI_{n-1} からの転送を実行し、リンクリスト・レジスタ・ファイルを部分的に上書きして（GPDMA_CxBR1.BNDT[15:0] を null とし、GPDMA_CxLLR が新しい LLI_n を指すようにする）、メモリから既に作成した LLI_n をロードした後、メモリの新しい LLI_n と LLI_{n+1} を設定して挿入することができます。

図 79. リンク・ステップ・モード（メモリ内）での新しい LLI_n 及び LLI_{n+1} の置き換え（オプション1）

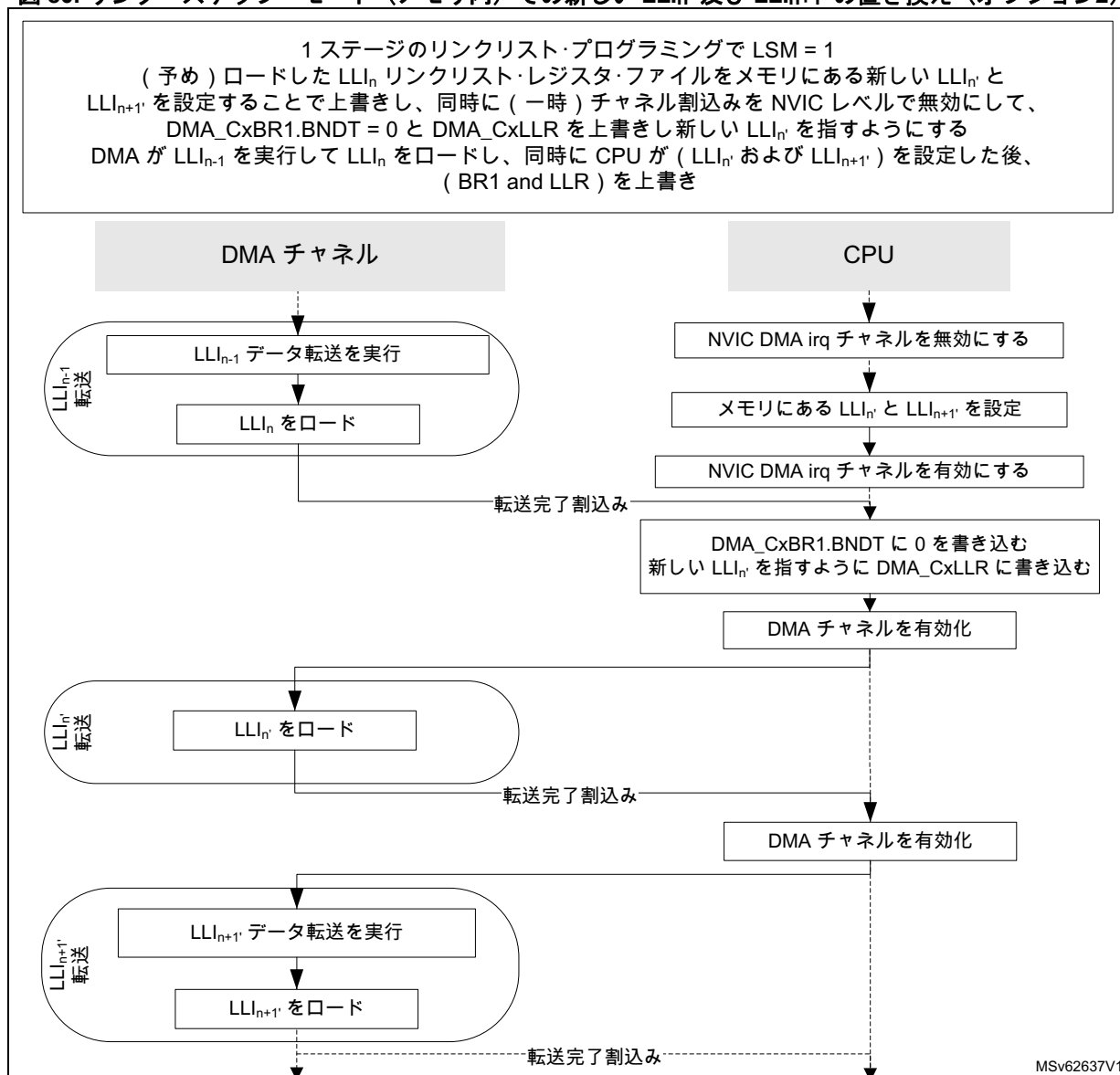


リンク・ステップ・モード（実行時）での LLI_n を新しい $LLI_{n'}$ へ置き換え

他のソフトウェア実装が存在しますが、一方で GPDMA は LLI_{n-1} からの転送を実行して、既に作成した LLI_n をメモリからロードし、（あるいはそれ以前に）ソフトウェアは次の項目を実行することができます。

1. 割込み操作による中断を防ぐために NVIC を無効にする。
2. 新しい $LLI_{n'}$ と $LLI_{n+1'}$ を設定する。
3. チャンネル割込み（転送完了）通知のために NVIC を再び有効にする。

LLI_{n-1} 割込みハンドラのソフトウェアはその後、GPDMA_CxBR1.BNDT[15:0] を null に上書きすることと、GPDMA_CxLLR が下の図に示すように新しい $LLI_{n'}$ を指すようにすることに限定されます。

図 80. リンク・ステップ・モード（メモリ内）での新しい $LLI_{n'}$ 及び $LLI_{n+1'}$ の置き換え（オプション2）

16.4.9 GPDMA チャネル状態およびリンクリスト・プログラミング

ソフトウェアは、チャネルが無効になっている時に (GPDMA_CxCR.EN = 0) チャネルを再設定し、実行モード (GPDMA_CxCR.LSM) を更新して run-to-completion モードとリンク・ステップ・モードとを相互に切り替えることができます。

あらゆる実行モードにおいてソフトウェアは次の項目を行うことができます。

- LLI_{n+1} が GPDMA チャネルによってロード/使用される前に、メモリ内の LLI_{n+1} を再プログラムして、この LLI_{n+1} によって最終的にチャネルを完了します (この LLI_{n+1} の GPDMA_CxLLR をクリアします)。
- チャネルを LSM 更新によりアポートおよび再設定 (セクション 16.4.4 を参照)

リンク・ステップ・モードでは、ソフトウェアが LLI_{n-1} の間、あらゆる LLI の各シングル実行後に LSM をクリアすることができます。

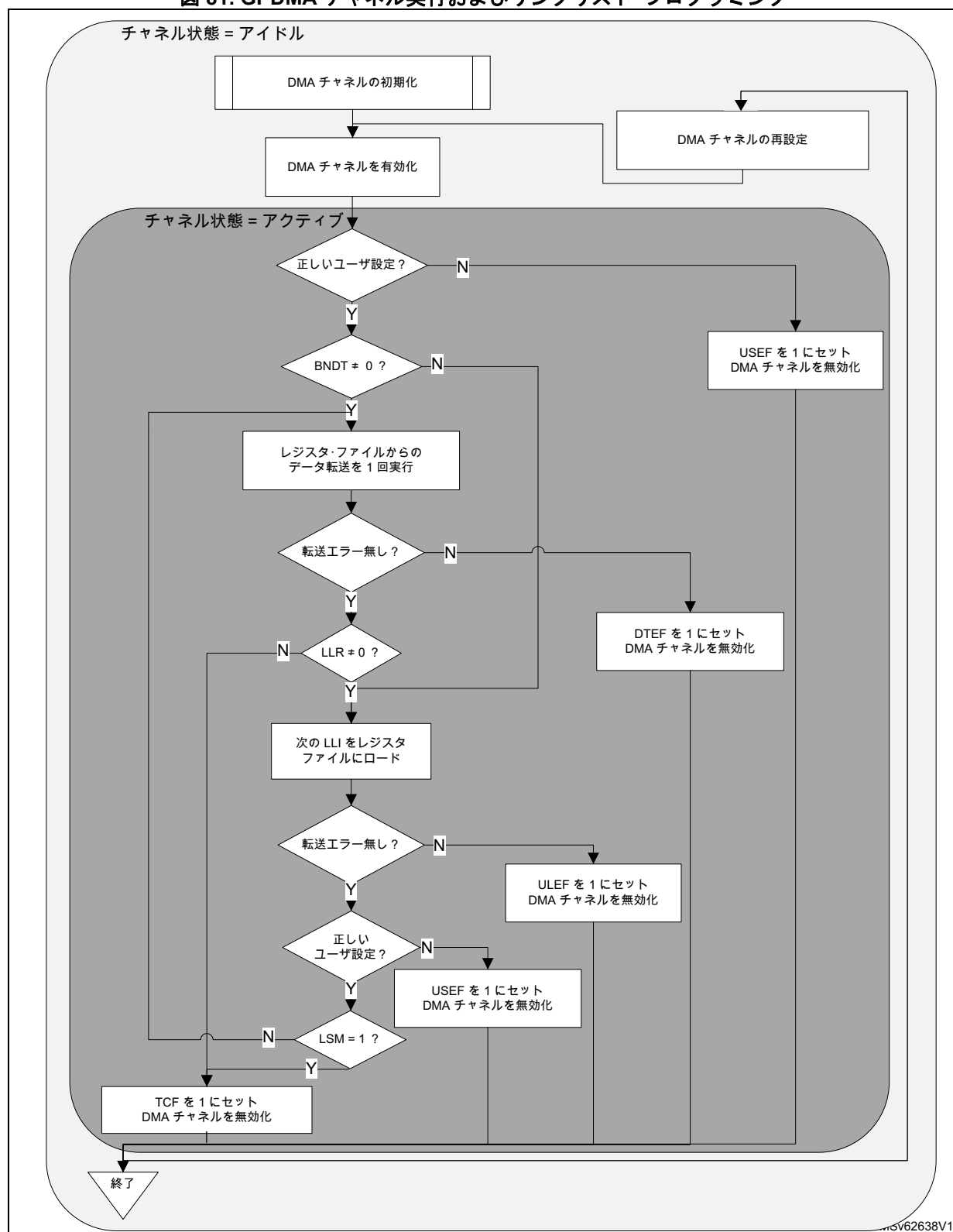
図 81 に実行モードに関係のない、全体の統合した GPDMA リンクリスト・プログラミングを示します。

注 :

図 81 は、GPDMA_CxTR2 の TCEM[1:0] のプログラムされた値により、TCEF をどれぐらい頻繁に発生させることができるかを示すことを意図したものではありません。これは (各) ブロックの完了時、(各) 2D ブロックの完了時、(各) LLI の完了時、あるいは最後の LLI データ転送の完了時のみに発生させることができます。run-to-completion モードではチャネル完了時の TCEM[1:0] の値に関わらず、ハードウェアは常に TCEF = 1 を設定し、そのチャネルを無効にします。リンク・ステップ・モードでは、LLI の各シングル実行後にチャネルが無効になり、TCEM[1:0] の値により TCEF を立てるかどうかが変わります。

図 81 では BNDT ≠ 0 が、最初のデータ転送開始の典型的な条件です。この条件は、ペリフェラルが早期に終了するデータ転送をリクエストすると (BNDT ≠ 0 および PFREQ = 1) となります (セクション 16.3.6 参照)。

図 81. GPDMA チャンネル実行およびリンクリスト・プログラミング



16.4.10 GPDMA FIFO を使用した転送

これは FIFO モードという 1 つの転送操作モードの説明です。FIFO を使用した FIFO ベースの転送は複数存在します。あらゆるチャネル x は専用の FIFO により実装され、そのサイズは `dma_fifo_size[x]` により定義されます（詳細は [セクション 16.3.2](#) を参照）。

GPDMA バースト

もっとも単純なプログラム転送は GPDMA バーストです。

GPDMA バーストは転送元から受け取ったデータ・バースト、または転送先へ送信されたデータ・バーストです。転送元（転送先）バーストは GPDMA_CxTR1 レジスタの `SBL_1[5:0]` (`DBL_1[5:0]`) フィールドにより定義されたバースト長、`SDW_LOG2[1:0]` (`DDW_LOG2[1:0]`) フィールドにより定義されたデータ幅でプログラムされます。

GPDMA バースト後の各データのアドレス指定モード (Beat (ビート) と呼ばれる) は GPDMA_CxTR1 の `SINC` および `DINC` によりそれぞれ転送元、転送先に対して、固定アドレス指定、あるいは隣接データに対するインクリメントアドレス指定のどちらかが定義されます。

GPDMA ソース/デスティネーション・バーストの開始および次のアドレス (GPDMA_CxSAR および GPDMA_CxDAR で定義される) は、それぞれのデータ幅に合っていないことはありません。

下の表に GPDMA バーストの主な特徴を列記します。

表 130. プログラムされた GPDMA ソース/デスティネーション・バースト

SDW_LOG2[1:0] DDW_LOG2[1:0]	データ幅 (バイト)	SINC/DINC	SBL_1[5:0] DBL_1[5:0]	バースト長 (データ／ ビート)	次のデータ／ ビート・ アドレス	次の バースト・ アドレス	バースト・ アドレス整合
00	1	0 (固定)	n = 0 から 63 ⁽¹⁾	n+1	+ 0	+ 0	1
01	2						2
10	4						4
00	1	1 (連続的な インクリメント)			+ 1	+ (n + 1)	1
01	2				+ 2	+ 2 × (n + 1)	2
10	4				+ 4	+ 4 × (n + 1)	4
11	ユーザ設定禁止。USEF が生成されバーストが発行されません。						

1. `S/DBL_1[5:0] = 0` の場合、バースト長は 1。この時、バーストはシングルとも呼ばれます。

上の表の次のバースト・アドレスは、プログラムされたソース/デスティネーション・バーストが一旦、完了したときに、GPDMA_CxSAR または GPDMA_CxDAR により指定された次の転送元/転送先のデフォルト・アドレスです。このデフォルト値は固定/連続的インクリメント・アドレスを参照します。

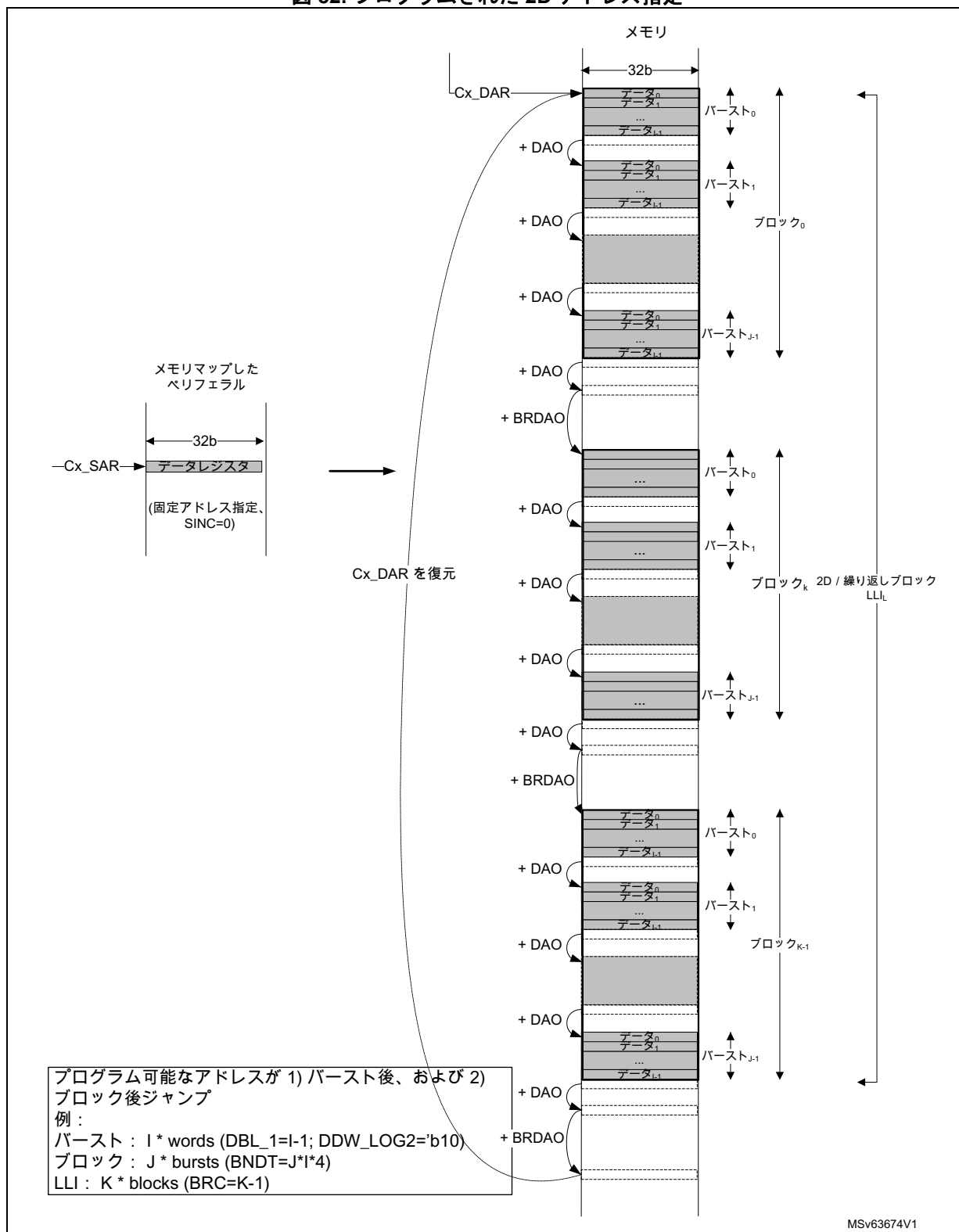
2D アドレス指定による GPDMA バースト (チャンネル x = 6、7)

チャンネルに追加の 2D アドレス指定機能がある場合、このデフォルト値は 2 個のプログラムされたインクリメントまたはデクリメント・オフセットを考慮しない値を参照します。これら 2 つの追加オフセット (デフォルト値は null) は次の場合に適用されます。

- それぞれ GPDMA_CxTR2.SAO[12:0]/DAO[12:0] および GPDMA_CxBR1.SDEC/DDEC で定義される、完了した各ソース／デスティネーション・バーストの後
- それぞれ GPDMA_CxBR2.BRSAO[15:0]/BRDAO[15:0] および GPDMA_CxBR1.BRSDEC/BRDDEC で定義される、完了した各ブロックの後

下に示す 2D 転送先バッファの図に示したとおり、各バースト完了後、最初にプログラムされたアドレス・ジャンプと各ブロックの後の 2 番目にプログラムされたアドレス・ジャンプで 2D／繰り返しブロックをアドレス指定することが可能です。

図 82. プログラムされた 2D アドレス指定



GPDMA FIFO を使用した転送

FIFO モードの転送は一般的に 2 つのパイプライン化され分離したバースト転送からなります。

- 転送元から FIFO へのバーストの 1 つは、GPDMA_CxTR1.SAP で定義されるとおり、割り当てられた転送元マスタ・ポートを経由します
- FIFO から転送先への 1 つのバーストは、GPDMA_CxTR1.DAP で定義されるとおり、割り当てられた転送先マスタ・ポートを経由します

GPDMA ソース・バースト

リクエストされた FIFO へのソース・バースト転送は、割り当てられたポートを経由して、プログラムされたバースト・サイズに対する現在の FIFO レベルにより、できるだけ早くスケジュールすることが可能です (FIFO が転送元から新しいバーストを取得する準備ができたとき)。

FIFO レベル $\leq 2^{\text{dma_fifo_size}[x]} - (\text{SBL_1}[5:0] + 1) \times 2^{\text{SDW_LOG2}[1:0]}$ のとき。

ここで、

- FIFO レベルは現在の FIFO のフィル・レベル (バイト) です。
- $2^{\text{dma_fifo_size}[x]}$ はチャンネル x の FIFO サイズの半分 (バイト) です (実装の詳細と $\text{dma_fifo_size}[x]$ 値は [セクション 16.3.2](#) 参照)。
- $(\text{SBL_1}[5:0] + 1) \times 2^{\text{SDW_LOG2}[1:0]}$ はプログラムされたソース・バースト転送のサイズ (バイト) です。

チャンネルの優先順位 (GPDMA_CxCR.PRIO[1:0]) に基づき、この準備された FIFO ベースのソース転送は、内部で他のリクエストされ、アクティブなチャンネルに対してアービトレーションされます。

GPDMA デスティネーション・バースト

リクエストされた FIFO からのデスティネーション・バースト転送は、割り当てられたポートを経由して、プログラムされたバースト・サイズに対する現在の FIFO レベルにより、できるだけ早くスケジュールすることが可能です (FIFO が転送先に 1 つの新しいバーストをブッシュする準備ができたとき)。

FIFO レベル $\geq (\text{DBL_1}[5:0] + 1) \times 2^{\text{DDW_LOG2}[1:0]}$ のとき。

ここで、

- FIFO レベルは現在の FIFO のフィル・レベル (バイト) です。
- $(\text{DBL_1}[5:0] + 1) \times 2^{\text{DDW_LOG2}[1:0]}$ はプログラムされたデスティネーション・バースト転送のサイズ (バイト) です。

チャンネルの優先順位に基づき、この準備された FIFO ベースのデスティネーション転送は、内部で他のリクエストされ、アクティブなチャンネルに対してアービトレーションされます。

GPDMA バーストと転送元ブロックサイズ、1 KB アドレス境界および FIFO サイズ

プログラムされたソース/デスティネーション GPDMA バーストは、次のいずれかの条件を満足しない場合には、AHB バーストとしてそのまま実装されます。

- チャンネル x の FIFO サイズの半分がプログラムされたソース/デスティネーション・バースト・サイズより小さい場合、プログラムされたソース/デスティネーション GPDMA バーストが、ユーザの一切の制約なく、各転送が FIFO サイズの半分以下の小さいサイズの一連のシングルまたはバースト転送で実装されます。
- 転送元のブロック・サイズ (GPDMA_CxBR1.BNDT[15:0]) が転送元のバースト・サイズの倍数ではなく、ソース・バーストのデータ幅 (GPDMA_CxTR1.SDW_LOG2[1:0]) の倍数の場合、GPDMA は、ユーザの一切の制約なく、シングルまたは、より短いバーストへと変更、短縮を行い、転送元ブロックの厳密なサイズを転送します。

- ソース/デスティネーション・バースト転送が AHB 転送の 1 KB のアドレス境界を越えた場合、GPDMA はプログラムされたバーストをシングルまたは、より短いバーストへと変更、短縮を行い、一切のユーザによる制約なく、AHB プロトコルに準拠するようにします。
- AHB 転送におけるソース/デスティネーション・バースト長が 16 を超えた場合、GPDMA はプログラムされたバーストをより短いシングルまたはバーストに変更、短縮を行い、一切のユーザの制約なく、AHB プロトコルに準拠するようにします。

いずれの場合も GPDMA は転送元/転送先データ（とアドレス）の整合性を一切のユーザの制約なく確保します。（GPDMA_CxSR でソフトウェア読取り可能な）現在の FIFO レベルは、実効的な転送サイズと比較して更新され、変更された可能性のある、AHB の各シングルまたはバースト転送間のアービトレーションを GPDMA が再度行います。

各シングル、またはプログラムされたバーストより小さいバースト・サイズのバースト転送は、他のリクエストされた有効なチャンネルに対し、チャンネルの優先順位に基づいて内部でアービトレーションされます。

注： リンクリスト・モードでのリンクリスト・パラメータの更新に関係したメモリから内部 GPDMA レジスタへの GPDMA 読取り転送は、GPDMA_CxCR.LAP によるプログラムに従って、リンクが割り当てられたポートでスケジュールされます。

GPDMA データ・ハンドリング：バイト・ベースの再整列、パッキング/アンパッキング、パディング/切捨て、符号拡張、左/右詰め

データ・ハンドリングは GPDMA_CxTR1 により制御されます。プログラムされたバーストの転送元/転送先データ幅は、SDW_LOG2[21:0] および DDW_LOG2[1:0] フィールドに従って、バイト、ハーフワードまたはワードです（表 131 を参照）。

ユーザは、転送データの転送元と転送先の間でのデータ・ハンドリングを設定することができます。具体的には、プログラムされたデータ・ハンドリングは次のように整然と実行されます。

1. バイトベースの転送元、再整列
 - SBX = 1 で転送元のデータ幅がワードの場合、各転送元データ・ワードの中央のアンアラインド・ハーフワードの 2 バイトを交換します。
2. 転送先データ幅が転送元データ幅と異なる場合、PAM[1:0] に従った、パッキングやアンパッキング、パディング、または切捨てによるデータ幅の変換。
 - 転送先データ幅 > 転送元データ幅の場合、ポスト SBX 転送元データは、右寄せで 0s でパディングするか、転送先データ幅まで符号拡張するか、あるいは、FIFO キューに入れて転送先データ幅までパッキングします。
 - 転送先データ幅 < 転送元データ幅の場合、ポスト SBX データは、右寄せで左に転送先データ幅で切捨てるか、FIFO キューに入れてアンパッキングし、転送先データ幅までストリーム・ダウンします。
3. バイト・ベースの転送先の再整列。
 - DBX = 1 で転送先データ幅がバイトではない場合、整列したポスト PAM[1:0] ハーフワード内の 2 バイトを交換します。
 - DHX = 1 で転送先データ幅がバイトでもハーフワードでもない場合、整列したポスト PAM[1:0] ワード内の 2 つの整列したハーフワードを交換します。

注： 0s のパディングによる左詰めは、0s のパディングによる右詰めと、転送先のバイト・ベースの再整列の両方をプログラミングすることで行うことができます。

下の表は転送元から転送先への考えられるデータ・ハンドリングを列記したものです。

表 131. プログラムされたデータ・ハンドリング

SDW_ LOG2 [1:0]	転送元 データ	転送元 データ・ ストリーム ⁽¹⁾	SBX	DDW_ LOG2 [1:0]	転送先データ	PAM[1:0] ⁽²⁾	DBX	DHX	転送先データ・ストリーム (1)
00	バイト	B ₇ 、B ₆ 、B ₅ 、 B ₄ 、B ₃ 、B ₂ 、 B ₁ 、B ₀	x	00	バイト	xx	x		B ₇ 、B ₆ 、B ₅ 、B ₄ 、B ₃ 、B ₂ 、B ₁ 、 B ₀
				01	ハーフワード	00 (RA, 0P)	0	x	0B ₃ 、0B ₂ 、0B ₁ 、0B ₀
							1		B ₃ 0、B ₂ 0、B ₁ 0、B ₀ 0
						01 (RA, SE)	0		SB ₃ 、SB ₂ 、SB ₁ 、SB ₀
							1		B ₃ S、B ₂ S、B ₁ S、B ₀ S
						1x (PACK)	0		B ₇ B ₆ 、B ₅ B ₄ 、B ₃ B ₂ 、B ₁ B ₀
							1		B ₆ B ₇ 、B ₄ B ₅ 、B ₂ B ₃ 、B ₀ B ₁
				10	ワード	00 (RA, 0P)	0	0	000B ₁ 、000B ₀
							1		00B ₁ 0、00B ₀ 0
							0	1	0B ₁ 00、0B ₀ 00
							1		B ₁ 000、B ₀ 000
						01 (RA, SE)	0	0	SSSB ₁ 、SSSB ₀
							1		SSB ₁ S、SSB ₀ S
							0	1	SB ₁ SS、SB ₀ SS
							1		B ₁ SSS、B ₀ SSS
						1x (PACK)	0	0	B ₇ B ₆ B ₅ B ₄ 、B ₃ B ₂ B ₁ B ₀
							1		B ₆ B ₇ B ₄ B ₅ 、B ₂ B ₃ B ₀ B ₁
							0	1	B ₅ B ₄ B ₇ B ₆ 、B ₁ B ₀ B ₃ B ₂
							1		B ₄ B ₅ B ₆ B ₇ 、B ₀ B ₁ B ₂ B ₃
01	ハーフ ワード	B ₇ B ₆ 、B ₅ B ₄ 、 B ₃ B ₂ 、B ₁ B ₀		00	バイト	00 (RA, LT)	x	x	B ₆ 、B ₄ 、B ₂ 、B ₀
						01 (LA, RT)			B ₇ 、B ₅ 、B ₃ 、B ₁
						1x (UNPACK)			B ₇ 、B ₆ 、B ₅ 、B ₄ 、B ₃ 、B ₂ 、B ₁ 、 B ₀

表 131. プログラムされたデータ・ハンドリング (続き)

SDW_ LOG2 [1:0]	転送元 データ	転送元 データ・ ストリーム ⁽¹⁾	SBX	DDW_ LOG2 [1:0]	転送先データ	PAM[1:0] ⁽²⁾	DBX	DHX	転送先データ・ストリーム (1)
01	ハーフ ワード	B ₇ B ₆ 、B ₅ B ₄ 、 B ₃ B ₂ 、B ₁ B ₀	x	01	ハーフワード	xx	0	x	B ₇ B ₆ 、B ₅ B ₄ 、B ₃ B ₂ 、B ₁ B ₀
							1		B ₆ B ₇ 、B ₄ B ₅ 、B ₂ B ₃ 、B ₀ B ₁
				10	ワード	00 (RA, 0P)	0	0	00B ₃ B ₂ 、00B ₁ B ₀
							1		00B ₂ B ₃ 、00B ₀ B ₁
							0	1	B ₃ B ₂ 00、B ₁ B ₀ 00
							1		B ₂ B ₃ 00、B ₀ B ₁ 00
						01 (RA, SE)	0	0	SSB ₃ B ₂ 、SSB ₁ B ₀
							1		SSB ₂ B ₃ 、SSB ₀ B ₁
							0	1	B ₃ B ₂ SS、B ₁ B ₀ SS
							1		B ₂ B ₃ SS、B ₀ B ₁ SS
						1x (PACK)	0	0	B ₇ B ₆ B ₅ B ₄ 、B ₃ B ₂ B ₁ B ₀
							1		B ₆ B ₇ B ₄ B ₅ 、B ₂ B ₃ B ₀ B ₁
							0	1	B ₅ B ₄ B ₇ B ₆ 、B ₁ B ₀ B ₃ B ₂
							1		B ₄ B ₅ B ₆ B ₇ 、B ₀ B ₁ B ₂ B ₃
10	ワード	B ₇ B ₆ B ₅ B ₄ 、 B ₃ B ₂ B ₁ B ₀	0	00	バイト	00 (RA, LT)	x	x	B ₁₂ 、B ₈ 、B ₄ 、B ₀
						01 (LA, RT)			B ₁₅ 、B ₁₁ 、B ₇ 、B ₃
						10 (UNPACK)			B ₇ 、B ₆ 、B ₅ 、B ₄ 、B ₃ 、B ₂ 、B ₁ 、 B ₀
				01	ハーフワード	00 (RA, LT)	0		B ₅ B ₄ 、B ₁ B ₀
							1		B ₄ B ₅ 、B ₀ B ₁
						01 (LA, RT)	0		B ₇ B ₆ 、B ₃ B ₂
							1		B ₆ B ₇ 、B ₂ B ₃
						1x (UNPACK)	0		B ₇ B ₆ 、B ₅ B ₄ 、B ₃ B ₂ 、B ₁ B ₀
							1		B ₆ B ₇ 、B ₄ B ₅ 、B ₂ B ₃ 、B ₀ B ₁

表 131. プログラムされたデータ・ハンドリング (続き)

SDW_ LOG2 [1:0]	転送元 データ	転送元 データ・ ストリーム ⁽¹⁾	SBX	DDW_ LOG2 [1:0]	転送先データ	PAM[1:0] ⁽²⁾	DBX	DHX	転送先データ・ストリーム (1)
10	ワード	B ₇ B ₆ B ₅ B ₄ 、 B ₃ B ₂ B ₁ B ₀	0	10	ワード	xx	0	0	B ₇ B ₆ B ₅ B ₄ 、 B ₃ B ₂ B ₁ B ₀
							1		B ₆ B ₇ B ₄ B ₅ 、 B ₂ B ₃ B ₀ B ₁
							0	1	B ₅ B ₄ B ₇ B ₆ 、 B ₁ B ₀ B ₃ B ₂
							1		B ₄ B ₅ B ₆ B ₇ 、 B ₀ B ₁ B ₂ B ₃
			1	00	バイト	00 (RA, LT)	x	x	B ₁₂ 、 B ₈ 、 B ₄ 、 B ₀
						01 (LA, RT)			B ₁₅ 、 B ₁₁ 、 B ₇ 、 B ₃
						1x (UNPACK)			B ₇ 、 B ₅ 、 B ₆ 、 B ₄ 、 B ₃ 、 B ₁ 、 B ₂ 、 B ₀
				01	ハーフワード	00 (RA, LT)	0		B ₆ B ₄ 、 B ₂ B ₀
							1		B ₄ B ₆ 、 B ₀ B ₂
						01 (LA, RT)	0		B ₇ B ₅ 、 B ₃ B ₁
							1		B ₅ B ₇ 、 B ₁ B ₃
						1x (UNPACK)	0		B ₇ B ₅ 、 B ₆ B ₄ 、 B ₃ B ₁ 、 B ₂ B ₀
							1		B ₅ B ₇ 、 B ₄ B ₆ 、 B ₁ B ₃ 、 B ₀ B ₂
				10	ワード	xx	0	0	B ₇ B ₅ B ₆ B ₄ 、 B ₃ B ₁ B ₂ B ₀
							1		B ₅ B ₇ B ₄ B ₆ 、 B ₁ B ₃ B ₀ B ₂
							0	1	B ₆ B ₄ B ₇ B ₅ 、 B ₂ B ₀ B ₃ B ₁
							1		B ₄ B ₆ B ₅ B ₇ 、 B ₀ B ₂ B ₁ B ₃

- データ・ストリームを最低インデックス (B₀) のバイトから始めて、タイミングで整列します。
- RA = 右詰め、LA = 左詰め、RT = 右切捨て、LT = 左切捨て、0P = 転送先データ幅までゼロ・ビット・パディング、SE = 転送先データ幅まで符号ビット拡張。

16.4.11 GPDMA 転送リクエストとアービトレーション

GPDMA 転送リクエスト

GPDMA_CxTR2 の定義に従って、プログラムされた GPDMA データ転送を次のいずれかの方法でリクエストします。

- 制御ビット SWREQ = 1 の場合、ソフトウェア・リクエスト：これは通常 CPU により、あるメモリ・マッピングされたアドレスから別のメモリ・マッピングされたアドレスへのデータ転送 (メモリ間、GPIO とメモリの相互転送) に使用されます。
- SWREQ = 0 の場合、ペリフェラルから来る入力ハードウェア・リクエスト：GPDMA ハードウェア・ペリフェラル・リクエストの選択は REQSEL[7:0] フィールドにより行われます ([セクション 16.3.4](#) 参照)。選択したハードウェア・リクエストは次のいずれかです。
 - GPDMA モードで設定されたペリフェラルからのハードウェア・リクエスト (ペリフェラル・データ・レジスタとメモリの相互転送)
 - メモリから制御レジスタを更新するための、ペリフェラルからのハードウェア・リクエスト
 - メモリに転送されたステータス・レジスタの読出しのための、ペリフェラルからのハードウェア・リクエスト

注意： ユーザーは、特定の時点でこの要求がペリフェラルによってアサートされ、各チャネルがこの要求されたデータ転送を実行する準備ができている場合、GPDMA_CxTR.REQSEL[7:0] を介して同じ入力ハードウェア ペリフェラル GPDMA 要求を 2 つの異なるチャネルに割り当ててはなりません。ユーザ設定エラーのレポートはありません。

アービトレーションのための GPDMA 転送リクエスト

割り当てられたマスタ・ポート (GPDMA_CxTR1.SAP) を使ってスケジュールされる、準備の整った (転送元アドレスから FIFO への) FIFO ベースの GPDMA ソース・シングル/バースト転送は、他の同時にリクエストされた、同じマスタ・ポートへの GPDMA 転送に対して、チャネルの優先順位 (GPDMA_CxCR.PRIO[1:0]) に基づいてアービトレーションされます。

割り当てられたマスタ・ポート (GPDMA_CxTR1.DAP) を使ってスケジュールされる、準備の整った (FIFO から転送先アドレスへの) FIFO ベースの GPDMA デスティネーション・シングル/バースト転送は、他の同時にリクエストされた、同じマスタ・ポートへの GPDMA 転送に対して、チャネルの優先順位 (GPDMA_CxCR.PRIO[1:0]) に基づいてアービトレーションされます。

GPDMA にリクエストされ、アービトレーションされたリンク転送は、メモリのリンクリスト・データ構造からリンクリスト・レジスタ (GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR または GPDMA_CxLLR に加え GPDMA_CxTR3、GPDMA_CxBR2) の 1 つへの、1 回の 32 bit 読出しにより構成されます。メモリからの各 32 bit 読出しは、データ転送と同じチャネルの優先順位でアービトレーションされ、割り当てられたマスタ・ポート (GPDMA_CxCR.LAP) でスケジューリングされます。

メモリ間転送のためにソフトウェア・リクエストでプログラムされたデータ転送 (GPDMA_CxTR2.SWREQ = 1) が何であれ、あるいはメモリとペリフェラル間のハードウェアがリクエストした転送 (GPDMA_CxTR2.SWREQ = 0) が何であれ、または、そのハードウェア・リクエストの種類が何であれ、許可された各シングル/バースト転送後に再アービトレーションが行われます。

転送先のペリフェラルでハードウェア・リクエストがプログラムされると (GPDMA_CxTR2.SWREQ = 0 および GPDMA_CxTR2.DREQ = 1)、最初のメモリ読出し (おそらく 2D/繰り返し) ブロック (最初のレディになっている FIFO ベースのソース・バースト・リクエスト) は、対応する選択されたハードウェア・リクエストの発生によりゲートされます。この最初のメモリへの読出しリクエストは、早い段階では (ブロック転送が有効になって実行可能になって直ぐには) アービタでは考慮されません。

GPDMA アービトレーション

GPDMA アービトレーションは 4 グレードが割り当てられたチャネルの優先順位 (GPDMA_CxCR.PRIO[1:0]) から行われます。図 83 に図示したアービトレーション・ポリシーは次のように定義されます。

- 優先順位の高い 1 つのトラフィック・クラス (キュー 3)。優先順位 3 が割り当てられたタイミングが重要なチャネル専用。

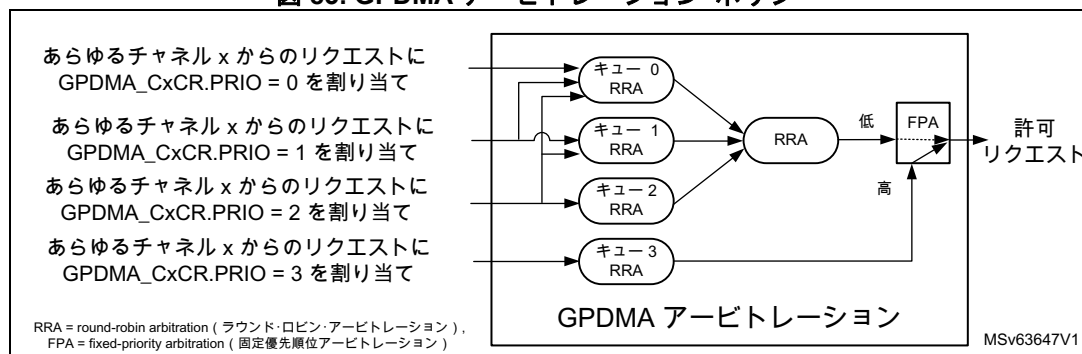
このトラフィック・クラスは、他のあらゆる低優先順位のトラフィック・クラスに対して、固定優先順位アービトレーションで許可されます。このクラス内では、リクエストされたシングル/バースト転送はラウンド・ロビン (持ち回り) でアービトレーションされます。
- 優先順位の低い 3 つのトラフィック・クラス (キュー 0、1、2)。タイミングの制約が緩い、優先順位 0、1、2 のチャネル。

このクラスの、リクエストされたそれぞれのシングル/バースト転送は、プログラムされた優先順位で単純な重み付けを行い、ラウンド・ロビン (持ち回り) でアービトレーションされます：

 - 優先順位 0 のリクエストはキュー 0 に割り当てられます。
 - 優先順位 1 のリクエストはキュー 0 と 1 に割り当て、複製されます。
 - 優先順位 2 のリクエストはキュー 0、1、2 に割り当て、複製されます。

- キュー 0、1、2 は、同時リクエストがある場合、その有効な入力リクエストをラウンド・ロビン（持ち回り）方式で平等に許可します。
- さらに、低優先順位のトラフィックの第 2 ステージがあり、そこにはキュー 0、1、2 から同時に選択されたリクエスト間で公平にオルタネートするラウンド・ロビン（持ち回り）のアービタがあります。

図 83. GPDMA アービトレーション・ポリシー



GPDMA アービトレーションとバンド幅

このアービトレーション・ポリシーでは、次の事項が保証されます。

- 同じポリシーのリクエストの最大バンド幅は等しい
- タイミングが重要なリクエスト（優先順位 3）の予約済みのバンド幅 (B_{Q3} と表記)
- 様々な、低い優先順位のリクエスト（優先順位 0、1、2 の間）に対する残りの重み付けバンド幅。

次の 2 つの例では、重み付けラウンド・ロビン（持ち回り）アービトレーションがプログラムされた優先順位で実施されていることがよく分かります。

- 例 1:** 2 つのタイミングの制約が緩い GPDMA リクエスト、req0 と req1 の基本アプリケーション。次のようなプログラミングの可能性があります。
 - 同じ優先順位が割り当てられている場合、アービタにより req0 に割り当てられるバンド幅 (B_{req0}) は req1 に割り当てられるバンド幅 (B_{req1}) と同じです。

$$B_{req0} = B_{req1} = 1/2 \times (1 - B_{Q3})$$
 - req0 が優先順位 0 に、req1 が優先順位 1 に割り当てられると、req0 に割り当てられるバンド幅 (B_{P0}) は req1 に割り当てられるバンド幅 (B_{P1}) の **3 分の 1** になります。

$$B_{req0} = B_{P0} = 1/2 \times 1/2 \times (1 - B_{Q3}) = 1/4 \times (1 - B_{Q3})$$

$$B_{req1} = B_{P1} = (1/2 + 1) \times 1/3 \times (1 - B_{Q3}) = 3/4 \times (1 - B_{Q3})$$
 - req0 が優先順位 0 に、req1 が優先順位 2 に割り当てられると、req0 に割り当てられるバンド幅 (B_{P0}) は req1 に割り当てられるバンド幅 (B_{P2}) の **5 分の 1** になります。

$$B_{req0} = B_{P0} = 1/2 \times 1/3 \times (1 - B_{Q3}) = 1/6 \times (1 - B_{Q3})$$

$$B_{req1} = B_{P2} = (1/2 + 1 + 1) \times 1/3 \times (1 - B_{Q3}) = 5/6 \times (1 - B_{Q3})$$

上で計算したバンド幅の計算は理論的な入力リクエストに基づいており、あらゆる GPDMA クロック・サイクルで常に有効です。アービタから計算されたこのバンド幅は、アプリケーションにより得られたリクエストの頻度により重み付けする必要があり、これは、常に有効とは限らず、ある GPDMA クライアント（例えば 400 kHz で I2C）と別のクライアント（1 kHz の PWM）とでは、上で示した 3 倍や 5 倍とは大きく異なる場合があります。

- **例 2 :** ユーザが同じ null ではない GPDMA リクエストの N 数を、すべてのタイミングの制約が緩い優先順位 0、1、2 に配布するアプリケーション。このときのバンド幅の計算は次のとおりです。
 - 優先順位 0 の一連のリクエストに対して割り当てたバンド幅 (B_{P0}) は、

$$B_{P0} = 1/3 \times 1/3 \times (1 - B_{Q3}) = 1/9 \times (1 - B_{Q3})$$
 - 優先順位 1 の一連のリクエストに対して割り当てたバンド幅 (B_{P1}) は、

$$B_{P1} = (1/3 + 1/2) \times 1/3 \times (1 - B_{Q3}) = 5/18 \times (1 - B_{Q3})$$
 - 優先順位 2 の一連のリクエストに対して割り当てたバンド幅 (B_{P2}) は、

$$B_{P2} = (1/3 + 1/2 + 1) \times 1/3 \times (1 - B_{Q3}) = 11/18 \times (1 - B_{Q3})$$
 - その優先順位 P_i ($i = 0$ から 2) の N 個のリクエストの中の n リクエストに割り当てるバンド幅 (B_n) は $B_n = 1/N \times B_{P_i}$
 - 優先順位 0_i のリクエスト n に割り当てたバンド幅 (B_{n, P_i}) は、

$$B_{n, P0} = 1/N \times 1/9 \times (1 - B_{Q3})$$

$$B_{n, P1} = 1/N \times 5/18 \times (1 - B_{Q3})$$

$$B_{n, P2} = 1/N \times 11/18 \times (1 - B_{Q3})$$

この例では、マスタ・ポート・バスのバンド幅がタイミングが重要なキュー 3 により完全には消費されていない場合、残りのバンド幅は、優先順位 0 のリクエストに割り当てられているバンド幅が優先順位 1 のリクエストに割り当てられているバンド幅の 2.5 分の 1、優先順位 2 のリクエストに割り当てられているバンド幅に対しては 5.5 分の 1 になるように割り当てられます。

より一般的には、次のリクエストが存在することを想定します。

- I 個のリクエスト ($I \geq 0$) を優先順位 0 に割り当て。
 $I > 0$ の場合、これらのリクエストは $i = 0$ から $I-1$ で記録されます。
- J 個のリクエスト ($J \geq 0$) を優先順位 1 に割り当て。
 $J > 0$ の場合、これらのリクエストは $j = 0$ から $J-1$ で記録されます。
- K 個のリクエスト ($K \geq 0$) を優先順位 2 に割り当て。
 これらのリクエストは $k = 0$ から $K-1$ で記録されます。
- L 個のリクエスト ($L \geq 0$) を優先順位 3 に割り当て。
 $L > 0$ 場合、これらのリクエストは $l = 0$ から $L-1$ で記録されます。

B_{Q3} はタイミングが重要なリクエストのために予約されているバンド幅なので、優先順位 3 の各リクエスト L のバンド幅は、

- $L > 0$ の場合、 $B_l = B_{Q3} / L$ (その他の場合は、 $B_l = 0$)

タイミングが重要ではないキューのそれぞれのバンド幅は次のとおりです。

- $B_{Q0} = 1/3 \times (1 - B_{Q3})$
- $B_{Q1} = 1/3 \times (1 - B_{Q3})$
- $B_{Q2} = 1/3 \times (1 - B_{Q3})$

優先順位 0 の一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P0} = I / (I + J + K) \times B_{Q0}$

優先順位 0 の各リクエスト i に対するバンド幅は次のとおりです。

- $L > 0$ の場合、 $B_i = B_{P0} / I$ (その他の場合は $B_{P0} = 0$)

優先順位 1 で キュー 0 にルーティングされる一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P1, Q0} = J / (I + J + K) \times B_{Q0}$

優先順位 1 で キュー 1 にルーティングされる一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P1,Q1} = J / (J + K) \times B_{Q1}$

優先順位 1 の一連のリクエストに対するバンド幅の合計は次のとおりです。

- $B_{P1} = B_{P1,Q0} + B_{P1,Q1}$

優先順位 1 のリクエスト j に対するバンド幅は次のとおりです。

- $J > 0$ の場合、 $B_j = B_{P1} / J$ (その他の場合は $B_j = 0$)

優先順位 2 で キュー 0 にルーティングされる一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P2,Q0} = K / (I + J + K) \times B_{Q0}$

優先順位 2 で キュー 1 にルーティングされる一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P2,Q1} = K / (J + K) \times B_{Q1}$

優先順位 2 で キュー 2 にルーティングされる一連のリクエストに対するバンド幅は次のとおりです。

- $B_{P2,Q2} = B_{Q2}$

優先順位 2 の一連のリクエストに対するバンド幅の合計は次のとおりです。

- $B_{P2} = B_{P2,Q0} + B_{P2,Q1} + B_{P2,Q2}$

優先順位 2 のリクエスト k に対するバンド幅は次のとおりです。

- $B_k = B_{P2} / K$ (一般的には $K > 0$)

そのため、タイミングが重要ではないリクエスト i 、 j 、 k に割り当てられる残留バンド幅の最大値は最終的に次のようになります。

- 一般的な場合 (優先順位 2 のリクエスト k が最低 1 つある場合 ($K > 0$))。

- $B_i = 1/I \times 1/3 \times I/(I + J + K) \times (1 - B_{Q3})$
- $B_j = 1/J \times 1/3 \times [J/(I + J + K) + J/(J + K)] \times (1 - B_{Q3})$
- $B_k = 1/K \times 1/3 \times [K/(I + J + K) + K/(J + K) + 1] \times (1 - B_{Q3})$

- 特別な場合 (優先順位 2 のリクエスト k がない場合 ($K = 0$))。

- $B_i = 1/I \times 1/2 \times I/(I + J) \times (1 - B_{Q3})$
- $B_j = 1/J \times 1/2 \times [J/(I + J) + 1] \times (1 - B_{Q3})$

結果として GPDMA アービタは、各キューや、より一般的には各リクエスト／チャンネルに対するプログラム可能な重み付けバンド幅のリミッターとして使用することができます。重み付けの差はプログラムされたチャンネルの優先順位から単純に与えられることになるのです。

16.4.12 GPDMA トリガ転送

プログラムされた GPDMA 転送は、選択された入力トリガ・イベントの立上がり／立下りエッジにより、GPDMA_CxTR2.TRIGPOL[1:0] および GPDMA_CxTR2.TRIGSEL[5:0] に従ってトリガすることができます (セクション 16.3.7 の「トリガ選択 (trigger selection)」参照)。

GPDMA_CxTR2.TRIGM[1:0] のトリガ・モードで定義されたトリガ転送は、LLI のデータ転送レベルで、ブロックの最初のバースト読出し、チャンネル x の 2D／繰返しブロックの最初のバースト読出し ($x = 6$ から 7)、あるいはプログラムされた各シングル読出しの条件付けを行います。トリガ・モードは LLI リンク転送の条件付けをプログラムをすることも可能です (詳細は GPDMA_CxTR2 の TRIGM[1:0] 参照)。

トリガ・ヒットのメモリ書込みとトリガ・オーバーラン・フラグの生成

新しい有効なトリガ設定（選択したトリガの立上がり、または立下りエッジ（それぞれTRIGPOL[1:0] = 01 と TRIGPOL[1:0] = 10））でチャンネルが有効化／ロードされると、チャンネル x のトリガに対する GPDMA モニタリングを開始します。

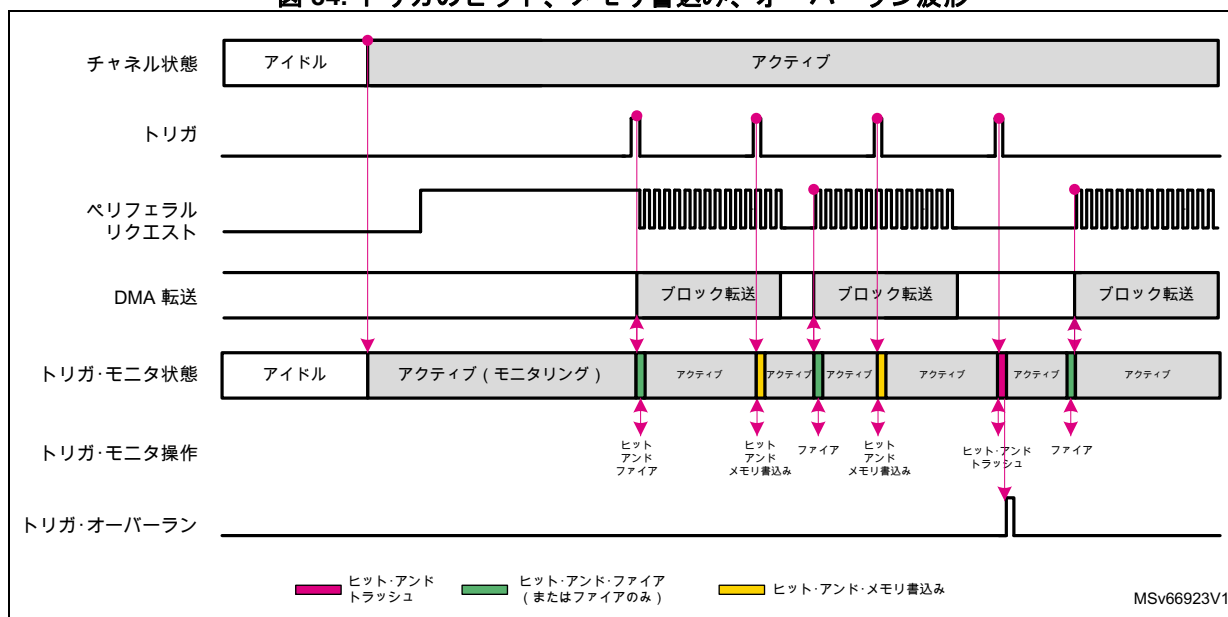
このトリガのモニタリングは、トリガされた完了していない（データまたはリンクの）転送を行っている間、アクティブなままとなります。新しいトリガを検出すると、このヒットは内部でメモリに書き込まれ、定義された立上がり／立下りエッジと TRIGSEL[5:0] が変更されずチャンネルが有効になっている限り、次の転送を許可します。

次の $LLIn_{+1}$ を転送すると、GPDMA_CxTR2 が TRIGSEL[5:0] または TRIGPOL[1:0] のいずれかの新しい値で更新され、モニタリングがリセットされ、以前に定義された $LLIn$ トリガの記憶されたヒットの可能性が破棄されます。

注意： 最初の新しいトリガー hit_{n+1} が記憶された後、別のトリガー hit_{n+2} が検出され、 hit_n トリガーによる転送がまだ完了していない場合、 hit_{n+2} は失われ、記憶されません。トリガー オーバーラン フラグが報告され (GPDMA_CxSR.TOF=1)、有効な場合は割り込みが生成されます (ifGPDMA_CxCR.TOIE=1) トリガのオーバーランでハードウェアにより、チャンネルが自動的に無効になることはありません。

以下の図は、ブロックレベル トリガ モードと立上がりエッジ トリガ極性を使用した構成例におけるトリガ ヒット、記憶、オーバーランを示しています。

図 84. トリガのヒット、メモリ書込み、オーバーラン波形



注： ユーザは、異なるチャンネルに同じ入力トリガ・イベントを割り当てることができます。これは、ブロードキャスト・トリガ・イベントに異なるチャンネルをトリガするために使用可能です。

16.4.13 リンクリスト・プログラミングでの GPDMA サーキュラ・バッファリング

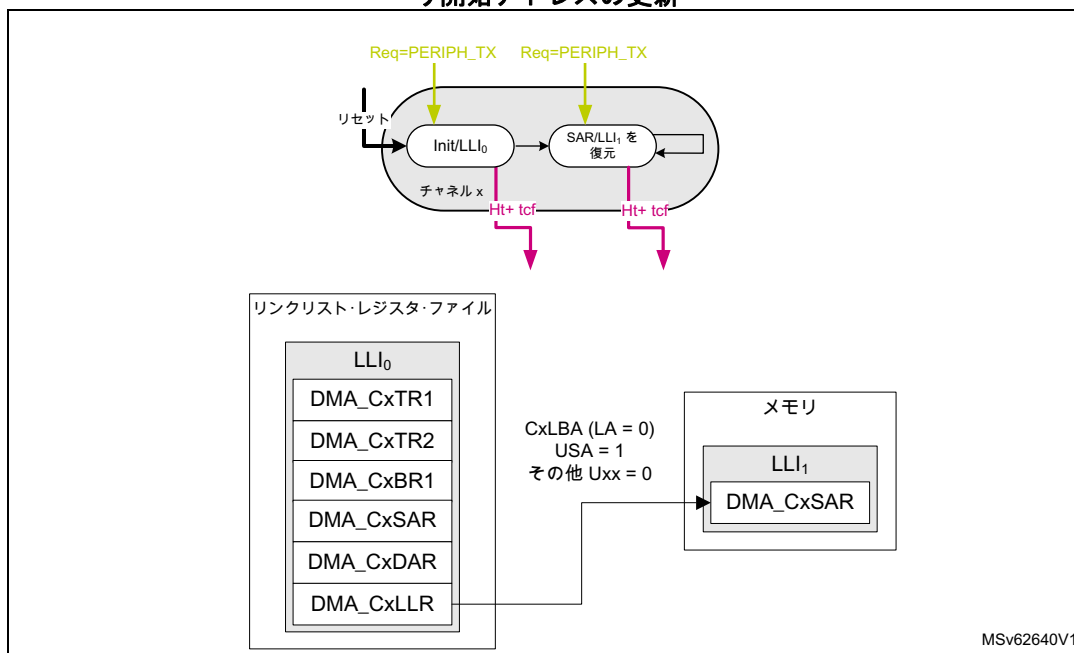
リニア・アドレス指定チャンネルによる、メモリとペリフェラル間の相互転送の GPDMA サーキュラ・バッファリング

サーキュラ・バッファで、連続したメモリからペリフェラル（またはペリフェラルからメモリ）への転送を行うには、ソフトウェアが 1/2 転送と完全な転送イベント／割込み生成 (GPDMA_CxCR.HTIE = 1 および GPDMA_CxCR.TCIE = 1) でチャンネルを設定して、同時バッファ・ソフトウェア・プロセッシングを可能にする必要があります。

LLI₀ を、リニア・アドレス指定チャンネルで最初のブロック転送に設定します。メモリからペリフェラルへの転送（ペリフェラルからメモリへの転送）で、メモリの転送元（転送先）の開始アドレスを復元するには、連続実行の LLI₁ が必要です。GPDMA は、ブロック転送が完了し、GPDMA_CxBR1 の復元が必要無くなると、最初にプログラムされた GPDMA_CxBR1.BNDT[15:0] を自動的にリロードします。

リニア・アドレス指定の GPDMA チャンネルと転送元のサーキュラ・バッファによる、このプログラミングを下の図に示します。

図 85. GPDMA サーキュラ・バッファ・プログラミング：リニア・アドレス指定によるメモリ開始アドレスの更新

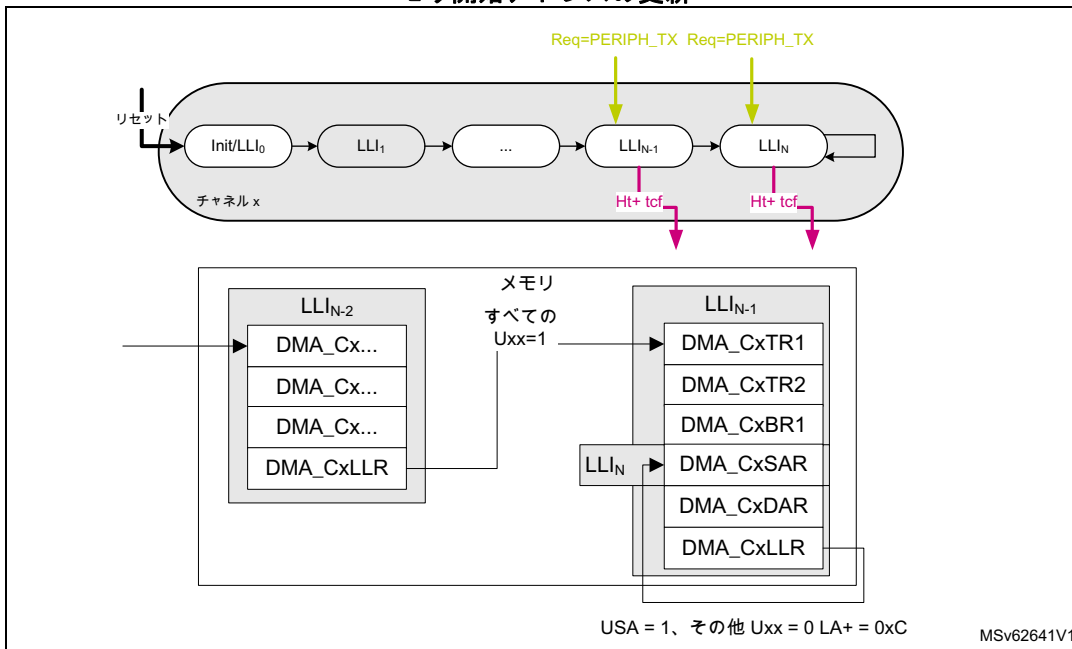


注： 2D アドレス指定チャンネルでは、ユーザが GPDMA_CxBR1.BRC[10:0] = 1 でシングル LLI を使用し、GDMA_CxBR2 と GDMA_CxBR1 で負のメモリ・ブロック・アドレス・オフセットをプログラムして、メモリ送信元か送信先の開始アドレスへジャンプすることができます。

共有 GPDMA のチャンネル x を経由した、他の転送後にサーキュラ・バッファリングを実行しなくてはならない場合、最初のブロック転送の設定にはメモリにある直前の LLI_{N-1} が必要です。最後の LLI_N で、メモリからペリフェラルへの転送（ペリフェラルからメモリへの転送）で、メモリの転送元（転送先）の開始アドレスを復元します。

リニア・アドレス指定の 共有 GPDMA チャンネル、と転送元のサーキュラ・バッファによる、このプログラミングを下の図に示します。

図 86. サーキュラ・バッファのある共有 GPDMA チャンネル：リニア・アドレス指定によるメモリ開始アドレスの更新



16.4.14 ペリフェラルのフロー制御モードでの GPDMA 転送

ペリフェラル・フロー制御モード機能を持ったペリフェラルでは、割り当てられたチャンネルがこの機能を実装している場合、GPDMA ブロック転送の早期終了を決定することができます（[セクション 16.3.6](#) 参照）。

関連した GPDMA チャンネル x もペリフェラル・フロー制御モード（GPDMA_CxTR2.PFREQ = 1）がプログラムされている場合は、次のとおりです。

- GPDMA ブロック転送を次のように開始します。
 - GPDMA_CxBR1.BNDT[15:0] ≠ 0 の場合、プログラムされた値は GPDMA ハードウェアにより内部的に考慮されます。
 - GPDMA_CxBR1.BNDT[15:0] = 0 の場合、GPDMA ハードウェアは、転送するブロックの転送元でのサイズの最大値を、内部的に 64 KB と見なします。
- GPDMA ブロック転送は次の条件のどれかが発生すると、その時点で直ぐに完了します。
 - GPDMA_CxBR1.BNDT[15:0] = 0 になった時。
 - ペリフェラルがブロックを早期に終了した時。プログラムされている場合、GPDMA_CxTR2 に従って完了転送イベントが生成されます（[GPDMA チャンネル x 転送レジスタ 2 \(GPDMA_CxTR2\)](#) 参照）。その後、ソフトウェアは転送された現在のバイト数を転送元（GPDMA_CxBR1.BNDT[15:0]）から読み出す、そして／または、メモリにあるバッファの転送元あるいは転送先の現在のアドレス（GPDMA_CxSAR[31:0] または GPDMA_CxDAR[31:0]）を読み出すことが可能です。

ペリフェラル・フローコントロール・モード。

- ハードウェアによりリクエストされた転送の転送先ペリフェラルはサポートされていません。メモリからペリフェラルへの転送はサポートされていません。
- 転送元ペリフェラルからのデータ・パッキングはサポートされていません。
- 2D/繰り返しブロックはサポートされていません。
- GPDMA_CxBR1.BNDT[15:0] は転送元（ペリフェラル）バースト・サイズの倍数でプログラムされなくてはなりません。

16.4.15 GPDMA セキュア／非セキュア・チャネル

GPDMA コントローラは、チャネル・レベルで TrustZone ハードウェア・アーキテクチャに準拠しています。このアーキテクチャでは、コントローラの全リソースが、常にセキュア・ワールドおよび非セキュア・ワールドのどちらか一方に存在できるようにパーティション分割されています。

あらゆるチャネル x がセキュアまたは非セキュアのハードウェア・リソースで、GPDMA_SECCFGR.SEC x により設定されます。

チャネル x がセキュア特権エージェントによりセキュア状態に設定されている場合、以下のアクセス制御規則が適用されます。

- このチャネルのレジスタ・フィールドへの非セキュア読み出しアクセスは、非セキュア・エージェントで読み出すことができる GPDMA_SECCFGR、GPDMA_PRIVCFGR、GPDMA_RCFGLOCKR を除いて、強制的に 0 を返します。
- このチャネルのレジスタ・フィールドへの非セキュア書き込みアクセスによる影響はありません。

チャネル x がセキュア状態に設定されている場合、セキュア・エージェントは、転送元からの GPDMA データ転送 (GPDMA_CxTR1.SSEC) と、転送先への GPDMA データ転送 (GPDMA_CxTR1.DSEC) を個別にセキュアまたは非セキュアに設定することができます。

チャネル x がセキュア状態のリンクリスト・モードに設定されている場合、GPDMA メモリから、そのレジスタ・ファイルへの次のリンクリスト・データ構造のロードは、マスタ・ポートに配置された GPDMA_CxCR.LAP を経由したセキュア転送により自動的に実行されます。

GPDMA は GPDMA_SECCFGR を反映したセキュア・パスを生成して、他のペリフェラルに各 GPDMA チャネル x のセキュア／非セキュア状態を常に通知します。

また、GPDMA は、セキュア GPDMA レジスタに対する不正な非セキュア・アクセスに関し、セキュリティの不正アクセス・パルス信号を生成します。この信号は TrustZone 割込みコントローラに送られます。

セキュア・ソフトウェアがチャネルをセキュア状態から非セキュア状態に切り替える必要がある場合、セキュア・ソフトウェアはそのチャネルをアポートするか、セキュア・チャネルが完了するのを待ってから切り替えなくてはなりません。これは、チャネルを次の非セキュア転送に動的に再割り当てを行うために必要です。非セキュア・ソフトウェアは、これを行うことが許可されておらず、次の転送に向け GPDMA_CxCR を再プログラムできるようになるには GPDMA_CxCR.EN = 0 でなくてはなりません。セキュア・ソフトウェアは、チャネル x (GPDMA_CxCR.RESET = 1) だけではなく、 x レジスタ・ファイルの全チャネルもリセット値にリセットする場合があります。

16.4.16 GPDMA 特権／非特権チャンネル

あらゆるチャンネル x は、GPDMA_PRIVCFGR.PRIV x による特権エージェントの設定に従って、特権または非特権ハードウェア・リソースとなります。

チャンネル x が特権エージェントにより特権状態に設定されている場合、次のアクセス制御規則が適用されます。

- このチャンネルのレジスタ・フィールドへの非特権読み出しアクセスは、非特権・エージェントで読み出すことができる GPDMA_PRIVCFGR、GPDMA_SECCFGR、GPDMA_RCFGLOCKR を除いて、強制的に 0 を返します。
- このチャンネルのレジスタフィールドへの非特権書き込みアクセスによる影響はありません。

チャンネルが特権（非特権）状態に設定されると、ソースおよびデスティネーション・データ転送は、AHB マスタ・ポート経由の特権（非特権）転送です。

チャンネルが特権（非特権）状態のリンクリスト・モードに設定されている場合、GPDMA メモリからのそのレジスタ・ファイルへの次のリンクリスト・データ構造のロードは、マスタ・ポートに割り当てられた GPDMA_CxCR.LAP を経由した特権（非特権）転送で自動的に実行されます。

GPDMA は GPDMA_PRIVCFGR を反映した特権バスを生成して、他のペリフェラルに各 GPDMA チャンネル x の特権／非特権状態を常に通知します。

特権ソフトウェアがチャンネルを特権状態から非特権状態に切り替える必要がある場合、特権ソフトウェアはそのチャンネルをアポートするか、特権チャンネルが完了するのを待ってから切り替えなくてはなりません。これは、チャンネルを次の非特権転送に動的に再割り当てを行うために必要です。非特権ソフトウェアは、これを行うことが許可されておらず、次の転送に向け、GPDMA_CxCR を再プログラムできるようになるには GPDMA_CxCR.EN = 0 でなくてはなりません。特権ソフトウェアは、チャンネル x (GPDMA_CxCR.RESET = 1) だけではなく、 x レジスタ・ファイルの全チャンネルもリセット値にリセットする場合があります。

16.4.17 GPDMA エラー管理

GPDMA は次に示すように、根本原因に応じて転送エラーを管理しユーザに報告することができます。

データ転送エラー

転送元または転送先への（AHB のシングルまたはバースト転送としての）バス・アクセスに関して。

- 転送元または転送先のターゲットは AHB エラーを報告します。
- プログラムしたチャンネル転送が停止します（GPDMA_CxCR.EN が GPDMA ハードウェアでクリア）。チャンネル状態レジスタはアイドル状態（GPDMA_CxSR.IDLEF = 1）とデータ・エラー（GPDMA_CxSR.DTEF = 1）を報告します。
- GPDMA データ転送エラーの後、ユーザは、保護属性を含む、製品が定義した転送元と転送先のメモリ・マッピングを考慮して、デバッグ・セッションを実行しなくてはなりません。
- GPDMA データ転送エラーの後、ユーザはチャンネルのリセットを発行して（GPDMA_CxCR.RESET をセット）、ハードウェアの GPDMA チャンネル・データ経路と FIFO の内容をリセットしてから、そのチャンネルを次の転送に対して有効にしなくてはなりません。

リンク転送エラー

メモリにあるプログラムされた LLI からの GPDMA チャンネル・レジスタの暫定的な更新に関して。

- リンクリスト・メモリは AHB エラーを報告します。
- プログラムされたチャンネル転送は停止し (GPDMA_CxCR.EN が GPDMA ハードウェアでクリアされる)、チャンネル状態レジスタがアイドル状態 (GPDMA_CxSR.IDLEF = 1) と、リンク・エラー (GPDMA_CxSR.ULEF = 1) を報告します。
- GPDMA のリンクエラーの後、ユーザは、保護属性を含む、製品が定義したリンクリスト・データ構造 (GPDMA_CxLBAR および GPDMA_CxLLR) のメモリ・マッピングを考慮して、デバッグ・セッションを実行しなくてはなりません。
- GPDMA リンク・エラーの後、ユーザはリンクリスト・レジスタ・ファイル (GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR、に加え、GPDMA_CxTR3 および GPDMA_CxBR2) を明示的に書き出してから、そのチャンネルを次の転送のために有効にしなくてはなりません。

ユーザ設定エラー

非認証ユーザ設定での GPDMA 転送の暫定実行に関して。

- プログラムされたチャンネル転送を無効にし (GPDMA ハードウェアにより GPDMA_CxCR.EN を強制しクリアし)、次の非認証プログラム・データ転送が実行されることを回避します。チャンネル状態レジスタはアイドル状態 (GPDMA_CxSR.IDLEF = 1) とユーザ設定エラー (GPDMA_CxSR.USEF = 1) を報告します。
- GPDMA ユーザ設定エラーの後、ユーザは GPDMA チャンネル・プログラミングを考慮に入れて、デバッグ・セッションを実行しなくてはなりません。ユーザ設定エラーは次のどれかが原因となっている可能性があります。
 - 次の LLI_i によるプログラムされたこの値の更新が無い、プログラムされた null の転送元ブロック・サイズ (GPDMA_CxBR1.BNDT[15:0] = 0 および GPDMA_CxLLR.UB1 = 0)
 - プログラムされた null ではない転送元ブロック・サイズが、ソース・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxBR1.BNDT[2:0] に対して GPDMA_CxTR1.SDW_LOG2[1:0])
 - パッキング／アンパッキング・モード (PAM[1] = 1) の場合、プログラムされた null ではない転送元ブロック・サイズが、デスティネーション・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxBR1.BNDT[2:0] に対して GPDMA_CxTR1.DDW_LOG2[1:0])
 - プログラムされたアンアラインド転送元開始アドレスが、ソース・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxSAR[2:0] に対して GPDMA_CxTR1.SDW_LOG2[1:0])
 - チャンネル x (x = 6 から 7) : プログラムされたアンアラインド転送元アドレス・オフセットがソース・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxTR3.SAO[2:0] に対して GPDMA_CxTR1.SDW_LOG2[1:0])
 - チャンネル x (x = 6 から 7) : プログラムされたブロックのアンアラインド繰り返し転送元アドレス・オフセットが、ソース・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxBR2.BRSAO[2:0] に対して GPDMA_CxTR1.SDW_LOG2[1:0])
 - プログラムされたアンアラインド転送先開始アドレスが、デスティネーション・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxDAR[2:0] に対して GPDMA_CxTR1.DDW_LOG2[1:0])
 - チャンネル x (x = 6 から 7) : プログラムされたアンアラインド転送先アドレス・オフセットがデスティネーション・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxTR3.DAO[2:0] に対して GPDMA_CxTR1.DDW_LOG2[1:0])

- チャンネル x ($x = 6$ から 7) : プログラムされたブロックのアンアラインド繰り返し転送先アドレス・オフセットが、デスティネーション・バースト転送のプログラムされたデータ幅の倍数ではない (GPDMA_CxBR2.BRDAO[2:0] に対して GPDMA_CxTR1.DDW_LOG2[1:0])
- プログラムされたダブルワードの転送元データ幅 (GPDMA_CxTR1.SDW_LOG2[1:0] = 11)
- プログラムされたダブルワードの転送先データ幅 (GPDMA_CxTR1.DDW_LOG2[1:0] = 11)
- null データ転送による、プログラムされたリンクリスト項目 LLI_{n+1} (GPDMA_CxLLR.UB1 = 1 および GPDMA_CxBR1.BNDT = 0)

16.4.18 GPDMA 自律モード

プログラムされたリンクリスト転送を GPDMA が実行する際の動的電力消費を削減するため、RCC が対応する GPDMA イネーブル制御ビットでプログラムされている場合、GPDMA ハードウェアはそのクロック・ゲーティングを自動で管理し、デバイスが Run または低消費電力モードのときには必ず RCC にクロック・リクエスト出力信号を生成します。

RCC プログラミングの詳細についてはリファレンス・マニュアルの RCC セクションを参照してください。

GPDMA 自律機能に対する低消費電力モードの可用性の詳細に関しては [セクション 16.3.3](#) を参照してください。

次に示すように、ユーザは GPDMA_CxTR2 により、特定の GPDMA 転送の実行を GPDMA チャンネル x の LLI_n レベルでプログラムしてスケジューリングすることができます。

- ソフトウェアが、入力トリガとリクエストの為の TRIGM[1:0]、TRIGPOL[1:0]、TRIGSEL[5:0]、SWREQ、REQSEL[7:0] で、転送の入力を制御し条件付けします。
- ソフトウェアが、TCM[1:0] で転送出力の制御および信号出力を行い、転送完了イベントを生成するかどうか、および関連する 1/2 データ転送イベントを生成するかどうかを決めています。

詳細は、[GPDMA チャンネル \$x\$ 転送レジスタ 2 \(GPDMA_CxTR2\)](#) を参照してください。

低消費電力モードで使用する場合、この機能は、有効になった GPDMA 転送完了割り込み (GPDMA_CxCR.TCIE = 1) により、特定の転送の完了で CPU ウェイクアップを有効にし、および/または、同じチャンネルでの別の LLI_{n+1} 転送を操作するために自律 GPDMA の継続を有効にします。

出力チャンネル x 転送完了イベント `gpdma_chx_tc` は、このイベントがループバックし、GPDMA レベルで接続されている場合 ([セクション 16.3.7](#) 参照)、チャンネルの選択した入力トリガとしてプログラムすることが可能で、クリアされた転送完了フラグ (TCF) なしで、自律的に細かい GPDMA チャンネル間転送スケジューリングを可能にします。

任意の GPDMA チャンネル x は、次のいずれかの条件でそのクロック・リクエストをアサートします。

- 実行する予定の次の転送がトリガ (GPDMA_CxTR2.TRIGPOL[1:0] および GPDMA_CxTR2.TRIGM[1:0]) により条件付けされるようにプログラムされている場合、そのトリガのヒットが発生している時のみ。
- 実行する予定の次の転送がトリガにより条件付けされない場合は次のとおりです。
 - GPDMA_CxTR2.SWREQ = 0 の場合、選択されたペリフェラルによりハードウェア・リクエストがアサートされた時のみ。
 - GPDMA_CxTR2.SWREQ = 1 (メモリ間、GPIO とメモリ相互の間) の場合、GPDMA が有効になると直ぐ。

次の条件をすべて満足すると直ぐに GPDMA チャネル x がそのクロック・リクエストをリリースします。

- 実行予定の転送が完了している。
- GPDMA チャネル x はすぐには準備が整わず、次の転送の実行がリクエストされていない。
- チャネル x 割込みが立っている場合、この割込みを発生させる可能性のある状態レジスタのすべてのフラグがソフトウェア・エージェントによりクリアされます。

1 つのチャネルがそのクロック・リクエストをアサートすると、GPDMA はそのクロック・リクエストを RCC にアサートします。どのチャネルもそのクロック・リクエストをアサートしないと、GPDMA はそのクロック・リクエストを RCC にリリースします。

16.5 デバッグ・モードでの GPDMA

マイクロコントローラがデバッグ・モードに入ると（コアが停止）、あらゆるチャネル x を、DBGMCU モジュールのプログラマブル制御ビットに応じて、個別に継続（デフォルト）またはサスペンドにすることが可能です。

注：デバッグ・モードでは、GPDMA_CxSR.SUSPF は DBGMCU モジュールのプログラマブル制御ビットによるサスペンドの影響は受けません。この場合、GPDMA_CxSR.IDLEF をチェックしてチャネル・サスペンドの完了状態を知ることができます。

16.6 低消費電力モードの GPDMA

表 132. 低消費電力モードが GPDMA に与える影響

モード	説明
SLEEP	影響はありません。GPDMA 割込みによって、デバイスは SLEEP モードを終了します。
STOP ⁽¹⁾	STOP モードに入っても、GPDMA レジスタの内容は保持されます。GPDMA レジスタの内容はメモリから次のリンクリスト項目により自律的に更新され、自律的なデータ転送を実行することが可能です。デバイスは、GPDMA 割込みによって STOP モード ⁽¹⁾ から復帰することが可能です。
STANDBY	GPDMA はパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. STOP モードが何かサポートされるかについては、[セクション 16.3.3](#) を参照してください。



16.7 GPDMA 割込み

各チャネルと各 CPU には個別に 1 つの GPDMA 割込みラインがあります（デバイスにいくつかある場合）。

表 133. GPDMA 割込みリクエスト

項目 (割込みの略称)	割込みイベント	割込みイネーブル	イベントフラグ	イベントのクリア方法
GPDMA_CHx	転送完了	GPDMA_CxCR.TCIE	GPDMA_CxSR.TCF	GPDMA_CxFCR.TCF に 1 を書き込む
	1/2 転送	GPDMA_CxCR.HTIE	GPDMA_CxSR.HTF	GPDMA_CxFCR.HTF に 1 を書き込む
	データ転送エラー	GPDMA_CxCR.DTEIE	GPDMA_CxSR.DTEF	GPDMA_CxFCR.DTEF に 1 を書き込む
	更新リンク・エラー	GPDMA_CxCR.ULEIE	GPDMA_CxSR.ULEF	GPDMA_CxFCR.ULEF に 1 を書き込む
	ユーザ設定エラー	GPDMA_CxCR.USEIE	GPDMA_CxSR.USEF	GPDMA_CxFCR.USEF に 1 を書き込む
	サスペンド状態	GPDMA_CxCR.SUSPIE	GPDMA_CxSR.SUSPF	GPDMA_CxFCR.SUSPF に 1 を書き込む
	トリガ・オーバーラン	GPDMA_CxCR.TOFIE	GPDMA_CxSR.TOF	GPDMA_CxFCR.TOF に 1 を書き込む

GPDMA チャネル x イベントは次のいずれかの場合があります。

- 転送完了
- 1/2 転送完了
- 転送エラー、その原因は次のいずれか。
 - データ転送エラー
 - 更新リンク・エラー
 - ユーザ設定エラー、完了サスペンション
- トリガ・オーバーラン

注： チャネル x 転送完了イベントが発生すると、出力信号 `gpdma_chx_tc` が 1 クロック・サイクルのハイ・パルスとして生成されます。

次の両方の条件を満足すると、あらゆる xx イベントに続いて割込みが生成されます。

- 対応する割込みイベント xx が有効である ($GPDMA_CxCR.xxIE = 1$)。
- 対応するイベント・フラグがクリアされている ($GPDMA_CxSR.xx F = 0$)。これは、前の同じ xx イベント発生後に、ソフトウェア・エージェントが対応する xx フラグのクリア制御ビットに 1 を書き込んだはずだということを意味します ($GPDMA_CxFCR.xx F$ に 1 を書き込む)。

TCF (転送完了) と HTF (1/2 転送) イベントの生成は次のように $GPDMA_CxTR2.TCEM[1:0]$ によって制御されます。

- 転送完了イベントは、転送完了イベント・モード ($GPDMA_CxTR2.TCEM[1:0]$) に応じて、ブロック転送完了、2D／繰り返しブロック転送完了、または、次の LLI (がある場合) のアップロードを含む LLI 転送完了、あるいはフル・リンクリスト完了のいずれかです。
- 1/2 転送イベントは、転送完了イベントモード $GPDMA_CxTR2.TCEM[1:0]$ により、1/2 ブロック転送または、2D／繰り返しブロックの 1/2 転送です。

転送元ブロックのバイト・サイズの半分 ($GPDMA_CxBR1.BNDT[15:0] / 2$ を整数値に切り上げ) が転送先に転送されたときに 1/2 ブロック転送となります。

繰り返しブロックの半分 ($(GPDMA_CxBR1.BRC[10:0] + 1) / 2$ を整数値に切り上げ) が転送先に転送されたときに、2D／繰り返し 1/2 ブロック転送となります。

詳細は、[GPDMA チャネル x 転送レジスタ 2 \(GPDMA_CxTR2\)](#) を参照してください。

次のいずれかの状況で転送エラーが立ちます。

- ・ 転送元から、または転送先へのシングル／バースト・データ転送時（DTEF）
- ・ メモリにあるプログラムされた LLI による GPDMA チャネル・レジスタの更新時（ULEF）
- ・ 非認証設定での GPDMA チャネルの暫定実行時（USEF）

ユーザは GPDMA チャネル・プログラミングを USEF の根本原因リストに対して修正するデバッグ・セッションを行わなくてはなりません（[セクション 16.4.17](#) 参照）。

トリガ・オーバーランは [トリガ・ヒットのメモリ書込みとトリガ・オーバーラン・フラグの生成](#)に記載します。

16.8 GPDMA レジスタ

GPDMA レジスタは 32 bit ワード・データ・でのアラインド・アクセスでアクセスしなくてはなりません。

16.8.1 GPDMA セキュア設定レジスタ（GPDMA_SECCFGR）

アドレスオフセット：0x00

リセット値：0x0000 0000

このレジスタへの書込みアクセスはセキュアおよび特権でなくてはなりません。読出しアクセスはセキュアまたは非セキュア、特権または非特権です。

対応するチャネル x がロックされている場合（GPDMA_RCFGLOCKR.LOCK x = 1）、書込みアクセスはビット・レベルで無視されます。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

このレジスタは、GPDMA チャネルの初期化／クローズ時（GPDMA_CxCR.EN = 0 のとき）にビット・レベルでプログラムし、あらゆるチャネル x を個別にセキュアまたは非セキュア・ワールドに安全に割り当てる必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **SEC x** ：チャネル x （ $x = 7$ から 0）のセキュア状態

0：非セキュア

1：セキュア

16.8.2 GPDMA 特権設定レジスタ (GPDMA_PRIVCFGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

このレジスタへの書き込みアクセスは特権でなくてはなりません。読出しアクセスは特権または非特権、セキュアまたは非セキュアで行うことができます。

このレジスタにはセキュアな情報と非セキュアな情報を混在させることができます。チャンネル x がセキュアに設定された場合 (GPDMA_SECCFGR.SEC x = 1)、PRIV x ビットはセキュア・(および特権) エージェントによってのみ書き込むことが可能です。

対応するチャンネル x がロックされている場合 (GPDMA_RCFGLOCKR.LOCK x = 1)、書き込みアクセスはビット・レベルで無視されます。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

このレジスタは、GPDMA チャンネルの初期化／閉鎖時 (GPDMA_CxCR.EN = 0) にビット・レベルでプログラムし、あらゆるチャンネル x を特権または非特権ワールドに割り当てる必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PRIV x** : チャンネル x ($x = 7$ から 0) の特権状態

0 : 非特権

1 : 特権

16.8.3 GPDMA 設定ロック・レジスタ (GPDMA_RCFGLOCKR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

このレジスタは、例えばブート時などに、あらゆる GPDMA チャンネル／リソースのセキュア特権属性を個別にロックするために (例えばブート時に、あらゆるチャンネル x の GPDMA_CxSECCFGR および GPDMA_CxPRIVCFGR の設定をロックするために)、セキュア特権属性のソフトウェア・エージェントにより書き込むことが可能です。

読出しアクセスは特権または非特権、セキュアまたは非セキュアで行うことができます。

注 : TZEN = 0 の場合、このレジスタは書き込みできません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
								rS	rS	rS	rS	rS	rS	rS	rS

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **LOCKx** : グローバル GPDMA がリセットされるまで、GPDMA_SECCFGR.SECx および GPDMA_PRIVCFGR.PRIVx の設定をロックします (x = 7 から 0)。

このビットはリセット後にクリアされ、一旦セットされると、グローバル GPDMA がリセットされるまでリセットすることはできません。

0 : チャンネル x のセキュア特権設定が書き込み可能です。

1 : チャンネル x のセキュア特権設定が書き込み不可です。

16.8.4 GPDMA 非セキュア・マスク済み割込みステータス・レジスタ (GPDMA_MISR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

レジスタは読出しレジスタです。

これは非セキュア・レジスタで、各非セキュア・チャンネル x のマスク済み割込みステータス・ビット MISx を含みます (GPDMA_SECCFGR.SECx = 0 のチャンネル x 設定)。これは GPDMA_CxSR のすべてのフラグの論理和で、各転送元フラグは対応する GPDMA_CxCR の割込みイネーブル・ビットにより有効になっています。

すべてのビットは、GPDMA_CxFCR の対応するフラグ・クリアビットに 1 を書き込むと、ハードウェアによりデアサートされます。

チャンネル x がセキュア状態の場合 (GPDMA_SECCFGR.SECx = 1)、このチャンネル x のマスク済み割込みステータス・ビット MISx への読出しアクセスは 0 を返します。

このレジスタには、各チャンネルの特権状態 GPDMA_PRIVCFGR.PRIVx によっては、特権情報と非特権情報が混在する可能性があります。特権ソフトウェアは非セキュア割込みステータスをすべて読み出すことができます。非特権ソフトウェアに許可されるのは非特権 (および非セキュア) チャンネルのステータス読出しのみで、その他の特権ビット・フィールドはゼロが返されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIS7	MIS6	MIS5	MIS4	MIS3	MIS2	MIS1	MIS0
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **MISx** : チャンネル x のマスク済み割込みステータス (x = 7 から 0)

0 : チャンネル x で割込みは発生していません

1 : チャンネル x で割込みが発生しました

16.8.5 GPDMA セキュア・マスク済み割込みステータス・レジスタ (GPDMA_SMISR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

これはセキュア読出しレジスタで、各セキュア・チャネル x のマスク済み割込みステータス・ビット MIS x を含みます (GPDMA_SECCFGR.SEC x = 1)。これは GPDMA_CxSR フラグすべての論理和で、各転送元フラグは、対応する GPDMA_CxCR 割込みイネーブル・ビットにより有効になっています。

各ビットは、対応する GPDMA_CxFCR フラグ・クリア・ビットに 1 をセキュアに書き込むと、ハードウェアによりデアサートされます。

このレジスタは非セキュア・チャネルに関する情報は何も含みません。

このレジスタには、各チャネルの特権状態 GPDMA_PRIVCFGR.PRIV x によっては、特権情報と非特権情報が混在する可能性があります。特権ソフトウェアはセキュア割込みステータスをすべて読み出すことができます。非特権ソフトウェアに許可されるのは非特権およびセキュア・チャネルのステータス読出しのみで、その他の特権ビット・フィールドはゼロが返されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIS7	MIS6	MIS5	MIS4	MIS3	MIS2	MIS1	MIS0
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 MIS x : セキュア・チャネル x のマスク済み割込みステータス ($x = 7$ から 0)

0 : セキュア・チャネル x で割込みは発生していません

1 : セキュア・チャネル x で割込みが発生しました

16.8.6 GPDMA チャネル x のリンクリスト・ベースのアドレス・レジスタ (GPDMA_CxLBAR)

アドレスオフセット : $0x50 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは特権ソフトウェアによって書き込まれる必要があります。これは、チャネル x GPDMA_PRIVCFGR.PRIVx の特権ステータスにより、特権読取りが可能かどうか異なります。

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) により、セキュアまたは非セキュアとなります。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャネルをベースとするレジスタは、あるチャネル x に対して、メモリ領域のリンクリスト・ベースのアドレスで、ここから GPDMA 転送のプログラムされたシーケンスを記述する LLI が、条件付きで自動的に更新されます。

64 KB に合わせたチャネル x リンクリストのベース・アドレスは、各 LLI のワード整列アドレス・オフセットを定義する、16 bit GPDMA_CxLLR レジスタによりオフセットされています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LBA[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:16 **LBA[31:16]** : GPDMA チャネル x のリンクリスト・ベース・アドレス

ビット 15:0 予約済みであり、リセット値に保持する必要があります。



16.8.7 GPDMA チャネル x フラグ・クリア・レジスタ (GPDMA_CxFCR)

アドレスオフセット : $0x5C + 0x80 \times x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0000$

これは書き込みレジスタで、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TOF	SUSPF	USEF	ULEF	DTEF	HTF	TCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	w	w	w	w	w	w	w								

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TOF** : トリガ・オーバーラン・フラグのクリア

0 : 影響なし。

1 : 対応する TOF フラグがクリアされました

ビット 13 **SUSPF** : 完了したサスペンド・フラグのクリア

0 : 影響なし。

1 : 対応する SUSPF フラグがクリアされました

ビット 12 **USEF** : ユーザ設定エラー・フラグのクリア

0 : 影響なし。

1 : 対応する USEF フラグがクリアされました

ビット 11 **ULEF** : 更新リンク転送エラー・フラグのクリア

0 : 影響なし。

1 : 対応する ULEF フラグがクリアされました

ビット 10 **DTEF** : データ転送エラー・フラグのクリア

0 : 影響なし。

1 : 対応する DTEF フラグがクリアされました

ビット 9 **HTF** : 1/2 転送フラグのクリア

0 : 影響なし。

1 : 対応する HTF フラグがクリアされました

ビット 8 **TCF** : 転送完了フラグのクリア

0 : 影響なし。

1 : 対応する TCF フラグがクリアされました

ビット 7:0 予約済みであり、リセット値に保持する必要があります。

16.8.8 GPDMA チャネル x ステータス・レジスタ (GPDMA_CxSR)

アドレスオフセット : $0x60 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0001$

これはチャネル・ステータスを報告する読出しレジスタです。

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFOL[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TOF	SUSPF	USEF	ULEF	DTEF	HTF	TCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLEF
	r	r	r	r	r	r	r								r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **FIFOL[7:0]** : モニタされた FIFO レベル

FIFO で利用可能な書込みビートの数。単位はプログラムされた転送先データ幅です (バイト、ハーフワード、ワードの単位については GPDMA_CxTR1.DDW_LOG2[1:0] 参照)。

注: 有効な転送をサスペンドした後、転送先にどれぐらいのデータが転送されたかを知るには、GPDMA_CxBR1.BDNT[15:0] と GPDMA_CxBR1.BRC[10:0] に加え、FIFOL[7:0] を読み出すことが必要な場合があります。読み出す前に転送のサスペンド (GPDMA_CxSR.SUSPF = 1) を待たなくてはならない場合があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TOF** : トリガ・オーバーラン・フラグ

- 0 : トリガ・オーバーラン・イベントはありません
- 1 : トリガ・オーバーラン・イベントが発生しました

ビット 13 **SUSPF** : 完了したサスペンド・フラグ

- 0 : 完了したサスペンド・イベントはありません
- 1 : サスペンド・イベントの完了が発生しました

ビット 12 **USEF** : ユーザ設定エラー・フラグ

- 0 : ユーザ設定エラー・イベントはありません
- 1 : ユーザ設定エラー・イベントが発生しました

ビット 11 **ULEF** : 更新リンク転送エラー・フラグ

- 0 : 更新リンク転送エラー・イベントはありません
- 1 : メモリからリンクリスト・レジスタを更新しているときに、マスタ・バス・エラー・イベントが発生しました

ビット 10 **DTEF** : データ転送エラー・フラグ

- 0 : データ転送エラー・イベントはありません
- 1 : データ転送でマスタ・バス・エラー・イベントが発生しました

ビット 9 HTF : 1/2 転送フラグ

0 : 1/2 転送イベントは発生していません

1 : 1/2 転送イベントが発生しました。

1/2 転送イベントは、転送完了イベントモード (GPDMA_CxTR2.TCEM[1:0]) に応じて、1/2 ブロック転送または、2D／繰り返しブロックの 1/2 転送です。

転送元ブロック・サイズの半分のバイト数 (GPDMA_CxBR1.BNDT[15:0]/2 を整数値に切り上げ) が転送先に転送されたときに 1/2 ブロック転送となります。

繰り返しブロックの半分 ((GPDMA_CxBR1.BRC[10:0] + 1) / 2) を整数値に切り上げ) が転送先に転送されたときに、2D／繰り返し 1/2 ブロック転送となります。

ビット 8 TCF : 転送完了フラグ

0 : 転送完了イベントは発生していません

1 : 転送完了イベントが発生しました

転送完了イベントは、転送完了イベント・モード (GPDMA_CxTR2.TCEM[1:0]) により、ブロック転送完了、2D／繰り返しブロック転送完了、または、次の LLI (がある場合) のアップロードを含む LLI 転送完了、あるいはフル・リンクリスト完了のいずれかです。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 IDLEF : アイドル・フラグ

0 : チャンネルはアイドル状態ではありません

1 : チャンネルはアイドル状態です

このアイドル・フラグは、チャンネルが有効なチャンネル設定 (USEF がすぐには報告されない) でイネーブルになっていると (GPDMA_CxCR.EN = 1)、ハードウェアによってデアサートされます。

このアイドル・フラグは、チャンネルが (サスペンドまたはディスエーブル状態で) アイドル状態に戻った時に、ハードウェアによりアサートされるか、ハード・リセット後にアサートされます。

16.8.9 GPDMA チャンネル x 制御レジスタ (GPDMA_CxCR)

アドレスオフセット : $0x64 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : 0x0000 0000

このレジスタは、チャンネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャンネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタはチャンネルの制御 (アクティベート、サスペンド、アボート、ディスエーブル) に使われます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIO[1:0]		Res.	Res.	Res.	Res.	LAP	LSM
								rw	rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TOIE	SUSPI E	USEIE	ULEIE	DTEIE	HTIE	TCIE	Res.	Res.	Res.	Res.	Res.	SUSP	リセッ ト	EN
	rw	rw	rw	rw	rw	rw	rw						rw	w	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **PRIQ[1:0]** : チャネル x GPDMA 転送の、他に対する優先レベル

- 00 : 優先度低、重み低
- 01 : 優先度低、重み中
- 10 : 優先度低、重み高
- 11 : 優先度高

注 : このビットは **EN = 0** のときには書き込まなくてはなりません。このビットは、**EN = 1** のときは読み出し専用です。

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **LAP** : リンクリスト割り当てポート

このビットは、メモリからの GPDMA のリンクリスト・レジスタの更新にマスタ・ポートを割り当てるために使用します。

- 0 : ポート 0 (AHB) 割り当て済み
- 1 : ポート 1 (AHB) 割り当て済み

注 : このビットは **EN = 0** のときには書き込まなくてはなりません。このビットは、**EN = 1** のときは読み出し専用です。

ビット 16 **LSM** : リンク・ステップ・モード

0 : フル・リンクリストで実行し、最後の LLI の終わりで完了したチャネル (**GPDMA_CxLLR** = 0)。このリンク・アドレスの下位 16 bit は null (**LA[15:0]** = 0) で、すべての更新ビットも null (**UT1** = **UB1** = **UT2** = **USA** = **UDA** = **ULL** = 0 および **UT3** = **UB2** = 0) です。この時、**GPDMA_CxBR1.BNDT[15:0]** = 0 および **GPDMA_CxBR1.BRC[10:0]** = 0 となります。

1 : 現在の LLI で 1 回実行されたチャネル

最初に、(おそらく、1Dノ繰り返し) ブロック転送が、現在の内部レジスタファイルの定義により、**GPDMA_CxBR1.BNDT[15:0]** = 0 および **GPDMA_CxBR1.BRC[10:0]** = 0 まで実行されます。その後、次のリンクリスト・データ構造が **GPDMA_CxLLR** の定義に従い、条件付きでメモリからアップロードされます。チャネルの実行が終了します。

注 : このビットは **EN = 0** のときには書き込まなくてはなりません。このビットは、**EN = 1** のときは読み出し専用です。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TOIE** : トリガ・オーバーラン割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 13 **SUSPIE** : 完了サスペンド割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 12 **USEIE** : ユーザ設定エラー割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 11 **ULEIE** : 更新リンク転送エラー割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 10 **DTEIE** : データ転送エラー割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 9 **HTIE** : 1/2 転送完了割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 8 **TCIE** : 転送完了割込みイネーブル

- 0 : 割込みは無効です。
- 1 : 割込みは有効です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **SUSP** : サスペンド

RESET (ビット 1) フィールドへ 1 を書き込むと、ビット 2 への書き込みに関係なく、ハードウェアはこのビットをデアサートします。その他の場合 :

ソフトウェアはアクティブなチャンネル (マスタ・ポートを経由した GPDMA 転送が進行中のチャンネル) をサスペンドするために 1 を書き込まなくてはなりません。

ソフトウェアはサスペンドしたチャンネルをレジュームするために、図 68 で詳細を説明しているプログラミング・シーケンスに従って 0 を書き込まなくてはなりません。

- 0 : 書き込み : チャンネルのレジューム、読出し : チャンネルのサスペンドなし
- 1 : 書き込み : チャンネルをサスペンド、読出し : サスペンドしたチャンネル

ビット 1 **RESET** : リセット

このビットは書き込み専用です。0 を書き込んでも影響はありません。1 の書き込みは次の項目のリセットを意味します : FIFO、チャンネルの内部状態、SUSP および EN ビット (それぞれ、ビット 2 およびビット 0 への書き込み内容には関係ありません)。

リセットは、チャンネルが定常状態のときに有効で、これは次のどれかを意味しています。

- サスペンドした状態のアクティブなチャンネル (GPDMA_CxSR.SUSPF = 1 および GPDMA_CxSR.IDLEF = GPDMA_CxCR.EN = 1)
- 無効状態のチャンネル (GPDMA_CxSR.IDLEF = 1 および GPDMA_CxCR.EN = 0)

RESET を書き込んだ後、このチャンネルを継続して使用するためには、ハードウェアにより変更される設定レジスタ (GPDMA_CxBR1, GPDMA_CxSAR および GPDMA_CxDAR) など、チャンネルを明示的に再設定してから、チャンネルを再度イネーブルにすることが必要です (図 69 のプログラミング・シーケンス参照)。

- 0 : チャンネルのリセットなし
- 1 : チャンネルのリセット

ビット 0 **EN** : イネーブル

RESET (ビット 1) フィールドへ 1 を書き込むと、ビット 0 への書き込みに関係なく、ハードウェアはこのビットをデアサートします。その他の場合 :

このビットは、転送エラー (マスタ・パス・エラーまたはユーザ設定エラー) またはチャンネル転送完了 (例えば、LLI のシングル実行の最後に LSM = 1 になっている場合など、チャンネルの設定準備が完了している) 時にハードウェアによりデアサートされます。

その他の場合、このビットは、ソフトウェアでアサートできます。

この EN ビットへ 0 を書き込んでも無視されます。

- 0 : 書き込み : 無視、読出し : チャンネル無効化
- 1 : 書き込み : チャンネルの有効化、読出し : 有効になったチャンネル

16.8.10 GPDMA チャネル x 転送レジスタ 1 (GPDMA_CxTR1)

アドレスオフセット : $0x90 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは、セキュア DSEC と SSEC を除き、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) に従って、セキュアまたは非セキュアとなり、GPDMA_PRIVCFGR.PRIVx のチャネル x の特権ステートに従って、特権または非特権となります。

このレジスタは、チャネル x の転送を制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャネルが完了したら、このレジスタに書き込まなくてはなりません。その後、ハードウェアは GPDMA_CxCR.EN をデアサートします。チャネル転送は、ブロック、2D/繰返しブロック、LLI、あるいはフル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードのリンク転送時に、GPDMA_CxLLR.UT1 = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DSEC	DAP	Res.	Res.	DHX	DBX	DBL_1 [5:0]						DINC	Res.	DDW_LOG2 [1:0]	
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSEC	SAP	SBX	PAM[1:0]		Res.	SBL_1 [5:0]						SINC	Res.	SDW_LOG2 [1:0]	
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw		rw	rw

ビット 31 **DSEC** : 転送先への GPDMA 転送のセキュリティ属性

GPDMA_SECCFGR.SECx = 1 で、アクセスがセキュアな場合。

0 : GPDMA は非セキュア転送を行います

1 : GPDMA はセキュア転送を行います

これはセキュア・レジスタ・ビットです。このビットはセキュア・ソフトウェアからのみ読み出すことができます。GPDMA_SECCFGR.SECx = 1 のとき、このビットはセキュア・ソフトウェアにより書き込まなくてはなりません。GPDMA_SECCFGR.SECx = 0 のとき、セキュア書込みは無視されます。

GPDMA_SECCFGR.SECx がデアサートされると、この DSEC ビットも (チャネルのセキュア再設定で非セキュアとして) ハードウェアによりデアサートされ、転送先への GPDMA 転送は非セキュアになります。

ビット 30 **DAP** : 転送先割り当てポート

このビットはデスティネーション転送のマスター・ポートの割り当てに使用されます。

0 : ポート 0 (AHB) 割り当て済み

1 : ポート 1 (AHB) 割り当て済み

注 : このビットは EN = 0 のときには書き込まなくてはなりません。このビットは、EN = 1 のときは読出し専用です。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **DHX** : 転送先ハーフワード交換

転送先データ・サイズがワードより短い場合、このビットは無視されます。

転送先データ・サイズがワードの場合。

0 : ワード内でのハーフワードベースの交換は無し

1 : 2 つの連続した (PAM 以降) ハーフワードが、各転送先ワードで交換されます。

ビット 26 **DBX** : 転送先バイト交換

転送先データ・サイズがバイト単位の場合、このビットは無視されます。
転送先データ・サイズがバイトではない場合。
0 : ハーフワード内でのバイトベースの交換なし
1 : 2 つの連続した (PAM 以降) バイトが、各転送先ハーフワードで交換されます。

ビット 25:20 **DBL_1[5:0]** : 転送先のバースト長から 1 を引いた 0 から 63 の値。

バースト長の単位は、ビートと呼ばれるバーストのデータ 1 つ分です。DBL_1[5:0] = 0 の場合、バーストはシングルと呼ばれることがあります。各データ/ビートの幅は転送先データ幅 DDW_LOG2[1:0] で定義されます。

注 : バースト転送が AHB 転送の 1 KB のアドレス境界を越えた場合、GPDMA は AHB プロトコルに準拠するよう、プログラムされたバーストをシングルまたは、より短いバーストへと変更、短縮を行います。

バースト転送の長さがチャネル x の FIFO サイズより大きい場合、GPDMA は FIFO サイズに準拠するよう、プログラムされたバーストをシングルまたは、より短いバーストへと変更、短縮を行います。GPDMA の再アービトレーションが有効長とより短いシングル/バースト長の間で行われるため、転送効率は低いですが、データの整合性は保証されます。

ビット 19 **DINC** : 転送先インクリメント・バースト

0 : 固定バースト
1 : 隣接インクリメント・バースト

GPDMA_CxDAR が指す転送先アドレスは、バースト・ビート/シングル転送後も保持されるか、隣接データに対応するオフセット分だけ転送後にインクリメントされます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **DDW_LOG2[1:0]** : バーストの転送先データ幅の二進対数 (バイト単位)

00 : バイト
01 : ハーフワード (2 バイト)
10 : ワード (4 バイト)
11 : ユーザ設定エラーが報告され、転送は行われませんでした

注 : 8 バイト・データ幅を設定すると、ユーザ設定エラーが報告され、転送は行われません。デスティネーション・バースト転送は、そのデータ幅に合ったアドレスでなくてはなりません (開始アドレス GPDMA_CxDAR[2:0] で、DDW_LOG2[1:0] に対するアドレス・オフセット GPDMA_CxTR3.DAO[2:0])。そうしないとユーザ設定エラーが報告され、転送は行われません。

ビット 15 **SSEC** : 転送元からの GPDMA 転送のセキュリティ属性

GPDMA_SECCFGR.SECx = 1 で、アクセスがセキュアな場合。

0 : GPDMA は非セキュア転送を行います

1 : GPDMA はセキュア転送を行います

これはセキュア・レジスタ・ビットです。このビットはセキュア・ソフトウェアからのみ読み出すことができます。GPDMA_SECCFGR.SECx = 1 のとき、このビットはセキュア・ソフトウェアにより書き込まなくてはなりません。GPDMA_SECCFGR.SECx = 0 のとき、セキュア書込みは無視されます。

GPDMA_SECCFGR.SECx がデアサートされると、この SSEC ビットもハードウェアによりデアサートされ (チャネルのセキュア再設定で非セキュアとして)、転送元からの GPDMA 転送は非セキュアになります。

ビット 14 **SAP** : 転送元割り当てポート

このビットはソース転送のマス・ポートの割り当てに使用されます。

0 : ポート 0 (AHB) 割り当て済み

1 : ポート 1 (AHB) 割り当て済み

注 : このビットは EN = 0 のときには書き込まなくてはなりません。このビットは、EN = 1 のときは読み出し専用です。

ビット 13 **SBX** : 各転送元のワードのアンアラインド・ハーフワード内での転送元のバイトを交換

転送元データ幅がワードより短い場合、このビットは無視されます。

転送元データ幅がワードの場合。

0 : 各転送元ワードのアンアラインド・ハーフワード内でのバイト・ベースの交換なし

1 : 各転送元ワードのアンアラインド・ハーフワード内の2 つの連続したバイトが交換されます

ビット 12:11 **PAM[1:0]** : パディング/配置モード

DDW_LOG2[1:0] = SDW_LOG2[1:0] の場合 : パースト・デスティネーション転送とパースト・ソース転送のデータ幅が同じときには、このビットは無視されます。

それ以外の場合、次のエニュメレーションした値になり、条件 PAM_1 は転送先データ幅が転送元データ幅より大きい場合、条件 PAM_2 は転送元データ幅が転送先データ幅より大きい場合です。

条件 : PAM_1

00 : 転送元データは、右寄せで、転送先データ幅まで 0s でパディングされた値で転送されます

01 : 転送元データは、右寄せで、転送先データ幅まで符号拡張された値で転送されます

10-11 : 連続した転送元データを FIFO キューに入れ、左 (LSB) から右 (MSB) の順で転送先データ幅でパックしてから (リトル・エンディアンと呼ばれます)、デスティネーション転送します

条件 : PAM_2

00 : 転送元データを、右寄せし、転送先データ幅に左側で切捨てをして転送します

01 : 転送元データを、左寄せし、転送先データ幅に右側で切捨てをして転送します

10-11 : 転送元データを FIFO キューに入れ、転送先データ幅にアンパックし、左 (LSB) から右 (MSB) の順で転送します (リトル・エンディアンと呼ばれます)

注 : 転送元ペリフェラルからの転送がペリフェラル・フローコントロール・モードで設定され (SWREQ = 0、PFREQ = 1 および DREQ = 0)、転送先データ幅が転送元データ幅より大きい場合、パッキングはサポートされません。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:4 **SBL_1[5:0]** : 転送元のパースト長から 1 を引いた 0 から 63 の値。

パースト長の単位は、ビートと呼ばれるパーストのデータ 1 つ分です。SBL_1[5:0] = 0 の場合、パーストはシングルと呼ばれることがあります。各データ/ビートの幅は転送先データ幅 SDW_LOG2[1:0] で定義されます。

注 : パースト転送が AHB 転送の 1 KB のアドレス境界を越えた場合、GPDMA は AHB プロトコルに準拠するよう、プログラムされたパーストをシングルまたは、より短いパーストへと変更、短縮を行います。

パースト転送の長さがチャネル x の FIFO サイズより大きい場合、GPDMA は FIFO サイズに準拠するよう、プログラムされたパーストをシングルまたは、より短いパーストへと変更、短縮を行います。GPDMA の再アービトレーションが有効長とより短いシングル/パースト長の間で行われるため、転送効率は低いですが、データの整合性は保証されます。

ビット 3 **SINC** : 転送元インクリメント・パースト

0 : 固定パースト

1 : 隣接インクリメント・パースト

GPDMA_CxSAR が指す転送元アドレスは、パースト・ビート/シングル転送後も保持されるか、隣接データに対応するオフセット分だけ転送後にインクリメントされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **SDW_LOG2[1:0]** : バーストの転送元データ幅の二進対数 (バイト単位)

00 : バイト

01 : ハーフワード (2 バイト)

10 : ワード (4 バイト)

11 : ユーザ設定エラーが報告され、転送は行われませんでした

注 : 8 バイト・データ幅を設定すると、ユーザ設定エラーが報告され、転送は行われません。

転送元のブロック・サイズは転送元のデータ幅の倍数でなくてはなりません (GPDMA_CxBR1.BNDT[2:0] に対し SDW_LOG2[1:0])。そうしないとユーザ設定エラーが報告され、転送は行われません。

ソース・バースト転送は、そのデータ幅による整列アドレスでなくてはなりません (開始アドレス GPDMA_CxSAR[2:0] に対して SDW_LOG2[1:0])。そうしないとユーザ設定エラーが報告され、転送は行われません。

16.8.11 GPDMA チャネル x 転送レジスタ 2 (GPDMA_CxTR2)

アドレスオフセット : $0x94 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : 0x0000 0000

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、チャネル x の転送を制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャネルが完了したら (ハードウェアが GPDMA_CxCR.EN をデアサートしたら)、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.UT2 = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCM[1:0]		Res.	Res.	Res.	Res.	TRIGPOL[1:0]		Res.	Res.	TRIGSEL[5:0]					
rw	rw					rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGM[1:0]		Res.	PFREQ	BREQ	DREQ	SWREQ	Q	Res.	REQSEL[7:0]						
rw	rw		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 **TCEM[1:0]** : 転送完了イベント・モード

このビットは転送完了と 1/2 転送完了イベント生成の粒度を規定します。

00 : ブロック・レベル (GPDMA_CxBR1.BNDT[15:0] = 0 のとき) : 完全な転送 (1/2 転送) イベントがブロックの終わり (1/2 の終わり) に生成されます。

注 : 最初の LLI₀ データ転送が null/void の (GPDMA_CxBR1.BNDT[15:0] = 0 で内部のレジスタ・ファイルにより直接プログラムされる) 場合、完全な転送イベントも 1/2 転送イベントも生成されません。

01 : チャンネル x (x = 0 から 5)、00 同様チャンネル x (x = 6 から 7) で、2D/繰返しブロック・レベル (GPDMA_CxBR1.BRC[10:0] = 0 および GPDMA_CxBR1.BNDT[15:0] = 0 の場合)。完全な転送 (1/2 転送) イベントが 2D/繰返しブロック (1/2 ブロック) の最後に生成されます。

注 : 最初の LLI₀ データ転送が null/void の (GPDMA_CxBR1.BNDT[15:0] = 0 で内部のレジスタ・ファイルにより直接プログラムされる) 場合、完全な転送イベントも 1/2 転送イベントも生成されません。

10 : LLI レベル : LLI の更新 (がある場合は) を含む、完全な転送イベントが LLI 転送の最後に生成されます。LLI データ転送の半分で 1/2 転送イベントが生成されます。データ転送がある場合、LLI データ転送はチャンネル x (x = 6 から 7) のブロック転送または 2D/繰返しブロック転送です。

注 : 最初の LLI₀ データ転送が null/void の (内部のレジスタ・ファイルにより、GPDMA_CxBR1.BNDT[15:0] = 0 で直接プログラムされている) 場合、1/2 転送イベントは生成されず、LLI₁ のロードが完了したときに、転送完了イベントが生成されます。

11 : チャンネル・レベル : 最後の LLI 転送の終わりに、完全な転送イベントが生成されます。1/2 転送イベントが最後の LLI データ転送の半分で生成されます。最後の LLI が、リンク・アドレス GPDMA_CxLLR.LA[15:2] を 0 に更新し、すべての GPDMA_CxLLR 更新ビット (UT1、UT2、UB1、USA、UDA、ULL、および UT3 と UB2) をクリアします。チャンネル転送が連続/無限の場合、イベントは生成されません。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **TRIGPOL[1:0]** : トリガ・イベント極性

このビットは、TRIGSEL[5:0] で定義される、選択したトリガ・イベント入力の極性を定義します。

00 : トリガなし (マスク済みトリガ・イベント)

01 : 立上がりエッジでのトリガ

10 : 立下がりエッジでのトリガ

11 : 00 と同じ

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:16 **TRIGSEL[5:0]** : トリガ・イベント入力選択

このビットは、TRIGPOL[1:0] ≠ 00 の場合、有効なトリガ・イベントで GPDMA 転送のトリガ・イベント入力を(セクション 16.3.7 に従って) 選択します。

ビット 15:14 TRIGM[1:0]: トリガ・モード

このビットはトリガにより、転送粒度の条件付けを定義します。

チャンネル x が TRIGPOL[1:0] = 00 または 11 で有効になっている (GPDMA_CxCR.EN がアサートされている) 場合、TRIGM[1:0] ビットは無視されます。

その他の場合、GPDMA 転送が最低 1 つのトリガ・ヒットにより条件付けされます。

00: ブロックレベル: 最初の各ブロック転送のバースト読出しは、1 つのヒット・トリガ (2D/繰り返しブロックが GPDMA_CxBR1.BRC[10:0] \neq 0 に設定された場合、各ブロックに対し、チャンネル x = 12 から 15) により条件付けされます。

01: チャンネル x (x = 0 から 5) は 00 と同じ。チャンネル x (x = 6 から 7) は 2D/繰り返しブロック・レベル。2D/繰り返しブロック転送の最初のバースト読出しは、1 つのヒット・トリガで条件付けされます。

10: リンク・レベル: LLI リンク転送は 1 つのヒット・トリガにより条件付けされます。LLI データ転送 (がある場合) は条件付けされません。

11: プログラム・バースト・レベル: SWREQ = 1 の場合、プログラムされたバースト読出しはそれぞれ、1 つのヒット・トリガにより条件付けされます。SWREQ = 0 の場合、選択されたペリフェラルによりリクエストされたプログラムされたバーストは、それぞれ、1 つのヒット・トリガにより条件付けされます。

- ペリフェラルが LLI データ転送の転送元としてプログラムされている場合 (DREQ = 0)、プログラムされたそれぞれのバースト読出しが条件付けされます。

- ペリフェラルが LLI データ転送の転送先としてプログラムされている場合 (DREQ = 1)、プログラムされたそれぞれのバースト書込みが条件付けされます。最初の準備された FIFO ベースのソース・バーストとも呼ばれる、(おそらく 2D/繰り返し) ブロックの最初のメモリ・バースト読出しは、ハードウェア・リクエストと最初のトリガ・ヒットの両方の発生によりゲートされます。

チャンネルが新しいアクティブなトリガ設定 (選択したトリガの立上がり/立下りエッジ (それぞれ TRIGPOL[1:0] = 01、TRIGPOL[1:0] = 10)) でチャンネルがイネーブル/ロードされると、チャンネル x のトリガに対する GPDMA モニタリングを開始します。

このトリガのモニタはトリガされ完了していない (データまたはリンク) 転送の間、有効のままになります。新しいトリガが検出されると、このヒットは内部で記憶され、定義された立上がり、または立下りエッジが変更されておらず、TRIGSEL[5:0] が変更されていない限り、次の転送を許可しチャンネルが有効になります。

TRIGSEL[5:0] または TRIGPOL[1:0] の新しい値で GPDMA_CxTR2 を更新する次の LLI _{$n+1$} の転送が、前に定義した LLI _{n} トリガの記憶したヒットのモニタリング、トラッシングをリセットします。

最初の新しいトリガ hit _{$n+1$} を記憶した後、別の 2 つめのトリガ hit _{$n+2$} を検出し、hit _{n} にトリガされた転送が完了していない場合、hit _{$n+2$} は失われ記憶されません。トリガ・オーバーラン・フラグが報告され (GPDMA_CxSR.TOF = 1)、イネーブル (GPDMA_CxCR.TOIE = 1) になると割込みが生成されます。トリガのオーバーランでハードウェアにより、チャンネルが自動的に無効になることはありません。

注: 転送元ブロック・サイズがソース・バースト・サイズの倍数ではなく、転送元データ幅の倍数である場合、プログラムされた最後のソース・バーストは完了せず、ブロック・サイズに合うように内部で短縮されます。この場合、TRIGM[1:0] = 11 がかつ (SWREQ = 1 または (SWREQ = 0 かつ DREQ = 0)) の場合、(シングルおよび/または、より短いバーストによる) 短縮されたバースト転送は、トリガにより一旦、条件付けされます。

プログラムされたデスティネーション・バーストは、シングルおよび/または、(FIFO サイズ、ブロックサイズ、1 KB のアドレスの境界のはみ出しに対し) より短いバーストにより内部で短縮されます。トリガがプログラムされたデスティネーション・バーストを条件付けする (TRIGM[1:0] = 11 かつ SWREQ = 0 かつ DREQ = 1) 場合、この短縮されたデスティネーション・バースト転送は、一旦トリガにより条件付けされます。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 PFREQ : ペリフェラル・フロー制御モードでのハードウェアリクエスト

重要 : あるチャネル x にこの機能が実装されていない場合、このビットは予約され、PFREQ は存在しません (この機能が実装されているチャネルのリストは [セクション 16.3.2](#) を参照してください)。

チャネル x が SWREQ = 1 (メモリ間転送のソフトウェア・リクエスト) で有効になっている場合 (GPDMA_CxCR.EN がアサートされている場合)、このビットは無視されます。その他の場合 :

0 : 選択されたハードウェア・リクエストが、GPDMA 制御モードのハードウェア・リクエスト/確認プロトコルでペリフェラルによりドライブされます。GPDMA は GPDMA_CxTR1.BNDT[15:0] によりプログラムされ、ハードウェアによりブロック転送完了のために内部的に使用されます。

1 : 選択されたハードウェア・リクエストが、ペリフェラル制御モードのハードウェア・リクエスト/確認プロトコルでペリフェラルによりドライブされます。GPDMA ブロック転送はペリフェラルそのものにより早期に完了することが可能です (詳細は [セクション 16.3.6](#) 参照)。

注 : ペリフェラル・フロー・コントロール・モードでは次の制約があります。

- 2D/繰り返しブロックのサポートがありません (GPDMA_CxBR1.BRC[10:0] は 0 に設定しなくてはなりません)。
- ペリフェラルを転送元として設定しなくてはなりません (DREQ = 0)。
- より広い転送先データ幅へのデータ・パッキングはサポートされていません (転送先データ幅が転送元データ幅より大きい場合、GPDMA_CxTR1.PAM[1] を 0 に設定しなくてはなりません)。
- GPDMA_CxBR1.BNDT[15:0] はソース (ペリフェラル) ・バースト・サイズの倍数にプログラムされなくてはなりません。

ビット 11 BREQ : ブロック・ハードウェア・リクエスト

チャネル x が SWREQ = 1 (メモリ間転送のソフトウェア・リクエスト) で有効になっている場合 (GPDMA_CxCR.EN がアサートされている場合)、このビットは無視されます。その他の場合 :

0 : 選択されたハードウェア・リクエストが、バースト・レベルでのハードウェア・リクエスト/確認プロトコルでペリフェラルによりドライブされます。

1 : 選択されたハードウェア・リクエストが、ブロック・レベルでのハードウェア・リクエスト/確認プロトコルでペリフェラルによりドライブされます ([セクション 16.3.4](#) を参照)。

ビット 10 DREQ : 転送先ハードウェア・リクエスト

チャネル x が SWREQ = 1 (メモリ間転送のソフトウェア・リクエスト) で有効になっている場合 (GPDMA_CxCR.EN がアサートされている場合) このビットは無視されます。その他の場合 :

0 : 転送元のペリフェラルでドライブされる選択されたハードウェア・リクエスト (リクエスト信号が転送元/読み出しポート経由で GPDMA 転送スケジューラにより考慮されます)

1 : 転送先のペリフェラルでドライブされる選択されたハードウェア・リクエスト (リクエスト信号が転送先/書き出しポート経由で GPDMA 転送スケジューラにより考慮されます)

注 : SWREQ = 0 および PFREQ = 1 (ペリフェラル・フローコントロール・モードのペリフェラル・ハードウェア・リクエスト) でチャネル x が有効な (GPDMA_CxCR.EN がアサートされている) 場合、この DREQ ビットへのあらゆるソフトウェアのアサーションは無視されます。ペリフェラル・フローコントロール・モードでは、ペリフェラルからメモリへの転送のみがサポートされています。

ビット 9 SWREQ : ソフトウェア・リクエスト

このビットは GPDMA_CxCR.EN アサートされた場合、内部的に考慮されます。

0 : ソフトウェア・リクエストは無し。選択されたハードウェア・リクエスト、REQSEL[7:0]、は考慮されます。

1 : メモリ間転送のソフトウェア・リクエスト。REQSEL[7:0] に従ったデフォルトで選択されたハードウェア・リクエストは無視されます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REQSEL[7:0]** : GPDMA ハードウェア・リクエスト選択

チャンネル x が $SWREQ = 1$ (メモリ間転送のソフトウェア・リクエスト) で有効になっている場合 (GPDMA_CxCR.EN がアサートされている場合)、これらのビットは無視されます。その他の場合、選択されたハードウェア・リクエストは [セクション 16.3.4](#) に従って内部的に考慮されます。

注意 : ユーザは同じ入力ハードウェア・リクエスト (同じ REQSEL[7:0] 値) を異なる有効な GPDMA チャンネルに割り当ててはなりません (これらのチャンネルには GPDMA_CxCR.EN = 1 および GPDMA_CxTR2.SWREQ = 0 を割り当てます)。GPDMA では、同時に有効になっているチャンネルが同じハードウェア・ペリフェラル・リクエスト信号により誤って設定されているケースを、ハードウェアがサポートすることは意図していません。また、ユーザ設定エラーは報告されません。

16.8.12 GPDMA チャンネル x ブロック・レジスタ 1 (GPDMA_CxBR1)

アドレスオフセット : $0x98 + 0x80 * x$ ($x = 0$ から 5)

リセット値 : 0x0000 0000

このレジスタは、チャンネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャンネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、チャンネル x の転送をブロック・レベルで制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読み出し専用です。

チャンネル x が完了し (その後、ハードウェアが GPDMA_CxCR.EN をデアサートし) たら、このレジスタに書き込まなくてはなりません。チャンネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードのリンク転送時。

- GPDMA_CxLLR.UB1 = 1 の場合、このレジスタは、メモリにある次の LLI から GPDMA により、自動的に更新されます。
- GPDMA_CxLLR.UB1 = 0 で、少なくとも 1 つのリンクリスト・レジスタがメモリの中の次の LLI から更新されることになっている場合、このレジスタは自動的に、内部的にフィールド BNDT[15:0] のプログラムされた値で復元されます。
- すべての更新ビット GPDMA_CxLLR.Uxx が null で、GPDMA_CxLLR.LA[15:0] $\neq 0$ の場合、現在の LLI は最後の 1 つで、連続的に実行されます。このレジスタは、この最後の各 LLI の実行後に、BNDT[15:0] にプログラムされた値で自動的に、内部的に復元されます。
- GPDMA_CxLLR = 0 の場合、このレジスタと BNDT[15:0] は null のままとなり、チャンネル x は完了します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BNDT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **BNDT[15:0]** : 転送元から転送するデータ・バイトのブロック数

転送元から転送されてきたブロックのサイズ。チャンネルが有効な場合、このフィールドは読出し専用となりデクリメントされます。これは現在の転送元ブロックにある転送する残りのデータ項目の数を示すものです。BNDT[15:0] はバイト数でプログラムされ、最大の転送元ブロック・サイズは 64 KB - 1 です。一旦、最後のデータ転送が完了したとき (BNDT[15:0] = 0)。

- GPDMA_CxLLR.UB1 = 1 の場合、このフィールドはメモリの LLI により更新されます。
- GPDMA_CxLLR.UB1 = 0 で、null ではない Uxx 更新ビットが最低 1 つある場合、このフィールドはプログラムされた値に内部的に復元されます。
- すべての GPDMA_CxLLR.Uxx が 0 で、GPDMA_CxLLR.LA[15:0] = 0 の場合、このフィールドはプログラムされた値 (無限/連続の最後の LLI) に内部的に復元されます。
- GPDMA_CxLLR = 0 の場合、このフィールドは最後の LLI データ転送後に 0 に維持されます。

注 : null ではない転送元のブロック・サイズは、転送元のデータ幅の倍数でなくてはなりません (BNDT[2:0] に対し GPDMA_CxTR1.SDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

パッキング・モードに設定されている (GPDMA_CxTR1.PAM[1] = 1 で転送先データ幅が転送元データ幅とは異なる) 場合、(null ではない) 転送元ブロック・サイズは、転送先データ幅の倍数でなくてはなりません (BNDT[2:0] に対し GPDMA_CxTR1.DDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

16.8.13 GPDMA チャンネル x オルタネート・ブロック・レジスタ 1 (GPDMA_CxBR1)

アドレスオフセット : 0x98 + 0x80 × x (x = 6 から 7)

リセット値 : 0x0000 0000

このレジスタは、チャンネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャンネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、チャンネル x の転送をブロック・レベルで制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャンネル x が完了し (その後、ハードウェアが GPDMA_CxCR.EN をデアサートし) たら、このレジスタに書き込まなくてはなりません。チャンネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードのリンク転送時。

- GPDMA_CxLLR.UB1 = 1 の場合、このレジスタは、メモリにある次の LLI から GPDMA により、自動的に更新されます。
- GPDMA_CxLLR.UB1 = 0 で、少なくとも 1 つのリンクリスト・レジスタがメモリの中の次の LLI から更新されることになっている場合、このレジスタは自動的に、内部的にフィールド BNDT[15:0] および BRC[10:0] のプログラムされた値で復元されます。
- すべての更新ビット GPDMA_CxLLR.Uxx が null で、GPDMA_CxLLR.LA[15:0] ≠ 0 の場合、現在の LLI は最後の 1 つで、連続的に実行されます。このレジスタは、この最後の LLI の実行後に、フィールド BNDT[15:0] および BRC[10:0] にプログラムされた値で自動的に、内部的に復元されます。
- GPDMA_CxLLR = 0 の場合、BNDT[15:0] および BRC[10:0] は null のままとなり、チャンネル x は完了します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRDDEC	BRSDEC	DDEC	SDEC	Res.	BRC[10:0]										
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BNDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **BRDDEC** : ブロック繰り返し転送先アドレス・デクリメント

0 : ブロック転送の最後に、GPDMA_CxDAR レジスタは現在の GPDMA_CxDAR 値（現在の転送先アドレス）に、プログラムされたオフセット GPDMA_CxBR2.BRDAO を加えることで更新されます。

1 : ブロック転送の最後に、GPDMA_CxDAR レジスタは現在の GPDMA_CxDAR 値（現在の転送先アドレス）からプログラムされたオフセット GPDMA_CxBR2.BRDAO を減算することで更新されます。

注 : （BRDDEC に従った）このインクリメント／デクリメントに加え、GPDMA_CxDAR は同時に GPDMA_CxTR3.DAO 値のインクリメント／デクリメント（DDEC による）によっても更新されます。これは通常、各プログラムされたバースト転送の最後に行われます。

ビット 30 **BRSDEC** : ブロック繰り返し転送元アドレス・デクリメント

0 : ブロック転送の最後に、GPDMA_CxSAR レジスタは現在の GPDMA_CxSAR 値（現在の転送元アドレス）に、プログラムされたオフセット GPDMA_CxBR2.BRSAO を加えることで更新されます。

1 : ブロック転送の最後に、GPDMA_CxSAR レジスタは現在の GPDMA_CxSAR 値（現在の転送元アドレス）から、プログラムされたオフセット GPDMA_CxBR2.BRSAO を減算することで更新されます。

注 : （BRSDEC に従った）このインクリメント／デクリメントに加え、GPDMA_CxSAR は同時に GPDMA_CxTR3.SAO 値のインクリメント／デクリメント（SDEC による）によっても更新されます。これは、あらゆるプログラムされたバースト転送の後に行われます。

ビット 29 **DDEC** : 転送先アドレス・デクリメント

0 : 転送先へのプログラムされたバースト転送の最後に、GPDMA_CxDAR レジスタは現在の GPDMA_CxDAR 値（現在の転送先アドレス）に、プログラムされたオフセット GPDMA_CxTR3.DAO を加えることで更新されます。

1 : 転送先へのプログラムされたバースト転送の最後に、GPDMA_CxDAR レジスタは現在の GPDMA_CxDAR 値（現在の転送先アドレス）から、プログラムされたオフセット GPDMA_CxTR3.DAO を減算することで更新されます。

ビット 28 **SDEC** : 転送元アドレス・デクリメント

0 : 転送元からのプログラムされたバースト転送の最後に、GPDMA_CxSAR レジスタは現在の GPDMA_CxSAR 値（現在の転送元アドレス）に、プログラムされたオフセット GPDMA_CxTR3.SAO を加えることで更新されます。

1 : 転送元からのプログラムされたバースト転送の最後に、GPDMA_CxSAR レジスタは現在の GPDMA_CxSAR 値（現在の転送先アドレス）から、プログラムされたオフセット GPDMA_CxTR3.SAO を減算することで更新されます。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:16 **BRC[10:0]** : ブロック繰り返しカウンタ

このフィールドには、現在のブロックの繰り返し数が含まれます (0 から 2047)。

チャンネルが有効になっている場合、このフィールドは読出し専用になります。デクリメント後、このフィールドは現在のブロックを除く、残りのブロックの数を示します。このカウンタはそれぞれの完了したブロック転送に対してハードウェアによりデクリメントされます。

一旦、最後のブロック転送が完了した場合 (BRC[10:0] および BNDT[15:0] = 0)。

- GPDMA_CxLLR.UB1 = 1 の場合、全ての GPDMA_CxBR1 フィールドはメモリの中の次の LLI により更新されます。
- GPDMA_CxLLR.UB1 = 0 で、null ではない Uxx 更新ビットが最低 1 つある場合、このフィールドはプログラムされた値に内部的に復元されます。
- すべての GPDMA_CxLLR.Uxx が 0 で、GPDMA_CxLLR.LA[15:0] ≠ 0 の場合、このフィールドはプログラムされた値 (無限/連続の最後の LLI) に内部的に復元されます。
- GPDMA_CxLLR = 0 の場合、このフィールドは最後の LLI およびデータ転送後に 0 に維持されます。

ビット 15:0 **BNDT[15:0]** : 転送元から転送するデータ・バイトのブロック数

転送元から転送されてきたブロックのサイズ。チャンネルが有効な場合、このフィールドは読出し専用となりデクリメントされます。これは現在の転送元ブロックにある転送する残りのデータ項目の数を示すものです。BNDT[15:0] はバイト数でプログラムされ、最大の転送元ブロック・サイズは 64 KB - 1 です。

一旦、最後のデータ転送が完了したとき (BNDT[15:0] = 0)。

- GPDMA_CxLLR.UB1 = 1 の場合、このフィールドはメモリの LLI により更新されます。
- GPDMA_CxLLR.UB1 = 0 で、null ではない Uxx 更新ビットが最低 1 つある場合、このフィールドはプログラムされた値に内部的に復元されます。
- すべての GPDMA_CxLLR.Uxx が 0 で、GPDMA_CxLLR.LA[15:0] ≠ 0 の場合、このフィールドはプログラムされた値 (無限/連続の最後の LLI) に内部的に復元されます。
- GPDMA_CxLLR = 0 の場合、このフィールドは最後の LLI データ転送後に 0 に維持されます。

注 : null ではない転送元のブロック・サイズは、転送元のデータ幅の倍数でなくてはなりません (BNDT[2:0] に対し GPDMA_CxTR1.SDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

パッキング・モードに設定されている (GPDMA_CxTR1.PAM[1] = 1 で転送先データ幅が転送元データ幅とは異なる) 場合、(null ではない) 転送元ブロック・サイズは、転送先データ幅の倍数でなくてはなりません (BNDT[2:0] に対し GPDMA_CxTR1.DDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

16.8.14 GPDMA チャネル x 転送元アドレスレジスタ (GPDMA_CxSAR)

アドレスオフセット : $0x9C + 0x80 * x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは転送の転送元開始アドレスを設定します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

このレジスタは GPDMA_CxCR.EN = 1 のときには読み出し専用で、転送元からの次のバースト転送のアドレスを反映するように、ハードウェアにより継続的に更新されます。

チャネルが完了し（その後、ハードウェアが GPDMA_CxCR.EN をデアサートし）たら、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、2D／繰り返しブロック、LLI、あるいはフル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.USA = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **SA[31:0]** : 転送元アドレス

このフィールドは次のデータを読み込むアドレスを示すものです。

チャネルの操作に際し、このフィールドは、転送元のアドレス指定モード (GPDMA_CxTR1.SINC) に従い、各バースト転送の転送元データの後ろに固定、またはデータ幅 (GPDMA_CxTR1.SDW_LOG2[1:0]) によりインクリメントし、データを読み出す次のアドレスを反映するようになっています。

チャネルの操作に際して、このアドレスは完了した各ソース・バーストの後に更新され、結果的に次のようになります。

- 固定アドレス指定モードまたは隣接データインクリメント・モードのプログラムされたソース・バースト。連続インクリメント (GPDMA_CxTR1.SINC = 1) の場合、追加のアドレス・オフセット値はプログラムされたバースト・サイズで、GPDMA_CxTR1.SBL_1[5:0] および GPDMA_CxTR1.SDW_LOG2[21:0] で定義されます。
- GPDMA_CxBR1.SDEC および GPDMA_CxTR3.SAO[12:0] でプログラムされた、追加の転送元のインクリメント／デクリメント・オフセット値。
- 2D アドレス指定機能のあるチャネル x ($x = 6$ から 7) の、完了したソース・ブロック転送の場合、追加ブロックは、GPDMA_CxBR1.BRSDEC および GPDMA_CxBR2.BRSAO[15:0] によるプログラムに従って、転送元のインクリメント／デクリメント・オフセット値を繰り返します。

リンクリスト・モードでは、LLI が GPDMA_CxLLR.USA = 1 に設定されている場合、LLI データ転送が完了した後、このレジスタが GPDMA によりメモリから自動的に更新されます。

注： 転送元アドレスは、プログラムされたソース・バーストのデータ幅に合わせなくてはなりません (SA[2:0] に対し GPDMA_CxTR1.SDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

転送元ブロック・サイズがソース・バースト・サイズの倍数ではなく、転送元データ幅の倍数である場合、最後のプログラムされたソース・バーストは完了せず、ブロック・サイズに合うように内部で短縮されます。この場合、追加の GPDMA_CxTR3.SAO[12:0] は適用されません。

16.8.15 GPDMA チャネル x 転送先アドレス・レジスタ (GPDMA_CxDAR)

アドレスオフセット : $0xA0 + 0x80 * x$ ($x = 0$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは転送の転送先開始アドレスを設定します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

このレジスタは GPDMA_CxCR.EN = 1 のときには読出し専用で、転送先への次のバースト転送のアドレスを反映するように、ハードウェアにより継続的に更新されます。

チャネルが完了し（その後、ハードウェアが GPDMA_CxCR.EN をデアサートし）たら、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、2D／繰り返しブロック、LLI、あるいはフル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.UDA = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DA[31:0]** : 転送先アドレス

このフィールドは次のデータを書き出すアドレスを示すものです。

チャネルの操作に際し、このフィールドは転送先のアドレス指定モード (GPDMA_CxTR1.DINC) に従い、各バースト転送の転送先データの後ろに固定、またはデータ幅 (GPDMA_CxTR1.DDW_LOG2[21:0]) によりインクリメントし、データを書き出す次のアドレスを反映するようになっています。

チャネルの操作に際して、このアドレスは完了した各デスティネーション・バーストの後に更新され、結果的に次のようになります。

- 固定アドレス指定モードまたは隣接データインクリメント・モードのプログラムされたデスティネーション・バースト。連続インクリメント (GPDMA_CxTR1.DINC = 1) の場合、追加のアドレス・オフセット値はプログラムされたバースト・サイズで、GPDMA_CxTR1.DBL_1[5:0] および GPDMA_CxTR1.DDW_LOG2[1:0] で定義されます。
- GPDMA_CxBR1.DDEC および GPDMA_CxTR3.DAO[12:0] でプログラムされた、追加の転送先のインクリメント／デクリメント・オフセット値。
- 2D アドレス指定機能のあるチャネル x ($x = 6$ から 7) の、完了したデスティネーション・ブロック転送の場合、追加ブロックは、GPDMA_CxBR1.BRDDEC および GPDMA_CxBR2.BRDAO[15:0] によるプログラムに従って、転送先のインクリメント／デクリメント・オフセット値を繰り返します。

リンクリスト・モードでは、LLI が GPDMA_CxLLR.UDA = 1 に設定されている場合、LLI データ転送が完了した後、このレジスタが GPDMA によりメモリから自動的に更新されます。

注： 転送先アドレスは、プログラムされたデスティネーション・バーストのデータ幅に合わせなくてはなりません (DA[2:0] に対し GPDMA_CxTR1.DDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

16.8.16 GPDMA チャネル x 転送レジスタ 3 (GPDMA_CxTR3)

アドレスオフセット : $0xA4 + 0x80 \times x$ ($x = 6$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、チャネル x の転送を制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャネルが完了し (その後、ハードウェアが GPDMA_CxCR.EN をデアサートし) たら、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.UT3 = 1 の場合、このレジスタは GPDMA よりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	DAO[12:0]												
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SAO[12:0]												
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:16 **DAO[12:0]** : 転送先アドレス・オフセット・インクリメント

それぞれのプログラムされたデスティネーション・バーストの GPDMA_CxDAR が指す転送先アドレスは、このオフセット DAO[12:0] により、(GPDMA_CxBR1.DDEC に従って) インクリメントまたはデクリメントします。このオフセットは、完了したバーストがインクリメント・モード (GPDMA_CxTR1.DINC = 1) でアドレス指定される際に、プログラムされたバースト・サイズは含んでおらず、これに追加されるものです。

注 : 転送先アドレス・オフセットは、プログラムされたデスティネーション・バーストのデータ幅に合わせなくてはなりません (DAO[2:0] に対し GPDMA_CxTR1.DDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:0 **SAO[12:0]** : 転送元アドレス・オフセット・インクリメント

それぞれのプログラムされたソース・バーストの GPDMA_CxSAR が指す転送元アドレスは、このオフセット SAO[12:0] により、(GPDMA_CxBR1.SDEC に従って) インクリメントまたはデクリメントします。このオフセットは、完了したバーストがインクリメント・モード (GPDMA_CxTR1.SINC = 1) でアドレス指定される際に、プログラムされたバースト・サイズは含んでおらず、これに追加されるものです。

注 : 転送元アドレス・オフセットは、プログラムされたソース・バーストのデータ幅に合わせなくてはなりません (SAO[2:0] に対し GPDMA_CxTR1.SDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。

転送元ブロック・サイズがデスティネーション・バースト・サイズの倍数ではなく、転送元データ幅の倍数である場合、最後のプログラムされたソース・バーストは完了せず、ブロック・サイズに合うように内部で短縮されます。この場合、追加の GPDMA_CxTR3.SAO[12:0] は適用されません。

16.8.17 GPDMA チャネル x ブロック・レジスタ 2 (GPDMA_CxBR2)

アドレスオフセット : $0xA8 + 0x80 \times x$ ($x = 6$ から 7)

リセット値 : $0x0000\ 0000$

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、チャネル x の転送を 2D／繰り返しブロック・レベルで制御します。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読出し専用です。

チャネルが完了し（その後、ハードウェアが GPDMA_CxCR.EN をデアサートし）たら、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、2D／繰り返しブロック、LLI、あるいはフル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.UB2 = 1 の場合、このレジスタは GPDMA よりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRDAO[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRSAO[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **BRDAO[15:0]** : ブロック繰り返し転送先アドレス・オフセット

2D アドレス指定機能があるチャネルのこのフィールドは、ブロック転送の最後に現在の転送先アドレス (GPDMA_CxDAR) を (GPDMA_CxBR1.BRDDEC に従って加算または減算すること) で更新するために使用されます。

注 : ブロックが繰り返されている転送先アドレス・オフセットは、プログラムされたデスティネーション・パーストのデータ幅に合わせなくてはなりません (BRDAO[2:0] に対し GPDMA_CxTR1.DDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。
BRDAO[15:0] は、ペリフェラル・フローコントロール・モードでは 0 に設定しなくてはなりません (GPDMA_CxTR2.PFREQ = 1 の場合)。

ビット 15:0 **BRSAO[15:0]** : ブロック繰り返し転送元アドレス・オフセット

2D アドレス指定機能があるチャネルのこのフィールドは、ブロック転送の最後に現在の転送元アドレス (GPDMA_CxSAR) を (GPDMA_CxBR1.BRSDEC に従って加算または減算すること) で更新するために使用されます。

注 : ブロック繰り返し転送元アドレス・オフセットは、プログラムされたソース・パーストのデータ幅に合わせなくてはなりません (BRSAO[2:0] に対し GPDMA_CxTR1.SDW_LOG2[1:0])。それ以外の場合、ユーザ設定エラーが報告され、転送は行われません。
BRSAO[15:0] は、ペリフェラル・フローコントロール・モードでは 0 に設定しなくてはなりません (GPDMA_CxTR2.PFREQ = 1 の場合)。

16.8.18 GPDMA チャネル x のリンクリスト・アドレス・レジスタ (GPDMA_CxLLR)

アドレスオフセット : $0xCC + 0x80 * x$ ($x = 0$ から 5)

リセット値 : $0x0000\ 0000$

このレジスタは、チャネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、メモリの次の LLI のデータ構造と、そのアドレス・ポインタの設定を行います。このレジスタが null の場合、チャネル転送は完了です。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読み出し専用です。

チャネルが完了し（その後、ハードウェアが GPDMA_CxCR.EN をデアサートし）たら、このレジスタに書き込まなくてはなりません。チャネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.ULL = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UT1	UT2	UB1	USA	UDA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ULL
rw	rw	rw	rw	rw											rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LA[15:2]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

ビット 31 **UT1** : メモリから GPDMA_CxTR1 を更新

このビットはリンク転送時にメモリからの GPDMA_CxTR1 の更新を制御します。

0 : GPDMA_CxTR1 の更新は無し

1 : GPDMA_CxTR1 更新

ビット 30 **UT2** : メモリから GPDMA_CxTR2 を更新

このビットはリンク転送時にメモリからの GPDMA_CxTR2 の更新を制御します。

0 : GPDMA_CxTR2 の更新は無し

1 : GPDMA_CxTR2 更新

ビット 29 **UB1** : メモリから GPDMA_CxBR1 を更新

このビットはリンク転送時にメモリからの GPDMA_CxBR1 の更新を制御します。

UB1 = 0 で GPDMA_CxLLR ≠ 0 の場合、リンクリストは完了していません。

GPDMA_CxBR1.BNDT[15:0] は、データ転送の完了後、リンク転送の前にプログラムされた値に復元されます。

0 : メモリからの GPDMA_CxBR1 の更新は無し（リンク転送が何かあった場合は、GPDMA_CxBR1.BNDT[15:0] が復元されます）

1 : GPDMA_CxBR1 更新

ビット 28 **USA** : メモリから GPDMA_CxSAR を更新

このビットはリンク転送時にメモリからの GPDMA_CxSAR の更新を制御します。

0 : GPDMA_CxSAR の更新は無し

1 : GPDMA_CxSAR の更新有り

ビット 27 **UDA** : メモリから GPDMA_CxDAR レジスタを更新

このビットはリンク転送時にメモリからの GPDMA_CxDAR の更新の制御に使用されます。

0 : GPDMA_CxDAR の更新は無し

1 : GPDMA_CxDAR の更新有り

ビット 26:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **ULL** : メモリから GPDMA_CxLLR レジスタを更新

このビットはリンク転送時にメモリからの GPDMA_CxLLR の更新の制御に使用されます。

0 : GPDMA_CxLLR の更新は無し

1 : GPDMA_CxLLR の更新有り

ビット 15:2 **LA[15:2]** : 次のリンクリスト・データ構造へのポインタ (16 bit の下位アドレス)

UT1 = UT2 = UB1 = USA = UDA = ULL = 0 で LA[15:20] = 0 の場合、現在の LLI が最後です。チャンネル転送が、リンクリスト GPDMA レジスタ・ファイルの更新が一切なく完了します。

それ以外の場合、このフィールドは、データ転送が一旦完了すると、リンクリスト GPDMA 内部レジスタファイル (GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR) を条件付きで更新するため、次のリンクリスト・データ構造を自動的にフェッチする、メモリ・アドレス・オフセットへのポインタです。

注 : ユーザはこのポインタを 32 bit に合わせてプログラムしなくてはなりません。下位 2 bit の書き込みは無視されます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

16.8.19 GPDMA チャンネル x オルタネート・リンクリスト・アドレス・レジスタ (GPDMA_CxLLR)

アドレスオフセット : $0xCC + 0x80 \times x$ ($x = 6$ から 7)

リセット値 : 0x0000 0000

このレジスタは、チャンネル x のセキュア・ステート (GPDMA_SECCFGR.SECx) によって、セキュアまたは非セキュアとなり、チャンネル x の特権ステート (GPDMA_PRIVCFGR.PRIVx) によって、特権または非特権となります。

このレジスタは、メモリの次の LLI のデータ構造と、そのアドレス・ポインタの設定を行います。このレジスタが null の場合、チャンネル転送は完了です。

GPDMA_CxCR.EN = 0 の場合、このレジスタに書き込まなくてはなりません。

GPDMA_CxCR.EN = 1 のとき、このレジスタは読み出し専用です。

チャンネルが完了し (その後、ハードウェアが GPDMA_CxCR.EN をデアサートし) たら、このレジスタに書き込まなくてはなりません。チャンネル転送は、ブロック、LLI、フル・リンクリストと、様々なレベルで完了したり、プログラムされたりする場合があります。

リンクリスト・モードでは、リンク転送時、GPDMA_CxLLR.ULL = 1 の場合、このレジスタは GPDMA によりメモリから自動的に更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UT1	UT2	UB1	USA	UDA	UT3	UB2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ULL
rw	rw	rw	rw	rw	rw	rw									rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LA[15:2]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

ビット 31 **UT1** : メモリから GPDMA_CxTR1 を更新

このビットはリンク転送時にメモリからの GPDMA_CxTR1 の更新を制御します。

0 : GPDMA_CxTR1 の更新は無し

1 : GPDMA_CxTR1 更新

ビット 30 **UT2** : メモリから GPDMA_CxTR2 を更新

このビットはリンク転送時にメモリからの GPDMA_CxTR2 の更新を制御します。

0 : GPDMA_CxTR2 の更新は無し

1 : GPDMA_CxTR2 更新

ビット 29 **UB1** : メモリから GPDMA_CxBR1 を更新

このビットはリンク転送時にメモリからの GPDMA_CxBR1 の更新を制御します。

UB1 = 0 で GPDMA_CxLLR ≠ 0 の場合、リンクリストは完了していません。

GPDMA_CxBR1.BNDT[15:0] は、データ転送の完了後、リンク転送の前にプログラムされた値に復元されます。

0 : メモリからの GPDMA_CxBR1 の更新は無し (リンク転送が何かあった場合は、GPDMA_CxBR1.BNDT[15:0] が復元されます)

1 : GPDMA_CxBR1 更新

ビット 28 **USA** : メモリから GPDMA_CxSAR を更新

このビットはリンク転送時にメモリからの GPDMA_CxSAR の更新を制御します。

0 : GPDMA_CxSAR の更新は無し

1 : GPDMA_CxSAR の更新有り

ビット 27 **UDA** : メモリから GPDMA_CxDAR レジスタを更新

このビットはリンク転送時にメモリからの GPDMA_CxDAR の更新の制御に使用されます。

0 : GPDMA_CxDAR の更新は無し

1 : GPDMA_CxDAR の更新有り

ビット 26 **UT3** : メモリから GPDMA_CxTR3 を更新

このビットはリンク転送時にメモリからの GPDMA_CxTR3 の更新を制御します。

0 : GPDMA_CxTR3 の更新は無し

1 : GPDMA_CxTR3 更新

ビット 25 **UB2** : メモリから GPDMA_CxBR2 を更新

このビットはリンク転送時にメモリからの GPDMA_CxBR2 の更新を制御します。

0 : GPDMA_CxBR2 の更新は無し

1 : GPDMA_CxBR2 更新

ビット 24:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **ULL** : メモリから GPDMA_CxLLR レジスタを更新

このビットはリンク転送時にメモリからの GPDMA_CxLLR の更新の制御に使用されます。

0 : GPDMA_CxLLR の更新は無し

1 : GPDMA_CxLLR の更新有り

ビット 15:2 **LA[15:2]** : 次のリンクリスト・データ構造へのポインタ (16 bit の下位アドレス)

UT1 = UT2 = UB1 = USA = UDA = ULL = 0 で LA[15:20] = 0 の場合、現在の LLI が最後です。チャネル転送が、リンクリスト GPDMA レジスタ・ファイルの更新が一切なく完了します。

それ以外の場合、このフィールドは、データ転送が一旦完了すると、リンクリスト GPDMA 内部レジスタファイル (GPDMA_CxTR1、GPDMA_CxTR2、GPDMA_CxBR1、GPDMA_CxSAR、GPDMA_CxDAR、GPDMA_CxLLR) を条件付きで更新するため、次のリンクリスト・データ構造を自動的にフェッチする、メモリ・アドレス・オフセットへのポインタです。

注 : ユーザはこのポインタを 32 bit に合わせてプログラムしなくてはなりません。下位 2 bit の書込みは無視されます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

16.8.20 GPDMA レジスタマップ

表 134. GPDMA レジスタのマップとリセット値

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	GPDMA_SECCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0	
	リセット値																									0	0	0	0	0	0	0	0	
0x04	GPDMA_PRIVCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0	
	リセット値																									0	0	0	0	0	0	0	0	
0x08	GPDMA_RCFGLOCKR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0	
	リセット値																									0	0	0	0	0	0	0	0	
0x0C	GPDMA_MISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MIS7	MIS6	MIS5	MIS4	MIS3	MIS2	MIS1	MIS0	
	リセット値																									0	0	0	0	0	0	0	0	
0x10	GPDMA_SMISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MIS7	MIS6	MIS5	MIS4	MIS3	MIS2	MIS1	MIS0	
	リセット値																									0	0	0	0	0	0	0	0	
0x14 - 0x4C	予約済み	予約済み																																
0x50+0x80 × x (x=0 から 7)	GPDMA_CxLBAR	LBA[31:16]																Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	
0x54 から 0x58+0x80 × x (x=0 から 7)	予約済み	予約済み																																
0x5C+0x80 × x (x=0 から 7)	GPDMA_CxFCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TOF	SUSPF	USEF	ULEF	DTEF	HTF	TCF	Res	Res	Res	Res	Res	Res	Res		
	リセット値																		0	0	0	0	0	0	0									
0x60+0x80 × x (x=0 から 7)	GPDMA_CxSR	Res	Res	Res	Res	Res	Res	Res	Res	FIFOL[7:0]								Res	TOF	SUSPF	USEF	ULEF	DTEF	HTF	TCF	Res	Res	Res	Res	Res	Res	Res		
	リセット値									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0							IDLEF	
0x64+0x80 × x (x=0 から 7)	GPDMA_CxCRCR	Res	Res	Res	Res	Res	Res	Res	Res	PRIO[1:0]		Res	Res	Res	Res	Res	LAP	LSM	TOIE	SUSPIE	USEIE	ULEIE	DTEIE	HTIE	TCIE	Res	Res	Res	Res	SUSP	リセット	EN		
	リセット値									0	0						0	0	0	0	0	0	0	0	0					0	0	0		
0x68 から 0x8C+0x80 × x (x=0 から 7)	予約済み	予約済み																																
0x90+0x80 × x (x=0 から 7)	GPDMA_CxTR1	DSEC	DAP	Res	Res	DHX	DBX	DBL_1 [5:0]					DINC	Res	DDW_LOG2 [1:0]	SSEC	SAP	SBX	PAM[1:0]			Res	SBL_1 [5:0]					SINC	Res	SDW_LOG2 [1:0]				
	リセット値	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x94+0x80 × x (x=0 から 7)	GPDMA_CxTR2	TCM[1:0]		Res	Res	Res	Res	TRIGPOL[1:0]		Res	Res	TRIGSEL[5:0]					TRIGM[1:0]		Res	PFREQ	BREQ	DREQ	SWREQ	Res	REQSEL[7:0]									
	リセット値	0	0					0	0			0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	
0x98+0x80 × x (x=0 から 5)	GPDMA_CxBR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BNDT[15:0]																	
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x98+0x80 × x (x=6 から 7)	GPDMA_CxBR1	BRDDEC	BRSEDEC	DDEC	SDEC	BRC[10:0]												BNDT[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 134. GPDMA レジスタのマップとリセット値 (続き)

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x9C+0x80 × x (x=0 から 7)	GPDMA_CxSAR	SA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA0+0x80 × x (x=0 から 7)	GPDMA_CxDAR	DA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4+0x80 × x (x=6 から 7)	GPDMA_CxTR3	Res	Res	Res	DAO[12:0]												Res	Res	Res	SAO[12:0]													
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8+0x80 × x (x=6 から 7)	GPDMA_CxBR2	BRDAO[15:0]																BRSAO[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xCC+0x80 × x (x=0 から 5)	GPDMA_CxLLR	UT1	UT2	UB1	USA	UDA	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ULL	LA[15:2]												Res	Res		
	リセット値	0	0	0	0	0											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xCC+0x80 × x (x=6 から 7)	GPDMA_CxLLR	UT1	UT2	UB1	USA	UDA	UT3	UB2	Res	Res	Res	Res	Res	Res	Res	Res	ULL	LA[15:2]												Res	Res		
	リセット値	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

17 ネスト化されたベクタ割り込みコントローラ (NVIC)

17.1 NVIC の主な機能

- 131 のマスク可能な割り込みチャネル (16 本の FPU 割り込みラインを持つ Cortex-M33 は数に含まれていない)
- 16 のプログラム可能な優先レベル (4 ビットの割り込み優先順位を使用)
- 遅延時間の少ない例外および割り込み処理
- 電源管理制御
- システム制御レジスタの実装

NVIC とプロセッサ・コアのインタフェースを密に結合することで、割り込み処理の遅延を低減し、後着割り込みを効率的に処理することができます。

NVIC レジスタは、セキュアステートと非セキュアステートにまたがってバンク分けされています。

コア例外を含むすべての割り込みは、NVIC によって管理されます。

17.2 SysTick 較正值レジスタ

TrustZone メインライン・セキュリティ拡張機能を備えた Cortex-M33 には、2 つの SysTick タイマが組み込まれています。

TrustZone がアクティブなときは、次に示す SysTick タイマが使用可能です。

- SysTick、セキュア・インスタンス
- SysTick、非セキュア・インスタンス

TrustZone が無効なときは、1 つの SysTick タイマだけが使用可能です。

SysTick タイマの較正值 (STCALIB) は 0x3E8 です。1 MHz の SysTick クロック周波数に基づいて、1 ms の基準タイムベースが得られます。特定の周波数で動作しているアプリケーションの 1 ms タイムベースを一致させるには、SYSTick 再ロード値を SYST_RVR レジスタに次のようにプログラムする必要があります。

- SysTick クロック・ソースが CPU クロック HCLK の場合
再ロード値 = $(HCLK \times STCALIB) - 1$
- SysTick クロック・ソースが 外部 クロック (HCLK/8) の場合
再ロード値 = $((HCLK/8) \times STCALIB) - 1$

HCLK は MHz 単位の AHB 周波数値を表します。

例 : SysTick クロックソースが 100 MHz の CPU クロック HCLK の場合、1 ms のタイムベースを一致させるには :

$$\text{SysTick 再ロード値} = (100 \times STCALIB) - 1 = 0x1869F$$

17.3 割込みベクタと例外ベクタ

次の表の灰色の行は、特定の位置を持たないベクタを示しています。

表 135. STM32H563/H573および STM32H562 ベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-4	固定	SettableReset	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスカブル割込み。RCC クロックセキュリティシステム (CSS) は NMI ベクタにリンクされます。	0x0000 0008
-	-3 または -1	固定	Secure HardFault	セキュア・ハード・フォールト	0x0000 : 000C
-	-1	固定	Non-secure HardFault	非セキュア・ハード・フォールト。あらゆる種類の異常	0x0000 : 000C
-	0	設定可能	MemManage	メモリ管理	0x0000 0010
-	1	設定可能	BusFault	プリフェッチ異常、メモリアクセス異常	0x0000 0014
-	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
-	3	設定可能	SecureFault	セキュアフォールト	0x0000 : 001C
-	-	-	-	予約済み	0x0000 0020 - 0x0000 0028
-	4	-	SVC	SWI 命令によるシステムサービスコール	0x0000 : 002C
-	5	-	デバッグモニタ	監視	0x0000 0030
-	-	-	-	予約済み	0x0000 0034
-	6	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038
-	7	設定可能	SysTick	システムティックタイマ	0x0000 : 003C
0	8	設定可能	WWDG	ウィンドウ型ウォッチドッグ割込み	0x0000 0040
1	9	設定可能	PVD_AVD	電源電圧モニタ／アナログ電圧モニタ	0x0000 0044
2	10	設定可能	RTC	RTC 非セキュア・グローバル割込み	0x0000 0048
3	11	設定可能	RTC_S	RTC セキュア・グローバル割込み	0x0000 : 004C
4	12	設定可能	TAMP	タンパ・グローバル割込み	0x0000 0050
5	13	設定可能	RAMCFG	RAM 設定グローバル割込み	0x0000 0054
6	14	設定可能	FLASH	Flash 非セキュア・グローバル割込み	0x0000 0058
7	15	設定可能	Flash_S	Flash セキュア・グローバル割込み	0x0000 : 005C
8	16	設定可能	GTZC	GTZC グローバル割込み	0x0000 0060
9	17	設定可能	RCC	RCC 非セキュア・グローバル割込み	0x0000 0064
10	18	設定可能	RCC_S	RCC セキュアグローバル割込み	0x0000 0068
11	19	設定可能	EXTI0	EXTI ライン 0 割込み	0x0000 : 006C
12	20	設定可能	EXTI1	EXTI ライン 1 割込み	0x0000 0070
13	21	設定可能	EXTI2	EXTI ライン 2 割込み	0x0000 0074
14	22	設定可能	EXTI3	EXTI ライン 3 割込み	0x0000 0078

表 135. STM32H563/H573および STM32H562 ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
15	23	設定可能	EXTI4	EXTI ライン 4 割り込み	0x0000 : 007C
16	24	設定可能	EXTI5	EXTI ライン 5 割り込み	0x0000 0080
17	25	設定可能	EXTI6	EXTI ライン 6 割り込み	0x0000 0084
18	26	設定可能	EXTI7	EXTI ライン 7 割り込み	0x0000 0088
19	27	設定可能	EXTI8	EXTI ライン 8 割り込み	0x0000 : 008C
20	28	設定可能	EXTI9	EXTI ライン 9 割り込み	0x0000 0090
21	29	設定可能	EXTI10	EXTI ライン 10 割り込み	0x0000 0094
22	30	設定可能	EXTI11	EXTI ライン 11 割り込み	0x0000 0098
23	31	設定可能	EXTI12	EXTI ライン 12 割り込み	0x0000 : 009C
24	32	設定可能	EXTI13	EXTI ライン 13 割り込み	0x0000 : 00E4
25	33	設定可能	EXTI14	EXTI ライン 14 割り込み	0x0000 : 00A0
26	34	設定可能	EXTI15	EXTI ライン 15 割り込み	0x0000 : 00A4
27	35	設定可能	GPDMA1_CH0	GPDMA1 チャンネル 0 グローバル割り込み	0x0000 : 00A8
28	36	設定可能	GPDMA1_CH1	GPDMA1 チャンネル 1 グローバル割り込み	0x0000 : 00AC
29	37	設定可能	GPDMA1_CH2	GPDMA1 チャンネル 2 グローバル割り込み	0x0000 : 00B0
30	38	設定可能	GPDMA1_CH3	GPDMA1 チャンネル 3 グローバル割り込み	0x0000 : 00B4
31	39	設定可能	GPDMA1_CH4	GPDMA1 チャンネル 4 グローバル割り込み	0x0000 : 00B8
32	40	設定可能	GPDMA1_CH5	GPDMA1 チャンネル 5 グローバル割り込み	0x0000 : 00C0
33	41	設定可能	GPDMA1_CH6	GPDMA1 チャンネル 6 グローバル割り込み	0x0000 : 00C4
34	42	設定可能	GPDMA1_CH7	GPDMA1 チャンネル 7 グローバル割り込み	0x0000 : 00C8
35	43	設定可能	IWDG	独立型ウォッチドッグ割り込み	0x0000 : 00CC
36	44	設定可能	SAES	セキュアAES	0x0000 : 00D0
37	45	設定可能	ADC1	ADC1 グローバル割り込み	0x0000 : 00D4
38	46	設定可能	DAC1	DAC1 グローバル割り込み	0x0000 : 00D8
39	47	設定可能	FDCAN1_IT0	FDCAN1 割り込み 0	0x0000 : 00DC
40	48	設定可能	FDCAN1_IT1	FDCAN1 割り込み 1	0x0000 : 00E0
41	49	設定可能	TIM1_BRK/TIM1_TERR/TIM1_IERR	TIM1 ブレーク/TIM1 遷移エラー/TIM1 インデックス・エラー	0x0000 : 00E4
42	50	設定可能	TIM1_UP	TIM1 更新	0x0000 : 00E8
43	51	設定可能	TIM1_TRG_COM/TIM1_DIR/TIM1_IDX	TIM1 トリガおよび転流/TIM1 方向変更割り込み/TIM1 インデックス	0x0000 : 00EC
44	52	設定可能	TIM1_CC	TIM1 キャプチャ/比較割り込み	0x0000 : 00F0
45	53	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 : 00F4
46	54	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 : 00F8
47	55	設定可能	TIM4	TIM4 グローバル割り込み	0x0000 : 00FC
48	56	設定可能	TIM5	TIM5 グローバル割り込み	0x0000 0100
49	57	設定可能	TIM6	TIM6 グローバル割り込み	0x0000 0104
50	58	設定可能	TIM7	TIM7 グローバル割り込み	0x0000 0108

表 135. STM32H563/H573および STM32H562 ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
51	59	設定可能	I2C1_EV	I2C1 イベント割り込み	0x0000 : 010C
52	60	設定可能	I2C1_ER	I2C1 エラー割り込み	0x0000 0110
53	61	設定可能	I2C2_EV	I2C2 イベント割り込み	0x0000 0114
54	62	設定可能	I2C2_ER	I2C2 エラー割り込み	0x0000 0118
55	63	設定可能	SPI1	SPI1 グローバル割り込み	0x0000 : 011C
56	64	設定可能	SPI2	SPI2 グローバル割り込み	0x0000 0120
57	65	設定可能	SPI3	SPI3 グローバル割り込み	0x0000 0124
58	66	設定可能	USART1	USART1 グローバル割り込み	0x0000 0128
59	67	設定可能	USART2	USART2 グローバル割り込み	0x0000 : 012C
60	68	設定可能	USART3	USART3 グローバル割り込み	0x0000 0130
61	69	設定可能	UART4	UART4 グローバル割り込み	0x0000 0134
62	70	設定可能	UART5	UART5 グローバル割り込み	0x0000 0138
63	71	設定可能	LPUART1	LPUART1 グローバル割り込み、または LPUART1 R ウェイクアップ、あるいは LPUART1 T ウェイクアップ (EXTI ライン経由)	0x0000 : 013C
64	72	設定可能	LPTIM1 または LPTIM1_AIT	LPTIM1 グローバル割り込み、または EXTI ラ インを経由した LPTimer1 AIT	0x0000 0140
65	73	設定可能	TIM8_BRK/TIM8_TERR/TIM 8_IERR	TIM8 ブレーク割り込み/TIM8 遷移エラー /TIM8 インデックス・エラー	0x0000 0144
66	74	設定可能	TIM8_UP	TIM8 更新割り込み	0x0000 0148
67	75	設定可能	TIM8_TRG_COM/TIM8_DIR /TIM8_IDX	TIM8 トリガおよび転流割り込み/TIM8 方 向変更割り込み/TIM8 インデックス	0x0000 : 014C
68	76	設定可能	TIM8_CC	TIM8 キャプチャ/比較割り込み	0x0000 0150
69	77	設定可能	ADC2	ADC2 グローバル割り込み	0x0000 0154
70	78	設定可能	LPTIM2 または LPTIM2_AIT	LPTIM2 グローバル割り込み、または EXTI ラ インを経由した LPTimer2 AIT	0x0000 0158
71	79	設定可能	TIM15	TIM15 グローバル割り込み	0x0000 : 015C
72	80	設定可能	TIM16	TIM16 グローバル割り込み	0x0000 0160
73	81	設定可能	TIM17	TIM17 グローバル割り込み	0x0000 0164
74	82	設定可能	USB_FS	USB FS グローバル割り込み	0x0000 0168
75	83	設定可能	CRS	クロックリカバリシステムグローバル割 込み	0x0000 : 016C
76	84	設定可能	UCPD1	UCPD1 グローバル割り込み	0x0000 0170
77	85	設定可能	FMC	FMC グローバル割り込み	0x0000 0174
78	86	設定可能	OCTOSPI1	OCTOSPI1 グローバル割り込み	0x0000 0178
79	87	設定可能	SDMMC1	SDMMC1 グローバル割り込み	0x0000 : 017C
80	88	設定可能	I2C3_EV	I2C3 イベント割り込み	0x0000 0180
81	89	設定可能	I2C3_ER	I2C3 エラー割り込み	0x0000 0184

表 135. STM32H563/H573および STM32H562 ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
82	90	設定可能	SPI4	SPI4 グローバル割り込み	0x0000 0188
83	91	設定可能	SPI5	SPI5 グローバル割り込み	0x0000 0174
84	92	設定可能	SPI6	SPI6 グローバル割り込み	0x0000 0178
85	93	設定可能	USART6	USART6 グローバル割り込み	0x0000 : 017C
86	94	設定可能	USART10	USART10 グローバル割り込み	0x0000 0180
87	95	設定可能	USART11	USART11 グローバル割り込み	0x0000 0184
88	96	設定可能	SAI1	SAI1 グローバル割り込み	0x0000 0188
89	97	設定可能	SAI2	SAI2 グローバル割り込み	0x0000 : 018C
90	98	設定可能	GPDMA2_CH0	GPDMA2 チャンネル 0 グローバル割り込み	0x0000 0190
91	99	設定可能	GPDMA2_CH1	GPDMA2 チャンネル 1 グローバル割り込み	0x0000 0194
92	100	設定可能	GPDMA2_CH2	GPDMA2 チャンネル 2 グローバル割り込み	0x0000 0198
93	101	設定可能	GPDMA2_CH3	GPDMA2 チャンネル 3 グローバル割り込み	0x0000 : 019C
94	102	設定可能	GPDMA2_CH4	GPDMA2 チャンネル 4 グローバル割り込み	0x0000 : 01A0
95	103	設定可能	GPDMA2_CH5	GPDMA2 チャンネル 5 グローバル割り込み	0x0000 : 01A4
96	104	設定可能	GPDMA2_CH6	GPDMA2 チャンネル 6 グローバル割り込み	0x0000 : 01A8
97	105	設定可能	GPDMA2_CH7	GPDMA2 チャンネル 7 グローバル割り込み	0x0000 : 01AC
98	106	設定可能	UART7	UART7 グローバル割り込み	0x0000 : 01B0
99	107	設定可能	UART8	UART8 グローバル割り込み	0x0000 : 01B4
100	108	設定可能	UART9	UART9 グローバル割り込み	0x0000 : 01B8
101	109	設定可能	UART12	UART12 グローバル割り込み	0x0000 : 01BC
102	110	設定可能	SDMMC2	SDMMC2 グローバル割り込み	0x0000 : 01C0
103	111	設定可能	FPU	浮動小数点割り込み	0x0000 : 01C4
104	112	設定可能	ICACHE	命令キャッシュグローバル割り込み	0x0000 : 01C8
105	113	設定可能	DCACHE	データ・キャッシュ・グローバル割り込み	0x0000 : 01CC
106	114	設定可能	ETH	Ethernet 割り込み	0x0000 : 01D0
107	115	設定可能	ETH_WKUP	Ethernet ウェイクアップ割り込み (EXTI ライン経由)	0x0000 : 01D4
108	116	設定可能	DCMI_PSSI	DCMI/PSSI グローバル割り込み	0x0000 : 01D8
109	117	設定可能	FDCAN2_IT0	FDCAN2 割り込み 0	0x0000 : 01DC
110	118	設定可能	FDCAN2_IT1	FDCAN2 割り込み 1	0x0000 : 01E0
111	119	設定可能	CORDIC	CORDIC 割り込み	0x0000 : 01E4
112	120	設定可能	FMAC	FMAC 割り込み	0x0000 : 01E8
113	121	設定可能	DTS OR DTS_WKUP	DTS 割り込み、または EXTI ラインを経由した DTS AIT	0x0000 : 01EC
114	122	設定可能	RNG	RNG グローバル割り込み	0x0000 : 01F0
115	123	設定可能	OTFDEC1	OTFDEC1 セキュアグローバル割り込み	0x0000 : 01F4
116	124	設定可能	AES	AES グローバル割り込み	0x0000 : 01F8
117	125	設定可能	HASH	HASH 割り込み	0x0000 : 01FC

表 135. STM32H563/H573および STM32H562 ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
118	126	設定可能	PKA	PKA グローバル割り込み	0x0000 0200
119	127	設定可能	CEC	HDMI-CEC グローバル割り込み	0x0000 0204
120	128	設定可能	TIM12	TIM12 グローバル割り込み	0x0000 0208
121	129	設定可能	TIM13	TIM13 グローバル割り込み	0x0000 : 020C
122	130	設定可能	TIM14	TIM14 グローバル割り込み	0x0000 0210
123	131	設定可能	I3C1_EV	I3C1 イベント割り込み	0x0000 0214
124	132	設定可能	I3C1_ER	I3C1 エラー割り込み	0x0000 0218
125	133	設定可能	I2C4_EV	I2C4 イベント割り込み	0x0000 : 021C
126	134	設定可能	I2C4_ER	I2C4 エラー割り込み	0x0000 0220
127	135	設定可能	LPTIM3 または LPTIM3_AIT	LPTIM3 グローバル割り込み、または EXTI ラインを経由した LPTimer3 AIT	0x0000 0224
128	136	設定可能	LPTIM4 または LPTIM4_AIT	LPTIM4 グローバル割り込み、または EXTI ラインを経由した LPTimer4 AIT	0x0000 0228
129	137	設定可能	LPTIM5 または LPTIM5_AIT	LPTIM5 グローバル割り込み、または EXTI ラインを経由した LPTimer5 AIT	0x0000 : 022C
130	138	設定可能	LPTIM6 または LPTIM6_AIT	LPTIM6 グローバル割り込み、または EXTI ラインを経由した LPTimer6 AIT	0x0000 0230

18 拡張割込み／イベントコントローラ (EXTI)

拡張割込み／イベントコントローラ (EXTI) は、設定可能なイベントおよびダイレクトイベントの入力によって個々の CPU およびシステムのウェイクアップを管理します。電源制御にウェイクアップ・リクエストを供給し、CPU の NVIC に割込みリクエストを生成し、CPU イベント入力へのイベントを生成します。CPU イベント信号を生成するには、CPU は追加のイベント生成ブロック (EVG) が必要になります。

EXTI ウェイクアップリクエストにより、システムは STOP モードからウェイクアップできます。

RUN モードでも、割込みリクエストとイベントリクエストの生成を使用できます。

EXTI には EXTI マルチプレクサの IO ポートの選択も含まれています。

18.1 EXTI の主な機能

EXTI の主な機能は以下のとおりです。

- 58 の入力イベントをサポート
- すべてのイベント入力はシステムをウェイクアップできる可能性があります。
- ペリフェラルに関連付けられたウェイクアップ・フラグがないイベントは、EXTI にフラグを持ち、EXTI から CPU への割込みを生成します。
- CPU ウェイクアップイベントを生成するためにイベントを使用できます。

非同期イベント入力は下記の 2 つのグループに分類されます。

- 設定可能なイベント (I/O またはペリフェラルからパルスを生成できる信号) には、以下の特徴があります
 - 選択可能なアクティブトリガエッジ
 - 立上がりおよび立下がりエッジに依存しない割込みペンディング・ステータス・レジスタ・ビット
 - CPU のウェイクアップ、割込み、およびイベント生成を調整するために使用される個々の割込みおよびイベント生成マスク
 - ソフトウェア・トリガの可能
 - セキュア・イベント：セキュア入力イベントの制御および設定ビットへのアクセスは、セキュアおよび／または特権で行うことができます。
 - EXTI の IO ポートの選択
- ダイレクト・イベント (ペリフェラルでクリアする必要がある関連フラグを持つペリフェラルからの割込みおよびウェイクアップ・ソース) は、次の機能を持ちます。
 - 固定立ち上がりエッジアクティブトリガ
 - EXTI に割込みペンディング・ステータス・レジスタ・ビットなし (割込みペンディング・ステータス・フラグは、イベントを生成するペリフェラルから供給される)
 - CPU のウェイクアップ、およびイベント生成を調整するために使用される個々の割込みおよびイベント生成マスク
 - ソフトウェア・トリガの機能無し

18.2 EXTI ブロック図

図 87 に示すように、EXTI は AHB インタフェースを介してアクセスするレジスタ・ブロック、イベント入力トリガ・ブロック、マスキング・ブロック、および EXTI マルチプレクスから成ります。

レジスタブロックは、すべての EXTI レジスタを含んでいます。

イベント入力トリガブロックは、イベント入力エッジのトリガロジックを供給します。

マスキングブロックは、さまざまなウェイクアップ、割込み、およびイベント出力に、イベント入力
とそれらのマスキングを供給します。

EXTI マルチプレクサは、EXTI イベント信号が IO ポートを選択できるようにします。

図 87. EXTI ブロック図

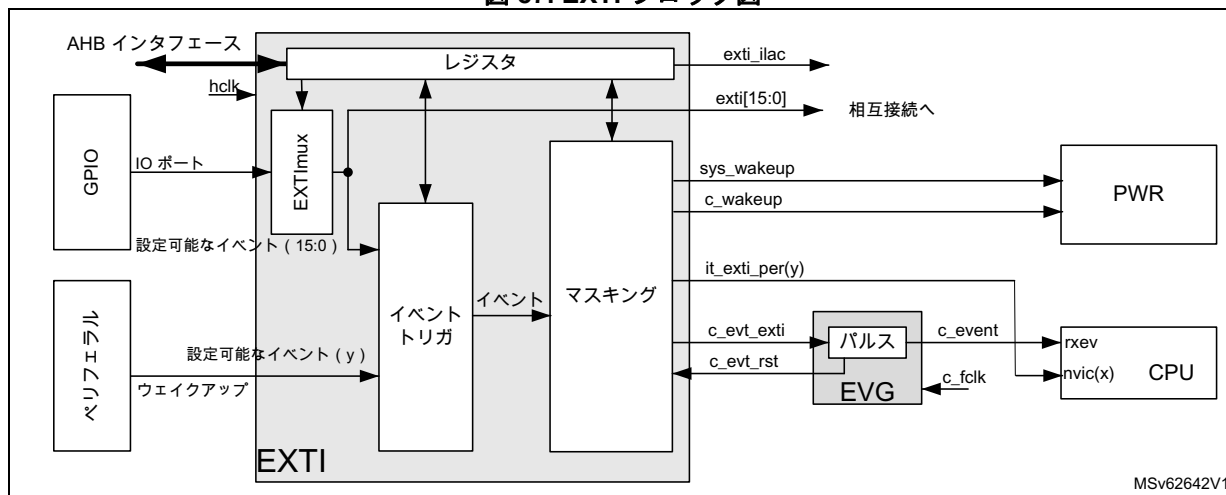


表 136. EXTI 信号

ピン名	I/O	説明
AHB インタフェース	I/O	EXTI レジスタのバスインタフェース。1 つのイベントがセキュリティを有効にするように設定されている場合、AHB インタフェースはセキュアなアクセスをサポートします。
hclk	┃	AHB バスクロックおよび EXTI システムクロック
設定可能なイベント (y)	┃	関連した割込みやフラグの無い、ペリフェラルからの非同期ウェイクアップ・イベント
ダイレクトイベント (x)	┃	関連した割込みやフラグの有る、ペリフェラルからの同期および非同期ウェイクアップ・イベント
exti_ilac	O	不正アクセスイベント
IOPort (n)	┃	GPIO のブロック IO ポート [15:0]
exti[15:0]	O	他のペリフェラルをトリガするための EXTI GPIO 出力ポート
it_exti_per (y)	O	設定可能なイベント(y) に伴う CPU への割込み
c_evt_exti	O	CPU への、ハイレベルに反応するイベント出力、hclk と同期
c_evt_rst	┃	c_evt_exti をクリアする非同期リセット入力
sys_wakeup	O	ck_sys および hclk 用の PWR への非同期システムウェイクアップリクエスト
c_wakeup	O	CPU 用 PWR へのウェイクアップリクエスト、hclk と同期

表 137. EVG 信号

ピン名	I/O	説明
c_fclk	I	CPU フリー・ランニング・クロック
c_evt_in	I	CPU クロックに非同期の、EXTI からのハイレベル対応イベント入力
c_event	O	CPU クロックに同期したイベントパルス
c_evt_rst	O	CPU クロックに同期したイベントリセット信号

18.2.1 ペリフェラルと CPU 間の EXTI 接続

システムの STOP モード時にウェイクアップまたは割り込みイベントを生成できるペリフェラルを EXTI に接続します。

- パルスを生成する、またはペリフェラルに割り込みステータス・ビットがないペリフェラルのウェイクアップ信号は、EXTI の設定可能なイベント入力に接続されます。これらのイベントに、EXTI はクリアすべきステータス・ペンディング・ビットを供給します。CPU に割り込むのはステータス・ビットに関連した EXTI 割り込みです。
- クリアすべきペリフェラルのステータス・ビットを持つペリフェラルの割り込みやウェイクアップ信号は、EXTI のダイレクト・イベント入力に接続されます。EXTI にはステータス・ペンディング・ビットはありません。割り込みまたはウェイクアップは、ペリフェラルの CPU によってクリアされます。CPU に直接割り込むのはペリフェラル割り込みです。

すべての GPIO ポートは EXTI マルチプレクサに入力されるため、ポートピンを選択して設定可能なイベントによってシステムをウェイクアップすることができます。

EXTI の設定可能なイベント割り込みは、NVIC に接続されています。

専用の EXTI/EVG CPU イベントを、CPU rxev 入力に接続します。

EXTI の CPU のウェイクアップ信号を PWR に接続し、システムおよび CPU サブシステム・バスのクロックをウェイクアップするために使用します。

18.2.2 EXTI 割り込み／イベントのマッピング

EXTI ラインは下の表に示すように接続されています。

表 138. EXTI ラインの接続

EXTI ライン	ラインソース	ラインタイプ
0~15	GPIO	設定可能
16	PVD/AVD 出力	設定可能
17	RTC 非セキュア	-
18	RTC セキュア	-
19	TAMP 非セキュア	-
20	TAMP セキュア	-
21	I2C1 ウェイクアップ	-
22	I2C2 ウェイクアップ	-
23	I2C3 ウェイクアップ	-
24	I3C ウェイクアップ	-
25	USART1 ウェイクアップ	-

表 138. EXTI ラインの接続 (続き)

EXTI ライン	ラインソース	ラインタイプ
26	USART2 ウェイクアップ	-
27	USART3 ウェイクアップ	-
28	UART4 ウェイクアップ	-
29	UART5 ウェイクアップ	-
30	USART6 ウェイクアップ	-
31	UART7 ウェイクアップ	-
32	UART8 ウェイクアップ	-
33	UART9 ウェイクアップ	-
34	USART10 ウェイクアップ	-
35	USART11 ウェイクアップ	-
36	UART12 ウェイクアップ	-
37	LPUART1 ウェイクアップ	-
38	LPTIM1	-
39	LPTIM2	-
40	SPI1 ウェイクアップ	-
41	SPI2 ウェイクアップ	-
42	SPI3 ウェイクアップ	-
43	SPI4 ウェイクアップ	-
44	SPI5 ウェイクアップ	-
45	SPI6 ウェイクアップ	-
46	ETH ウェイクアップ	設定可能
47	USB FS ウェイクアップ	-
48	USBPD1 ウェイクアップ	-
49	LPTIM2 : CH1	-
50	DTS ウェイクアップ	設定可能
51	HDMI-CEC ウェイクアップ	-
52	I2C4 ウェイクアップ	-
53	UVM 出力	設定可能
54	LPTIM3	-
55	LPTIM4	-
56	LPTIM5	-
57	LPTIM6	-

18.3 EXTI の機能説明

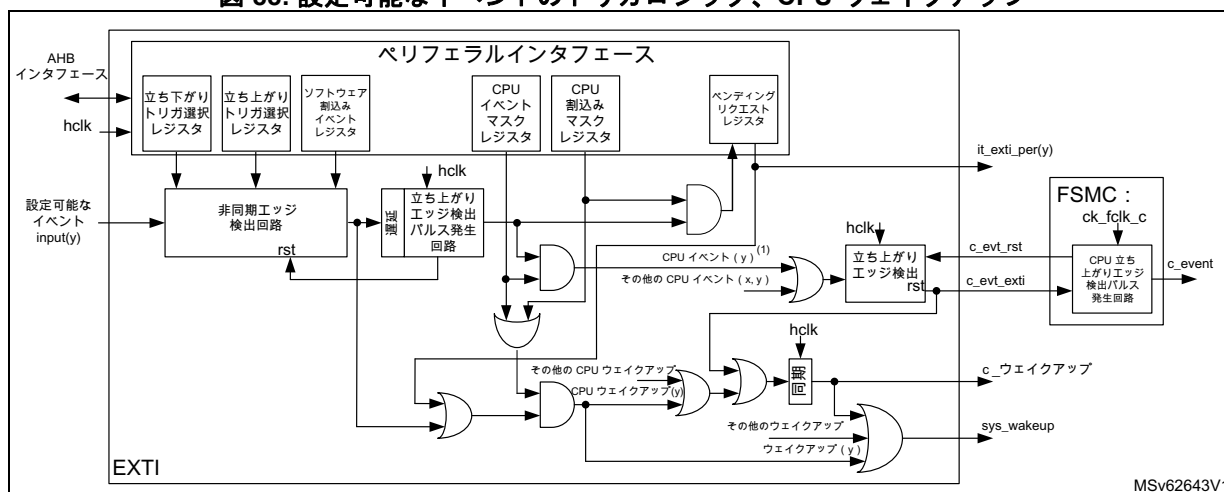
イベントの機能は、次のとおりレジスタ・ビットで制御されます。

- アクティブ・トリガ・エッジの有効化
 - EXTI 立上りトリガ選択レジスタ (EXTI_RTSR1) および EXTI 立上りトリガ選択レジスタ 2 (EXTI_RTSR2) で立上がりエッジを選択
 - EXTI 立下りトリガ選択レジスタ (EXTI_FTSR1) および EXTI 立下りトリガ選択レジスタ 2 (EXTI_FTSR2) で立下がりエッジを選択
- EXTI ソフトウェア割込みイベントレジスタ (EXTI_SWIER1) および EXTI ソフトウェア割込みイベント・レジスタ 2 (EXTI_SWIER2) でソフトウェア・トリガ
- 割込みペンディング・フラグ
 - EXTI 立上りエッジペンディングレジスタ (EXTI_RPR1) および EXTI 立上りエッジ・ペンディング・レジスタ 2 (EXTI_RPR2)
 - EXTI 立下りエッジペンディングレジスタ 1 (EXTI_FPR1) および EXTI 立下りエッジ・ペンディング・レジスタ 2 (EXTI_FPR2)
- CPU ウェイクアップと割込みの有効化
 - 割込みマスクレジスタによる EXTI CPU ウェイクアップレジスタ 1 (EXTI_IMR1) および 割込みマスク・レジスタ 2 による EXTI CPU ウェイクアップ (EXTI_IMR2)
- CPU ウェイクアップおよびイベントの有効化
 - イベントマスクレジスタによる EXTI CPU ウェイクアップレジスタ 1 (EXTI_EMR1) および イベント・マスク・レジスタ 2 による EXTI CPU ウェイクアップ (EXTI_EMR2)

18.3.1 EXTI の設定可能なイベント入力ウェイクアップ

下の図は、CPU サブシステム・バス・クロックをウェイクアップさせ、EXTI ペンディング・フラグを生成し、また CPU や CPU ウェイクアップ・イベントに割込みをかける、設定可能なイベント入力に関連したロジックを詳しく表しています。

図 88. 設定可能なイベントのトリガロジック、CPU ウェイクアップ



1. CPU の rxeve を生成する c_event をサポートしている入カイベントに対してのみ。

対応するレジスタ・ビットを書き込むことで、エッジ選択の設定に関係なく、ソフトウェア割込みイベント・レジスタにより、設定可能なイベントをソフトウェアでトリガできます。

設定可能なイベント・アクティブ・トリガ・エッジ（あるいは両方のエッジ）を選択して、立上がり／立下がりエッジ選択レジスタで有効にします。

CPU はそれ専用のウェイクアップ（割込み）マスク・レジスタと専用のイベント・マスク・レジスタを備えています。イベントが有効になると、これが CPU に生成されます。CPU のすべてのイベントの論理和がとられ、ひとつの CPU イベント信号に集約されます。CPU のイベント・ペンディング・レジスタ（EXTI_RPR および EXTI_FPR）は、マスクされていない CPU イベントにはセットされません。

設定可能なイベントには、固有の割込みペンディング・リクエスト・レジスタが割り当てられます。ペンディングレジスタは、マスクされていない割込みについてのみセットされます。各設定イベントは、CPU に対しそれらに共通した割込みをかけます。設定可能なイベント割込みは、EXTI_RPR および EXTI_FPR はレジスタでソフトウェアにより確認しなくてはなりません。

CPU ウェイクアップ（割込み）または CPU イベントが有効化されると、非同期エッジ検出回路は、クロック遅延および立上がりエッジ検出パルス発生回路によってリセットされます。これにより、EXTI hclk クロックは、非同期エッジ検出回路がリセットされる前にウェイクアップされることが保証されます。

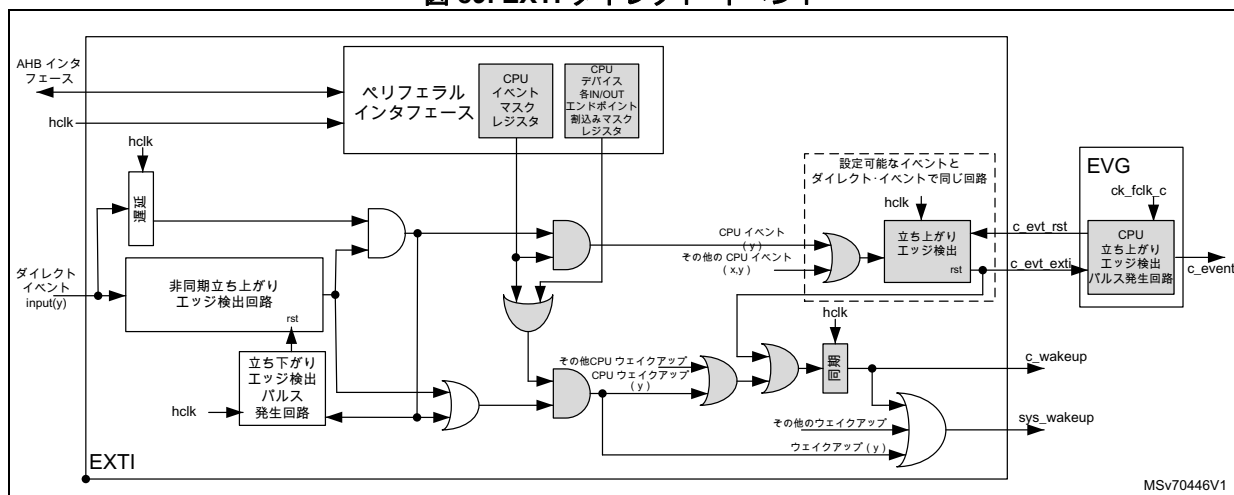
注： 検出された設定可能なイベント割込みペンディングリクエストは正しいアクセス許可を有する CPU によってクリアされることがあります。割込みペンディングリクエストがアクティブである限り、システムは低電力モードに入ることができません。

18.3.2 EXTI のダイレクトイベント入力、ウェイクアップ

ダイレクトイベントには関連する EXTI 割込みがありません。EXTI はシステム・クロックと CPU サブシステム・クロックをウェイクアップするだけで、CPU ウェイクアップ・イベントを生成することができます。ダイレクト・ウェイクアップ・イベントに関連するペリフェラル同期割込みは、CPU をウェイクアップします。

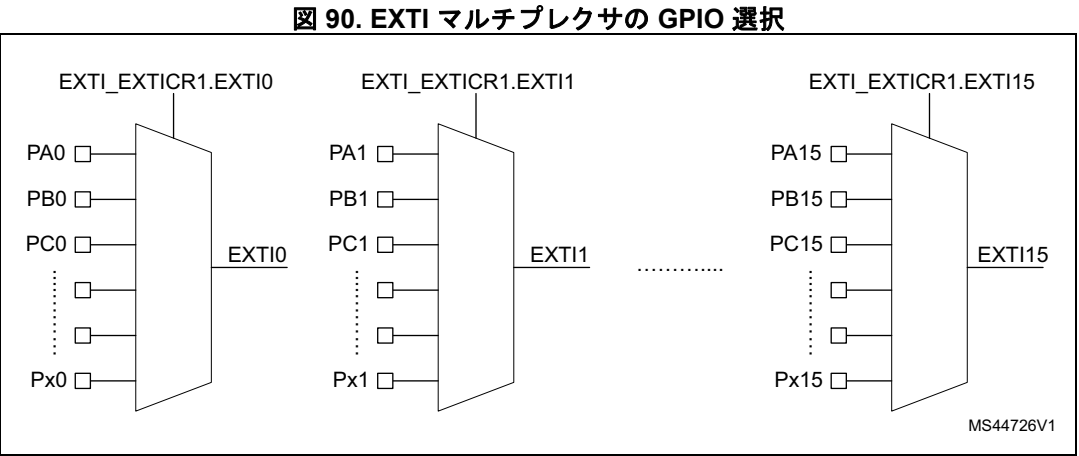
EXTI のダイレクト・イベントは CPU をウェイクアップする CPU イベントを生成できます。CPU イベントは、関連するペリフェラルの割込みフラグがセットされる前に、発生することがあります。

図 89. EXTI ダイレクト・イベント



18.3.3 EXTI マルチプレクサの選択

EXTI マルチプレクサにより、割り込みおよびウェイクアップとして GPIO の選択ができます。GPIOは、16 EXTI マルチプレクサ・ラインを介し設定可能なイベントとして、最初の 16 EXTI イベントに接続されます。EXTI マルチプレクサ出力としての GPIO ポートの選択は、[EXTI 外部割り込み選択レジスタ 1（EXTI_EXTICR1）](#)により制御されます。



EXTI マルチプレクサ出力は、EXTI からの出力信号として、EXTI_IMR や EXTI_EMR レジスタのマスキングに関係なく、他のペリフェラルをトリガするために利用可能です。

18.4 EXTI の機能的挙動

設定可能なイベントは、少なくとも 1 つのトリガエッジを有効にすることで有効になります。

イベント入力が有効になると、CPU ウェイクアップの生成は CPU 割り込みマスクおよび CPU イベント・マスクの条件により決定されます。

表 139. マスキングの機能

CPU 割り込み有効 (EXTI_IMR.IMn において)	CPU イベント有効 (EXTI_EMR.EMn において)	設定可能なイベント入力 (EXTI_RPR.RPIFn および EXTI_FPR.FPIFn において)	Exti (n) 割り込み ⁽¹⁾	CPU イベント	CPU ウェイクアップ
0	0	不可	マスク	マスク	マスク
	1	不可	マスク	はい	はい
1	0	ステータスラッチ	はい	マスク	はい ⁽²⁾
	1	ステータスラッチ	はい	はい	はい

1. ひとつの exti(n) 割り込みが CPU に送信されます。CPU(m) に割り込みが必要ない場合、exti(n) 割り込みは CPU NVIC 内でマスクしなければなりません。

2. CPU 割り込みが EXTI_IMR.IMn で有効化されている場合のみです。

設定可能なイベント入力の場合、イベント入力時にイネーブル・エッジが発生すると、イベント・リクエストが生成されます。関連する CPU 割り込みがマスクされていない場合、対応するペンディング・ビット EXTI_RPR.RPIFn および／または EXTI_FPR.FPIFn がセットされ、CPU サブシステムがウェイクアップされ、CPU 割り込み信号がアクティブになります。EXTI_RPR.RPIFn および／または

EXTI_FPR.FPIF_n ペンディング・ビットは、ソフトウェアで 1 を書き込んで、クリアする必要があります。これによって CPU 割込みもクリアされます。

設定可能なイベント入力の場合、ソフトウェア割込み／イベント・レジスタ EXTI_SWIER、に 1 を書き込むと（これによってイベント時に立上がりエッジが生成されます）、ソフトウェアによってイベント・リクエストを生成することができます。立上がりエッジ・イベント・ペンディング・ビットは、EXTI_RTISR の設定にかかわらず、EXTI_RPR にセットされます。

18.5 EXTI イベント保護

EXTI は、イベント・レジスタ・ビットが非セキュアおよび非特権のアクセスによって変更されないように保護することができます。保護は、入力イベントごとに EXTI_SECCFGR および EXTI_PRIVCFGR のレジスタ・ビットによって個別にアクティブになります。EXTI レベルでは、この保護は次の非認証書込みアクセスを回避する形になっています。

- セキュアおよび／または特権の設定可能なイベントの設定の変更。
- セキュアおよび／または特権の入力イベントのマスクの変更。
- セキュアおよび／または特権の入力イベントのペンディング・ステータスのクリア。

表 140. レジスタ保護の概要

レジスタ名	アクセスタイプ	保護 ⁽¹⁾⁽²⁾
EXTI_RTISR	RW	セキュリティおよび特権は、EXTI_SECCFGR および EXTI_PRIVCFGR でビット単位に有効にできます。
EXTI_FTISR	RW	
EXTI_SWIER	RW	
EXTI_RPR	RW	
EXTI_FPR	RW	
EXTI_SECCFGR	RW	常にセキュア。特権は EXTI_PRIVCFGR でビット単位で有効にできます。
EXTI_PRIVCFGR	RW	常に特権。セキュリティは EXTI_SECCFGR でビット単位で有効にできます。
EXTI_EXTICR _n	RW	セキュリティおよび特権は、EXTI_SECCFGR および EXTI_PRIVCFGR でビット単位に有効にできます。
EXTI_LOCKR	RW	常にセキュア。
EXTI_IM	RW	セキュリティおよび特権は、EXTI_SECCFGR および EXTI_PRIVCFGR でビット単位に有効にできます。
EXTI_EMR	RW	

1. セキュリティは個々の入力イベントで有効になります (EXTI_SECCFGR レジスタ)。

2. 特権は個々の入力イベントで有効になります (EXTI_PRIVCFGR レジスタ)。

18.5.1 EXTI セキュリティ保護

入カイベントに対してセキュリティが有効になっている場合、関連する入カイベントの設定ビットおよび制御ビットはセキュア・アクセスによってのみ変更および読出しが可能です。非セキュア書込みアクセスは破棄され、読出しには 0 が返されます。

入カイベントが非セキュアな場合、セキュリティは無効になります。関連する入カイベントの設定ビットおよび制御ビットはセキュアアクセスおよび非セキュアアクセスによって変更および読出しができます。

レジスタ EXTI_SECCFGR のセキュリティ設定は、EXTI_LOCKR.LOCK によってリセット後にグローバルにロックすることができます。

18.5.2 EXTI 特権保護

入カイベントに対して特権が有効になっている場合、関連する入カイベントの設定ビットおよび制御ビットは特権アクセスによってのみ変更および読出しが可能です。非特権書込みアクセスは破棄され、読出しには 0 が返されます。

入カイベントが非特権の場合、特権は無効になります。関連する入カイベントの設定ビットおよび制御ビットは特権アクセスおよび非特権アクセスによって変更および読出しができます。

レジスタ EXTI_PRIVCFGR の特権設定は、EXTI_LOCKR.LOCK によってリセット後にグローバルにロックすることができます。

18.6 EXTI レジスタ

EXTI レジスタマップは、以下のセクションに分けられています。

表 141. EXTI レジスタマップセクション

アドレスオフセット	説明
0x000 - 0x01C	一般の設定可能なイベント [31:0] の設定
0x020 - 0x03C	一般の設定可能なイベント [57:32] の設定
0x060 - 0x06C	EXTI IO ポートのマルチプレクサ選択
0x070	EXTI 保護ロック設定
0x080 - 0x0BC	CPU 入力イベント設定

すべてのレジスタには、ワード (32 ビット)、ハーフワード (16 ビット)、およびバイト (8 ビット) 単位でアクセスできます。

18.6.1 EXTI 立上りトリガ選択レジスタ 1 (EXTI_RTSR1)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RT16
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:0 **RTx** : 設定可能なイベント入力 $x^{(1)}$ ($x = 16 \sim 0$) の立上りトリガ・イベント設定ビット

EXTI_SECCFGR.SECx が無効のとき、RTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RTx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、RTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、RTx は特権アクセスによってのみアクセスできます。このビット x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立上りトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立上りトリガは有効です。

- 設定可能なイベント入力はエッジトリガであるため、これらの入力でグリッチが生成されないようにする必要があります。レジスタへの書込み中に設定可能なイベント入力で立上りエッジが発生した場合、関連するペンディングビットはセットされません。同じ設定可能なイベント入力に対して、立上りエッジトリガと立下りエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

18.6.2 EXTI 立下りトリガ選択レジスタ 1 (EXTI_FTSR1)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT16
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:0 **FTx** : 設定可能なイベント入力 x⁽¹⁾ (x = 16~0) の立下がりトリガ・イベント設定ビット

EXTI_SECCFGR.SECx が無効のとき、FTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FTx はセキュア・アクセスによってのみアクセスできます。この FTx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、FTx は特権アクセスによってのみアクセスできます。この FTx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立下がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立下がりトリガは有効です。

1. 設定可能なイベント入力エッジトリガであるため、これらの入力でグリッチが生成されないようにする必要があります。レジスタへの書込み中に設定可能なイベント入力立下がりエッジが発生した場合、関連するペンディングビットはセットされません。同じ設定可能なイベント入力に対して、立上がりエッジトリガと立下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

18.6.3 EXTI ソフトウェア割込みイベントレジスタ 1 (EXTI_SWIER1)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI16
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:0 **SWIx** : イベント x (x = 16~0) のソフトウェア割込み

EXTI_SECCFGR.SECx が無効のとき、SWIx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、SWIx はセキュア・アクセスによってのみアクセスできます。この SWIx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、SWIx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、SWIx は特権アクセスによってのみアクセスできます。この SWIx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

ソフトウェア割込みは、EXTI_RTISR および EXTI_FTSR の設定とは無関係に生成されます。読出し時には必ず 0 が返ります。

0 : 0 を書き込んでも、ビットの値は変化しません。

1 : 1 を書き込むと、イベント x で立上がりエッジ・イベントをトリガします。このビットは、ハードウェアによって自動的にクリアされます。

18.6.4 EXTI 立上がりエッジペンディングレジスタ 1 (EXTI_RPR1)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:0 **RPIFx** : 設定可能なイベント入力 x の立上がりエッジ・ペンディング・ビット (x = 16~0)

EXTI_SECCFGR.SECx が無効のとき、RPIFx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RPIFx はセキュア・アクセスによってのみアクセスできます。この RPIFx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、RPIFx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、RPIFx は特権アクセスによってのみアクセスできます。この RPIFx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立上がりエッジトリガリクエストは発生していません。

1 : 立上がりエッジトリガリクエストが発生しました。

このビットは、立上がりエッジイベントまたは EXTI_SWIER ソフトウェアトリガが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

18.6.5 EXTI 立下りエッジペンディングレジスタ1 (EXTI_FPR1)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF16
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:0 **FPIF_x** : 設定可能なイベント入力 x 立下りエッジ・ペンディング・ビット (x = 16~0)

EXTI_SECCFGR.SEC_x が無効のとき、FPIF_x は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SEC_x が有効なとき、FPIF_x はセキュア・アクセスによってのみアクセスできます。この FPIF_x への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIV_x が無効のとき、FPIF_x は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、FPIF_x は特権アクセスによってのみアクセスできます。この FPIF_x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立下りエッジトリガリクエストは発生していません。

1 : 立下りエッジトリガリクエストが発生します。

このビットは、立下りエッジイベントが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

18.6.6 EXTI セキュリティ設定レジスタ1 (EXTI_SECCFGR1)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

このレジスタは書込みアクセスセキュリティを提供します。非セキュア書込みアクセスは無視され、不正アクセスイベントが生成されます。非セキュア読出しにはレジスタのデータが返ります。

セキュリティ対応入力イベント用のレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEC31	SEC30	SEC29	SEC28	SEC27	SEC26	SEC25	SEC24	SEC23	SEC22	SEC21	SEC20	SEC19	SEC18	SEC17	SEC16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **SEC_x** : イベント入力 x (x = 31~0) のセキュリティ有効化

EXTI_PRIVCFGR.PRIV_x が無効のとき、SEC_x は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、SEC_x は特権アクセスによってのみ書込みができます。この SEC_x への非特権書込みは破棄されます。

0 : イベントのセキュリティは無効です (非セキュア)。

1 : イベントのセキュリティは有効です (セキュア)。

18.6.7 EXTI 特権設定レジスタ1 (EXTI_PRIVCFGR1)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタは特権書込みアクセス保護を提供します。非特権読出しにはレジスタのデータが返ります。

特権対応入力イベント用のレジスタ・ビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIV31	PRIV30	PRIV29	PRIV28	PRIV27	PRIV26	PRIV25	PRIV24	PRIV23	PRIV22	PRIV21	PRIV20	PRIV19	PRIV18	PRIV17	PRIV16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV15	PRIV14	PRIV13	PRIV12	PRIV11	PRIV10	PRIV9	PRIV8	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **PRIVx** : イベント入力 x (x = 31~0) のセキュリティ有効化

EXTI_SECCFGR.SECx が無効のとき、PRIVx はセキュアおよび非セキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、PRIVx はセキュア・アクセスによってのみ書込みができます。この PRIVx への非セキュア書込みは破棄されます。

0 : イベントの特権は無効です (非特権)。

1 : イベントの特権は有効です (特権)。

18.6.8 EXTI 立上りトリガ選択レジスタ 2 (EXTI_RTSR2)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT53	Res.	Res.	RT50	Res.	Res.
										rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RT46	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:22、20:19、
17:15、13:0 予約済みであり、リセット値に保持する必要があります。

ビット 21 **RT53** : 設定可能なイベント入力 x の立上がりトリガ・イベント設定ビット⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、RTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RTx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、RTx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、RTx は特権アクセスによってのみアクセスできます。このビット x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立上がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立上がりトリガは有効です。

ビット 18 **RT50** : 設定可能なイベント入力 x の立上がりトリガ・イベント設定ビット ⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、RTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RTx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、RTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、RTx は特権アクセスによってのみアクセスできます。このビット x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立上がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立上がりトリガは有効です。

ビット 14 **RT46** : 設定可能なイベント入力 x の立上がりトリガ・イベント設定ビット ⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、RTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RTx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、RTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、RTx は特権アクセスによってのみアクセスできます。このビット x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立上がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立上がりトリガは有効です。

1. 設定可能なイベント入力エッジトリガであるため、これらの入力でグリッチが生成されないようにする必要があります。レジスタへの書込み中に設定可能なイベント入力で立上がりエッジが発生した場合、関連するペンディングビットはセットされません。同じ設定可能なイベント入力に対して、立上がりエッジトリガと立下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

18.6.9 EXTI 立下りトリガ選択レジスタ 2 (EXTI_FTSR2)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT53	Res.	Res.	FT50	Res.	Res.
										rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FT46	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:22、20:19、17:15、13:0 予約済みであり、リセット値に保持する必要があります。

ビット 21 **FT53** : 設定可能なイベント入力 x の立下がりトリガ・イベント設定ビット ⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、FTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FTx はセキュア・アクセスによってのみアクセスできます。この FTx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、FTx は特権アクセスによってのみアクセスできます。この FTx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立下がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立下がりトリガは有効です。

ビット 18 **FT50** : 設定可能なイベント入力 x の立下がりトリガ・イベント設定ビット ⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、FTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FTx はセキュア・アクセスによってのみアクセスできます。この FTx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、FTx は特権アクセスによってのみアクセスできます。この FTx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立下がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立下がりトリガは有効です。

ビット 14 **FT46** : 設定可能なイベント入力 x の立下がりトリガ・イベント設定ビット ⁽¹⁾

EXTI_SECCFGR.SECx が無効のとき、FTx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FTx はセキュア・アクセスによってのみアクセスできます。この FTx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FTx は非特権および特権アクセスによってアクセスできます。

EXTI_PRIVCFGR.PRIVx が有効なとき、FTx は特権アクセスによってのみアクセスできます。この FTx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 入力ラインの (イベントと割込みについて) 立下がりトリガは無効です。

1 : 入力ラインの (イベントと割込みについて) 立下がりトリガは有効です。

1. 設定可能なイベント入力はエッジトリガであるため、これらの入力でグリッチが生成されないようにする必要があります。
レジスタへの書込み中に設定可能なイベント入力で立下がりエッジが発生した場合、関連するペンディングビットはセットされません。
同じ設定可能なイベント入力に対して、立上がりエッジトリガと立下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

18.6.10 EXTI ソフトウェア割込みイベント・レジスタ 2 (EXTI_SWIER2)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI53	Res.	Res.	SWI50	Res.	Res.
										rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SWI46	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:22、20:19、17:15、13:0 予約済みであり、リセット値に保持する必要があります。

ビット 21 **SWI53** : イベント x のソフトウェア割込み

EXTI_SECCFGR.SECx が無効のとき、SWIx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、SWIx はセキュア・アクセスによってのみアクセスできます。この SWIx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、SWIx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、SWIx は特権アクセスによってのみアクセスできます。この SWIx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

ソフトウェア割込みは、EXTI_RTISR および EXTI_FTSR の設定とは無関係に生成されます。読出し時には必ず 0 が返ります。

0 : 0 を書き込んでも、ビットの値は変化しません。

1 : 1 を書き込むと、イベント x で立上がりエッジ・イベントをトリガします。このビットは、ハードウェアによって自動的にクリアされます。

ビット 18 **SWI50** : イベント x のソフトウェア割込み

EXTI_SECCFGR.SECx が無効のとき、SWIx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、SWIx はセキュア・アクセスによってのみアクセスできます。この SWIx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、SWIx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、SWIx は特権アクセスによってのみアクセスできます。この SWIx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

ソフトウェア割込みは、EXTI_RTISR および EXTI_FTSR の設定とは無関係に生成されます。読出し時には必ず 0 が返ります。

0 : 0 を書き込んでも、ビットの値は変化しません。

1 : 1 を書き込むと、イベント x で立上がりエッジ・イベントをトリガします。このビットは、ハードウェアによって自動的にクリアされます。

ビット 14 **SWI46** : イベント x のソフトウェア割込み

EXTI_SECCFGR.SECx が無効のとき、SWIx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、SWIx はセキュア・アクセスによってのみアクセスできます。この SWIx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、SWIx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、SWIx は特権アクセスによってのみアクセスできます。この SWIx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

ソフトウェア割込みは、EXTI_RTISR および EXTI_FTSR の設定とは無関係に生成されます。読出し時には必ず 0 が返ります。

0 : 0 を書き込んでも、ビットの値は変化しません。

1 : 1 を書き込むと、イベント x で立上がりエッジ・イベントをトリガします。このビットは、ハードウェアによって自動的にクリアされます。

18.6.11 EXTI 立上リエッジ・ペンディング・レジスタ 2 (EXTI_RPR2)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットののみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF53	Res.	Res.	RPIF50	Res.	Res.
										rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RPIF46	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:22、20:19、
17:15、13:0 予約済みであり、リセット値に保持する必要があります。

ビット 21 **RPIF53** : 設定可能なイベント入力 x の立上がりエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、RPIF_x は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RPIF_x はセキュア・アクセスによってのみアクセスできます。この RPIF_x への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIV_x が無効のとき、RPIF_x は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、RPIF_x は特権アクセスによってのみアクセスできます。この RPIF_x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立上がりエッジトリガリクエストは発生していません。

1 : 立上がりエッジトリガリクエストが発生しました。

このビットは、立上がりエッジイベントまたは EXTI_SWIER ソフトウェアトリガが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

ビット 18 **RPIF50** : 設定可能なイベント入力 x の立上がりエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、RPIF_x は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RPIF_x はセキュア・アクセスによってのみアクセスできます。この RPIF_x への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIV_x が無効のとき、RPIF_x は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、RPIF_x は特権アクセスによってのみアクセスできます。この RPIF_x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立上がりエッジトリガリクエストは発生していません。

1 : 立上がりエッジトリガリクエストが発生しました。

このビットは、立上がりエッジイベントまたは EXTI_SWIER ソフトウェアトリガが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

ビット 14 **RPIF46** : 設定可能なイベント入力 x の立上がりエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、RPIF_x は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、RPIF_x はセキュア・アクセスによってのみアクセスできます。この RPIF_x への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIV_x が無効のとき、RPIF_x は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、RPIF_x は特権アクセスによってのみアクセスできます。この RPIF_x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立上がりエッジトリガリクエストは発生していません。

1 : 立上がりエッジトリガリクエストが発生しました。

このビットは、立上がりエッジイベントまたは EXTI_SWIER ソフトウェアトリガが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

18.6.12 EXTI 立下りエッジ・ペンディング・レジスタ 2 (EXTI_FPR2)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

設定可能なイベントに対するレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF53	Res.	Res.	FPIF50	Res.	Res.
										rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FPIF46	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:22、20:19、
17:15、13:0 予約済みであり、リセット値に保持する必要があります。

ビット 21 **FPIF53** : 設定可能なイベント入力 x の立下りエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、FPIF_x は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FPIF_x はセキュア・アクセスによってのみアクセスできます。この FPIF_x への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIV_x が無効のとき、FPIF_x は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIV_x が有効なとき、FPIF_x は特権アクセスによってのみアクセスできます。この FPIF_x への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立下りエッジトリガリクエストは発生していません。

1 : 立下りエッジトリガリクエストが発生します。

このビットは、立下りエッジイベントが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

ビット 18 **FPIF50** : 設定可能なイベント入力 x の立下がりエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、FPIFx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FPIFx はセキュア・アクセスによってのみアクセスできます。この FPIFx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FPIFx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、FPIFx は特権アクセスによってのみアクセスできます。この FPIFx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立下がりエッジトリガリクエストは発生していません。

1 : 立下がりエッジトリガリクエストが発生します。

このビットは、立下がりエッジイベントが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

ビット 14 **FPIF46** : 設定可能なイベント入力 x の立下がりエッジ・ペンディング・ビット

EXTI_SECCFGR.SECx が無効のとき、FPIFx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、FPIFx はセキュア・アクセスによってのみアクセスできます。この FPIFx への非セキュア書込みは破棄され、非セキュア読出しには 0 が返ります。

EXTI_PRIVCFGR.PRIVx が無効のとき、FPIFx は非特権および特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、FPIFx は特権アクセスによってのみアクセスできます。この FPIFx への非特権書込みは破棄され、非特権読出しには 0 が返ります。

0 : 立下がりエッジトリガリクエストは発生していません。

1 : 立下がりエッジトリガリクエストが発生します。

このビットは、立下がりエッジイベントが設定可能なイベントラインに到着したときにセットされます。このビットは 1 を書き込むことによってクリアされます。

18.6.13 EXTI セキュリティ設定レジスタ 2 (EXTI_SECCFGR2)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

このレジスタは書込みアクセスセキュリティを提供します。非セキュア書込みアクセスは無視され、不正アクセスイベントが生成されます。非セキュア読出しにはレジスタのデータが返ります。

特権対応入力イベント用のレジスタ・ビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	SEC57	SEC56	SEC55	SEC54	SEC53	SEC52	SEC51	SEC50	SEC49	SEC48
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC47	SEC46	SEC45	SEC44	SEC43	SEC42	SEC41	SEC40	SEC39	SEC38	SEC37	SEC36	SEC35	SEC34	SEC33	SEC32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:0 **SECx** : イベント入力 x (x = 57~32) のセキュリティ有効化

EXTI_PRIVCFGR.PRIVx が無効のとき、SECx は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、SECx は特権アクセスによってのみ書込みができます。この SECx への非特権書込みは破棄されます。

0 : イベントのセキュリティは無効です (非セキュア)。

1 : イベントのセキュリティは有効です (セキュア)。

18.6.14 EXTI 特権設定レジスタ 2 (EXTI_PRIVCFGR2)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

このレジスタは特権書込みアクセス保護を提供します。非特権読出しにはレジスタのデータが返ります。
セキュリティ対応入力イベント用のレジスタビットのみが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PRIV57	PRIV56	PRIV55	PRIV54	PRIV53	PRIV52	PRIV51	PRIV50	PRIV49	PRIV48
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV47	PRIV46	PRIV45	PRIV44	PRIV43	PRIV42	PRIV41	PRIV40	PRIV39	PRIV38	PRIV37	PRIV36	PRIV35	PRIV34	PRIV33	PRIV32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:0 **PRIVx** : イベント入力 x (x = 57~32) のセキュリティ有効化

EXTI_SECCFGR.SECx が無効のとき、PRIVx はセキュアおよび非セキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、PRIVx はセキュア・アクセスによってのみ書込みができます。この PRIVx への非セキュア書込みは破棄されます。

0 : イベントの特権は無効です (非特権)。

1 : イベントの特権は有効です (特権)。

18.6.15 EXTI 外部割込み選択レジスタ1 (EXTI_EXTICR1)

アドレスオフセット : 0x060 EXTI マルチプレクサ 0, 1, 2, 3

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI3 [7:0]								EXTI2 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI1 [7:0]								EXTI0 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **EXTI3[7:0]** : EXTI3 GPIO ポート選択

これらのビットは、EXTI3 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC3 が無効のとき、EXTI3 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC3 が有効なとき、EXTI3 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV3 が無効のとき、EXTI3 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV3 が有効なとき、EXTI3 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA3 ピン

0x01 : PB3 ピン

0x02 : PC3 ピン

0x03 : PD3 ピン

0x04 : PE3 ピン

0x05 : PF3 ピン

0x06 : PG3 ピン

0x07 : PH3 ピン

0x08 : PI3 ピン

その他 : 予約済み

ビット 23:16 **EXTI2[7:0]** : EXTI2 GPIO ポート選択

これらのビットは、EXTI2 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC2 が無効のとき、EXTI2 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC2 が有効なとき、EXTI2 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV2 が無効のとき、EXTI2 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV2 が有効なとき、EXTI2 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA2 ピン

0x01 : PB2 ピン

0x02 : PC2 ピン

0x03 : PD2 ピン

0x04 : PE2 ピン

0x05 : PF2 ピン

0x06 : PG2 ピン

0x07 : PH2 ピン

0x08 : PI2 ピン

その他 : 予約済み

ビット 15:8 **EXTI1[7:0]** : EXTI1 GPIO ポート選択

これらのビットは、EXTI1 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC1 が無効のとき、EXTI1 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC1 が有効なとき、EXTI1 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV1 が無効のとき、EXTI1 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV1 が有効なとき、EXTI1 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA1 ピン

0x01 : PB1 ピン

0x02 : PC1 ピン

0x03 : PD1 ピン

0x04 : PE1 ピン

0x05 : PF1 ピン

0x06 : PG1 ピン

0x07 : PH1 ピン

0x08 : PI1 ピン

その他 : 予約済み

ビット 7:0 **EXTI0[7:0]** : EXTI0 GPIO ポート選択

これらのビットは、EXTI0 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC0 が無効のとき、EXTI0 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC0 が有効なとき、EXTI0 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV0 が無効のとき、EXTI0 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV0 が有効なとき、EXTI0 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA0 ピン

0x01 : PB0 ピン

0x02 : PC0 ピン

0x03 : PD0 ピン

0x04 : PE0 ピン

0x05 : PF0 ピン

0x06 : PG0 ピン

0x07 : PH0 ピン

0x08 : PI0 ピン

その他 : 予約済み

18.6.16 EXTI 外部割り込み選択レジスタ2 (EXTI_EXTICR2)

アドレスオフセット : 0x064 EXTI マルチプレクサ 4, 5, 6, 7

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI7 [7:0]								EXTI6 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI5 [7:0]								EXTI4 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **EXTI7[7:0]** : EXTI7 GPIO ポート選択

これらのビットは、EXTI7 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC7 が無効のとき、EXTI7 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC7 が有効なとき、EXTI7 はセキュア・アクセスによってのみアクセスできます。非セキュア書き込みは破棄され、非セキュア読み出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV7 が無効のとき、EXTI7 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV7 が有効なとき、EXTI7 は特権アクセスによってのみアクセスできます。このビットへの非特権書き込みは破棄されます。

0x00 : PA7 ピン

0x01 : PB7 ピン

0x02 : PC7 ピン

0x03 : PD7 ピン

0x04 : PE7 ピン

0x05 : PF7 ピン

0x06 : PG7 ピン

0x07 : PH7 ピン

0x08 : PI7 ピン

その他 : 予約済み

ビット 23:16 **EXTI6[7:0]** : EXTI6 GPIO ポート選択

これらのビットは、EXTI6 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC6 が無効のとき、EXTI6 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC6 が有効なとき、EXTI6 はセキュア・アクセスによってのみアクセスできます。非セキュア書き込みは破棄され、非セキュア読み出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV6 が無効のとき、EXTI6 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV6 が有効なとき、EXTI6 は特権アクセスによってのみアクセスできます。このビットへの非特権書き込みは破棄されます。

0x00 : PA6 ピン

0x01 : PB6 ピン

0x02 : PC6 ピン

0x03 : PD6 ピン

0x04 : PE6 ピン

0x05 : PF6 ピン

0x06 : PG6 ピン

0x07 : PH6 ピン

0x08 : PI6 ピン

その他 : 予約済み

ビット 15:8 **EXTI5[7:0]** : EXTI5 GPIO ポート選択

これらのビットは、EXTI5 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC5 が無効のとき、EXTI5 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC5 が有効なとき、EXTI5 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV5 が無効のとき、EXTI5 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV5 が有効なとき、EXTI5 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA5 ピン

0x01 : PB5 ピン

0x02 : PC5 ピン

0x03 : PD5 ピン

0x04 : PE5 ピン

0x05 : PF5 ピン

0x06 : PG5 ピン

0x07 : PH5 ピン

0x08 : PI5 ピン

その他 : 予約済み

ビット 7:0 **EXTI4[7:0]** : EXTI4 GPIO ポート選択

これらのビットは、EXTI4 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC4 が無効のとき、EXTI4 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC4 が有効なとき、EXTI4 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV4 が無効のとき、EXTI4 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV4 が有効なとき、EXTI4 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA4 ピン

0x01 : PB4 ピン

0x02 : PC4 ピン

0x03 : PD4 ピン

0x04 : PE4 ピン

0x05 : PF4 ピン

0x06 : PG4 ピン

0x07 : PH4 ピン

0x08 : PI4 ピン

その他 : 予約済み

18.6.17 EXTI 外部割込み選択レジスタ3 (EXTI_EXTICR3)

アドレスオフセット : 0x068 EXTI マルチプレクサ 8, 9, 10, 11

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI11 [7:0]								EXTI10 [7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI9 [7:0]								EXTI8 [7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **EXTI11[7:0]** : EXTI11 GPIO ポート選択

これらのビットは、EXTI11 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC11 が無効のとき、EXTI11 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC11 が有効なとき、EXTI11 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV11 が無効のとき、EXTI11 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV11 が有効なとき、EXTI11 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA11 ピン

0x01 : PB11 ピン

0x02 : PC11 ピン

0x03 : PD11 ピン

0x04 : PE11 ピン

0x05 : PF11 ピン

0x06 : PG11 ピン

0x07 : PH11 ピン

0x08 : PI11 ピン

その他 : 予約済み

ビット 23:16 **EXTI10[7:0]** : EXTI10 GPIO ポート選択

これらのビットは、EXTI10 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC10 が無効のとき、EXTI10 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC10 が有効なとき、EXTI10 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV10 が無効のとき、EXTI10 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV10 が有効なとき、EXTI10 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA10 ピン

0x01 : PB10 ピン

0x02 : PC10 ピン

0x03 : PD10 ピン

0x04 : PE10 ピン

0x05 : PF10 ピン

0x06 : PG10 ピン

0x07 : PH10 ピン

0x08 : PI10 ピン

その他 : 予約済み

ビット 15:8 **EXTI9[7:0]** : EXTI9 GPIO ポート選択

これらのビットは、EXTI9 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC9 が無効のとき、EXTI9 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC9 が有効なとき、EXTI9 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV9 が無効のとき、EXTI9 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV9 が有効なとき、EXTI9 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA9 ピン

0x01 : PB9 ピン

0x02 : PC9 ピン

0x03 : PD9 ピン

0x04 : PE9 ピン

0x05 : PF9 ピン

0x06 : PG9 ピン

0x07 : PH9 ピン

0x08 : PI9 ピン

その他 : 予約済み

ビット 7:0 **EXTI8[7:0]** : EXTI8 GPIO ポート選択

これらのビットは、EXTI8 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC8 が無効のとき、EXTI8 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC8 が有効なとき、EXTI8 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV8 が無効のとき、EXTI8 は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR1.PRIV8 が有効なとき、EXTI8 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA8 ピン

0x01 : PB8 ピン

0x02 : PC8 ピン

0x03 : PD8 ピン

0x04 : PE8 ピン

0x05 : PF8 ピン

0x06 : PG8 ピン

0x07 : PH8 ピン

0x08 : PI8 ピン

その他 : 予約済み

18.6.18 EXTI 外部割込み選択レジスタ 4 (EXTI_EXTICR4)

アドレスオフセット : 0x060 EXTI マルチプレクサ 12, 13, 14, 15

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15 [7:0]								EXTI14 [7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI13 [7:0]								EXTI12 [7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **EXTI15[7:0]** : EXTI15 GPIO ポート選択

これらのビットは、EXTI15 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC15 が無効のとき、EXTI15 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC15 が有効なとき、EXTI15 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV15 が無効のとき、EXTI15 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV15 が有効なとき、EXTI15 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA15 ピン

0x01 : PB15 ピン

0x02 : PC15 ピン

0x03 : PD15 ピン

0x04 : PE15 ピン

0x05 : PF15 ピン

0x06 : PG15 ピン

0x07 : PH15 ピン

その他 : 予約済み

ビット 23:16 **EXTI14[7:0]** : EXTI14 GPIO ポート選択

これらのビットは、EXTI14 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC14 が無効のとき、EXTI14 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC14 が有効なとき、EXTI14 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV14 が無効のとき、EXTI14 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV14 が有効なとき、EXTI14 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA14 ピン

0x01 : PB14 ピン

0x02 : PC14 ピン

0x03 : PD14 ピン

0x04 : PE14 ピン

0x05 : PF14 ピン

0x06 : PG14 ピン

0x07 : PH14 ピン

その他 : 予約済み

ビット 15:8 **EXTI13[7:0]** : EXTI13 GPIO ポート選択

これらのビットは、EXTI13 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC13 が無効のとき、EXTI13 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC13 が有効なとき、EXTI13 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV13 が無効のとき、EXTI13 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV13 が有効なとき、EXTI13 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA13 ピン

0x01 : PB13 ピン

0x02 : PC13 ピン

0x03 : PD13 ピン

0x04 : PE13 ピン

0x05 : PF13 ピン

0x06 : PG13 ピン

0x07 : PH13 ピン

その他 : 予約済み

ビット 7:0 **EXTI12[7:0]** : EXTI12 GPIO ポート選択

これらのビットは、EXTI12 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。EXTI_SECCFGR1.SEC12 が無効のとき、EXTI12 は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR1.SEC12 が有効なとき、EXTI12 はセキュア・アクセスによってのみアクセスできます。非セキュア書込みは破棄され、非セキュア読出しは 0 を返します。

EXTI_PRIVCFGR1.PRIV12 が無効のとき、EXTI12 は特権および非特権アクセスによってアクセスできます。

EXTI_PRIVCFGR1.PRIV12 が有効なとき、EXTI12 は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0x00 : PA12 ピン

0x01 : PB12 ピン

0x02 : PC12 ピン

0x03 : PD12 ピン

0x04 : PE12 ピン

0x05 : PF12 ピン

0x06 : PG12 ピン

0x07 : PH12 ピン

その他 : 予約済み

18.6.19 EXTI ロックレジスタ (EXTI_LOCKR)

アドレスオフセット : 0x070

リセット値 : 0x0000 0000

このレジスタは書込みアクセス・セキュリティを提供します。非セキュア書込みアクセスは無視され、読出しアクセスにはゼロデータが返されます。そして両者は不正アクセスイベントを生成します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK
															rs

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **LOCK** : グローバル・セキュリティおよび特権設定レジスタ (EXTI_SECCFGR および EXTI_PRIVCFGR) のロック

このビットは、リセット後 1 度のみ書き込まれます。

0 : セキュリティおよび特権の設定はオープンで、変更可能です。

1 : セキュリティおよび特権の設定はロックされており、もはや変更できません。

18.6.20 割込みマスクレジスタによる EXTI CPU ウェイクアップレジスタ1 (EXTI_IMR1)

アドレスオフセット : 0x080

リセット値 : 0xFFFFE 0000

設定可能なイベントに対するレジスタ・ビットが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IMx** : イベント入力 x ⁽¹⁾ (x = 31~0) の割込みマスクによる CPU ウェイクアップ

EXTI_SECCFGR.SECx が無効のとき、IMx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、IMx はセキュア・アクセスによってのみアクセスできます。このビットへの非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、IMx は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、IMx は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0 : 入力イベント x からの割込みリクエストによるウェイクアップはマスクされます。

1 : 入力イベント x からの割込みリクエストによるウェイクアップはマスクされません。

1. デフォルトで割込みを無効にするために、設定可能なイベント入力のリセット値は 0 に設定されます。

18.6.21 イベントマスクレジスタによる EXTI CPU ウェイクアップレジスタ1 (EXTI_EMR1)

アドレスオフセット : 0x084

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EM31	EM30	EM29	EM28	EM27	EM26	EM25	EM24	EM23	EM22	EM21	EM20	EM19	EM18	EM17	EM16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **EMx** : イベント入力 x (x = 31~0) のイベント生成マスクによる CPU ウェイクアップ

EXTI_SECCFGR.SECx が無効のとき、EMx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、EMx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、EMx は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、EMx は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0 : ライン x からのイベント生成によるウェイクアップはマスクされます。

1 : ライン x からのイベント生成によるウェイクアップはマスクされません。

18.6.22 割り込みマスク・レジスタ 2 によるEXTI CPU ウェイクアップ(EXTI_IMR2)

アドレスオフセット : 0x090

リセット値 : 0x03DB BFFF

設定可能なイベントに対するレジスタ・ビットが含まれています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	IM57	IM56	IM55	IM54	IM53	IM52	IM51	IM50	IM49	IM48
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM47	IM46	IM45	IM44	IM43	IM42	IM41	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:0 **IMx** : イベント入力 x⁽¹⁾ (x = 57~0) の割り込みマスクによる CPU ウェイクアップ

EXTI_SECCFGR.SECx が無効のとき、IMx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、IMx はセキュア・アクセスによってのみアクセスできます。このビットへの非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、IMx は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、IMx は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0 : 入力イベント x からの割り込みリクエストによるウェイクアップはマスクされます。

1 : 入力イベント x からの割り込みリクエストによるウェイクアップはマスクされません。

1. デフォルトで割り込みを無効にするために、設定可能なイベント入力のリセット値は 0 に設定されます。

18.6.23 イベント・マスク・レジスタ 2 による EXTI CPU ウェイクアップ (EXTI_EMR2)

アドレスオフセット : 0x094

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	EM57	EM56	EM55	EM54	EM53	EM52	EM51	EM50	EM49	EM48
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM47	EM46	EM45	EM44	EM43	EM42	EM41	EM40	EM39	EM38	EM37	EM36	EM35	EM34	EM33	EM32
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:0 **EMx** : イベント入力 x (x = 57~32) のイベント生成マスクによる CPU ウェイクアップ

EXTI_SECCFGR.SECx が無効のとき、EMx は非セキュアおよびセキュア・アクセスによってアクセスできます。

EXTI_SECCFGR.SECx が有効なとき、EMx はセキュア・アクセスによってのみアクセスできます。このビット x への非セキュア書込みは破棄され、非セキュア読出しには 0 を返します。

EXTI_PRIVCFGR.PRIVx が無効のとき、EMx は特権および非特権アクセスによってアクセスできます。EXTI_PRIVCFGR.PRIVx が有効なとき、EMx は特権アクセスによってのみアクセスできます。このビットへの非特権書込みは破棄されます。

0 : ライン x からのイベント生成によるウェイクアップはマスクされます。

1 : ライン x からのイベント生成によるウェイクアップはマスクされません。

18.6.24 EXTI レジスタマップ

表 142. EXTI レジスタのマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	EXTI_RTSR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT16	RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	EXTI_FTSR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT16	FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	EXTI_SWIER1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI16	SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	EXTI_RPR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF16	RPIF15	RPIF14	RPIF13	RPIF12	RPIF11	RPIF10	RPIF9	RPIF8	RPIF7	RPIF6	RPIF5	RPIF4	RPIF3	RPIF2	RPIF1	RPIF0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	EXTI_FPR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF16	FPIF15	FPIF14	FPIF13	FPIF12	FPIF11	FPIF10	FPIF9	FPIF8	FPIF7	FPIF6	FPIF5	FPIF4	FPIF3	FPIF2	FPIF1	FPIF0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	EXTI_SECCFGR1	SEC31	SEC30	SEC29	SEC28	SEC27	SEC26	SEC25	SEC24	SEC23	SEC22	SEC21	SEC20	SEC19	SEC18	SEC17	SEC16	SEC15	SEC14	SEC13	SEC12	SEC11	SEC10	SEC9	SEC8	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	EXTI_PRIVCFGR1	PRIV31	PRIV30	PRIV29	PRIV28	PRIV27	PRIV26	PRIV25	PRIV24	PRIV23	PRIV22	PRIV21	PRIV20	PRIV19	PRIV18	PRIV17	PRIV16	PRIV15	PRIV14	PRIV13	PRIV12	PRIV11	PRIV10	PRIV9	PRIV8	PRIV7	PRIV6	PRIV5	PRIV4	PRIV3	PRIV2	PRIV1	PRIV0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	EXTI_RTSR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT53	Res.	Res.	RT50	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0			0																		
0x024	EXTI_FTSR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT53	Res.	Res.	FT50	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0			0																		
0x028	EXTI_SWIER2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI53	Res.	Res.	SWI50	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0			0																		
0x02C	EXTI_RPR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIF53	Res.	Res.	RPIF50	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0			0																		
0x030	EXTI_FPR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPIF53	Res.	Res.	FPIF50	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0			0																		
0x034	EXTI_SECCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	SEC57	SEC56	SEC55	SEC54	SEC53	SEC52	SEC51	SEC50	SEC49	SEC48	SEC47	SEC46	SEC45	SEC44	SEC43	SEC42	SEC41	SEC40	SEC39	SEC38	SEC37	SEC36	SEC35	SEC34	SEC33	SEC32
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	EXTI_PRIVCFGR2	Res.	Res.	Res.	Res.	Res.	Res.	PRIV57	PRIV56	PRIV55	PRIV54	PRIV53	PRIV52	PRIV51	PRIV50	PRIV49	PRIV48	PRIV47	PRIV46	PRIV45	PRIV44	PRIV43	PRIV42	PRIV41	PRIV40	PRIV39	PRIV38	PRIV37	PRIV36	PRIV35	PRIV34	PRIV33	PRIV32
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C から 0x05C	予約済み	予約済み																															
0x060	EXTI_EXTICR1	EXTI3 [7:0]								EXTI2 [7:0]								EXTI1 [7:0]								EXTI0 [7:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 142. EXTI レジスタのマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x064	EXTI_EXTICR2	EXTI7 [7:0]								EXTI6 [7:0]								EXTI5 [7:0]								EXTI4 [7:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x068	EXTI_EXTICR3	EXTI11 [7:0]								EXTI10 [7:0]								EXTI9 [7:0]								EXTI8 [7:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x06C	EXTI_EXTICR4	EXTI15 [7:0]								EXTI14 [7:0]								EXTI13 [7:0]								EXTI12 [7:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x070	EXTI_LOCKR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK
	リセット値																																0
0x074 から 0x07C	予約済み	予約済み																															
0x080	EXTI_IMR1	IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	EXTI_EMR1	EM31	EM30	EM29	EM28	EM27	EM26	EM25	EM24	EM23	EM22	EM21	EM20	EM19	EM18	EM17	EM16	EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x090	EXTI_IMR2	Res.	Res.	Res.	Res.	Res.	Res.	IM57	IM56	IM55	IM54	IM53	IM52	IM51	IM50	IM49	IM48	IM47	IM46	IM45	IM44	IM43	IM42	IM41	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
	リセット値							1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0x094	EXTI_EMR2	Res.	Res.	Res.	Res.	Res.	Res.	EM57	EM56	EM55	EM54	EM53	EM52	EM51	EM50	EM49	EM48	EM47	EM46	EM45	EM44	EM43	EM42	EM41	EM40	EM39	EM38	EM37	EM36	EM35	EM34	EM33	EM32
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

19 巡回冗長検査計算ユニット (CRC)

19.1 概要

CRC (Cyclic Redundancy Check : 巡回冗長検査) 計算ユニットは、8、16、または 32 ビットデータワードと、ある生成多項式から、CRC コードを得るために使用されます。

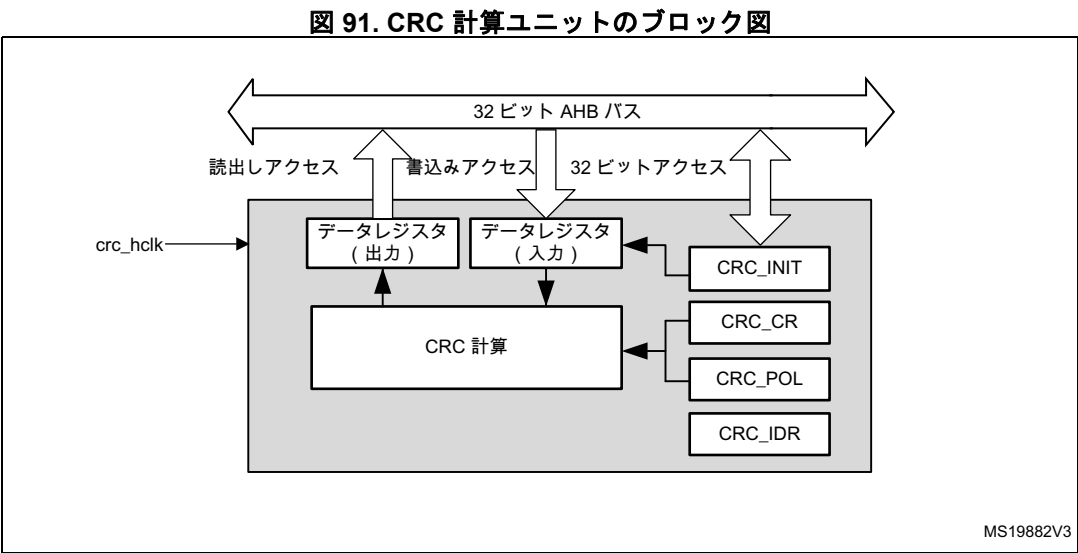
他のアプリケーションの中でも、CRC ベースのテクニックは、データ転送やストレージの整合性を確認するために使用されます。機能安全規格の範囲内では、このテクニックが Flash メモリの整合性を確認するひとつの手段となっています。CRC 計算ユニットは、実行時にソフトウェアのシグネチャ計算を支援します。リンク時に生成されて、特定のメモリ領域に保存されたリファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

19.2 CRC の主な機能

- 以下の CRC-32 (Ethernet) 多項式を使用します。0x4C11DB7
$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$
- あるいは、完全にプログラム可能な多項式をプログラム可能なサイズ (7、8、16、32 ビット) で使用します。
- サイズが 8、16、32 ビットのデータを取り扱います。
- プログラム可能な CRC の初期値
- シングル入力/出力 32 ビットデータレジスタ
- 計算時のバスのストールを避けるための入力バッファ
- データサイズが 32 ビットの場合、CRC の計算は AHB クロック 4 サイクル (HCLK) 以内に行われます。
- 汎用 8 ビットレジスタ (一時ストレージとして使用可能)
- I/O データの可逆性のオプション
- ワード、右詰めハーフワード、右詰めバイトでアクセスできる CRC_DR レジスタを除き、AHB スレーブ・ペリフェラル経由で 32 bit ワードのみでアクセスします。

19.3 CRC の機能説明

19.3.1 CRC ブロック図



19.3.2 CRC 内部信号

表 143. CRC 内部入力／出力信号

信号名	信号タイプ	説明
crc_hclk	デジタル入力	AHB クロック

19.3.3 CRC 操作

CRC 計算ユニットは、1 つの 32 ビットリード／ライトデータレジスタ (CRC_DR) を持っています。このレジスタを使用して、新しいデータを入力し (書き込みアクセス)、前回の CRC 計算結果を保持します (読出しアクセス)。

データレジスタへの書き込み操作のたびに、前回の CRC 値 (CRC_DR に格納) と新しい値の組み合わせが作成されます。CRC 計算は、書き込まれるデータのフォーマットに応じ、32 ビットデータワード全体に対して、またはバイト単位で行われます。

CRC_DR レジスタは、ワード、右詰め of ハーフワード、右詰め of バイトによってアクセスできます。他のレジスタについては、32 ビットアクセスのみ可能です。

計算の時間はデータ幅に依存します。

- 32 ビットの場合、4 AHB クロックサイクル
- 16 ビットの場合、2 AHB クロックサイクル
- 8 ビットの場合、1 AHB クロックサイクル

入力バッファを使うと、前回の CRC 計算によるウェイトステートを待つことなく、すぐに第 2 のデータを書き込むことができます。

特定のバイト数に対する書込みアクセス数を最小限に抑えるために、データサイズを動的に調節することができます。たとえば、5 バイトの CRC は、1 ワードの書込みと、それに続く 1 バイトの書込みで計算することができます。

入力データを逆にして、さまざまなエンディアンネス方式を管理することができます。逆転操作は、CRC_CR レジスタの REV_IN[1:0] ビットに応じて、8、16、および 32 ビットで行うことができます。

たとえば、入力データ 0x1A2B3C4D は、CRC 計算では以下のように使用されます。

- 0x58D43CB2 (バイト単位でビットが逆転)
- 0xD458B23C (ハーフワード単位でビットが逆転)
- 0xB23CD458 (フルワードでビットが逆転)

また、出力データも、CRC_CR レジスタの REV_OUT ビットをセットすることによって逆にすることができます。

この操作はビット・レベルで行われます。例えば、出力データ 0x11223344 は 0x22CC4488 に変換されます。

CRC 計算機は、CRC_CR レジスタの RESET 制御ビットを使用して、プログラム可能な値に初期化することができます (デフォルト値は 0xFFFFFFFF)。

CRC の初期値は、CRC_INIT レジスタでプログラムすることができます。CRC_DR レジスタは、CRC_INIT レジスタの書込みアクセス時に自動的に初期化されます。

CRC_IDR レジスタを使用して、CRC 計算に関する一時的な値を保持することができます。このレジスタは、CRC_CR レジスタの RESET ビットによる影響を受けません。

多項式のプログラマビリティ

多項式係数は CRC_POL レジスタを介して完全にプログラム可能であり、多項式のサイズは CRC_CR レジスタの POLYSIZE[1:0] ビットをプログラムすることにより、7、8、16、または 32 ビットに設定できます。偶多項式はサポートされていません。

注： 偶数多項式の種類は $X+X^2+..+X^n$ で、奇数多項式の種類は $1+X+X^2+..+X^n$ です。

CRC データが 32 ビット未満であれば、その値は CRC_DR レジスタの下位ビットから読み出すことができます。

信頼できる CRC 計算結果を得るために、CRC 計算の実行中に多項式の値やサイズを変更することはできません。そのため、CRC 計算が進行中である場合、アプリケーションは計算をリセットするか、または CRC_DR の読出しを行ってから、多項式を変更しなければなりません。

多項式のデフォルト値は、CRC-32 (Ethernet) 多項式、0x4C11DB7。

19.4 CRC レジスタ

CRC_DR レジスタは、ワード、右詰め of のハーフワード、右詰め of のバイトによってアクセスできます。他のレジスタについては、32 ビットアクセスのみ可能です。

19.4.1 CRCデータレジスタ (CRC_DR)

アドレスオフセット : 0x00

リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DR[31:0]** : データレジスタビット

このレジスタを使用して、CRC 計算機に新しいデータを書き込みます。

読出し時には、前回の CRC 計算結果を保持します。

データサイズが 32 ビット未満であれば、下位ビットを使用して正しい値の書き込み／読出しを行います。

19.4.2 CRC 独立型データレジスタ (CRC_IDR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **IDR[31:0]** : 汎用 32 ビットデータレジスタビット

これらのビットは、4 バイトの一時的なストレージ領域として使用できます。

このレジスタは、CRC_CR レジスタの RESET ビットによって生成される CRC リセットの影響を受けません。

19.4.3 CRC制御レジスタ (CRC_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	リセット
								rw	rw	rw	rw	rw			rs

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **REV_OUT** : 出力データを逆にします。

このビットは、出力データビット順序の反転を制御します。

0 : ビットの順序は変わりません。

1 : 出力フォーマットのビットが反転しています。

ビット 6:5 **REV_IN[1:0]** : 入力データを逆にします。

このビットフィールドは、入力データ・ビット順序の反転を制御します

00 : ビットの順序は変わりません

01 : バイト単位でビットが反転しています

10 : ハーフワード単位でビットが反転しています

11 : ワード単位でビットが反転しています

ビット 4:3 **POLYSIZE[1:0]** : 多項式のサイズ

これらのビットは、多項式のサイズを制御します。

00 : 32 ビットの多項式

01 : 16 ビットの多項式

10 : 8 ビットの多項式

11 : 7 ビットの多項式

ビット 2:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **RESET** : RESET ビット

このビットは、CRC 計算ユニットをリセットし、CRC_INIT レジスタに格納された値にデータレジスタをセットするために、ソフトウェアによってセットされます。このビットはセットのみが可能で、ハードウェアによって自動的にクリアされます。

19.4.4 CRC の初期値 (CRC_INIT)

アドレスオフセット : 0x10

リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INIT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INIT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **CRC_INIT[31:0]** : プログラム可能な CRC の初期値

このレジスタを使用して、CRC の初期値を書き込みます。

19.4.5 CRC 多項式 (CRC_POL)

アドレスオフセット : 0x14

リセット値 : 0x04C1 1DB7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POL[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **POL[31:0]** : プログラム可能な多項式

このレジスタを使用して、CRC 計算に使用される多項式の係数を書き込みます。

多項式のサイズが 32 ビット未満であれば、下位ビットを使用して正しい値をプログラムする必要があります。

19.4.6 CRC レジスタマップ

表 144. CRC レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	DR[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	IDR[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	CRC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REV_OUT	REV_IN[1:0]	POLYSIZE[1:0]		Res	Res	リセット	
	リセット値																									0	0	0	0	0			0
0x10	CRC_INIT	CRC_INIT[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	POL[31:0]																															
	リセット値	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

20 CORDIC コプロセッサ (CORDIC)

20.1 CORDIC の概要

CORDIC コプロセッサは、モータ制御、計量、信号処理、その他多くのアプリケーションに共通して使用される演算関数（主として三角関数）のハードウェア高速化を提供します。

ソフトウェア実装に比べてこれらの関数の計算が高速化され、低い動作周波数での使用が可能となったり、他のタスクを実行するためにプロセッササイクルが開放されたりします。

20.2 CORDIC の主な機能

- 24bit CORDIC ローテーションエンジン
- 円モードと双曲線モード
- 回転モードとベクタリングモード
- 関数 : sine、cosine、sinh、cosh、atan、atan2、atanh、modulus、平方根、自然対数
- プログラム可能な精度
- 低遅延 AHB スレーブインタフェース
- 結果はポーリングや割込みを行わずに準備が出来次第読出し可能
- DMA 読出し／書込みチャンネル
- DMA による複数レジスタの読出し／書込み

20.3 CORDIC の機能詳細

20.3.1 概要

CORDIC は、三角関数と双曲線関数の値を求めるための低コストな逐次近似アルゴリズムです。

三角関数（円）モードでは、ある角度 θ の正弦と余弦は、回転角の累積和が入力角 θ に等しくなるまで単位ベクトル $[1, 0]$ を回転させることで求められます。そこで、回転されたベクトルのデカルト座標成分 x と y は、 θ の余弦と正弦にそれぞれ対応しています。反対に、逆正接 (y / x) に対応するベクトル $[x, y]$ の角度は、単位ベクトル $[1, 0]$ を得るために角度を連続的に減少させて、 $[x, y]$ を回転させることにより求められます。回転角の累積和から元ベクトルの角度が得られます。

CORDIC アルゴリズムは、連続した円回転を双曲線に沿ったステップに置き換えることにより、双曲線関数（sinh、cosh、atanh）の計算にも使用できます。

その他の関数は、上記の基本関数から導出可能です。

20.3.2 CORDIC 関数

コプロセッサを使用する際の最初のステップは、CORDIC_CR レジスタの FUNC フィールドをしかるべくプログラムすることにより、必要な関数を選択することです。

CORDIC コプロセッサでサポートしている関数を 表 145 に列記します。

表 145. CORDIC 関数

機能	第 1 引数 (ARG1)	第 2 引数 (ARG2)	第 1 結果 (RES1)	第 2 結果 (RES2)
余弦	角度 θ	モジュラス m	$m \cdot \cos \theta$	$m \cdot \sin \theta$
正弦	角度 θ	モジュラス m	$m \cdot \sin \theta$	$m \cdot \cos \theta$
位相	x	y	$\text{atan2}(y,x)$	$\sqrt{x^2 + y^2}$
モジュラス	x	y	$\sqrt{x^2 + y^2}$	$\text{atan2}(y,x)$
逆正接	x	なし	$\tan^{-1} x$	なし
双曲線余弦	x	なし	$\cosh x$	$\sinh x$
双曲線正弦	x	なし	$\sinh x$	$\cosh x$
双曲線逆正接	x	なし	$\tanh^{-1} x$	なし
自然対数	x	なし	$\ln x$	なし
平方根	x	なし	\sqrt{x}	なし

いくつかの関数は 2 つの入力引数 (ARG1 と ARG2) をとりますし、2 つの結果 (RES1 と RES2) を同時に生成するものもあります。これはアルゴリズムの副作用であり、2 つの値を得るために必要な演算は 1 回のみであることを意味します。これは、たとえば極座標を直交座標に変換する場合です。 $\sin \theta$ によって $\cos \theta$ も生成され、 $\cos \theta$ によって $\sin \theta$ も生成されます。直交座標の極座標変換 ($\text{phase}(x,y)$ 、 $\text{modulus}(x,y)$) や、双曲線関数 ($\cosh \theta$ 、 $\sinh \theta$) でも同様です。

注： 指数関数 $\exp x$ は、 $\sinh x$ と $\cosh x$ の和として求められます。また、 N を底とする対数 $\log_N x$ は、 $\ln x$ に定数 K (ここで $K = 1/\ln N$) を乗じることにより導出できます。

特定の関数 (atan 、 \log 、 sqrt) では、スケーリングファクタ (セクション 20.3.4 参照) を適用することにより、q1.31 固定小数点に対応している最大値 $[-1, 1]$ を超えて関数の範囲を拡大できます。スケーリングファクタは、その他すべての円関数では 0 に、双曲線関数では 1 にセットする必要があります。

余弦

表 146. 余弦パラメータ

パラメータ	説明	範囲
ARG1	ラジアン単位の角度 θ を π で割ったもの	$[-1, 1]$
ARG2	モジュラス m	$[0, 1]$

表 146. 余弦パラメータ (続き)

パラメータ	説明	範囲
RES1	$m \cdot \cos \theta$	[-1, 1]
RES2	$m \cdot \sin \theta$	[-1, 1]
SCALE	適用外	0

この関数は、 $-\pi$ から π の範囲の角度の余弦を計算します。極座標から直交座標への変換にも使用できます。

第 1 引数は、ラジアン単位の角度 θ です。ARG1 をプログラムする前に、 π で割る必要があります。

第 2 引数はモジュラス m です。 m が 1 よりも大きい場合には、ソフトウェアでスケーリングを適用して q1.31 範囲の ARG2 とする必要があります。

第 1 結果 RES1 は、角度の余弦にモジュラスを掛けた値です。

第 2 結果 RES2 は、角度の正弦にモジュラスを掛けた値です。

正弦

表 147. 正弦パラメータ

パラメータ	説明	範囲
ARG1	ラジアン単位の角度 θ を π で割ったもの	[-1, 1]
ARG2	モジュラス m	[0, 1]
RES1	$m \cdot \sin \theta$	[-1, 1]
RES2	$m \cdot \cos \theta$	[-1, 1]
SCALE	適用外	0

この関数は、 $-\pi$ から π の範囲の角度の正弦を計算します。極座標から直交座標への変換にも使用できます。

第 1 引数は、ラジアン単位の角度 θ です。ARG1 をプログラムする前に、 π で割る必要があります。

第 2 引数はモジュラス m です。 m が 1 よりも大きい場合には、ソフトウェアでスケーリングを適用して q1.31 範囲の ARG2 とする必要があります。

第 1 結果 RES1 は、角度の正弦にモジュラスを掛けた値です。

第 2 結果 RES2 は、角度の余弦にモジュラスを掛けた値です。

位相

表 148. 位相パラメータ

パラメータ	説明	範囲
ARG1	x 座標	[-1, 1]
ARG2	y 座標	[-1, 1]
RES1	ラジアン単位の位相角 θ を π で割ったもの	[-1, 1]
RES2	モジュラス m	[0, 1]
SCALE	適用外	0

この関数は、ベクトル $\mathbf{v} = [x \ y]$ の $-\pi$ から π の範囲にある位相角を計算します ($\text{atan2}(y,x)$ とも言います)。直交座標から極座標への変換にも使用できます。

第 1 引数は x 座標、すなわち、ベクトルの x 軸方向の大きさです。 $|x| > 1$ の場合には、ソフトウェアでスケーリングを適用して $q1.31$ 範囲の ARG1 とする必要があります。

第 2 引数は y 座標、すなわち、ベクトルの y 軸方向の大きさです。 $|y| > 1$ の場合には、ソフトウェアでスケーリングを適用して $q1.31$ 範囲の ARG2 とする必要があります。

第 1 結果 RES1 はベクトル \mathbf{v} の位相角 θ で、 π を乗算して角度をラジアン単位で取得します。 π に近い値は、位相角の円の性質のために $-\pi$ に折り返す場合があることに注意してください。

第 2 結果 RES2 は、 $v = \text{SQRT}(x^2 + y^2)$ の式によって与えられるモジュラスです。 $|v| = \sqrt{x^2 + y^2}$ 。 $|v| > 1$ の場合には、RES2の結果は 1 に飽和しています。

モジュラス

表 149. モジュラス・パラメータ

パラメータ	説明	範囲
ARG1	x 座標	$[-1, 1]$
ARG2	y 座標	$[-1, 1]$
RES1	モジュラス m	$[0, 1]$
RES2	位相角 θ	$[-1, 1]$
SCALE	適用外	0

この関数は、ベクトル $\mathbf{v} = [x \ y]$ の大きさ（モジュラス）を計算します。直交座標から極座標への変換にも使用できます。

第 1 引数は x 座標、すなわち、ベクトルの x 軸方向の大きさです。 $|x| > 1$ の場合には、ソフトウェアでスケーリングを適用して $q1.31$ 範囲の ARG1 とする必要があります。

第 2 引数は y 座標、すなわち、ベクトルの y 軸方向の大きさです。 $|y| > 1$ の場合には、ソフトウェアでスケーリングを適用して $q1.31$ 範囲の ARG2 とする必要があります。

第 1 結果 RES1 は、 $v = \text{SQRT}(x^2 + y^2)$ の式によって与えられるモジュラスです。 $|v| = \sqrt{x^2 + y^2}$ 。 $|v| > 1$ の場合には、RES1の結果は 1 に飽和しています。

第 2 結果 RES2 はベクトル \mathbf{v} の位相角 θ で、 π を乗算して角度をラジアン単位で取得します。 π に近い値は、位相角の円の性質のために $-\pi$ に折り返す場合があることに注意してください。

逆正接

表 150. 逆正接パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[-1, 1]
ARG2	適用外	-
RES1	ラジアン単位の $2^{-n} \cdot \tan^{-1} x$ を π で割ったもの	[-1, 1]
RES2	適用外	-
SCALE	n	[0, 7]

この関数は、入力引数 x の逆正接を計算します。

第 1 引数 ARG1 は、入力値 $x = \tan \theta$ です。 $|x| > 1$ の場合には、 $-1 < x \cdot 2^{-n} < 1$ となるようにソフトウェアでスケールリングファクタ 2^{-n} を適用する必要があります。スケールリングされた値 $x \cdot 2^{-n}$ は ARG1 に、スケールファクタ n を SCALE パラメータにプログラムする必要があります。

許容される最大入力値は $\theta = 128$ であり、この値は角度 $\theta = 89.55$ 度に相当することに注意してください。 $|x| > 128$ の場合には、 $\tan^{-1} x$ を計算するためにソフトウェア手法を用いる必要があります。

第 2 引数 ARG2 は使用しません。

第 1 結果 RES1 は、 $\theta = \tan^{-1} x$ の角度です。ラジアン単位で角度を得るために、RES1 に $2^n \cdot \pi$ を掛ける必要があります。

第 2 結果 RES2 は使用しません。

双曲線余弦

表 151. 双曲線余弦パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[-0.559, 0.559]
ARG2	適用外	-
RES1	$2^{-n} \cdot \cosh x$	[0.5, 0.846]
RES2	$2^{-n} \cdot \sinh x$	[-0.683, 0.683]
SCALE	n	1

この関数は、双曲角 x の双曲線余弦を計算します。指数関数 $e^x = \cosh x + \sinh x$ と $e^{-x} = \cosh x - \sinh x$ を計算するためにも使用できます。

第 1 引数は、双曲角 x です。 $-1.118 \sim +1.118$ の範囲の x の値のみに対応しています。 $\cosh x$ の最小値は $q1.31$ フォーマットの範囲を超える 1 であるため、ソフトウェアでスケールリングファクタ 2^{-n} を適用する必要があります。SCALE パラメータにファクタ $n = 1$ をプログラムする必要があります。

第 2 引数は使用しません。

第 1 結果 RES1 は、双曲線余弦 $\cosh x$ です。正しい結果を得るために、RES1 に 2 を掛ける必要があります。

第 2 結果 RES2 は、双曲線正弦 $\sinh x$ です。正しい結果を得るために、RES2 に 2 を掛ける必要があります。

双曲線正弦

表 152. 双曲線正弦パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[-0.559, 0.559]
ARG2	適用外	-
RES1	$2^{-n} \cdot \sinh x$	[-0.683, 0.683]
RES2	$2^{-n} \cdot \cosh x$	[0.5, 0.846]
SCALE	n	1

この関数は、双曲角 x の双曲線正弦を計算します。指数関数 $e^x = \cosh x + \sinh x$ と $e^{-x} = \cosh x - \sinh x$ を計算するためにも使用できます。

第 1 引数は、双曲角 x です。-1.118~+1.118 の範囲の x の値のみに対応しています。すべての入力値に対して、ソフトウェアでスケールリングファクタ 2^{-n} ($n = 1$) を適用する必要があります。スケールリングされた値 $x \cdot 0.5$ を ARG1 に、ファクタ $n = 1$ を SCALE パラメータにプログラムする必要があります。

第 2 引数は使用しません。

第 1 結果 RES1 は、双曲線正弦 $\sinh x$ です。正しい結果を得るために、RES1 に 2 を掛ける必要があります。

第 2 結果 RES2 は、双曲線余弦 $\cosh x$ です。正しい結果を得るために、RES2 に 2 を掛ける必要があります。

双曲線逆正接

表 153. 双曲線逆正接パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[-0.403, 0.403]
ARG2	適用外	-
RES1	$2^{-n} \cdot \operatorname{atanh} x$	[-0.559, 0.559]
RES2	適用外	-
SCALE	n	1

この関数は、入力引数 x の双曲線逆正接を計算します。

第 1 引数 ARG1 は、入力値 x です。-0.806 to +0.806 の範囲の x の値のみに対応しています。値 x は、ファクタ 2^{-n} ($n = 1$) でスケールリングする必要があります。スケールリングされた値 $x \cdot 0.5$ を ARG1 に、ファクタ $n = 1$ を SCALE パラメータにプログラムする必要があります。

第 2 引数は使用しません。

第 1 結果 RES1 は、双曲線逆正接 $\operatorname{atanh} x$ です。正しい値を得るために、RES1 に 2 を掛ける必要があります。

第 2 結果は使用しません。

自然対数

表 154. 自然対数パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[0.054, 0.875]
ARG2	適用外	-
RES1	$2^{-(n+1)} \cdot \ln x$	[-0.279, 0.137]
RES2	適用外	-
SCALE	n	[1, 4]

この関数は、入力引数 x の自然対数を計算します。

第 1 引数 ARG1 は、入力値 x です。0.107~9.35 の範囲の x の値のみに対応しています。 x の値をファクタ 2^{-n} でスケーリングし、 $x \cdot 2^{-n} < 1 \cdot 2^{-n}$ となるようにします。スケーリングされた値 $x \cdot 2^{-n}$ を ARG1 にプログラムし、ファクタ n は SCALE パラメータにプログラムする必要があります。

有効なスケーリングファクタ n と、それに対応する角度 x と ARG1 の範囲を 表 155 に示します。

表 155. 自然対数のスケーリングファクタと対応範囲

n	x の範囲	ARG1 の範囲
1	$0.107 \leq x < 1$	$0.0535 \leq \operatorname{ARG1} < 0.5$
2	$1 \leq x < 3$	$0.25 \leq \operatorname{ARG1} < 0.75$
3	$3 \leq x < 7$	$0.375 \leq \operatorname{ARG1} < 0.875$
4	$7 \leq x \leq 9.35$	$0.4375 \leq \operatorname{ARG1} < 0.584$

第 2 引数 は使用しません。

第 1 結果は自然対数 $\ln x$ です。正しい値を得るために、RES1 に $2^{(n+1)}$ を掛ける必要があります。

第 2 結果は使用しません。

平方根

表 156. 平方根パラメータ

パラメータ	説明	範囲
ARG1	$x \cdot 2^{-n}$	[0.027, 0.875]
ARG2	適用外	-
RES1	$2^{-n} \sqrt{x}$	[0.04, 1]
RES2	適用外	-
SCALE	n	[0, 2]

この関数は、入力引数 x の平方根を計算します。

第 1 引数 ARG1 は、入力値 x です。0.027~2.34 の範囲の x の値のみに対応しています。値 x は、 $x \cdot 2^{-n} < (1 - 2^{-(n-2)})$ となるように、ファクタ 2^{-n} でスケーリングする必要があります。

スケーリングされた値 $x \cdot 2^{-n}$ は ARG1 に、ファクタ n を SCALE パラメータにプログラムする必要があります。

有効なスケーリングファクタ n と、それに対応する角度 x と ARG1 の範囲を [表 157](#) に示します。

表 157. 平方根のスケーリング・ファクタと対応範囲

n	x の範囲	ARG1 の範囲
0	$0.027 \leq x < 0.75$	$0.027 \leq \text{ARG1} < 0.75$
1	$0.75 \leq x < 1.75$	$0.375 \leq \text{ARG1} < 0.875$
2	$1.75 \leq x \leq 2.341$	$0.4375 \leq \text{ARG1} \leq 0.585$

第 2 引数 は使用しません。

第 1 結果は x の平方根です。正しい値を得るために、RES1 に 2^n を掛ける必要があります。

第 2 結果は使用しません。

20.3.3 固定小数点表現

CORDIC は符号付き固定小数点整数フォーマットで動作します。入出力値は q1.31 と q1.15 のいずれかです。

q1.31 フォーマットでは、数字は 1bit の符号部分と 31bit の小数部分（バイナリ小数位）によって表現されます。したがって、数値範囲は -1 (0x80000000) から $1 - 2^{-31}$ (0x7FFFFFFF) となります。

q1.15 フォーマットでは、数値範囲は 1 (0x8000) から $1 - 2^{-15}$ (0x7FFF) となります。このフォーマットには、2 個の入力引数が単一の 32bit 書込みに格納可能で、2 個の結果が 1 回の 32bit 読出しでフェッチ可能であるという利点があります。

20.3.4 スケーリングファクタ

[セクション 20.3.2](#) に示した関数のいくつかには、スケーリングファクタ SCALE が指定されています。これによって、入力、出力、内部レジスタを飽和させることなく、CORDIC が対応している値の範囲全体を対称として含むように、関数の入力範囲を拡張可能となります。スケーリングファクタが必要な場合には、ソフトウェアで計算して、CORDIC_CSR レジスタの SCALE フィールドにプログラムする必要があります。スケーリングされた値を CORDIC_WDATA レジスタにプログラムする前に、入力引数をそれに応じてスケーリングする必要があります。また、CORDIC_RDATA レジスタから読み出された結果に対して、スケーリングを元に戻す必要があります。

注： スケーリングされた値の切り捨てのため、必然的にスケーリングファクタには精度の低下が伴います。

20.3.5 Precision

結果の精度は、CORDIC の繰り返し回数に依存します。このアルゴリズムは、三角関数（正弦、余弦、位相、モジュラス）の繰り返し 1 回あたり 1 バイナリデジットの一定速度で収束します（[図 92](#) 参照）。

双曲線関数（双曲線正弦、双曲線余弦、自然対数）では、CORDIC アルゴリズムの特殊性のために収束速度の一定度は劣ります（[図 93](#) 参照）。平方根関数は、双曲線関数のおよそ 2 倍の速度で収束します（[図 94](#) 参照）。

図 92. 三角関数に対する CORDIC の収束

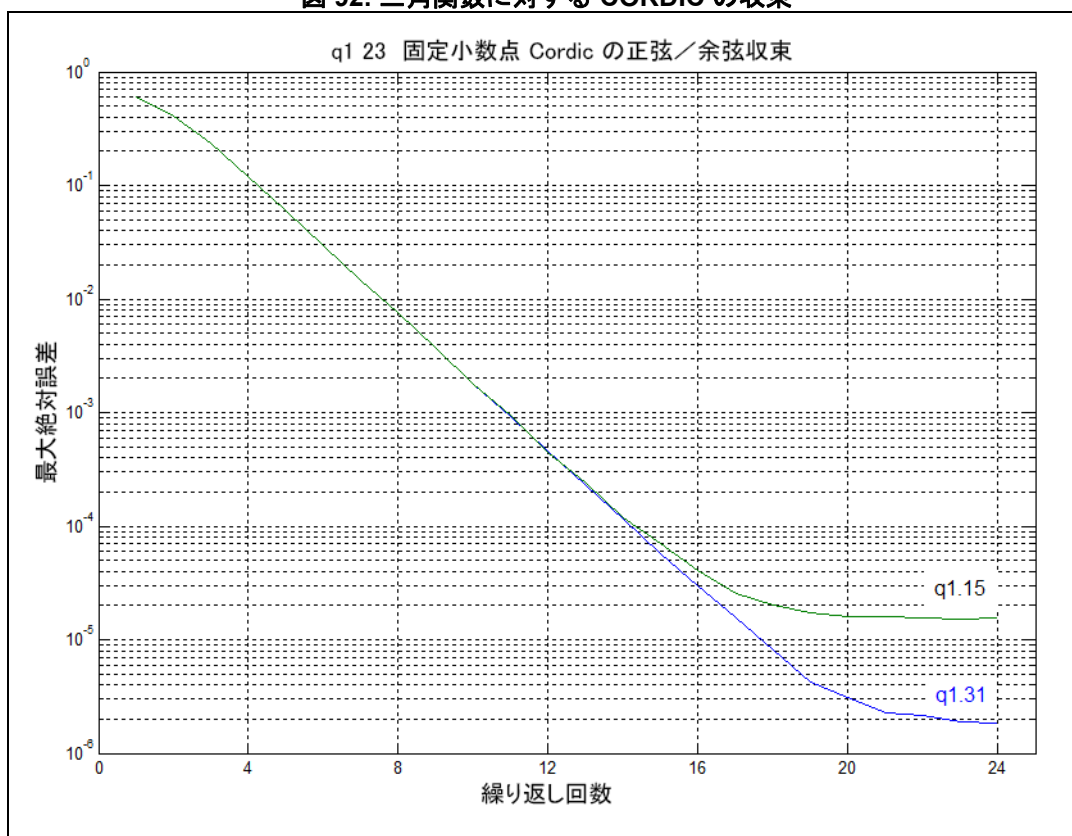


図 93. 双曲線関数に対する CORDIC の収束

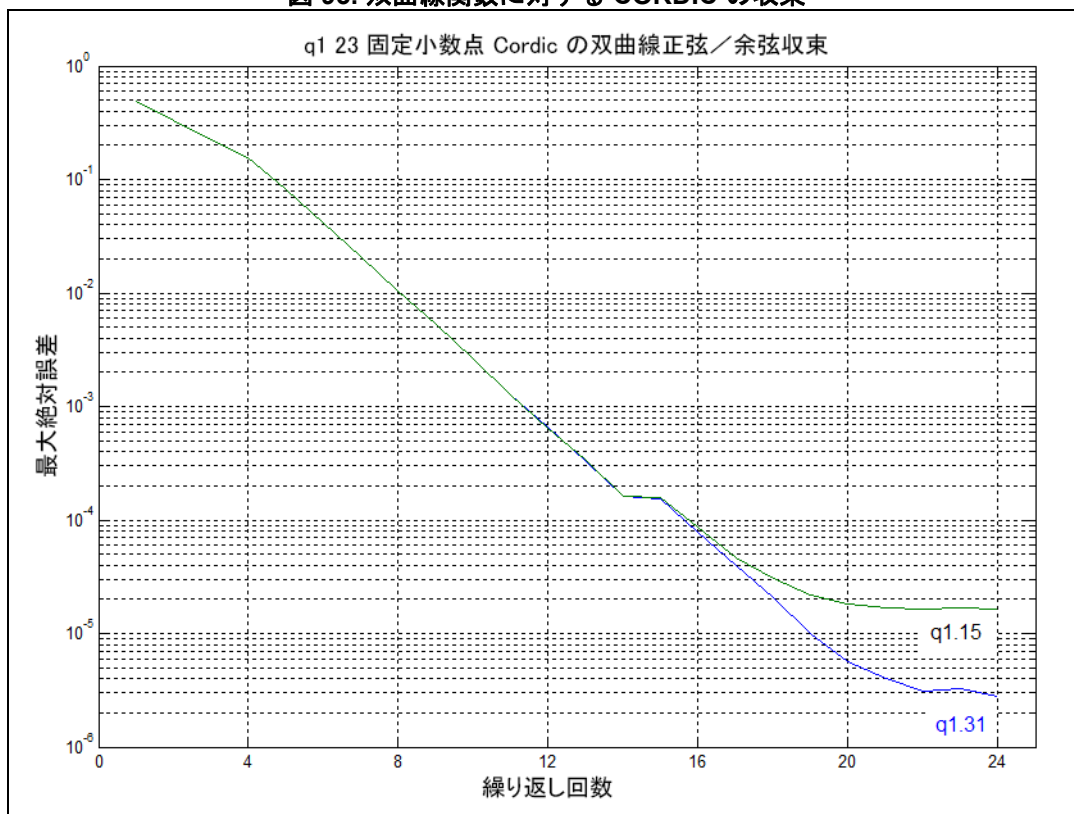
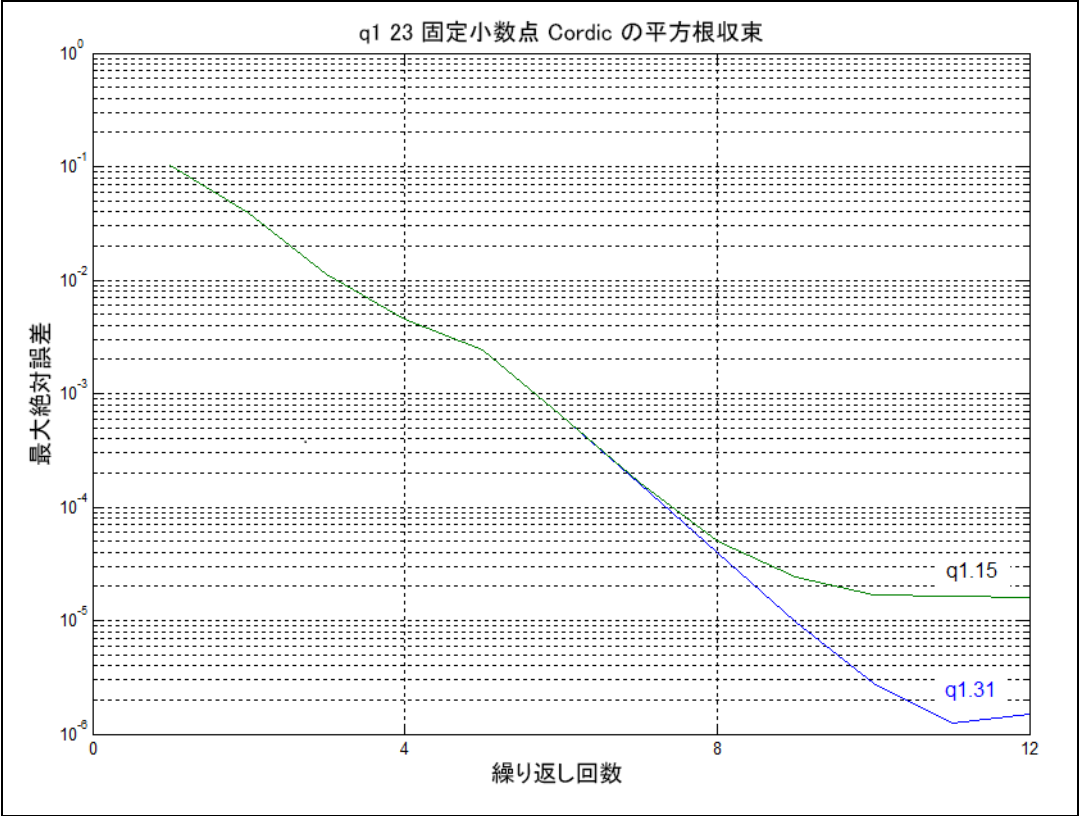


図 94. 平方根に対する CORDIC の収束



注： 収束速度は、量子化誤差が大きくなり始めるにつれて減少します。

CORDICは、1 クロックサイクルあたり 4 回の繰り返しを実行可能です。それぞれの関数に対して、繰り返しを 4 回ずつ実行した後に残留している最大誤差を、その精度に到達するために必要なクロックサイクル数とともに 表 158 に示します。この表から、必要なサイクル数を決定して、CORDIC_CR レジスタの PRECISION フィールドにプログラムすることができます。プログラムされた回数の繰り返しが完了するや否やコプロセッサが停止し、結果は直ちに読出し可能となります。

表 158. 精度対繰り返し回数

機能	繰り返し回数	サイクル数	最大残留誤差 ⁽¹⁾	
			q1.31 フォーマット	q1.15 フォーマット
Sin、Cos、 Phase ⁽²⁾ 、Mod、 Atan ⁽⁴⁾	4	1	2^{-3}	2^{-3}
	8	2	2^{-7}	2^{-7}
	12	3	2^{-11}	2^{-11}
	16	4	2^{-15}	2^{-15}
	20	5	2^{-18}	2^{-16}
	24	6	2^{-19}	2^{-16}

表 158. 精度対繰り返し回数 (続き)

機能	繰り返し回数	サイクル数	最大残留誤差 ⁽¹⁾	
			q1.31 フォーマット	q1.15 フォーマット
Sinh、Cosh、 Atanh、Ln ⁽³⁾	4	1	2^{-2}	2^{-2}
	8	2	2^{-6}	2^{-6}
	12	3	2^{-10}	2^{-10}
	16	4	2^{-13}	2^{-13}
	20	5	2^{-17}	2^{-15}
	24	6	2^{-18}	2^{-15}
Sqrt ⁽⁴⁾	4	1	2^{-7}	2^{-7}
	8	2	2^{-14}	2^{-14}
	12	3	2^{-19}	2^{-15}

1. 最大残留誤差とは、倍精度浮動小数点で実行された同じ計算と比較して、所定の繰り返し回数後に残っている最大誤差のことです。q15 フォーマットでは最大で 2^{-16} 、q31 フォーマットでは最大で 2^{-20} の追加の丸め誤差が生じることがあります。
2. モジュラスが 0.5 を上回る場合。到達可能な精度は、量子化誤差が大きくなるため、モジュラスの大きさに比例して減少します。
3. SCALE = 1。より大きなスケーリングファクタを用いると、それに比例して到達可能な精度が減少します。
4. SCALE = 0。より大きなスケーリングファクタを用いると、それに比例して到達可能な精度が減少します。

20.3.6 ゼロオーバーヘッドモード

コプロセッサを使用する最速の方法は、実行する関数 (FUNC)、必要なクロックサイクル数 (PRECISION)、入出力値のサイズ (ARGSIZE、RESSIZE)、入力引数の個数 (NARGS) や結果の個数 (NRES)、さらに、該当する場合にはスケーリングファクタ (SCALE) を CORDIC_CSR レジスタに前もってプログラムしておくことです。

その後、入力引数を CORDIC_WDATA レジスタに書き込むことにより、計算がトリガされます。正しい個数の入力引数が書き込まれると (かつ進行中の計算がすべて完了すると)、これらの入力引数ならびにその時点の CORDIC_CSR 設定値を用いて新しい計算が直ちに起動されます。変更がないのであれば、CORDIC_CSR レジスタを再プログラムする必要はありません。

32bit 入力引数が 2 個必要な場合 (ARGSIZE = 0、NARGS = 1)、最初に第 1 入力引数 ARG1 を書き込み、その後に第 2 引数 ARG2 を書き込む必要があります。一連の計算において第 2 引数が不変である場合には、最初の計算が始まったところで引数の個数を 1 (NARGS = 0) に再プログラムすることにより、2 番目の書き込みを行わないことができます。関数に変更されない限り、第 2 引数はプログラムされた値を保持します。

注： リセット後、ARG2 は +1 (0x7FFFFFFF) にセットされます。

16bit 引数が 2 個用いられる場合 (ARGSIZE = 1)、ARG1 を最下位ハーフワード、ARG2 を最上位ハーフワードとして 32bit ワード 1 個に格納する必要があります。すると、格納された 32bit ワードは CORDIC_WDATA レジスタに書き込まれます。この場合 (NARGS = 0) に必要な書き込み回数は 1 回のみです。

入力引数が 1 個のみ (ARG1) の関数では、NARGS = 0 とすることをお勧めします。NARGS = 1 とした場合、計算をトリガするには CORDIC_WDATA に対して 2 回目の書き込みを行う必要があります。この場合、ARG2 データは使用されません。

ひとたび計算が始まってから CORDIC_RDATA レジスタの読出しを試みると、結果を返す前に、計算が完了するまでバスにウェイト・ステートが挿入されます。したがって、ソフトウェアが入力を書き

込んだ直後に、有効かどうか確認するためのポーリングを行わずに結果を読み出すことができます。もしくは、プロセッサは、適切なクロックサイクル数だけ待ってから結果を読み出すこともできます。この時間は、次の計算のために CORDIC_CSR レジスタのプログラムを行い、必要に応じて次の入力データを準備することに使用可能です。CORDIC_CSR レジスタは計算の実行中に再プログラム可能であり、その計算結果に影響を与えることはありません。同様に、ひとたび前回の引数が反映されれば、次の引数で CORDIC_WDATA レジスタを更新できます。次の引数と設定値は、前回の計算が完了するまで保留されます。

計算が完了すると、その結果は CORDIC_RDATA レジスタから読み出すことができます。32bit の結果が 2 個期待されている場合 (NRES = 1、RESSIZE = 0)、最初に第 1 結果 (RES1) を読み出し、その後第 2 結果 (RES2) を読み出す必要があります。32bit の結果が 1 個だけ期待されている場合 (NRES = 0、RESSIZE = 0)、RES1 が 1 回目の読出しで出力されます。

16bit の結果が期待されている場合 (RESSIZE = 1)、CORDIC_RDATA を 1 回読み出すと、32bit ワードに格納された両方の結果がフェッチされます。RES1 は下位ハーフワード、RES2 は上位ハーフワードに格納されています。この場合には、NRES = 0 とプログラムすることをお勧めします。NRES = 1 とした場合、次の操作のために CORDIC を開放する目的で CORDIC_RDATA の 2 回目の読出しを行う必要があります。この 2 回目の読出しによるデータは破棄する必要があります。

次の計算は、期待される個数の引数が書き込み済みであることを前提として、期待される個数の結果が読み出されたときに始まります。このことは、常に、実行中または結果読出し待ちの計算が 1 つ、保留中の演算が 1 つ存在し得ることを意味しています。演算の保留中に CORDIC_WDATA にさらにアクセスすると、保留中の演算はキャンセルされ、データが上書きされます。

以下のシーケンスは、ゼロオーバーヘッドモードにおける CORDIC_IP の使用方法の要約です。

1. CORDIC_CSR レジスタに適切な設定値をプログラムします。
2. CORDIC_WDATA レジスタに最初の計算の引数をプログラムします。これによって最初の計算が起動されます。
3. 必要に応じて、次の計算のための設定値で CORDIC_CSR レジスタを更新します。
4. CORDIC_WDATA レジスタに次の計算の引数をプログラムします。
5. CORDIC_RDATA レジスタから結果を読み出します。これによって次の計算がトリガされます。
6. ステップ 3 に戻ります。

20.3.7 ポーリングモード

新しい結果が CORDIC_RDATA レジスタに有効になると、CORDIC_CSR レジスタに RRDY フラグがセットされます。このフラグは、そのレジスタを読み出すことによってポーリング可能です。CORDIC_RDATA レジスタを (CORDIC_CSR レジスタの NRES フィールドに応じて 1 回か 2 回) 読み出すと、このフラグはリセットされます。

有効になり次第結果が読み出されるのではないため、RRDY フラグのポーリングは、CORDIC_RDATA レジスタを直接読み出すよりも多少時間がかかります。しかしながら、CORDIC_CSR レジスタの読出しの間にプロセッサとバスインタフェースが停止しないことから、プロセッサの停止が許容されない場合 (低遅延割込みが必要な場合など) に、このモードが考慮の対象となる可能性があります。

20.3.8 割込みモード

CORDIC_CSR レジスタの割込みイネーブル (IE) ビットをセットすることにより、RRDY フラグがセットされるたびに割込みが生成されます。割込みは、このフラグがリセットされたときにクリアされます。

このモードによって、割込みサービスルーチンの中で計算結果を読み出すことが可能となり、他のタスクよりも高い優先度が与えられます。ただし、割込み処理の遅延のため、結果の直接読出しやフラグのポーリングよりも時間がかかります。

20.3.9 DMA モード

CORDIC_CSR レジスタに DMA 書込みイネーブル (DMAWEN) ビットがセットされ、保留となっている演算がない場合、DMA 書込みチャネルリクエストが生成されます。DMA コントローラは、メモリから CORDIC_WDATA レジスタへと第 1 入力引数 (ARG1) を転送可能となります。レジスタに書き込むと、DMA リクエストはデアサートされます。CORDIC_CSR レジスタが NARGS = 1 である場合、2 番目の DMA 書込みチャネルリクエストが生成され、第 2 入力引数 (ARG2) が CORDIC_WDATA レジスタに転送されます。入力引数がすべて書き込まれ、(結果の読出しによって) 実行中の計算がすべて完了すると、新しい計算が始まり、別の DMA 書込みチャネルリクエストが生成されます。

CORDIC_CSR レジスタに DMA 読出しイネーブル (DMAREN) ビットがセットされると、RRDY フラグがアクティブになり、DMA 読出しチャネルリクエストが生成されます。すると、DMA コントローラは、CORDIC_RDATA レジスタからメモリへと第 1 結果 (RES1) を転送可能となります。レジスタを読み出すと、DMA リクエストはデアサートされます。CORDIC_CSR レジスタが NRES = 1 である場合、2 番目の DMA リクエストが生成され、第 2 結果 (RES2) が読み出されます。結果がすべて読み出されると、RRDY フラグがデアサートされます。

DMA 読出しチャネルと書込みチャネルは、別々に有効化できます。両方のチャネルが有効化された場合、CORDIC は、プロセッサの介入なしに計算をデータバッファ上で自律的に繰り返すことができます。これにより、プロセッサは別のタスクを実行できます。DMA コントローラは、書込みチャネルでは“メモリからペリフェラル”モード、読出しチャネルでは“ペリフェラルからメモリ”モードで動作しています。プロセッサが DMAWEN フラグをセットするとシーケンスが開始されることに注意してください。それ以降、CORDIC がデータを処理可能な速度で DMA 読出しリクエストと DMA 書込みリクエストが生成されます。

場合によっては、入力データはメモリに格納され、出力は一定の時間間隔で D/A コンバータなどの別のペリフェラルへと転送されることがあります。この場合、転送先のペリフェラルは、新しいデータを必要とするたびに DMA リクエストを生成します。DMA コントローラは、次のサンプルを CORDIC_RDATA レジスタから直接フェッチすることができます (この場合、DMA コントローラは、転送元がペリフェラルレジスタであっても“メモリからペリフェラル”モードで操作しています)。結果を読み出す行為によって、CORDIC は新しい計算を開始できるようになりますが、その結果として DMA 書込みチャネルリクエストが生成され、DMA コントローラが次の入力値を CORDIC_WDATA レジスタに転送します。DMA 書込みチャネルは有効化 (DMAWEN = 1) されますが、読出しチャネルは有効化しないでください。

同様に、ADC など別ペリフェラルからのデータは、“ペリフェラルからメモリ”モードで CORDIC_WDATA レジスタに直接転送可能です。DMA 書込みチャネルは有効化しないでください。CORDIC は入力データを処理し、DMAREN = 1 の場合、それが完了すると DMA 読出しリクエストを生成します。すると、DMA コントローラは、CORDIC_RDATA レジスタからメモリへと結果を転送します (“ペリフェラルからメモリ”モード)。

注： CORDIC_CSR レジスタのプログラムのための DMA リクエストは生成されません。したがって、DMA モードが有用なのは、同じ関数と同じ設定で繰り返し実行する場合に限られます。スケーリング・ファクタは一連の DMA 転送中に変更することはできません。

注： CORDIC_WDATA レジスタや CORDIC_RDATA レジスタにアクセスする DMA の結果として、それぞれの DMA リクエストに確認応答を行う必要があります。これよりも前に当該レジスタへの無関係なアクセスが発生すると、確認応答が途中でアサートされ、DMA チャネルがブロックされることがあります。したがって、DMA 読出しチャネルが有効化されたときには、CPU による CORDIC_RDATA レジスタへのアクセスは避ける必要があります。同様に、DMA 書込みチャネルが有効化されたときには、プロセッサは CORDIC_WDATA レジスタへのアクセスを避ける必要があります。

20.4 CORDIC レジスタ

CORDIC レジスタには 32bit ワード・フォーマットでのみアクセスできます。

20.4.1 CORDIC 制御／ステータスレジスタ (CORDIC_CSR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARG SIZE	RES SIZE	NARGS	NRES	DMA WEN	DMA REN	IEN
r									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SCALE[2:0]			PRECISION[3:0]			FUNC[3:0]				
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **RRDY** : 結果レディ・フラグ

0 : 出力レジスタに新しいデータがありません。

1 : CORDIC_RDATA レジスタに新しいデータが入っています。

このビットは、CORDIC の演算が完了したときに、ハードウェアによってセットされます。CORDIC_RDATA レジスタが (NRES+1) 回読み込まれると、ハードウェアによってリセットされます。このビットがセットされたときに、IEN ビットもセットされていると、CORDIC 割込みがアサートされます。DMAREN ビットがセットされていると、DMA 読み出しチャネルリクエストが生成されます。このビットがセットされている間は、新しい計算は開始されません。

ビット 30:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **ARGSIZE** : 入力データの幅

0 : 32 ビット

1 : 16 ビット

ARGSIZE によって入力データの表現に用いられるビット数が選択されます。

32bit データが選択されると、CORDIC_WDATA レジスタは q1.31 フォーマットの引数を期待します。

16bit データが選択されると、CORDIC_WDATA レジスタは q1.15 フォーマットの引数を期待します。

最下位ハーフワードに第 1 引数 (ARG1)、最上位ハーフワードに第 2 引数 (ARG2) が書き込まれます。

ビット 21 **RESSIZE** : 出力データの幅

0 : 32 ビット

1 : 16 ビット

RESSIZE によって出力データの表現に用いられるビット数が選択されます。

32bit データが選択されると、CORDIC_RDATA レジスタには q1.31 フォーマットの結果が含まれます。

16bit データが選択されると、CORDIC_RDATA の最下位ハーフワードには第 1 結果 (RES1) が q1.15 フォーマットで含まれ、最上位ハーフワードには第 2 結果 (RES2) がこちらも q1.15 フォーマットで含まれます。

ビット 20

NARGS : CORDIC_WDATA レジスタが期待する引数の個数

0 : 次の計算には 1 回の 32bit 書き込みのみ (もしくは ARGSIZE = 1 の場合に 16bit 値 2 個) が必要です。

1 : 次の計算のトリガには、32bit 値を 2 個 CORDIC_WDATA レジスタに書き込む必要があります。

読み出すと、ビットの現在の状態が返されます。

ビット 19 **NRES** : CORDIC_RDATA レジスタ内の結果の個数

0 : 次の計算が完了すると、32bit 値 1 個のみ（もしくは RESSIZE = 1 の場合に 16bit 値 2 個）が CORDIC_RDATA レジスタに転送されます。CORDIC_RDATA から 1 回読み出すと、RRDY フラグがリセットされます。

1 : 次の計算が完了すると、32bit 値が 2 個 CORDIC_RDATA レジスタに転送されます。RRDY フラグのリセットには、CORDIC_RDATA から 2 回読み出す必要があります。

読み出すと、ビットの現在の状態が返されます。

ビット 18 **DMAWEN** : DMA 書込みチャネルの有効化

0 : 無効。DMA 書込みリクエストは生成されません。

1 : 有効となっています。保留中の演算が存在しない場合には、DMA 書込みチャネルにリクエストが常に生成されます。

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 17 **DMAREN** : DMA 読出しチャネルの有効化

0 : 無効。DMA 読出しリクエストは生成されません。

1 : 有効となっています。RRDY フラグがセットされるたびに、DMA 読出しチャネルにリクエストが生成されます。

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 16 **IEN** : 割込みの有効化

0 : 無効。割込みリクエストは生成されません。

1 : 有効となっています。RRDY フラグがセットされるたびに、割込みリクエストが生成されます。

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **SCALE[2:0]** : スケーリングファクタ

このフィールドの値は、引数や結果に適用されたスケーリングファクタを示します。値 n は、引数に係数 2^{-n} が掛けられているか、結果に 2^n を掛ける必要のあることを意味します。スケーリングファクタの各関数に対する適用可能性と適用範囲については、[セクション 20.3.2](#) を参照してください。

ビット 7:4 **PRECISION[3:0]** : 必要精度（繰り返し回数）

0 : 予約済み

1 から 15 : (繰り返し回数) / 4

所定の精度に対して必要となる繰り返し回数を求めるには [表 158](#) を参照してください。

多くの関数では、このフィールドの推奨範囲は 3~6 であることに注意してください。

ビット 3:0 **FUNC[3:0]** : 機能

0 : 余弦

1 : 正弦

2 : 位相

3 : モジュラス

4 : 逆正接

5 : 双曲線余弦

6 : 双曲線正弦

7 : 双曲線逆正接

8 : 自然対数

9 : 平方根

10 ~ 15 : 予約済み

20.4.2 CORDIC 引数レジスタ (CORDIC_WDATA)

アドレスオフセット : 0x04

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARG[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **ARG[31:0]** : 関数の入力引数

このレジスタには、CORDIC_CSR レジスタの FUNC フィールドで選択された関数に対する入力引数をプログラムします。

32bit フォーマットが選択され (CORDIC_CSR.ARGSIZE = 0)、入力引数が 2 個必要である場合 (CORDIC_CSR.NARGS = 1)、このレジスタに 2 回の連続した書込みが必要となります。1 回目には第 1 引数 (ARG1) を書き込み、2 回目には第 2 引数 (ARG2) を書き込みます。

32bit フォーマットが選択され、必要な入力引数は 1 個のみである場合 (NARGS = 0)、このレジスタに必要な書込みは第 1 引数 (ARG1) が含まれた 1 回のみとなります。

16bit フォーマットが選択された場合 (CORDIC_CSR.ARGSIZE = 1)、このレジスタに対する 1 回の書込みには両方の引数が含まれます。下位ハーフの ARG[15:0] に第 1 引数 (ARG1)、上位ハーフの ARG[31:16] に第 2 引数 (ARG2) が含まれます。この場合、NARGS は 0 にセットする必要があります。各関数に必要な引数とその許容範囲については、[セクション 20.3.2](#) を参照してください。

必要個数の引数を書き込まれると、前回の計算が完了していれば、CORDIC は、与えられた入力引数を用いて CORDIC_CSR.FUNC によって指定された関数を評価します。計算が実行中である場合、その計算が完了して結果が読み出されるまで、ARG1 と ARG2 の値は保留されます。この間にレジスタへの書込みがあると、保留中の演算はキャンセルされ、引数データが上書きされます。

20.4.3 CORDIC 結果レジスタ (CORDIC_RDATA)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 RES[31:0] : 関数の結果

32bit フォーマットが選択され (CORDIC_CSR.RESSIZE = 0)、出力値が 2 個期待されている場合 (CORDIC_CSR.NRES = 1)、このレジスタは RRDY がセットされたときに 2 回読み出される必要があります。1 回目の読み出しによって第 1 結果 (RES1) がフェッチされます。2 回目の読み出しによって第 2 結果 (RES2) がフェッチされ、RRDY がリセットされます。

32bit フォーマットが選択され、期待されている出力値は 1 個のみである場合 (NRES = 0)、第 1 結果 (RES1) のフェッチに必要なこのレジスタの読み出しは 1 回のみであり、RRDY がリセットされます。

16bit フォーマットが選択された場合 (CORDIC_CSR.RESSIZE = 1)、このレジスタの下位ハーフの RES[15:0] に第 1 結果 (RES1)、上位ハーフの RES[31:16] に第 2 結果 (RES2) が含まれます。この場合、NRES は 0 にセットする必要がある、実行される読み出しは 1 回のみとなります。

このレジスタを読み出すと、CORDIC_CSR レジスタの RRDY フラグがリセットされます。

20.4.4 CORDIC レジスタマップ

表 159. CORDIC レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CORDIC_CSR	RRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARG/SIZE	RESSIZE	NARGS	NRES	DMAWEN	DMAREN	IEN	Res.	Res.	Res.	Res.	SCALE [2:0]			PRECISION [3:0]			FUNC [3:0]					
	リセット値	0									0	0	0	0	0	0	0					0	0	0	0	0	1	0	1	0	0	0	0
0x04	CORDIC_WDATA	ARG[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x08	CORDIC_RDATA	RES[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

21 フィルタ演算アクセラレータ (FMAC)

21.1 FMAC の概要

フィルタ演算アクセラレータユニットは、ベクトルの算術演算を実行します。積和演算器 (MAC) ユニットから構成されており、ローカルメモリに保持されているベクトル要素のインデックス付けを可能とするアドレス生成ロジックを備えています。

このユニットには、デジタルフィルタの実装を可能とする、入出力上のサーキュラバッファサポートが含まれています。有限インパルス応答フィルタと無限インパルス応答フィルタのどちらも実現可能です。

このユニットによって、高頻度であったり時間がかかったりするフィルタリング動作を CPU からオフロードして、プロセッサを他のタスクに開放することができます。多くの場合には、このような計算をソフトウェアよりも高速に実行可能ですので、速度重視のタスクが高速化されます。

21.2 FMAC の主な機能

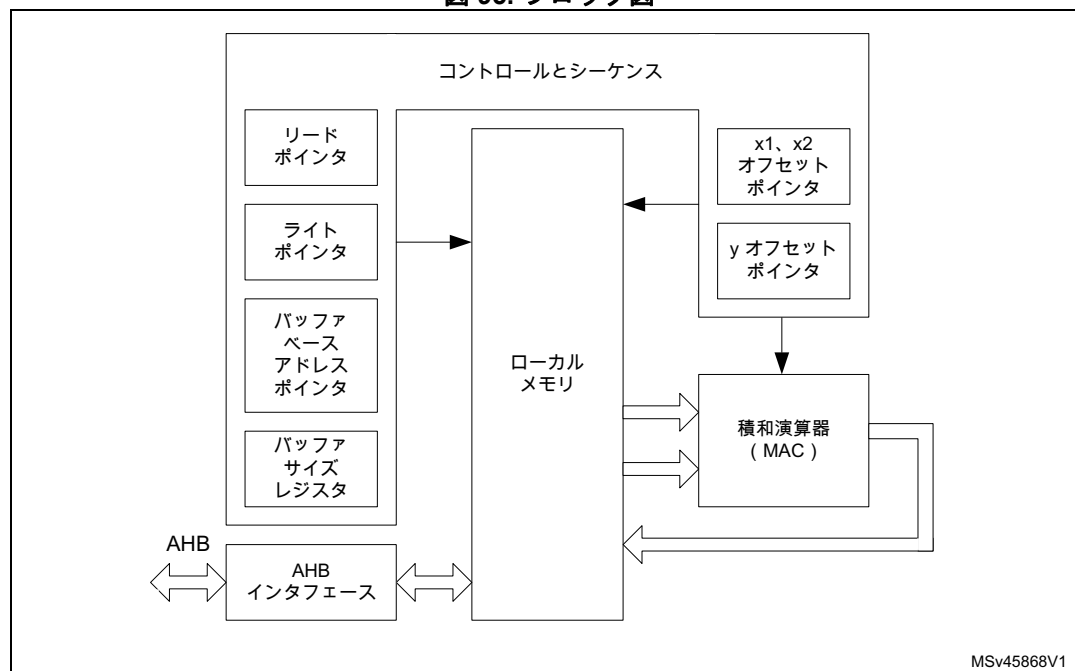
- 16 x 16bit 乗算器
- 加減算対応 24 + 2bit アキュムレータ
- 16bit 入出力データ
- 256 x 16bit ローカルメモリ
- メモリの最大 3 領域 (2 領域は入力、1 領域は出力) をデータバッファとして定義可能 (プログラム可能なベースアドレスポインタおよび付随するサイズレジスタによって定義)
- 入出力バッファはサーキュラバッファ対応可能
- フィルタ機能: FIR、IIR (直接型 1)
- ベクトル関数: ドット積、畳み込み、相関
- AHB スレーブインタフェース
- DMA 読出し/ライトデータチャネル

21.3 FMAC の機能詳細

21.3.1 概要

FMACを [図 95](#) に示します。

図 95. ブロック図



MSv45868V1

このユニットは、固定小数点積和演算器 (MAC) の周辺に構築されています。MAC は、メモリから 16bit 符号付き入力の値を 2 個とり、互いに掛け合わせてアキュムレータの内容に加算します。メモリの入力値アドレスは、ポインタへセットすることにより決定されます。これらのポインタは、内部ハードウェアによってロード、インクリメント、デクリメント、リセットが行えます。ポインタと MAC 演算は、要求された演算を実行するために、内蔵シーケンサによって制御されます。

ドット積を計算するために、プロセッサまたは DMA によって 2 本の入力ベクトルがローカルメモリにロードされ、要求された演算を選択して開始します。入力ベクトル要素の各ペアは、メモリからフェッチされ、互いに掛け合わされてから累積されます。ベクトル要素の処理がすべて完了するとアキュムレータの内容がローカルメモリに格納され、そこからプロセッサや DMA による読み出しが行えます。

有限インパルス応答 (FIR) フィルタ演算 (畳み込みとも呼ばれます) では、係数ベクトルと入力サンプル・ベクトルのドット積が繰り返し計算されます。繰り返しのたびに、後者を 1 サンプル・ディレイでシフトし、最も古いサンプルを破棄して新しいサンプルを追加します。

無限インパルス応答 (IIR) フィルタ演算では、フィードバック係数と前回の出力サンプルとの畳み込みが FIR 畳み込みの結果に加算されます。

フィルタ演算の詳細は、[セクション 21.3.6 : フィルタ機能](#) に記載されています。

21.3.2 ローカルメモリとバッファ

このユニットには、ローカルストレージとして用いられる 256 x 16 bit リード／ライトメモリが含まれています。

- 入力値（入力ベクトルの要素）は、バッファ X1 と X2 に格納されます。
- 出力値（演算結果）は、別のバッファ Y に格納されます。
- バッファのサイズと位置は、次のように指定されています。
 - x1_base : X1 バッファのベースアドレス
 - x2_base : X2 バッファのベースアドレス
 - y_base : Y バッファのベースアドレス
 - x1_buf_size : X1 バッファに割り当てられたデータ数（16bit ワード単位）
 - x2_buf_size : X2 バッファに割り当てられたデータ数（16bit ワード単位）
 - y_buf_size : Y バッファに割り当てられたデータ数（16bit ワード単位）

これらのパラメータは、ユニットの設定時に、対応するレジスタにプログラムされます。

CPU（または DMA コントローラ）は、初期化機能（[セクション 21.3.5 : 初期化機能](#)）を用いてライト・データ・レジスタに書き込むことにより、各バッファの内容を初期化できます。データは、ライトポインタによって示されるターゲットバッファ内の位置に転送されます。新たな書込みが終わるたびに、ライトポインタはインクリメントされます。ライトポインタが割り当てられたバッファ空間の最後に達すると、ベースアドレスに戻って折り返します。この機能は、演算の前にベクトルの要素をロードしたり、フィルタを初期化してフィルタ係数をロードしたりするために用いられます。

バッファ設定

X1、X2 および Y バッファ設定レジスタでバッファサイズとベースアドレスオフセットを設定する必要があります。各機能に必要なバッファサイズは、[セクション 21.3.6 : フィルタ機能](#)の機能詳細に規定されています。ベースアドレスは内部メモリのどこにでも選択可能ですが、すべてのバッファが内部メモリのアドレス範囲（0x00～0xFF）に収まるのが条件となっていますので、ベースアドレス + バッファサイズは 256 未満である必要があります。

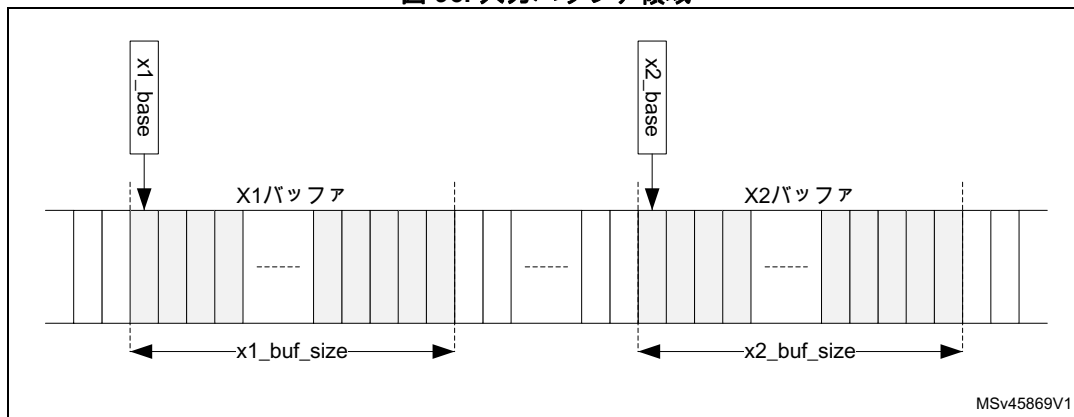
バッファのサイズと位置に関する制約はありません（オーバーラップしていても、完全に一致していても構いません）。フィルタ機能では、こうすると不正な動作を招くおそれがあるため、バッファをオーバーラップさせないことをお勧めします。

サーキュラバッファ演算が必要である場合には、オプションの「ヘッドルーム」*d* をバッファサイズに加算しても構いません。さらに、ウォーターマークレベルを設定して CPU や DMA の動作を調整することもできます。*d* の値とウォーターマークレベルは、アプリケーション性能要件に応じて選択する必要があります。スループットを最大とするためには、入力バッファは決して空となってはなりませんので、*d* はウォーターマーク・レベルよりも多少大きな値とし、割込みや DMA の遅延を可能とすべきです。その一方で、ユニットの処理可能速度に見合った速さで入力データを供給できない場合には、次のデータが書き込まれるのを待つためにバッファを空にすることができるので、（入力にオーバーフローが発生しないことを保証するために）*d* はウォーターマークレベルと等しくとも構いません。

21.3.3 入力バッファ

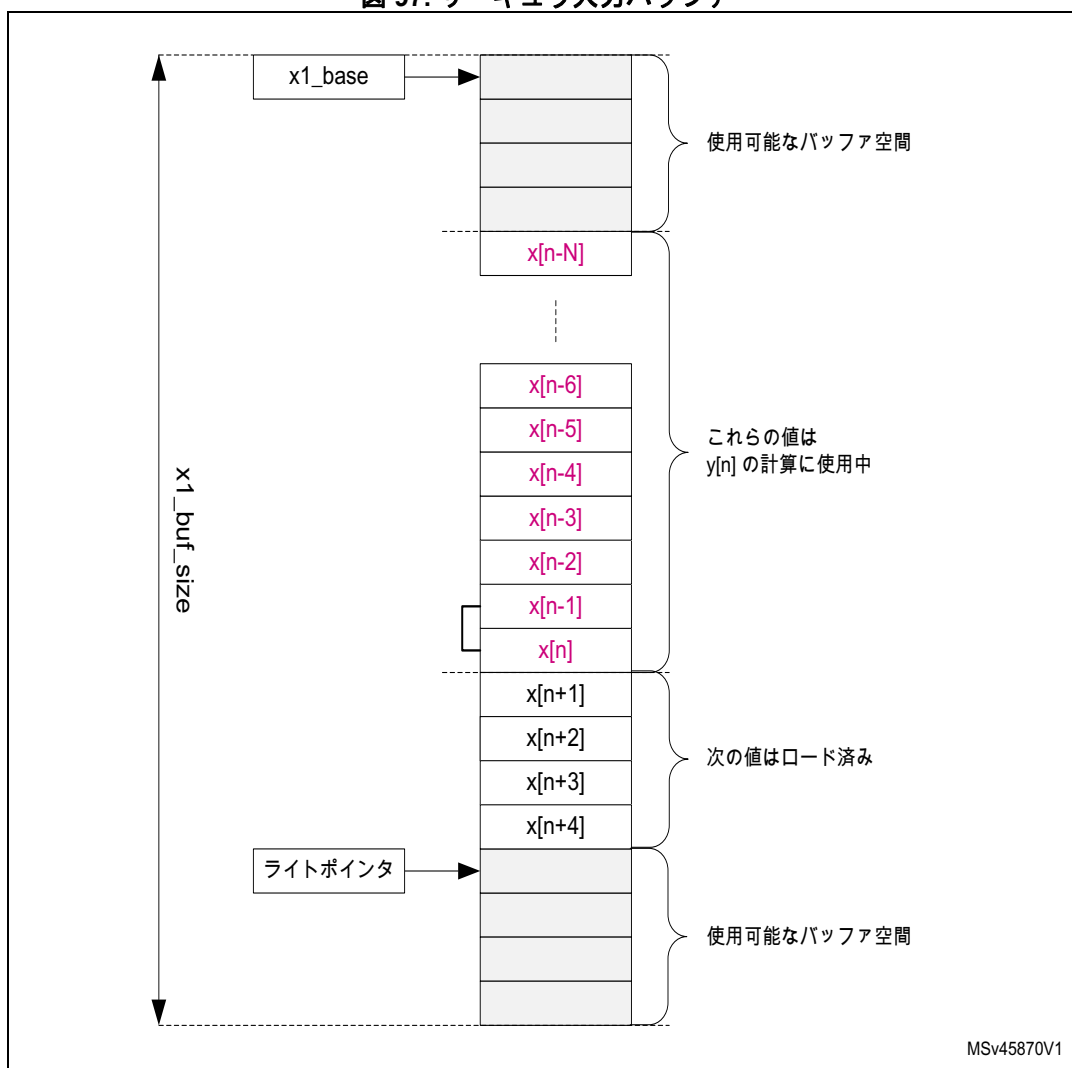
X1 バッファと X2 バッファは、MAC に入力するデータの格納に用いられます。乗算のたびに X1 バッファからの値と X2 バッファからの値が互いに掛け合わされます。制御ユニットのポインタによって、それぞれの値に対する（バッファベースアドレスを基準とした）リードアドレスオフセットが生成されます。ポインタは、現在の機能に応じてハードウェアによって管理されます。

図 96. 入力バッファ領域



X1 バッファはサーキュラバッファとして使用できますが、この場合、バッファに空きがあるときには必ず入力バッファに新しいデータが連続転送されます。デジタルフィルタでは、このバッファのプリロードは任意となります。それは、演算が開始されたときにバッファに入力サンプルが何も書き込まれていない場合にエンプティフラグが立ち、このフラグが、演算を開始するのに十分な量となるまで CPU または DMA が新しいサンプルをロードするトリガとなるためです。とはいえ、入力データがすでにシステムメモリの中にあってサーキュラ演算が必要ではないベクトル演算の場合は、プリロードは有用です。

図 97. サークュラ入力バッファ



MSv45870V1

X2 バッファは（サーキュラモードではなく）ベクトルモードでのみ使用可能であり、バッファの内容が 1 つの演算から次の演算まで変化しない場合を除いて、プリロードする必要があります。フィルタ機能に対しては、X2 バッファはフィルタ係数の格納に用いられます。

サーキュラバッファとしての動作時には、バッファ内に新しい値が常に存在しているようにするために、バッファに割り当てられたサイズ（ $x1_buf_size$ ）は、現在の計算に使用されている要素数よりも大きくしなければならないのが一般的です。図 97 にフィルタ動作のためのバッファレイアウトが図示されています。出力サンプル $y[n]$ を計算している間、このユニットは、 $x[n-N]$ から $x[n]$ まで $N+1$ 個の入力サンプルのセットを使用します。計算が完了すると、このユニットは、 $x[n-N+1]$ から $x[n+1]$ までの入力サンプルのセットを使用して $y[n+1]$ の計算を開始します。最も古い入力サンプル $x[n-N]$ が入力セットから抜け、新しいサンプル $x[n+1]$ が加わります。

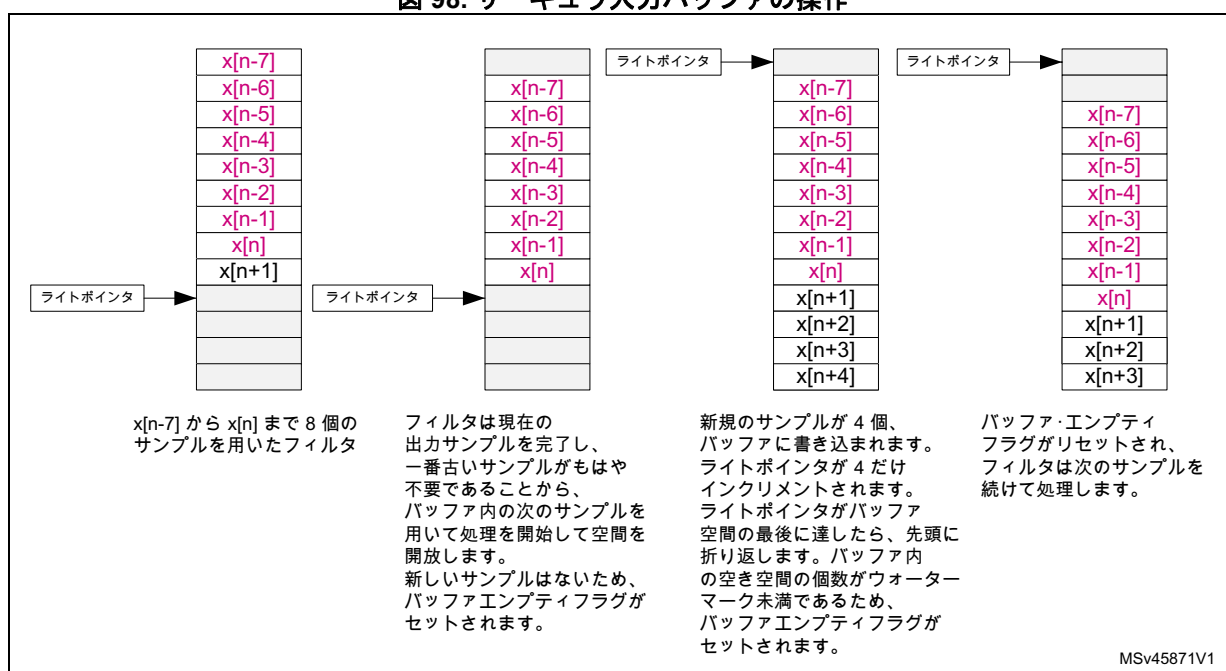
プロセッサや DMA コントローラは、必要時に新しいサンプル $x[n+1]$ がバッファの中で利用可能であることを保証する必要があります。そうならない場合、バッファにはエンプティフラグが立ち、新しいサンプルが追加されるまでユニットの実行は停止します。アンダーフロー状態は X1 バッファにシグナル送信されません。

注： タイマまたは ADC などその他のペリフェラルによってサンプルのフローが制御されている場合、転送元が供給可能な速度よりも、フィルタがそれぞれの新しいサンプルを処理する速度が速いため、バッファは定期的に空となります。これは、フィルタ演算の基本的な特徴です。

FMAC_X1BUFCFG レジスタの FULL_WM ビットフィールドにプログラムされたウォーターマーク閾値よりもバッファ内の空きスペースの個数の方が少ない場合、バッファにはフルフラグが立ちます。フルフラグがセットされていない限り、割込みが生成されて（有効となっている場合）バッファに追加データを要求します。ウォーターマークによって、オーバーフローの危険を伴わずに、1 回の割込みで複数のデータが転送可能となります。それにもかかわらずオーバーフローが発生した場合には、OVFL エラーフラグがセットされて、ライトデータは無視されます。オーバーフローが発生した場合には、ライト・ポインタがインクリメントされません。

フィルタ演算中の X1 バッファの操作を図 98 に図示します。この例には、ウォーターマークが 4 にセットされた 8 タップ FIR フィルタが示されています。

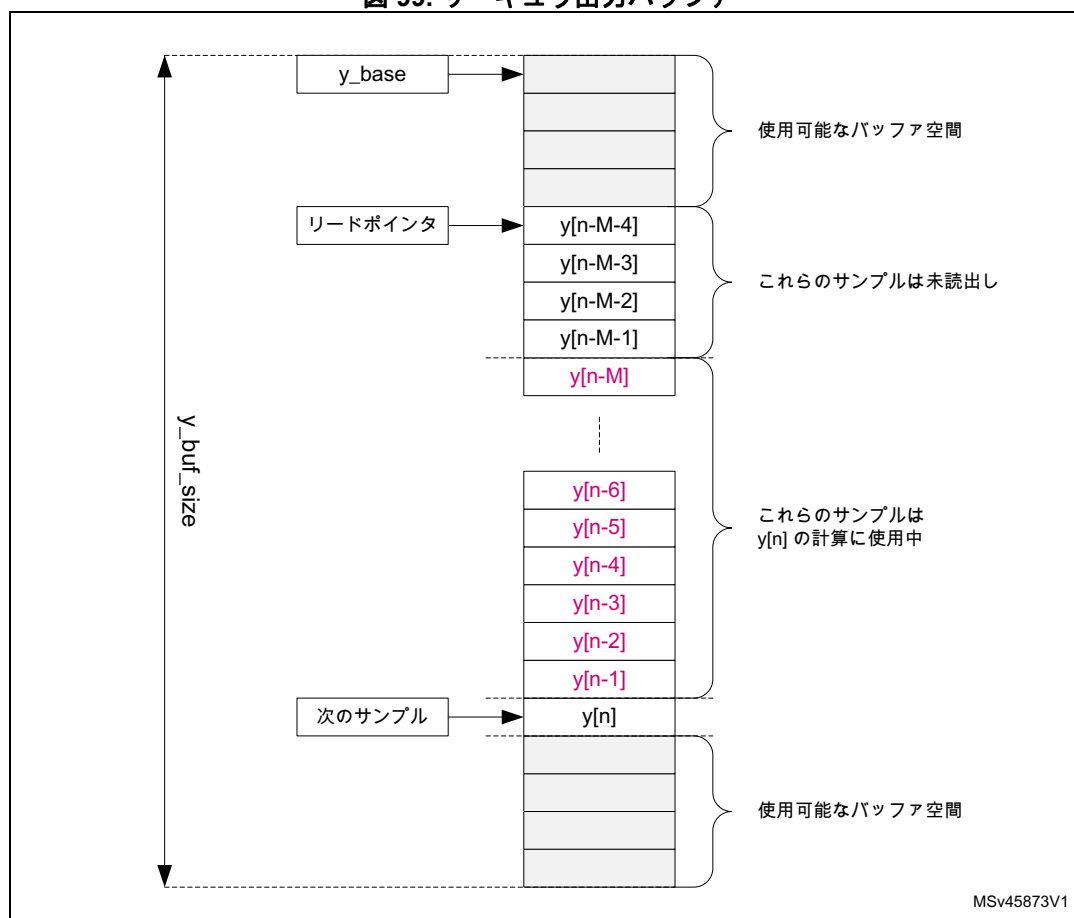
図 98. サークュラ入力バッファの操作



21.3.4 出力バッファ

Y（出力）バッファは、演算結果の格納に用いられます。それぞれの新しい出力値は、プロセッサが DMA コントローラによって読み出されるまでバッファの中に格納されています。リードデータレジスタに対してリードアクセスが行われるたびに、リードポインタで示されているアドレスからリードデータがフェッチされます。毎回の読出しの後にこのポインタはインクリメントされ、割り当てられた Y バッファ空間の最後に到達すると、ベースアドレスに戻って折り返されます。

図 99. サークュラ出力バッファ



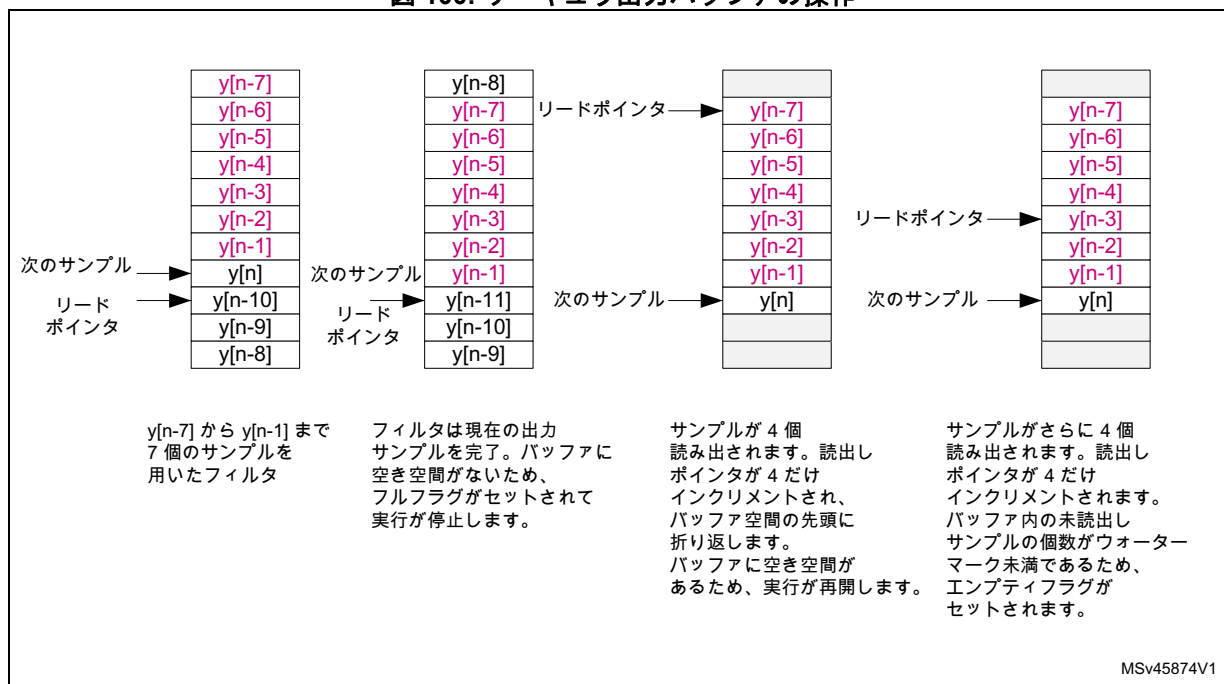
Y バッファはサーキュラバッファとしても使用可能です。次の出力値のアドレスがリード・ポインタで示されているものと同一である場合（未読出しのサンプル）には、バッファにはフルフラグが立ち、サンプルが読み出されるまで実行は停止します。

IIR フィルタの場合、Y バッファは、次の出力サンプル $y[n]$ の計算に使う $y[n-M]$ から $y[n-1]$ まで M 個の過去の出力サンプルセットの格納に用いられます。新しいサンプルがセットに追加されるたびに、最も古いサンプルである $y[n-M]$ が脱落します。

FMAC_YBUFCFG レジスタの EMPTY_WM ビットフィールドにプログラムされたウォーターマーク閾値よりもバッファ内の未読出しデータの個数の方が少ない場合、バッファにはエンプティフラグが立ちます。エンプティフラグがセットされていない限り、割り込みまたは DMA リクエストが生成されて（有効となっている場合）バッファからの読出しを要求します。ウォーターマークによって、アンダーフローの危険を伴わずに、1 回の割り込みで複数のデータが転送可能となります。それにもかかわらずアンダーフローが発生した場合には、UNFL エラーフラグがセットされます。この場合、リードポインタはインクリメントされずに、リードポインタアドレスにあるメモリの内容が読出し操作によって返されます。

サーキュラモードにおける Y バッファの操作を図 100 に図示します。この例には、ウォーターマークが 4 にセットされた 7 タップ IIR フィルタが示されています。

図 100. サークュラ出力バッファの操作



21.3.5 初期化機能

以下の機能によって FMAC ユニットが初期化されます。これらの機能は、START ビットをセットして、FMAC_PARAM レジスタの FUNC ビットフィールドに適切な値を書き込むことによってトリガされます。以下に示すように、P ビットフィールドと Q ビットフィールドにも各機能に適したパラメータ値が含まれている必要があります。R ビットフィールドは使用しません。この機能が完了すると、START ビットはハードウェアによって自動的にリセットされます。

初期化している間は、DMA リクエストと割り込みを無効化していくことをお勧めします。フロー制御は不要であるため、FMAC メモリへのデータ転送は、ソフトウェアでもメモリ間 DMA 転送でも行うことができます。

ロード X1 バッファ

この機能を使用すると、X1_BASE にあるアドレスから始まる N 個の値が X1 バッファにプリロードされます。FMAC_WDATA レジスタへの連続書き込みを行うと、X1 バッファにライト・データがロードされて、書き込みアドレスがインクリメントされます。この機能が完了すると、ライト・ポインタはアドレス X1_BASE + N を指します。

この機能は、あるベクトルの要素をバッファにプリロードしたり、あるフィルタの入カストレージ要素を初期化したりするのに使用できます。

パラメータ

- パラメータ P には、X1 バッファにロードされる値の個数 N が含まれています。
- パラメータ Q と パラメータ R は使用しません。

この機能は、FMAC_WDATA レジスタに N 回の書き込みが行われると完了します。

ロード X2 バッファ

この機能を使用すると、X2_BASE にあるアドレスから始まる N + M 個の値が X2 バッファにプリロードされます。FMAC_WDATA レジスタへの連続書込みを行うと、X2 バッファにライト・データがロードされて、書込みアドレスがインクリメントされます。

この機能は、ベクトルの要素やフィルタの係数をバッファにプリロードするのに使用できます。IIR の場合には、N 個のフィードフォワード係数と M 個のフィードバック係数が連結され、一緒に X2 バッファにロードされます。係数の総数は、N + M に等しくなります。FIR の場合には、フィードバック係数は存在しませんので、M = 0 となります。

パラメータ

- パラメータ P には、アドレス X2_BASE から始まって X2 バッファにロードされる値の個数 N が含まれています。
- パラメータ Q には、アドレス X2_BASE + N から始まって X2 バッファにロードされる値の個数 M が含まれています。
- パラメータ R は使用しません。

この機能は、FMAC_WDATA レジスタに N + M 回の書込みが行われると完了します。

ロード Y バッファ

この機能を使用すると、Y_BASE にあるアドレスから始まる N の値が Y バッファにプリロードされます。FMAC_WDATA レジスタへの連続書込みを行うと、Y バッファにライト・データがロードされて、書込みアドレスがインクリメントされます。この機能が完了すると、リード・ポインタはアドレス Y_BASE + N を指します。

この機能は、IIR フィルタのフィードバックストレージ要素をプリロードするのに使用できます。

パラメータ

- パラメータ P には、Y バッファにロードされる値の個数が含まれています。
- パラメータ Q と パラメータ R は使用しません。

この機能は、FMAC_WDATA レジスタに N 回の書込みが行われると完了します。

21.3.6 フィルタ機能

以下のフィルタ機能が FMAC ユニットによってサポートされています。これらの機能は、START ビットをセットして、FMAC_PARAM レジスタの FUNC ビットフィールドに該当する値を書き込むことによってトリガされます。以下に示すように、P、Q、R ビットフィールドにも各機能に適したパラメータ値が含まれている必要があります。フィルタ機能は、ソフトウェアによって START ビットがリセットされるまで継続します。

畳み込み (FIR フィルタ)

$$Y = B * X$$

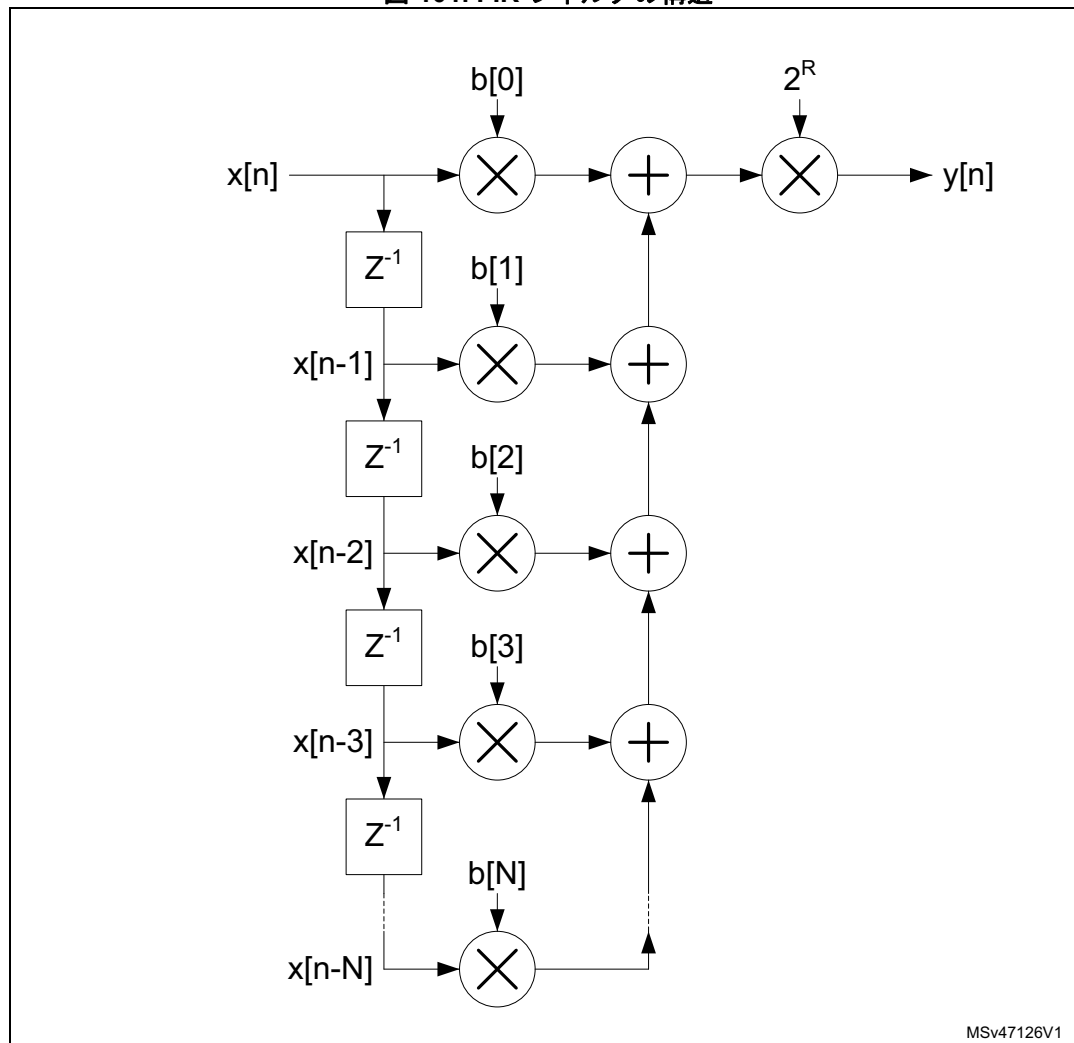
$$y_n = 2^R \cdot \sum_{k=0}^N b_k x_{n-k}$$

この機能は、長さが N+1 のベクトル **B** と不定長のベクトル **X** の畳み込みを行います。インクリメントする値 n に対する **Y** の要素は、ドット積 $y_n = \mathbf{B} \cdot \mathbf{X}_n$ として計算されます。ここで $\mathbf{X}_n = [x_{n-N}, \dots, x_n]$ は、インデックスが n - N から n まで N+1 個の **X** の要素から構成されます。

この機能は、有限インパルス応答 (FIR) フィルタに相当するものであり、ベクトル **B** にはフィルタ係数が、ベクトル **X** にはサンプルデータが含まれています。

フィルタの構造 (直接型) を 図 101 に示します。

図 101. FIR フィルタの構造



係数ベクトル **B** の次元を反対にすることにより、相互相関ベクトルの計算が可能であることに注意してください。

入力：

- X1 バッファにはベクトル **X** の要素が含まれています。長さが $N + 1 + d$ のサーキュラバッファとなります。
- X2 バッファにはベクトル **B** の要素が含まれています。長さが $N + 1$ の固定バッファとなります。

出力：

- Y バッファには出力値 y_n が含まれています。長さが d のサーキュラバッファとなります。

パラメータ :

- パラメータ P には、[2:127] の範囲の係数ベクトルの長さ N+1 が含まれています。
- パラメータ R には、アキュムレータ出力に適用されるゲインが含まれています。Y バッファに出力される値には 2^R が掛けられます。ここで、R は [0:7] の範囲です。
- パラメータ Q は使用しません。

この機能は、FMAC_PARAM レジスタの START ビットがソフトウェアによってリセットされると完了します。

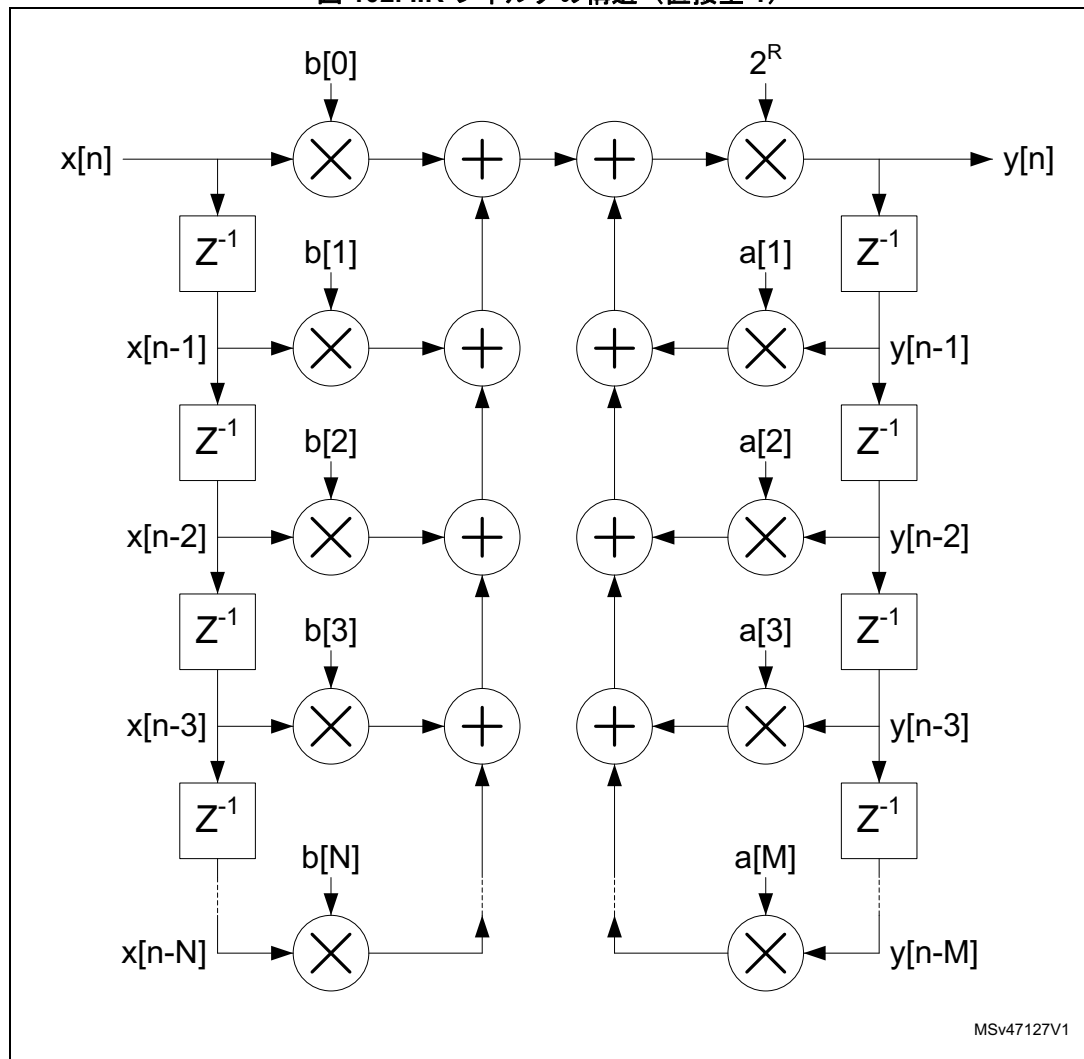
IIR フィルタ

$$\mathbf{Y} = \mathbf{B} * \mathbf{X} + \mathbf{A} * \mathbf{Y}'$$

$$y_n = 2^R \cdot \left(\sum_{k=0}^N b_k x_{n-k} + \sum_{k=1}^M a_k y_{n-k} \right)$$

この機能は、無限インパルス応答 (IIR) フィルタを実行します。フィルタ出力ベクトル \mathbf{Y} は、長さが N+1 の係数ベクトル \mathbf{B} と不定長のベクトル \mathbf{X} との畳み込みと、長さが M の 2 番目の係数ベクトル \mathbf{A} と遅延出力ベクトル \mathbf{Y}' との畳み込みの和です。インクリメントする値 n に対する \mathbf{Y} の要素は、 $y_n = \mathbf{B} * \mathbf{X}_n + \mathbf{A} * \mathbf{Y}_{n-1}$ として計算されます。ここで $\mathbf{X}_n = [x_{n-N}, \dots, x_n]$ は、インデックスが n - N から n までの \mathbf{X} の N+1 の要素から構成され、 $\mathbf{Y}_{n-1} = [y_{n-M}, \dots, y_{n-1}]$ はインデックスが n - M から n - 1 まで M 個の \mathbf{Y} の要素から構成されます。フィルタの構造 (直接型 1) を [図 102](#) に示します。

図 102. IIR フィルタの構造 (直接型 1)

**入力 :**

- X1 バッファにはベクトル **X** の要素が含まれています。長さが $N + 1 + d$ のサーキュラ・バッファとなります。
- X2 バッファには係数ベクトル **B** と **A** が連結された要素 ($b_0, b_1, b_2, \dots, b_N, a_1, a_2, \dots, a_M$) が含まれます。長さが $M + N + 1$ の固定バッファとなります。

出力 :

- Y バッファには出力値 y_n が含まれています。長さが $M + d$ のサーキュラバッファとなります。

パラメータ

- パラメータ P には、[2:64] の範囲の係数ベクトル **B** の長さ $N+1$ が含まれています。
- パラメータ Q には、[1:63] の範囲の係数ベクトル **A** の長さ M が含まれています。
- パラメータ R には、アキュムレータ出力に適用されるゲインが含まれています。Y バッファに出力される値には 2^R が掛けられます。ここで、R は [0:7] の範囲です。

この機能は、FMAC_PARAM レジスタの START ビットがソフトウェアによってリセットされると完了します。

21.3.7 固定小数点表現

FMAC は符号付き固定小数点整数フォーマットで動作します。入出力値は q1.15 です。

q1.15 フォーマットでは、数字は 1bit の符号部分と 15bit の小数部分（バイナリ小数位）によって表現されます。したがって、数値範囲は -1 (0x8000) から $1 - 2^{-15}$ (0x7FFF) となります。

アキュムレータは 26bit であり、そのうち 22bit が小数部分、4bit が整数／符号部分です (q4.22)。これによって、-8 (0x2000000) から +7.99999976 (0x1FFFFFFF) までの範囲の部分累積和に対応可能となっています。6dB ステップで 0dB から 42dB までプログラム可能なゲインをアキュムレータの出力に適用できます。

数値範囲を超過してもアキュムレータの内容は飽和しないことに注意してください。値が +7.99999976 を上回るか -8 を下回る部分積和は折り返されますが、それ以降の累積で折り返しを取り消されない限り、これは無害です。ただし、折り返しが発生すると FMAC_SR レジスタの SAT フラグがセットされ、FMAC_CR レジスタの SATIEN ビットがセットされている場合には割込みが発生します。これはフィルタのデバッグに役立ちます。

オプションとして、FMAC_CR レジスタの CLIPEN ビットをセットすることにより、プログラム可能なゲインを適用した後に、アキュムレータによるデータ出力を飽和させることも可能です。このビットがセットされると、q1.15 の数値範囲を超えるあらゆる値は、符合に応じて $1 - 2^{-15}$ または -1 に設定されます。クリッピングが有効化されていない場合、ゲインを適用した後に使われていないアキュムレータのビットは単純に切り詰められます。

21.3.8 FMAC を用いた FIR フィルタの実装

FMAC は 長さ N の FIR フィルタに対応しています。ここで、N はタップまたは係数の個数です。長さ N の FIR フィルタに対するローカルメモリの最小要件は $2N + 1$ です。

- 係数 N 個
- 入力サンプル N 個
- 出力サンプル 1 個

ローカルメモリのサイズは 256 です。N の最大値は 127 となります。

最大のスループットが必要な場合には、新しい入力サンプル待ちや出力サンプルの読出し待ちによりフィルタが停止することが絶対にないことを確実にするために、入力と出力のサンプルバッファそれぞれに対して、少量の追加スペース d1 と d2 を割り当てる必要があるかもしれません。この場合、ローカルメモリ要件は $2N + d1 + d2$ となります。

バッファは次のように設定する必要があります。

- $X1_BUF_SIZE = N + d1$
- $X2_BUF_SIZE = N$
- $Y_BUF_SIZE = d2$ (追加スペースが不要な場合は 1)

バッファ・ベース・アドレスはどこにでも割り当て可能ですが、X2 バッファは他のものとオーバーラップさせないでください。そうしないと、係数が上書きされます。次に設定例を示します。

- $X2_BASE = 0$
- $X1_BASE = N$
- $Y_BASE = 2N + d1$

ただし、メモリ空間が限られている場合には、各出力サンプルがもはや必要ではなくなった最も古い入力サンプルの場所を使用できるように、X1 バッファ領域と Y バッファ領域をオーバーラップさせることができます。

- X2_BASE = 0
- X1_BASE = N
- Y_BASE = N

この場合には Y_BUF_SIZE = X1_BUF_SIZE = N + d1 であり、バッファは同期したままとなります。

注： X1 バッファ設定レジスタの FULL_WM ビットフィールドは、 $\log_2(d1)$ 以下の値でプログラムする必要があります。そうしないと、入力サンプルが N 個書き込まれる前にバッファにフルフラグが立ってしまい、それ以上のサンプルがリクエストされません。同様に、Y バッファ設定レジスタの EMPTY_WM ビットフィールドは、 $\log_2(d2)$ 以下とする必要があります。

フィルタ係数は、ロード X2 バッファ機能を用いて X2 バッファにプリロードする必要があります。オプションとして、最大 N 個までの任意の個数のサンプルを X1 バッファにプリロードできます。FIR フィルタにはフィードバック・パスが存在しないため、Y バッファのプリロードは無意味です。

バッファの設定と初期化の後、FMAC メモリにデータを読み書きする方法に従って FMAC_CR レジスタをプログラムする必要があります。

以下の 3 種類の方法に対応しています。

- ポーリング：DMA リクエストや割込みリクエストは生成されません。ソフトウェアは、WDATA に書き込む前に X1_FULL フラグがローであることを、RDATA から読み出す前に Y_EMPTY フラグがローであることを確認する必要があります。
- 割込み：割込みリクエストは、書き込みでは X1_FULL フラグがロー、読出しでは Y_EMPTY フラグがローである間にアサートされます。
- DMA：DMA リクエストは、X1_FULL フラグがローである間に DMA ライトチャネルに、Y_EMPTY フラグがローである間にリードチャネルにアサートされます。

読出しと書き込みには別の方法を用いることもできます。ただし、同じ操作に対して割込みと DMA リクエストの両方を使用することは推奨されません^(a)。有効な組み合わせを表 160 に示します。

表 160. 読出し手法と書き込み手法の有効な組み合わせ

WIEN	RIEN	DMAWEN	DMAREN	書き込み	読出し
0	0	0	0	ポーリング	ポーリング
0	1	0	0	ポーリング	割込み
1	0	0	0	割込み	ポーリング
1	1	0	0	割込み	割込み
0	0	0	1	ポーリング	DMA
0	0	1	0	DMA	ポーリング
0	0	1	1	DMA	DMA
0	1	1	0	DMA	割込み
1	0	0	1	割込み	DMA

a. 割込みと DMA リクエストの両方が有効化された場合には、DMA のみが転送を行うようにしなくてはなりません。

フィルタは、FMAC_PARAM レジスタに次のビットフィールド値を書き込むことによって開始されます。

- FUNC = 8 (FIR フィルタ)
- P = N (係数の個数)
- Q = 「無視」
- R = ゲイン
- START = 1

X1 バッファにプリロードされている値が $N + d - 2^{\text{FULL_WM}}$ 個未満である場合、X1FULL フラグはローのままとなります。FMAC_CR レジスタの WIEN ビットがセットされると、直ちに割込みリクエストがアサートされて、追加で $2^{\text{FULL_WM}}$ 個のサンプルを FMAC_WDATA レジスタ経由でバッファに書き込むようにプロセッサにリクエストします。FMAC_SR レジスタで X1FULL フラグがハイになるまで、アサートされたままとなります。割込みサービスルーチンは、FMAC_WDATA レジスタへの $2^{\text{FULL_WM}}$ 回の書き込みのたびに X1FULL フラグを確認し、そのフラグがハイになるまで転送を繰り返す必要があります。同様に、FMAC_CR レジスタの DMAWEN ビットがセットされた場合には、X1FULL フラグがハイになるまで DMA 書き込みチャンネル・リクエストが生成されます。

(プリロードされたすべてのサンプルを含めて) 少なくとも N 個のサンプルが X1 バッファに書き込まれると、フィルタは最初の出力サンプルの計算を行います。

$2^{\text{EMPTY_WM}}$ 個の出力サンプルが Y バッファに書き込まれると、FMAC_SR レジスタの YEMPTY フラグがローになります。FMAC_CR レジスタの RIEN ビットがセットされると、割込みリクエストがアサートされて、追加で $2^{\text{EMPTY_WM}}$ 個のサンプルを FMAC_RDATA レジスタ経由でバッファから読み出すようにプロセッサにリクエストします。YEMPTY フラグがハイになるまで、アサートされたままとなります。割込みサービスルーチンは、FMAC_RDATA レジスタからの $2^{\text{EMPTY_WM}}$ 回の書き込みのたびに YEMPTY フラグを確認し、そのフラグがハイになるまで転送を繰り返す必要があります。FMAC_CR の DMAREN ビットがセットされた場合には、YEMPTY フラグがハイになるまで DMA 読み出しチャンネル・リクエストが生成されます。

ソフトウェアが START ビットをリセットして停止されるまで、フィルタはこの方法での動作を継続します。

21.3.9 FMAC を用いた IIR フィルタの実装

FMAC は長さが N の IIR フィルタに対応しています。ここで、N はフィードフォワードタップまたは係数の個数です。フィードバック係数の個数 M は 1 から N - 1 までの任意の値とできます。直接型 1 の実装のみが実現可能ですので、それ以外の型のために設計されたフィルタは変換する必要があります。

フィードフォワード係数 N 個、フィードバック係数 M 個の IIR フィルタに対するメモリの最小要件は $2N + 2M$ です。

- 係数 $N + M$ 個
- 入力サンプル N 個
- 出力サンプル M 個

$M = N - 1$ である場合、実装可能な最大フィルタ長は $N = 64$ となります。

FIR の場合、最大スループットのために、入力と出力のバッファサイズそれぞれに対して、少量の追加空間 d1 と d2 を与えるので、合計のメモリ要件は $2M + 2N + d1 + d2$ となります。

バッファは次のように設定する必要があります。

- X1_BUF_SIZE = $N + d1$
- X2_BUF_SIZE = $N + M$
- Y_BUF_SIZE = $M + d2$

バッファベースアドレスはどこにでも割り当て可能ですが、オーバーラップはさせないでください。次に設定例を示します。

- $X2_BASE = 0$
- $X1_BASE = N + M$
- $Y_BASE = 2N + M + d1$

注： X1 バッファ設定レジスタの FULL_WM ビットフィールドは、 $\log_2(d1)$ 以下の値でプログラムする必要があります。そうしないと、入力サンプルが N 個書き込まれる前にバッファにフルフラグが立ってしまい、それ以上のサンプルがリクエストされません。同様に、Y バッファ設定レジスタの EMPTY_WM ビットフィールドは、 $\log_2(d2)$ 以下とする必要があります。

フィルタ係数（フィードフォワード係数 N 個の後にフィードバック係数 M 個）は、ロード X2 バッファ機能を用いて X2 バッファにプリロードする**必要があります**。オプションとして、最大 N 個までの任意の個数のサンプルを X1 バッファにプリロードできます。オプションとして、最大 M 個までの任意の個数の値を Y バッファにプリロードできます。これは、フィードバック・ディレイ・ラインを初期化する効果があります。

バッファの設定の後、FIR フィルタと同じ方法で FMAC_CR レジスタをプログラムする必要があります（[セクション 21.3.8 : FMAC を用いた FIR フィルタの実装](#)参照）。

フィルタは、FMAC_PARAM レジスタに次のビットフィールド値を書き込むことによって開始されます。

- $FUNC = 9$ (IIR フィルタ)
- $P = N$ (フィードフォワード係数の個数)
- $Q = M$ (フィードバック係数の個数)
- $R = \text{ゲイン}$
- $START = 1$

X1 バッファにプリロードされている値が $N + d - 2^{FULL_WM}$ 個未満である場合、X1FULL フラグはローのままとなります。FMAC_CR レジスタの WIEN ビットがセットされると、直ちに割込みリクエストがアサートされて、追加で 2^{FULL_WM} 個のサンプルを FMAC_WDATA レジスタ経由でバッファに書き込むようにプロセッサにリクエストします。FMAC_SR レジスタで X1FULL フラグがハイになるまで、アサートされたままとなります。割込みサービスルーチンは、FMAC_WDATA レジスタへの 2^{FULL_WM} 回の書き込みのたびに X1FULL フラグを確認し、そのフラグがハイになるまで転送を繰り返す必要があります。同様に、FMAC_CR レジスタの DMAWEN ビットがセットされた場合には、X1FULL フラグがハイになるまで DMA 書き込みチャンネル・リクエストが生成されます。

(プリロードされたすべてのサンプルを含めて) 少なくとも N 個のサンプルが X1 バッファに書き込まれると、フィルタは最初の出力サンプルの計算を行います。X1 バッファの最初の N 個のサンプルと、Y バッファの最初の M 個のサンプルを使用して (プリロードされているか否かにはよりますが)、最初のサンプルが計算されます。最初の出力サンプルは、Y バッファの中の $Y_BASE + M$ に書き込まれます。

2^{EMPTY_WM} 個の新たな出力サンプルが Y バッファに書き込まれると、FMAC_SR レジスタの YEMPTY フラグがローになります。FMAC_CR レジスタの RIEN ビットがセットされると、割込みリクエストがアサートされて、追加で 2^{EMPTY_WM} 個のサンプルを FMAC_RDATA レジスタ経由でバッファから読み出すようにプロセッサにリクエストします。YEMPTY フラグがハイになるまで、アサートされたままとなります。割込みサービスルーチンは、FMAC_RDATA レジスタからの 2^{EMPTY_WM} 回の書き込みのたびに YEMPTY フラグを確認し、そのフラグがハイになるまで転送を繰り返す必要があります。FMAC_CR の DMAREN ビットがセットされた場合には、YEMPTY フラグがハイになるまで DMA 読出し・チャンネル・リクエストが生成されます。

ソフトウェアが START ビットをリセットして停止されるまで、フィルタはこの方法での動作を継続します。

21.3.10 フィルタ初期化の例

図 103. X1 バッファの初期化

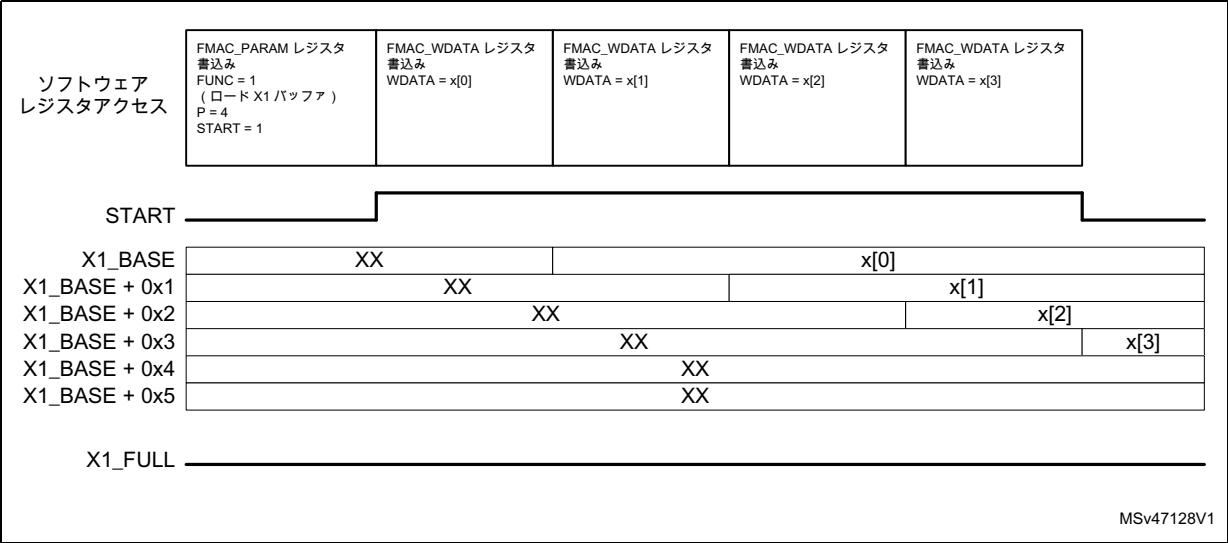


図 103 の例には、4 個のサンプル (P = 4) を用いた X1 バッファのプリロードが図示されています。バッファサイズは 6 です (X1_BUF_SIZE = 6)。初期化は、START ビットをセットして FMAC_PARAM レジスタをプログラムすることによって起動されます。すると、4 個のサンプルは FMAC_WDATA に書き込まれ、X1_BASE 以降のローカルメモリに転送されます。4 番目のサンプルが書き込まれると START ビットはリセットされます。この時点で、X1 バッファには書き込み順に 4 個のサンプルが格納されており、ライトポインタ (次の空き空間) は X1_BASE + 0x4 となっています。

21.3.11 フィルタ演算の例

図 104. フィルタ例 1

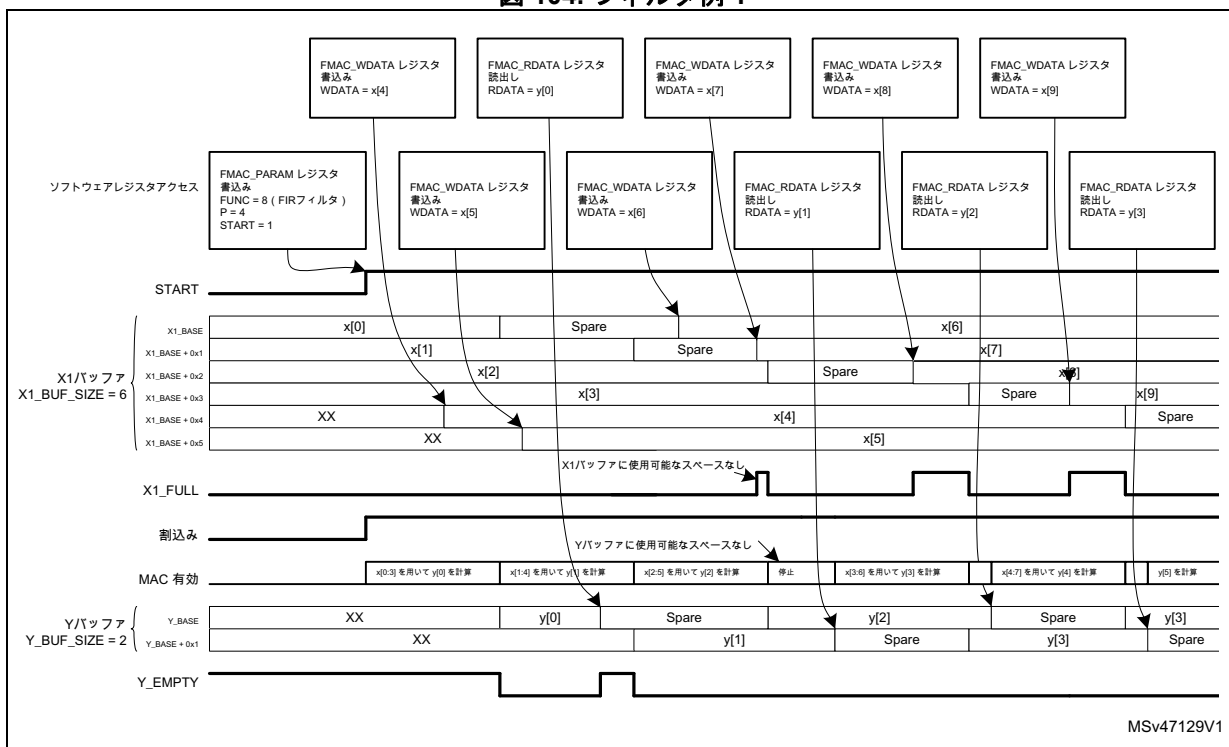


図 104 の例には、フィルタ演算の始まりが図示されています。フィルタは 4 タップです ($P = 4$)。X1 バッファサイズは 6、Y バッファサイズは 2 です。FULL_WM ビットフィールドと EMPTY_WM ビットフィールドはどちらも 0 にセットされています。図 103 に示すように、フィルタを開始する前に、X1 バッファには 4 個のサンプル $x[0:3]$ がプリロードされています。したがって、START ビットがセットされると、フィルタは最初の出力サンプル $y[0]$ の計算を直ちに開始します。X1_FULL はセットされていません (X1 バッファ内の 2 つの未初期化領域のため) ので、直ちに割込みがアサートされ、新しいデータをリクエストします。プロセッサが 2 個の新しいサンプル $x[4]$ および $x[5]$ を FMAC_WDATA レジスタに書き込むと、それらのサンプルは X1 バッファの空いている位置に転送されます。

そうしている間に FMAC は最初の出力サンプル $y[0]$ の計算を完了して Y バッファの中にも書き込むことにより、Y_EMPTY フラグがローとなります。同時に、 $x[0]$ サンプルはもはや必要ないため破棄され、メモリ内のその位置 (X1_BASE の位置) を開放します。必要な入力サンプル $x[1:5]$ はすべて X1 バッファの中に存在するので、FMAC は、2 番目の出力サンプル $y[1]$ に対する処理を直ちに開始可能となります。

Y_EMPTY フラグがローであるため、プロセッサが $x[5]$ の書き込みをし終わった後も割込みはアクティブのままとなります。プロセッサが FMAC_RDATA レジスタから $y[0]$ を読み出し、Y バッファ内のその位置を開放します。依然として $y[1]$ の計算中であるため、出力バッファの中にサンプルがなくなり、Y_EMPTY フラグがハイになります。ただし、依然として X1 バッファの中に空き空間が存在するために割込みはアクティブのままであり、この空間は次にプロセッサが $x[6]$ などを入れます。

注： この例では、FMAC が処理可能な速度以上で入力バッファにプロセッサがフィルできるため、X1_full フラグは定期的にはアクティブとなっています。ただし、Y バッファを十分な速度で読み出すことには苦勞しており、Y バッファの空間開放待ちで定期的に FMAC が停止しています。つまり、このフィルタは、最大スループットでは動作していません。その理由は、この例では、フィルタ長が短く、プロセッサが比較的低速であるためです。したがって、Y バッファサイズを増やしても効果はありません。

図 105. フィルタ例 2

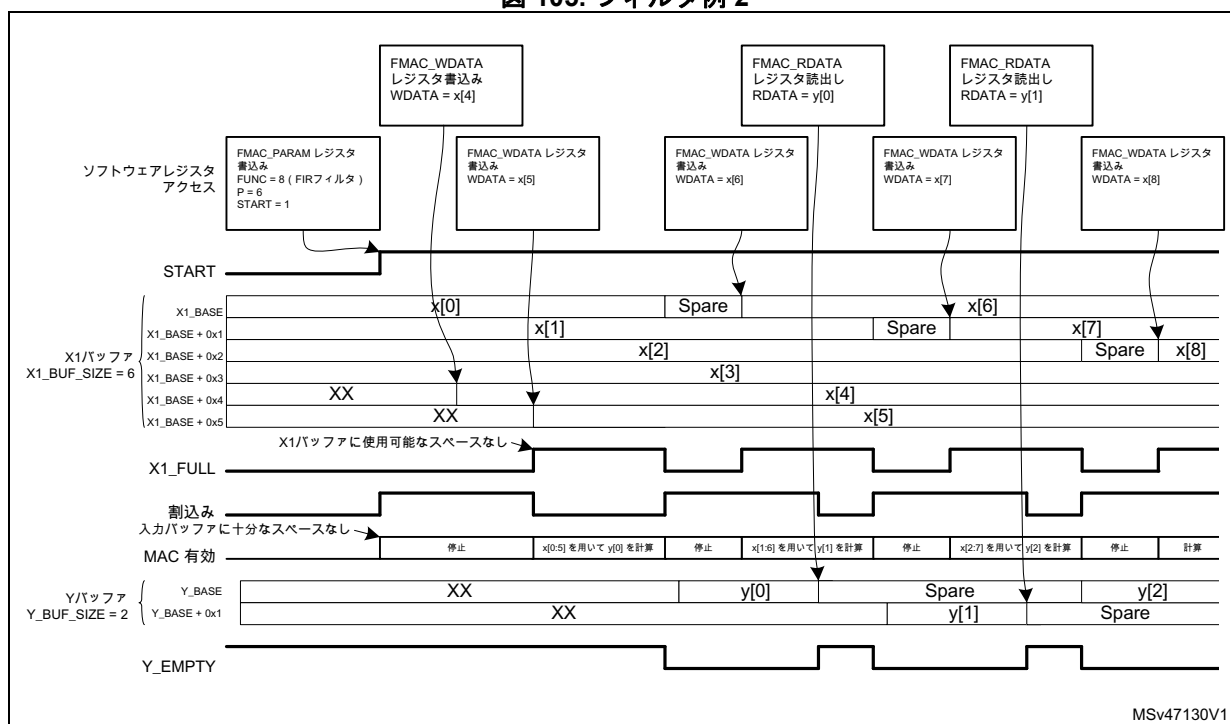


図 105 の例には、同じフィルタ演算の始まりが図示されていますが、今回のフィルタは 6 タップです (P = 6)。X1 バッファサイズは 6、Y バッファサイズは 2 です。FULL_WM ビットフィールドと EMPTY_WM ビットフィールドはどちらも 0 にセットされています。図 103 に示すように、フィルタを開始する前に、X1 バッファには 4 個のサンプル x[0:3] がプリロードされています。入力バッファに十分な数のサンプルがないことから、X1_FULL フラグはセットされませんので、直ちに割り込みがアサートされ、新しいデータをリクエストします。FMAC は停止します。

プロセッサが 2 個の新しいサンプル x[4] および x[5] を FMAC_WDATA レジスタに書き込むと、それらのサンプルは X1 バッファの空いている位置に転送されます。X1 バッファに未使用のサンプルが 6 個存在すると、X1_FULL フラグが直ちにアクティブとなり (バッファサイズが 6 であるため)、割り込むが無効になります。FMAC は最初の実出力サンプル y[0] の計算を開始します。これには 6 個の入力サンプルがすべて必要であるため、X1 バッファには空き空間が存在せず、X1_FULL フラグはアクティブのままとなります。FMAC が y[0] の計算を完了して Y バッファに書き込んで初めて x[0] の破棄が可能となり、X1 バッファの空間が開放されて、X1_FULL がデアサートされます。同時に、Y_EMPTY フラグは無効となります。これら両方のフラグの状態によって割り込みがアサートされ、最初に新しい入力サンプルの書き込みを、その次に計算されたばかりの実出力サンプルの読出しを、プロセッサにリクエストします。新しい入力サンプルが書き込まれるまで、FMAC は停止したままです。

この例では、プロセッサが新しい入力サンプルを書き込み可能となる前に、FMAC が現在の出力サンプルの計算を完了するのを待つ必要がありますので、X1 バッファが定期的に空となり、FMAC が停止します。入力バッファに追加空間をいくらか追加することにより、これは回避可能です。

21.3.12 フィルタ設計のヒント

FMAC のアーキテクチャにより、デジタルフィルタの設計には下に詳細を示したような、いくつかの制約条件が課されます。

1. 直接型 2 と転置型の実装は効率的ではありません。このような型のために設計されたフィルタは、直接型 1 に変換しなくてはなりません。
2. カスケードフィルタは、単一ステージに組み合わせるか、別のフィルタとして実装する必要があります。後者の場合には、複数セット（ステージあたり 1 セット）のフィルタ係数をメモリにプリロードしておき、X2_BASE アドレスレジスタを変更するだけで、どのセットを使用するか選択できます。複数ステージフィルタを実装する最も効率的な方法は、大きな X1 バッファに入力サンプルをプリロードしておき、最初のステージの係数を使用して IIR フィルタ機能を実行し、出力サンプルをメモリに戻して格納するというものです。次に、X2_BASE ポインタが 2 番目のステージの係数を指すように変更し、入力バッファに最初のステージの出力を再ロード（必要に応じてゲインを適用）してから IIR 機能を再実行します。すべてのステージに対して、この手順を繰り返します。最終ステージのサンプルがシステムメモリに転送されたら、入力バッファに次セットの入力サンプルをロードして、新しいラウンドの計算を開始できます。各ラウンドの間で連続性を維持するために、各ステージの N サンプル入力バッファには、最初に、前回ラウンドの N-1 個の最後の入力をすべてプリロードしてから、新しいサンプルを 1 個加える必要があることに注意してください。同様に、同じ理由により、各ステージの出力バッファには、前回ラウンドから最後の M 個のサンプルをロードする必要があります。
3. IIR 設計に直接型 1 を用いると、たとえば入力に大きなステップが生じたり、一部のフィルタ係数の絶対値が 1 を上回ったりするような場合に、アキュムレータの中に正または負の大きな部分和が生じることがあります。アキュムレータは 26bit に制限されているため、折り返し（符号の変化）なしに処理できる最大値は、プラス側で 0x1FFFFFFF、マイナス側で 0x20000000 となります。q3.23 固定小数点フォーマットでは、これらの値はそれぞれ 3.99999988 と -4 に対応します。累積の最後まで折り返しを取り消されない限り、折り返しが問題となることはありません。ただし、フィルタが開始されて予期しない結果に至る可能性がある場合には、このことは必ずしも当てはまりません。これを避けるためには、出力バッファに適切な値をプリロードすることを検討してください。
4. IIR フィルタには、フィードフォワード（分子）係数 $[b_0, b_1, \dots, b_{N-1}]$ とフィードバック（分母）係数 $[1, a_1, \dots, a_M]$ があります。多くの IIR フィルタでは、周波数応答において急峻なロールオフを得るために、一部の分母係数が 1 を上回る絶対値であることが要求されます。固定小数点の q1.15 フォーマットで係数がコード化されることから考えれば、これは不可能です。それにもかかわらず、 $2^{-R} \cdot [1, a_1, \dots, a_M]$ がすべて 1 未満となるようにファクタ 2^{-R} によって分母係数をスケールリングすることにより、このようなフィルタについても実装可能となります。ただし、スケールリングを打ち消すために、アキュムレータの出力に対して 2^R の逆ゲインを適用する必要があります。こうすることにより、信号対雑音比 (SNR) に悪影響が生じます。

21.4 FMAC レジスタ

21.4.1 FMAC X1 バッファ設定レジスタ (FMAC_X1BUFCFG)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

アクセス : ワード・アクセス

このレジスタは、FMAC_PARAM レジスタの START = 0 の場合にのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FULL_WM[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X1_BUF_SIZE [7:0]								X1_BASE [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **FULL_WM[1:0]** : バッファ・フル・フラグに対するウォーターマーク

サーキュラモードでの動作時の X1 バッファフルフラグの設定閾値を定義します。バッファ内の空き空間の個数が $2^{\text{FULL_WM}}$ 未満である場合にこのフラグがセットされます。

0 : 閾値 = 1

1 : 閾値 = 2

2 : 閾値 = 4

3 : 閾値 = 8

1 よりも大きい閾値を設定すると、1 回の割込みで複数のデータをバッファに転送可能となります。

DMA 書込みリクエストが有効である場合 (FMAC_CR レジスタの DMAWEN = 1)、閾値は 1 にセットしなくてはなりません。

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **X1_BUF_SIZE[7:0]** : X1 バッファに割り当てられたサイズ (16bit ワード単位)

最小バッファサイズは、フィルタのフィードフォワードタップ数 (+ ウォーターマーク閾値 - 1) です。

ビット 7:0 **X1_BASE[7:0]** : X1 バッファのベースアドレス

21.4.2 FMAC X2 バッファ設定レジスタ (FMAC_X2BUFCFG)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : ワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X2_BUF_SIZE [7:0]								X2_BASE [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **X2_BUF_SIZE[7:0]** : X2 バッファのサイズ (16 bit ワード単位)

機能の動作中 (START = 1) はこのビットフィールドは変更できません。

ビット 7:0 **X2_BASE[7:0]** : X2 バッファのベースアドレス

係数値の変更などのために、START = 1 の間にも X2 バッファベースアドレスは変更可能です。計算中に係数を変更すると結果に影響が生じるため、これを行っている間はフィルタを停止する必要があります。

21.4.3 FMAC Y バッファ設定レジスタ (FMAC_YBUFCFG)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : ワード・アクセス

このレジスタは、FMAC_PARAM レジスタの START = 0 の場合にのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EMPTY_WM[1:0]		Res	Res	Res	Res	Res	Res	Res	Res
						rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y_BUF_SIZE[7:0]								Y_BASE[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **EMPTY_WM[1:0]** : バッファ・エンプティ・フラグに対するウォーターマーク

サーキュラモードでの動作時の Y バッファエンプティフラグの設定閾値を定義します。バッファ内の未読値の個数が $2^{\text{EMPTY_WM}}$ 未満である場合にこのフラグがセットされます。

0 : 閾値 = 1

1 : 閾値 = 2

2 : 閾値 = 4

3 : 閾値 = 8

1 よりも大きい閾値を設定すると、1 回の割込みで複数のデータをバッファから転送可能となります。

DMA 読出しリクエストが有効である場合 (FMAC_CR レジスタの DMAREN = 1)、閾値は 1 に設定する必要があります。

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **Y_BUF_SIZE[7:0]** : Y バッファのサイズ (16 bit ワード単位)

FIR フィルタでは、最小バッファサイズは 1 (+ ウォーターマーク閾値) です。IIR フィルタでは、最小バッファサイズはフィードバックタップ数 (+ ウォーターマーク閾値) です。

ビット 7:0 **Y_BASE[7:0]** : Y バッファのベースアドレス

21.4.4 FMAC パラメータ・レジスタ (FMAC_PARAM)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : ワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
START	FUNC[6:0]							R[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Q[7:0]								P[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **START** : 実行の有効化

0 : 実行を停止します。

1 : 実行を開始します。

このビットを設定すると、FUNC ビットフィールドで選択された機能の実行がトリガされます。ソフトウェアでリセットすると、実行中のあらゆる機能が停止します。初期化機能では、このビットはハードウェアによってリセットされます。

ビット 30:24 **FUNC[6:0]** : 機能

0 : 予約済み

1 : ロード X1 パッファ

2 : ロード X2 パッファ

3 : ロード Y パッファ

4 から 7 : 予約済み

8 : 畳み込み (FIR フィルタ)

9 : IIR フィルタ (直接型 1)

10 から 127 : 予約済み

機能の動作中 (START = 1) はこのビットフィールドは変更できません。

ビット 23:16 **R[7:0]** : 入力パラメータ R

このパラメータの値は機能に依存します。

機能の動作中 (START = 1) はこのビットフィールドは変更できません。

ビット 15:8 **Q[7:0]** : 入力パラメータ Q

このパラメータの値は機能に依存します。

機能の動作中 (START = 1) はこのビットフィールドは変更できません。

ビット 7:0 **P[7:0]** : 入力パラメータ P

このパラメータの値は機能に依存します。

機能の動作中 (START = 1) はこのビットフィールドは変更できません。

21.4.5 FMAC 制御レジスタ (FMAC_CR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	リセット
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLIP EN	Res.	Res.	Res.	Res.	Res.	DMA WEN	DMA REN	Res.	Res.	Res.	SAT IEN	UNFL IEN	OVFL IEN	WIEN	RIEN
rw						rw	rw				rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **RESET** : FMAC ユニットのリセット

ライト・ポインタとリード・ポインタ、内部制御ロジック、FMAC_SR レジスタ、FMAC_PARAM レジスタ（アクティブである場合には START ビットを含む）がリセットされます。それ以外のレジスタ設定には影響しません。このビットは、ハードウェアによってリセットされます。

0 : リセットが非アクティブです。

1 : リセットがアクティブです。

ビット 15 **CLIPEN** : クリッピングの有効化

0 : クリッピングが無効です。q1.15 の範囲を超過したアキュムレータの出力の値は折り返します。

1 : クリッピングが有効です。q1.15 の範囲を超過したアキュムレータの出力の値は、符合に応じて正または負の最大値 (+1 または -1) に飽和します。

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **DMAWEN** : DMA ライトチャネルリクエストの有効化

0 : 無効にします。DMA リクエストは生成されません

1 : 有効にします。X1 バッファがフルではない間、DMA リクエストが生成されます。

このビットは、FMAC_PARAM レジスタの START = 0 の場合にのみ修正できます。読み出すと、ビットの現在の状態が返されます。

ビット 8 **DMAREN** : DMA リードチャネルリクエストの有効化

0 : 無効にします。DMA リクエストは生成されません

1 : 有効にします。Y バッファが空ではない間、DMA リクエストが生成されます。

このビットは、FMAC_PARAM レジスタの START = 0 の場合にのみ修正できます。読み出すと、ビットの現在の状態が返されます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **SATIEN** : 飽和エラー割込みの有効化

0 : 無効。飽和が検出されても割込みは生成されません。

1 : 有効となっています。SAT フラグがセットされると、割込みリクエストが生成されます

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 3 **UNFLIEN** : アンダーフロー・エラー割込みの有効化

0 : 無効。アンダーフローが検出されても割込みは生成されません。

1 : 有効となっています。UNFL フラグがセットされると、割込みリクエストが生成されます

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 2 **OVFLIEN** : オーバーフロー・エラー割込みの有効化

0 : 無効。オーバーフローが検出されても割込みは生成されません。

1 : 有効となっています。OVFL フラグがセットされると、割込みリクエストが生成されます

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 1 **WIEN** : 書き込み割込みの有効化

0 : 無効。書き込み割込みリクエストは生成されません。

1 : 有効となっています。X1 バッファ FULL フラグがセットされていない場合、割込みリクエストが生成されます。

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

ビット 0 **RIEN** : 読出し割込みの有効化

0 : 無効。読出し割込みリクエストは生成されません。

1 : 有効となっています。Y バッファ EMPTY フラグがセットされていない場合、割込みリクエストが生成されます。

このビットは、ソフトウェアによってセット／クリアされます。読み出すと、ビットの現在の状態が返されます。

21.4.6 FMAC ステータス・レジスタ (FMAC_SR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0001

アクセス : ワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SAT	UNFL	OVFL	Res.	Res.	Res.	Res.	Res.	Res.	X1 FULL	Y EMPTY
					r	r	r							r	r

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **SAT** : 飽和エラー・フラグ

飽和は、演算の結果がアキュムレータの数値範囲を超えたときに発生します。

0 : 飽和は検出されていません。

1 : 飽和が検出されています。SATIEN ビットがセットされている場合は、割込みが生成されます。

このフラグは、ユニットのリセットによってクリアされます。

ビット 9 **UNFL** : アンダーフロー・エラー・フラグ

Y バッファに有効なデータが存在しないときに FMAC_RDATA から読出しを行うとアンダーフローが発生します。

0 : アンダーフローは検出されていません。

1 : アンダーフローが検出されています。UNFLIEN ビットがセットされている場合は、割込みが生成されます。

このフラグは、ユニットのリセットによってクリアされます。

ビット 8 OVFL : オーバーフロー・エラー・フラグ

X1 バッファに空き空間が存在しないときに FMAC_WDATA への書込みを行うとオーバーフローが発生します。

0 : オーバーフローは検出されていません。

1 : オーバーフローが検出されています。OVFLIEN ビットがセットされている場合は、割込みが生成されます。

このフラグは、ユニットのリセットによってクリアされます。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 X1FULL : X1 バッファ・フル・フラグ

利用可能な空間の数が FULL_WM 閾値未満の場合に、バッファにフルフラグが立ちます。利用可能な空間の数は、ライトポインタと現在使用中のサンプルの中で最も古いものとの差となります。

0 : X1 バッファはフルではありません。WIEN ビットがセットされていると、フラグがセットされるまで割込みリクエストがアサートされます。DMAWEN がセットされていると、フラグがセットされるまで DMA 書込みチャネル・リクエストが生成されます。

1 : X1 バッファはフルです。

このフラグはハードウェアまたはリセットによってクリアされます。

注 : X1 バッファの中で利用可能な最後の空間がフィルされた後、X1FULL フラグがハイになるまでに、3 クロックサイクルの遅延があります。オーバーフローのリスクを回避するために、X1 バッファに書き込んでから FMAC_SR を読み出すまでの間にソフトウェアディレイを挿入することをお勧めします。その代わりに、FULL_WM 閾値を 2 とすることもできます。

ビット 0 YEMPTY : Y バッファ・エンプティ・フラグ

未読データの数 EMPTY_WM 閾値未満の場合に、バッファにエンプティフラグが立ちます。未読データの数、リードポインタと現在の出力転送先アドレスとの差となります。

0 : Y バッファは空ではありません。RIEN ビットがセットされていると、フラグがセットされるまで割込みリクエストがアサートされます。DMAREN がセットされていると、フラグがセットされるまで DMA 読出しチャネル・リクエストが生成されます。

1 : Y バッファは空です。

このフラグはハードウェアまたはリセットによってクリアされます。

注 : Y バッファから最後のサンプルが読み出された後、YEMPTY フラグがハイになるまでに、3 クロックサイクルの遅延があります。アンダーフローのリスクを回避するために、Y バッファから読み出してから FMAC_SR を読み出すまでの間にソフトウェアディレイを挿入することをお勧めします。その代わりに、EMPTY_WM 閾値を 2 とすることもできます。

21.4.7 FMAC ライト・データ・レジスタ (FMAC_WDATA)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

アクセス : ワード、およびハーフワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDATA[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **WDATA[15:0]** : 書き込みデータ

このレジスタへのライト・アクセスが発生すると、ライト・ポインタで示されているアドレス・オフセットにライト・データが転送されます。ポインタアドレスは、毎回のライトアクセスの後に自動的にインクリメントされます。

21.4.8 FMAC リード・データ・レジスタ (FMAC_RDATA)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : ワード、およびハーフワード・アクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RDATA[15:0]** : データ読出し

このレジスタへのリードアクセスが発生すると、リードデータは READ ポインタで示されているアドレスオフセットの Y 出力バッファの内容となります。ポインタアドレスは、毎回のリードアクセスの後に自動的にインクリメントされます。

21.4.9 FMAC レジスタマップ

表 161. FMAC レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMAC_X1BUFCFG	Res.	Res.	Res.	Res.	Res.	Res.	FULL_WM [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	X1_BUF_SIZE [7:0]					X1_BASE [7:0]										
	リセット値							0 0										0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0														
0x04	FMAC_X2BUFCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	X2_BUF_SIZE [7:0]					X2_BASE [7:0]										
	リセット値																	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0														
0x08	FMAC_YBUFCFG	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY_WM [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Y_BUF_SIZE[7:0]					Y_BASE[7:0]										
	リセット値							0 0										0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0														
0x0C	FMAC_PARAM	START	FUNC[6:0]						R[7:0]							Q[7:0]					P[7:0]												
	リセット値	0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0							0 0 0 0 0 0 0 0					0 0 0 0 0 0 0 0																
0x10	FMAC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	リセット	CLIPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SATIEN	UNFLIEN	OVFLIEN	WIEN	RIEN
	リセット値																0 0	0 0											0 0	0 0	0 0	0 0	0 0
0x14	FMAC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAT	UNFL	OVFL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	X1FULL	YEMPTY	
	リセット値																		0 0 0												0 0	0 1	
0x18	FMAC_WDATA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	リセット値																	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0														
0x1C	FMAC_RDATA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATA[15:0]															
	リセット値																	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0													

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

22 フレキシブル・スタティク・メモリ・コントローラ (FSMC)

22.1 概要

フレキシブル・スタティク・メモリ・コントローラ (FSMC) には、次の 3 つのメモリ・コントローラが含まれています。

- NOR/PSRAM メモリコントローラ
- NAND メモリコントローラ
- 同期 DRAM (SDRAM/モバイル LPDDR SDRAM) コントローラ

このメモリ・コントローラはフレキシブル・メモリ・コントローラ (FMC) とも呼ばれています。

22.2 FMC の主な特徴

FMC 機能ブロックは、同期および非同期のスタティクメモリ、SDRAM メモリ、NAND Flash メモリとのインタフェースを構成します。主要目的は、

- AHB トランザクションを該当する外部デバイスプロトコルに変換すること
- 外部メモリデバイスのアクセス時間条件を満たすようにすること

すべての外部メモリが、アドレス信号、データ信号、制御信号をコントローラと共用します。各外部デバイスは、一意なチップセレクトによってアクセスされます。FMCは外部デバイスへ一度に 1 回だけアクセスします。

FMCコントローラの主要機能は次の通りです。

- 次のスタティクメモリにマップされたデバイスとのインタフェース
 - スタティクランダムアクセスメモリ (SRAM)
 - NOR 型 Flash メモリ/OneNAND Flash メモリ
 - PSRAM (4 メモリバンク)
 - 強誘電性 RAM (FRAM)
 - 最大 8 KB のデータをチェックできる ECC ハードウェア付き NAND Flash メモリ
- 同期 DRAM (SDRAM/モバイル LPDDR SDRAM) メモリとのインタフェース
- Intel 8080 およびモトローラ 6800 モードをサポートしている並列 LCD モジュール用インタフェース
- NOR Flashメモリ、PSRAM、SDRAMなどの同期デバイスに対するアクセスを高速化するバーストモードのサポート
- 非同期アクセスと同期アクセス用のプログラマブルな連続クロック出力
- 8 ビット、16 ビット幅のデータバス
- メモリバンクごとに独立したチップセレクト制御
- メモリバンクごとに独立した設定
- PSRAM デバイス、SRAM デバイス、SDRAM デバイスで使用する書込みイネーブルとバイトレーン選択出力
- 外部非同期ウェイト制御
- 16x32 ビットの深さの書込み FIFO
- SDRAM コントローラ用の 6x32 ビットの深さ (6x14 ビットアドレスタグ) のキャッシュ可能読出し FIFO

書込み FIFO はすべてのメモリコントローラに共通であり、以下の構成となっています。

- ライトデータ FIFO は、メモリへ書き込む AHB データ（最大 32 ビット）と、AHB 転送（バーストモードまたは非シーケンシャルモード）用の 1 ビットを格納
- 書込みアドレス FIFO は、AHB アドレス（最大 28 ビット）と AHB データサイズ（最大 2 ビット）を格納バーストモードで動作する場合、ページ境界（PSRAM と SDRAM）をまたぐとき以外は、開始アドレスのみを格納します。この場合、AHB バーストは 2 つの FIFO エントリに分割されます。

開始時に、ユーザアプリケーションから FMC ピンを設定する必要があります。アプリケーションで使用しない FMC I/O ピンは、他の用途に使用することができます。

外部デバイスタイプおよびその特性を定義する FMC レジスタは、通常、起動時に設定され、次のリセットまたは電源投入まで変更されません。

ただし、以下のいくつかのビットだけなら動作中に変更可能です。

- FMC_BCRx レジスタの MBKEN、FMCEN、WEN ビット
- FMC_PCR レジスタの ECCEN および PBKEN ビット
- FMC_SR レジスタの IFS、IRS、および ILS ビット

以下のシーケンスに従って、FMC の有効時にパラメータを変更します。

1. 最初に、FMC コントローラを無効にすることで、レジスタの変更中はそれ以上メモリコントローラへアクセスできないようにします。
2. すべての必要な設定を更新します。
3. FMC コントローラを再び有効にします。

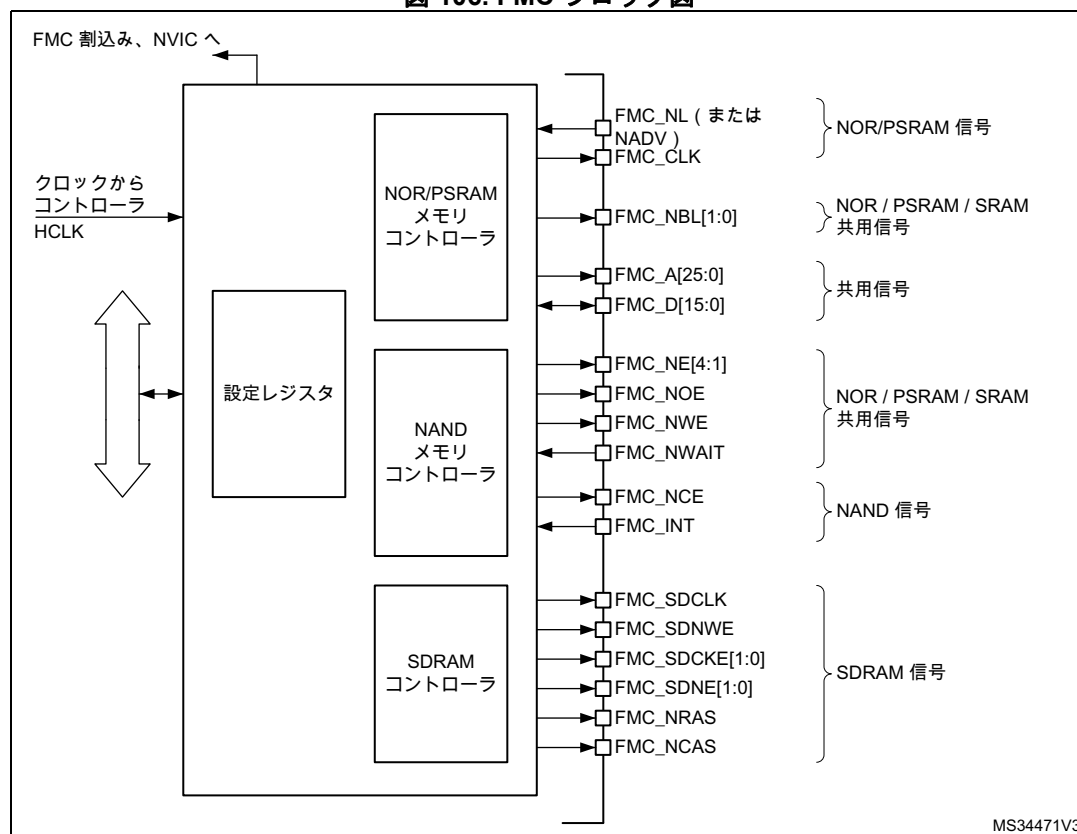
22.3 FMC ブロック図

FMC は次のメインブロックで構成されます。

- AHB インタフェース (FMC 設定レジスタを含む)
- NOR Flash/PSRAM/SRAM コントローラ
- SDRAM コントローラ

以下にブロック図を示します。

図 106. FMC ブロック図



22.4 AHB インタフェース

AHB スレーブインタフェースを使用することで、内部 CPU やその他のバスマスタペリフェラルから外部メモリへアクセスすることができます。

AHB トランザクションは、外部デバイスプロトコルに変換されます。特に、選択された外部メモリが 16 または 8 ビット幅の場合、AHB 上の 32 ビット幅のトランザクションは、16 または 8 ビットの連続した複数のアクセスに分割されます。FMC チップセレクト (FMC_NEX) は、拡張モードが有効になっている際にアクセスモード D の場合を除いて、複数の連続アクセス間にトグルしません。

FMC は、以下の条件下で AHB エラーを生成します。

- 有効化していない FMC バンク (バンク 1~4) の読出しまたは書込みを行ったとき。
- FMC_BCRx レジスタの FACCEN ビットがリセットされている状態で NOR Flash バンクの読出しまたは書込みを行ったとき。
- 書込み保護された SDRAM バンク (SDRAM_SDCRx レジスタの WP ビットがセット中) に対して書込みを行ったとき。
- SDRAM アドレス範囲に違反したとき (予約済みアドレス範囲へのアクセス)。

この AHB エラーの影響は、読出し/書込みアクセスを試みた AHB マスタに依存します。

- アクセスが Cortex-M33 CPU から行われた場合は、ハードフォールト割込みが発生します。
- アクセスが DMA コントローラから行われた場合は、DMA 転送エラーが発生し、対応する DMA チャンネルが自動的に無効化されます。

AHB クロック (HCLK) は、FMC のリファレンスクロックです。

22.4.1 サポートされるメモリおよびトランザクション

一般的なトランザクション規則

リクエストされる AHB トランザクションのデータサイズは、アクセスされる外部デバイスのデータ幅が固定であっても、8、16、または 32 ビット幅です。このため、転送に矛盾が生じることがあります。

したがって、次のようないくつかの単純なトランザクション規則を守る必要があります。

- AHB トランザクションサイズとメモリデータサイズが等しい場合 :
この場合は問題ありません。
- AHB トランザクションサイズがメモリサイズより大きい場合 :
この場合、FMC は、AHB トランザクションを外部データ幅に合わせて、連続した小さなメモリアクセスに分割します。FMC チップセレクト (FMC_NEX) は、複数の連続アクセス間にトグルしません。パスターンアラウンドタイミングが 0 以外の値に設定されている場合は、FMC チップセレクト (FMC_NEX) は複数の連続アクセス間にトグルします。この機能は、FRAM メモリとインタフェースするときに必要です。
- AHB トランザクションサイズがメモリサイズより小さい場合 :
外部デバイスのタイプによっては、転送に一貫性がなくなる場合があります :
 - バイト選択機能を持つデバイス (SRAM、ROM、PSRAM、SDRAM) に対するアクセス
この場合、FMC は、読出し/書込みトランザクションを許可して、バイト・レーン NBL[1:0] を通じて正しいデータにアクセスします。
書込み対象バイトは NBL[1:0] によりアドレス指定されます。
すべてのメモリバイトが読み出され (読出しトランザクション中 NBL[1:0] はローに駆動されます)、不要なバイトは無視されます。

- バイト選択機能を持たないデバイス（NOR および NAND Flashメモリ）に対するアクセス
この状況は、16 ビット幅の Flash メモリへのバイト・アクセスが要求されたときに発生します。バイト・モードでデバイスにアクセスできないため（Flashメモリに対しては 16 ビットワードの読出し／書込みのみが可能）、書込みトランザクションと読出しトランザクションが可能です（コントローラは 16 ビット・メモリ・ワード全体を読み出して、必要なバイトのみを使用）。

NOR 型Flash／PSRAM および SDRAMのラップ・サポート

同期メモリに対するラップバーストモードはサポートされていません。メモリは、不定長のリニアバーストモードに設定する必要があります。

設定レジスタ

FMC は、レジスタのセットとして設定することができます。NOR Flash／PSRAM コントローラ・レジスタの詳細については、[セクション 22.6.6](#) を参照してください。NAND Flash レジスタの詳細説明は [セクション 22.7.7](#) を、SDRAM コントローラ・レジスタの詳細説明は [セクション 22.8.5](#) を参照してください。

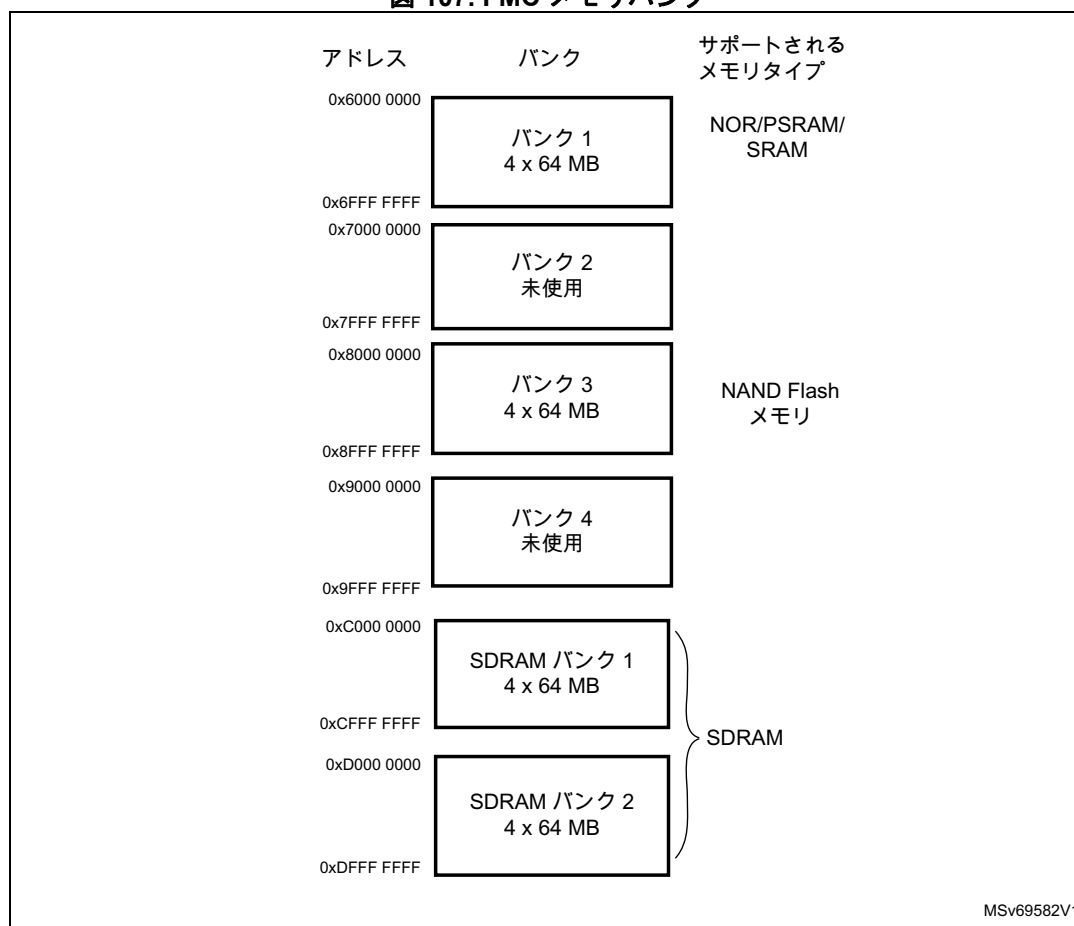
22.5 外部デバイスアドレスマッピング

FMC から見ると、外部メモリは、それぞれ 256 MB の固定サイズのバンクに分けられています（[図 107](#) を参照）。

- バンク 1 は、4 つまでの NOR Flash メモリまたは PSRAM メモリデバイスのアドレス指定に使用されます。このバンクは、次のように 4 つの専用チップセレクトを持つ 4 つの NOR/PSRAM サブバンクに分割されます。
 - バンク 1 - NOR/PSRAM 1
 - バンク 1 - NOR/PSRAM 2
 - バンク 1 - NOR/PSRAM 3
 - バンク 1 - NOR/PSRAM 4
- バンク 3 は、NAND Flash メモリデバイスのアドレス指定に使用されます。この空間の MPU メモリ属性をソフトウェアによってデバイスに再設定する必要があります。
- バンク 4 と 5 は、SDRAM デバイスのアドレス指定に使用されます（バンクごとに 1 つのデバイス）。

各バンクで使用されるメモリのタイプは、ユーザアプリケーションから設定レジスタを使って設定することができます。

図 107. FMC メモリバンク



22.5.1 NOR/PSRAM アドレスマッピング

HADDR[27:26] ビットは、表 162に示される 4 つのメモリバンクの 1 つを選択するために使用されます。

表 162. NOR/PSRAM バンク選択

HADDR[27:26] ⁽¹⁾	選択されるバンク
00	バンク 1 - NOR/PSRAM 1
01	バンク 1 - NOR/PSRAM 2
10	バンク 1 - NOR/PSRAM 3
11	バンク 1 - NOR/PSRAM 4

1. HADDR は、外部メモリに変換される内部 AHB アドレスラインです。

HADDR[25:0] ビットは、外部メモリアドレスは相当します。HADDR はバイトアドレスですが、メモリはワードレベルでアドレス指定されるので、メモリに対して実際に発行されるアドレスは、次の表に示されるように、メモリのデータ幅に応じて変わります。

表 163. NOR/PSRAM 外部メモリアドレス

メモリ幅 ⁽¹⁾	メモリに発行されるデータアドレス	最大メモリ容量 (ビット)
8 ビット	HADDR[25:0]	64 MB × 8 = 512 M bit
16 ビット	HADDR[25:1] >> 1	64 MB/2 × 16 = 512 M bit

1. 外部メモリが 16 ビット幅の場合、FMC は、内部で HADDR[25:1] を使用して、外部メモリ FMC_A[24:0] に対するアドレスを生成します。
外部メモリの幅にかかわらず、FMC_A[0] は外部メモリアドレス A[0] に接続されなくてはなりません。

22.5.2 NAND Flashメモリ・アドレス・マッピング

NAND バンクは、表 164 に示されるメモリ領域に分割されます。

表 164. NAND メモリマッピングおよびタイミングレジスタ

開始アドレス	終了アドレス	FMC バンク	メモリ空間	タイミングレジスタ
0x8800 0000	0x8BFF FFFF	バンク 3 - NAND Flash	属性	FMC_PATT (0x8C)
0x8000 0000	0x83FF FFFF		共通	FMC_PMEM (0x88)

NAND 型 Flash メモリでは、共通および属性メモリ空間は、下位 256 KB に位置する 3 つのセクションに分割されます (表 165 を参照)。

- データセクション (共通/属性メモリ空間の最初の 64 KB)
- コマンドセクション (共通/属性メモリ空間の 2 番目の 64 KB)
- アドレスセクション (共通/属性メモリ空間の次の 128 KB)

表 165. NAND バンク選択

セクション名	HADDR[17:16]	アドレス範囲
アドレスセクション	1X	0x020000-0x03FFFF
コマンドセクション	01	0x010000-0x01FFFF
データセクション	00	0x000000-0x00FFFF

アプリケーション・ソフトウェアは、3 つのセクションを使用して、NAND 型 Flash メモリにアクセスします。

- NAND Flash メモリにコマンドを送信するときは**、ソフトウェアからコマンド・セクションの任意のメモリ位置にコマンドの値を書き込む必要があります。
- 読出しまたは書込みを行う必要がある NAND Flash のアドレスを指定するには**、ソフトウェアは、アドレス・セクションの任意のメモリ位置にアドレス値を書き込む必要があります。アドレスは 4 または 5 バイト長なので (実際のメモリサイズに依存)、アドレス全体を指定するには、アドレスセクションへのいくつかの連続した書込みが必要です。
- データの読出しまたは書込みを行うときは**、ソフトウェアは、データセクションの任意のメモリ位置からデータを読み出すか、書き込む必要があります。

NAND 型 Flash メモリはアドレスを自動的にインクリメントするので、連続したメモリ位置にアクセスするには、データ・セクションのアドレスをインクリメントする必要はありません。

22.5.3 SDRAM アドレスマッピング

HADDR[28] ビット（内部 AHB アドレスライン 28）を使って、2 つのメモリバンクから 1 つを選択します（表 166 を参照）。

表 166. SDRAM バンク選択

HADDR[28]	選択されるバンク	制御レジスタ	タイミングレジスタ
0	SDRAM Bank1	FMC_SDCR1	FMC_SDTR1
1	SDRAM Bank2	FMC_SDCR2	FMC_SDTR2

次の表に、13 ビット行/11 ビット列、内部 4 バンク構成に対する SDRAM マッピングを示します。

表 167. SDRAM アドレスマッピング

メモリ幅 ⁽¹⁾	内部バンク	Rowアドレス	Columnアドレス ⁽²⁾	最大メモリ容量 (M バイト)
8 ビット	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 MB : 4 x 8K x 2K
16 ビット	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 MB : 4 x 8K x 2K x 2

- 16 ビットメモリとインタフェースする場合、FMC は内部で HADDR[11:1] 内部AHB アドレス・ラインを使用して外部アドレスを発生します。メモリの幅にかかわらず、FMC_A[0] は外部メモリアドレス A[0] に接続する必要があります。
- 自動プリチャージはサポートされていません。FMC_A[10] は外部メモリアドレス A[10] に接続する必要がありますが、常にローレベルに駆動されます。

HADDR[27:0] ビットは、SDRAM コントローラの設定に応じて外部SDRAM アドレスへ変換されます。

- データ・サイズ : 8 または 16 bit
- Rowサイズ : 11、12 または 13 ビット
- Columnサイズ : 8、9、10 または 11 ビット
- 内部バンク数 : 2 つまたは 4 つの内部バンク

次の表に、SDRAM アドレスマッピングと SDRAM コントローラ設定の関係を示します。

表 168. 8ビットデータバス幅での SDRAM アドレスマッピング⁽¹⁾⁽²⁾

Rowサイズ設定	HADDR (AHB 内部アドレスライン)																															
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
11 ビット Rowサイズ 設定	Res.							バンク [1:0]		ROW [10:0]										COLUMN [7:0]												
	Res.						バンク [1:0]		ROW [10:0]										COLUMN [8:0]													
	Res.					バンク [1:0]		ROW [10:0]										COLUMN [9:0]														
	Res.				バンク [1:0]		ROW [10:0]										COLUMN [10:0]															

表 168. 8ビットデータバス幅での SDRAM アドレスマッピング⁽¹⁾⁽²⁾ (続き)

Rowサイズ設定	HADDR (AHB 内部アドレスライン)																															
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
12 ビット Rowサイズ 設定	Res.						バンク [1:0]		ROW [11:0]											COLUMN [7:0]												
	Res.					バンク [1:0]		ROW [11:0]												COLUMN [8:0]												
	Res.				バンク [1:0]		ROW [11:0]												COLUMN [9:0]													
	Res.			バンク [1:0]		ROW [11:0]												COLUMN [10:0]														
13 ビット Rowサイズ 設定	Res.					バンク [1:0]		ROW [12:0]											COLUMN [7:0]													
	Res.				バンク [1:0]		ROW [12:0]												COLUMN [8:0]													
	Res.			バンク [1:0]		ROW [12:0]												COLUMN [9:0]														
	Res.		バンク [1:0]		ROW [12:0]												COLUMN [10:0]															

1. BANK[1:0] は、バンクアドレス BA[1:0] になります。2 つの内部バンクのみを使用する場合、BA1 は常に"0"に設定する必要があります。
2. 予約済み (Res.) アドレス範囲をアクセスすると、AHB エラーが発生します。

表 169. 16ビットデータバス幅での SDRAM アドレスマッピング⁽¹⁾⁽²⁾

Rowサイズ設定	HADDR (AHB アドレスライン)																											
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11 ビット Rowサイズ 設定	Res.						バンク [1:0]		ROW [10:0]											COLUMN [7:0]								BM0 ⁽³⁾
	Res.						バンク [1:0]		ROW [10:0]											COLUMN [8:0]								BM0
	Res.						バンク [1:0]		ROW [10:0]											COLUMN [9:0]								BM0
	Res.						バンク [1:0]		ROW [10:0]											COLUMN [10:0]								BM0
12 ビット Rowサイズ 設定	Res.						バンク [1:0]		ROW [11:0]											COLUMN [7:0]								BM0
	Res.						バンク [1:0]		ROW [11:0]											COLUMN [8:0]								BM0
	Res.						バンク [1:0]		ROW [11:0]											COLUMN [9:0]								BM0
	Res.						バンク [1:0]		ROW [11:0]											COLUMN [10:0]								BM0
13 ビット Rowサイズ 設定	Res.						バンク [1:0]		ROW [12:0]											COLUMN [7:0]								BM0
	Res.						バンク [1:0]		ROW [12:0]											COLUMN [8:0]								BM0
	Res.						バンク [1:0]		ROW [12:0]											COLUMN [9:0]								BM0
	Res.						バンク [1:0]		ROW [12:0]											COLUMN [10:0]								BM0

1. BANK[1:0] は、バンクアドレス BA[1:0] になります。2 つの内部バンクのみを使用する場合、BA1 は常に"0"に設定する必要があります。
2. 予約済み (Res.) アドレス空間をアクセスすると、AHB エラーが発生します。
3. BM0 は、16 ビットアクセスに対するバイトマスクです。

22.6 NOR 型 Flash／PSRAM コントローラ

FMC は、以下のタイプのメモリを駆動するのに適した信号タイミングを発生します。

- 非同期 SRAM、FRAM、および ROM
 - 8 ビット
 - 16 ビット
- PSRAM (CellularRAM™)
 - 非同期モード
 - 同期アクセスに対するバーストモード
 - マルチプレクスまたは非マルチプレクス
- NOR Flash メモリ
 - 非同期モード
 - 同期アクセスに対するバーストモード
 - マルチプレクスまたは非マルチプレクス

FMC は、バンクごとに 1 つのチップセレクト信号 NE[4:1] を出力します。その他の信号（アドレス、データ、および制御）はすべて、共有されます。

FMC は、次のようなプログラム可能なタイミングにより、広範囲なデバイスをサポートしています。

- プログラム可能なウェイトステート（15 まで）
- プログラム可能なバスターンアラウンドサイクル（15 まで）
- プログラム可能な出カインーブルおよび書込みインーブル遅延（15 まで）
- 独立した読出しおよび書込みタイミングとプロトコルにより、広範囲なメモリおよびタイミングをサポート
- プログラム可能な連続クロック (FMC_CLK) 出力

FMC クロック (FMC_CLK) は HCLK クロックの約数です。このクロックは、同期アクセス中にのみ、または FMC_BCR1 レジスタの CCKEN ビット設定に応じて非同期アクセスおよび同期アクセス中に、選択した外部デバイスへ供給することができます。

- CCKEN ビットがリセットされている場合、FMC は同期アクセス（読出し／書込みトランザクション）中にのみクロック (CLK) を生成します。
- CCKEN ビットがセットされている場合、FMC は非同期アクセス中および同期アクセス中に連続クロックを生成します。FMC_CLK 連続クロックを生成するためには、バンク 1 を同期モードに設定する必要があります（[セクション 22.6.6: NOR/PSRAM コントローラレジスタ](#)を参照）。すべての同期メモリに対して同じクロックが使用されるため、連続出力クロックを生成して同期アクセスを実行する場合、AHB データサイズはメモリ・データ幅 (MWID) に一致する必要があります。そうしないと、AHB データ・トランザクションに応じて、FMC_CLK 周波数が変化します（FMC_CLK 分周比の式については [セクション 22.6.5: 同期トランザクション](#)を参照）。

各バンクのサイズは 64 MB に固定されます。各バンクは、専用レジスタによって設定されます（[セクション 22.6.6: NOR/PSRAM コントローラレジスタ](#)を参照）。

プログラム可能なメモリ・パラメータは、アクセス・タイミング（[表 170](#)を参照）を含み、ウェイト管理をサポートします（バーストモードでの PSRAM および NOR Flash へのアクセス）。

表 170. プログラム可能な NOR/PSRAM のアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
アドレスセットアップ	アドレスセットアップ フェーズ時間	非同期	AHB クロック サイクル (HCLK)	0	15
アドレスホールド	アドレスホールドフェーズ 時間	非同期、 マルチプレクス I/O	AHB クロック サイクル (HCLK)	1	15
NBL セットアップ	バイトレーンセットアップ フェーズ時間	非同期	AHB クロック サイクル (HCLK)	0	3
データセットアップ	データセットアップ フェーズ時間	非同期	AHB クロック サイクル (HCLK)	1	256
データホールド	データホールド フェーズ時間	非同期	AHB クロック サイクル (HCLK)	0	3
バスターン	バスターンアラウンド フェーズ時間	非同期および 同期読出し／書込み	AHB クロック サイクル (HCLK)	0	15
クロック分周比	1つのメモリクロック サイクル (CLK) に必要な AHB クロックサイクル (HCLK) 数	同期	AHB クロック サイクル (HCLK)	2	16
データ遅延	バーストの最初のデータの 前にメモリに発行される クロックサイクル数	同期	メモリクロック サイクル (CLK)	2	17

22.6.1 外部メモリインタフェース信号

表 171、表 172 および表 173 に、NOR Flash メモリ、SRAM、および PSRAM へのインタフェースで一般に使用される信号を示します。

注： 接頭辞「N」は、アクティブローの信号を表します。

NOR Flash メモリ、非マルチプレクス I/O

表 171. 非マルチプレクス I/O NOR Flash メモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書込みインーブル

表 171. 非マルチプレクス I/O NOR Flash メモリ (続き)

FMC 信号名	I/O	機能
NL(= NADV)	O	ラッチ・イネーブル (この信号は、一部の NOR 型 Flash デバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FMC への NOR 型 Flash ウェイト入力信号

最大容量は、512 M ビット (26 アドレスライン) です。

NOR Flash メモリ、16 bit マルチプレクス I/O

表 172. 16 bit マルチプレクス I/O NOR Flash メモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書込みイネーブル
NL(= NADV)	O	ラッチ・イネーブル (この信号は、一部の NOR 型 Flash デバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FMC への NOR 型 Flash ウェイト入力信号

最大容量は 512 M ビットです。

PSRAM/FRAM/SRAM、非マルチプレクス I/O

表 173. 非マルチプレクス I/O PSRAM/SRAM

FMC 信号名	I/O	機能
CLK	O	クロック (PSRAM 同期アクセス専用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (CellularRAM™、すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインーブル
NWE	O	書込みイネーブル
NL(= NADV)	O	PSRAM 入力の場合のみアドレス有効 (メモリ信号名 : NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出力。バイト 0 およびバイト 1 制御 (上位および下位バイトイネーブル)

最大容量は 512 M ビットです。

PSRAM、16 ビットマルチプレクス I/O

表 174. 16 ビットマルチプレクス I/O PSRAM

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (CellularRAM™、すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインエーブル
NWE	O	書込みインエーブル
NL(= NADV)	O	アドレス有効 PSRAM 入力 (メモリ信号名 : NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出力。バイト 0 およびバイト 1 制御 (上位および下位バイトインエーブル)

最大容量は、512 M ビット (26 アドレスライン) です。

22.6.2 サポートされるメモリおよびトランザクション

以下の 表 175 に、NOR Flash メモリ、PSRAM、SRAM に対してメモリ・データ・バスが 16 bit 幅の場合にサポートされているデバイス、アクセスモード、トランザクションの例を示します。FMC では許容されない (すなわち非サポート) トランザクションは、この例で灰色表示しています。

表 175. NOR Flash/PSRAM : サポートされているメモリおよびトランザクションの例

デバイス	モード	読出し/ 書込み	AHB データ サイズ	メモリ データ サイズ	許可/禁止	コメント
NOR 型 Flash (マルチ プレクス I/O と 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされて いません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-

表 175. NOR Flash/PSRAM : サポートされているメモリおよびトランザクションの例

デバイス	モード	読出し/ 書込み	AHB データ サイズ	メモリ データ サイズ	許可/禁止	コメント
PSRAM (マルチプレクス I/O および 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされて いません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-
	同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	同期	W	16/32	16	Y	-
SRAM と ROM	非同期	R	8/16	16	Y	-
	非同期	W	8/16	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割、 バイトレーン NBL[1:0] の使用

22.6.3 一般的なタイミング規則

信号の同期

- すべてのコントローラ出力信号は、内部クロック (HCLK) の立上がりエッジで変化します。
- 同期モード (読出しまたは書込み) では、すべての出力信号が HCLK の立上がりエッジで変化します。CLKDIV の値によらず、すべての信号は次のように変化します。
 - NOEL/NWEL/ NEL/NADV L/ NADV H/ NBLL/ アドレスの有効出力は、FMC_CLK クロックの立下がりエッジで変化します。
 - NOEH/ NWEH/ NEH/ NOEH/ NBLH/ アドレスの無効出力は、FMC_CLK クロックの立上がりエッジで変化します。

22.6.4 NOR Flash/PSRAM コントローラ非同期トランザクション

非同期スタティク・メモリ (NOR Flash、PSRAM、SRAM、FRAM)

- 信号は、内部クロック HCLK によって同期されます。このクロックはメモリには発行されません。
- FMC は、NOE 信号をネゲートする前に、常にデータをサンプリングします。これにより、メモリのデータホールドタイミングに関する制約が満たされます (チップイネーブルハイからデータ遷移までの最小時間は通常 0 ns です)。
- 拡張モードを有効にした場合 (FMC_BCRx レジスタの EXTMOD ビットをセット)、最大 4 つの拡張モード (A、B、C、D) が使用できます。書込み動作および読出し動作で、A、B、C、D

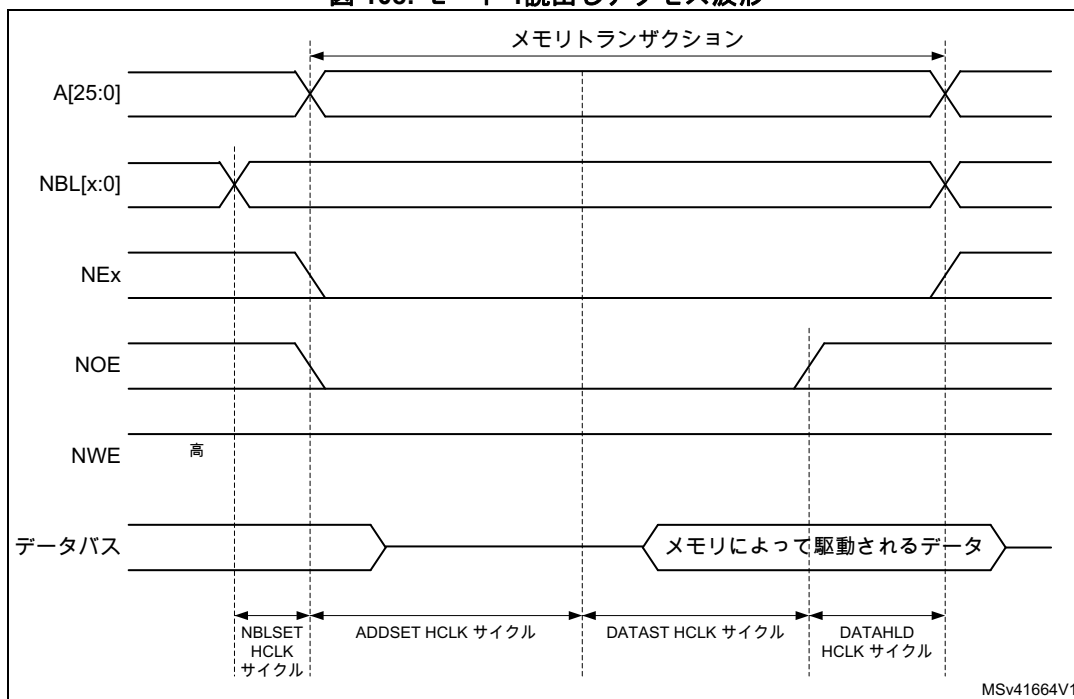
モードをミックスすることができます。例えば、読出し動作をモード A で実行し、書込み動作をモード B で実行することができます。

- 拡張モードを無効にした場合 (FMC_BCRx レジスタで EXTMOD ビットをリセット)、FMC は次のようにモード 1 またはモード 2 で動作することができます。
 - SRAM/PSRAM メモリタイプを選択した場合 (FMC_BCRx レジスタで MTYP = 0x0 または 0x01)、モード 1 がデフォルトモードです。
 - NOR メモリタイプを選択した場合 (FMC_BCRx レジスタで MTYP = 0x10)、モード 2 がデフォルトモードです。

モード 1 - SRAM/FRAM/PSRAM (CRAM)

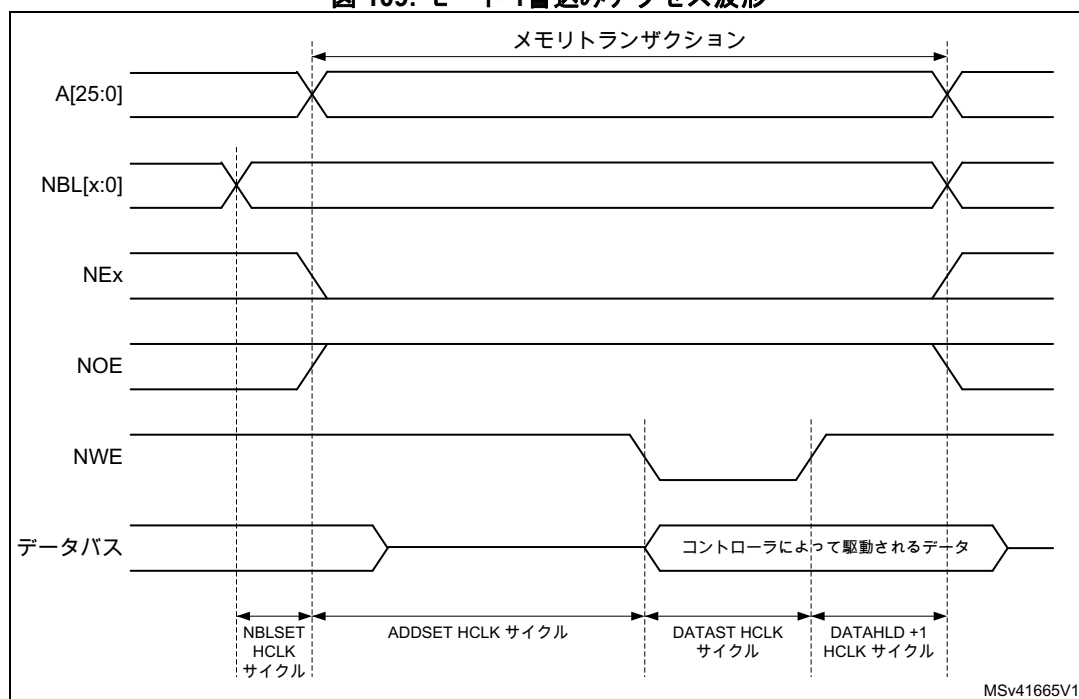
次の図に、サポートされているモードに対する読出しトランザクションと書込みトランザクションを示します。それに続いて、FMC_BCRx レジスタ、および FMC_BTRx/FMC_BWTRx レジスタの必要とされる設定を示します。

図 108. モード 1 読出しアクセス波形



MSv41664V1

図 109. モード 1 書込みアクセス波形



読出しおよび書込みトランザクションの最後の DATAHLD 時間は、NOE/NWE の立上がりエッジ後のアドレスおよびデータホールド時間を保証します。DATAST の値はゼロより大きくなければなりません (DATAST > 0)。

表 176. FMC_BCRx ビットフィールド (モード 1)

ビット番号	ビット名	設定値
31	FMCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5:4	MWID	必要に応じて設定します。

表 176. FMC_BCRx ビットフィールド (モード 1) (続き)

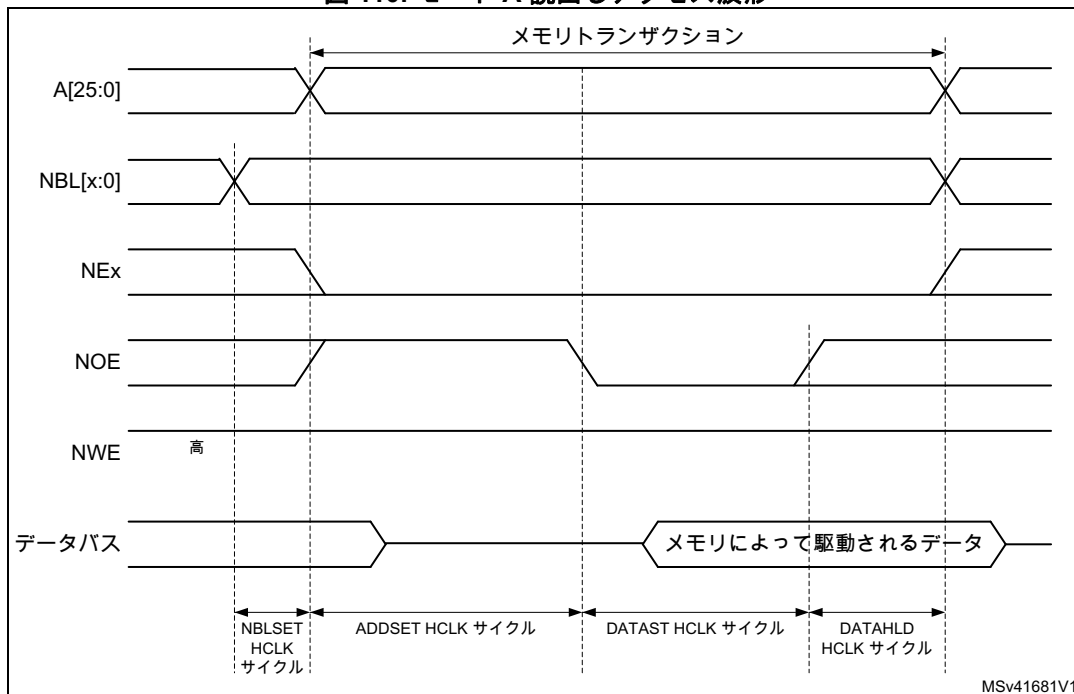
ビット番号	ビット名	設定値
3:2	MTYP	必要に応じて、0x2 (NOR Flash メモリ) を除きます。
1	MUXE	0x0
0	MBKEN	0x1

表 177. FMC_BTRx ビットフィールド (モード 1)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (読出しアクセスの場合は DATAHLD HCLK サイクル、書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	無視
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

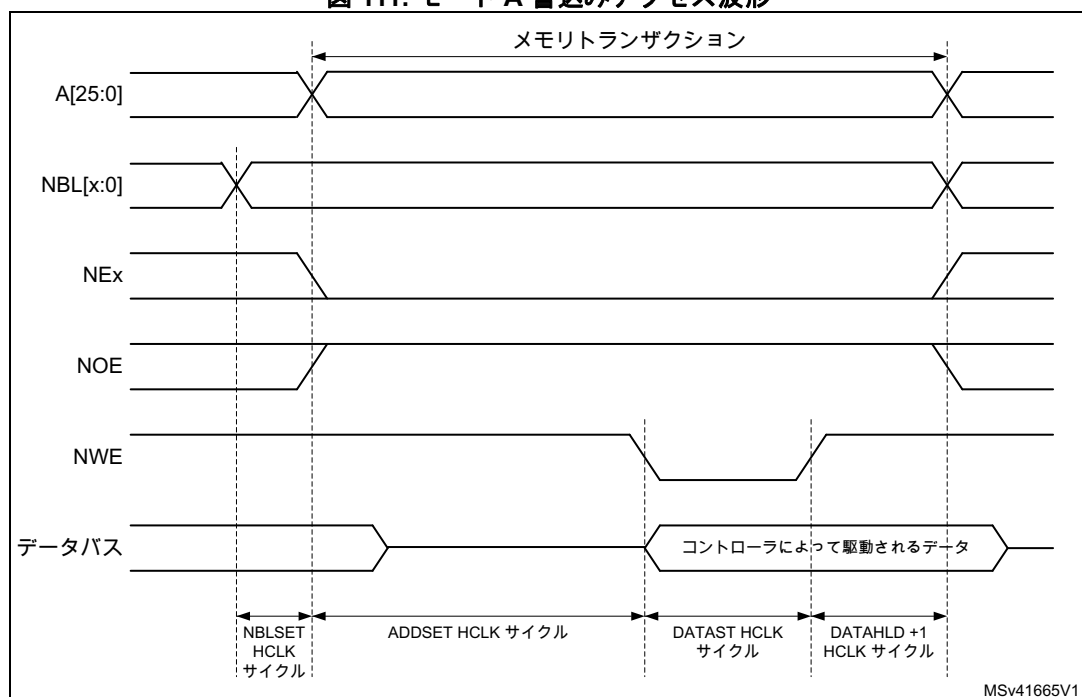
モード A - SRAM/FRAM/PSRAM (CRAM) OE トグル

図 110. モード A 読出しアクセス波形



1. NBL[1:0] は読出しアクセス中ローレベルに駆動されます。

図 111. モード A 書込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグルングと、独立した読出しおよび書込みのタイミングです。

表 178. FMC_BCRx ビットフィールド (モード A)

ビット番号	ビット名	設定値
31	FMCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5:4	MWID	必要に応じて設定します。

表 178. FMC_BCRx ビットフィールド (モード A) (続き)

ビット番号	ビット名	設定値
3:2	MTYP	必要に応じて、0x2 (NOR Flash メモリ) を除きます。
1	MUXEN	0x0
0	MBKEN	0x1

表 179. FMC_BTRx ビットフィールド (モード A)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (読出しアクセスの場合は DATAHLD HCLK サイクル)。
29:28	ACCMOD	0x0
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	読出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	読出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

表 180. FMC_BWTRx ビットフィールド (モード A)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	0x0
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	書込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	書込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

モード 2/B - NOR 型 Flash

図 112. モード 2およびモード B 読出しアクセス波形

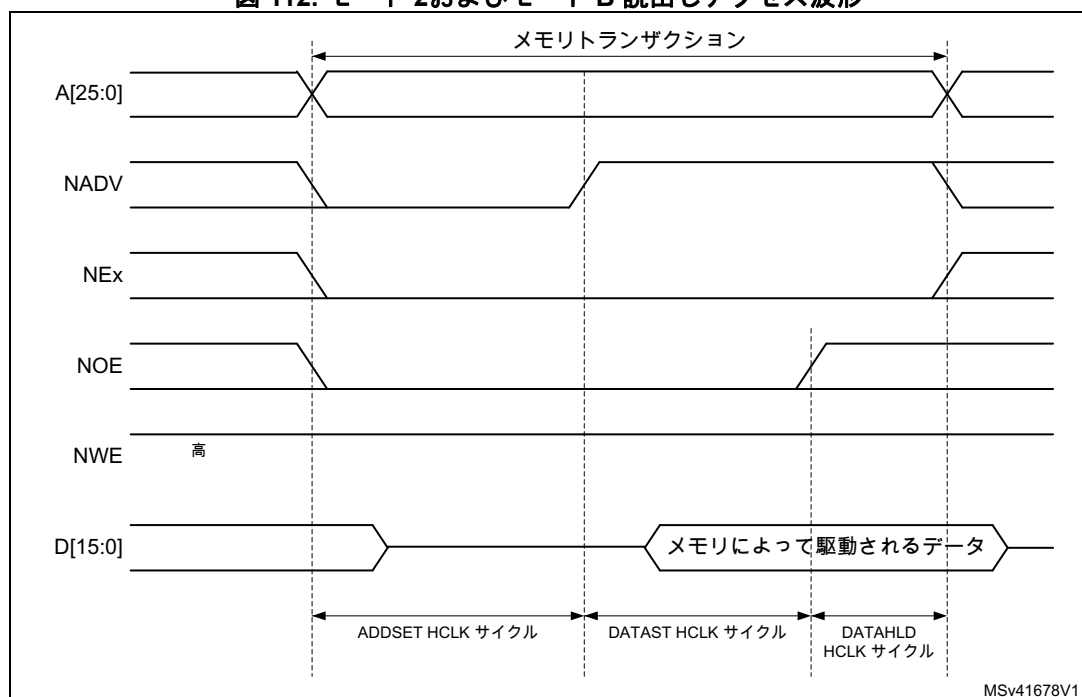


図 113. モード 2書き込みアクセス波形

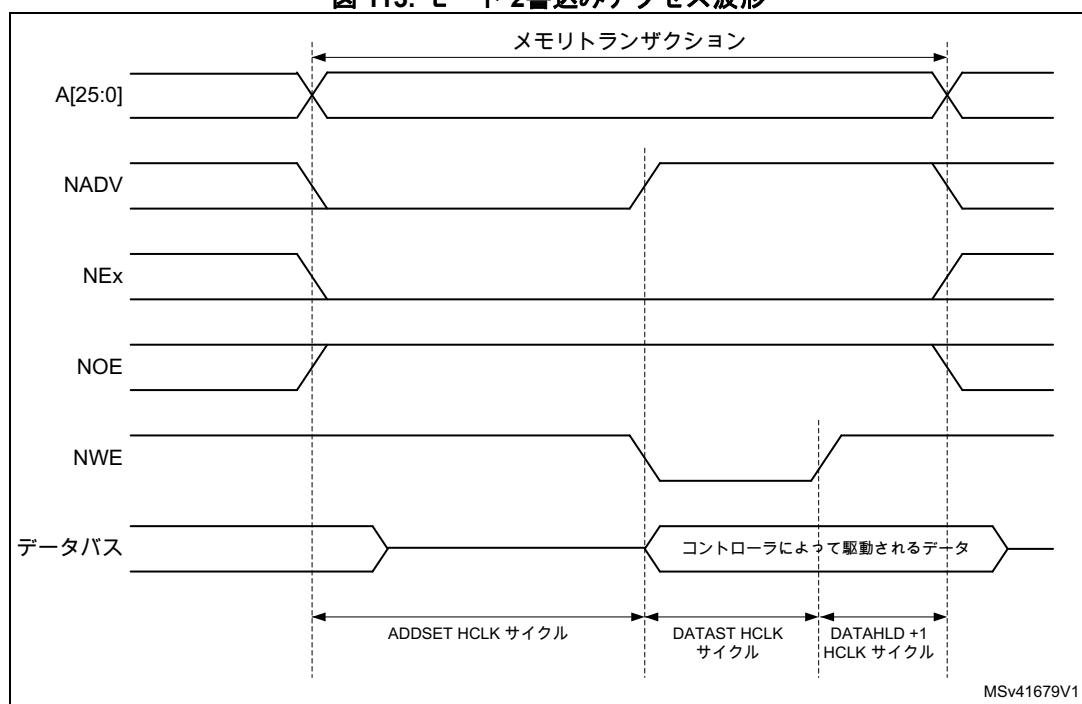
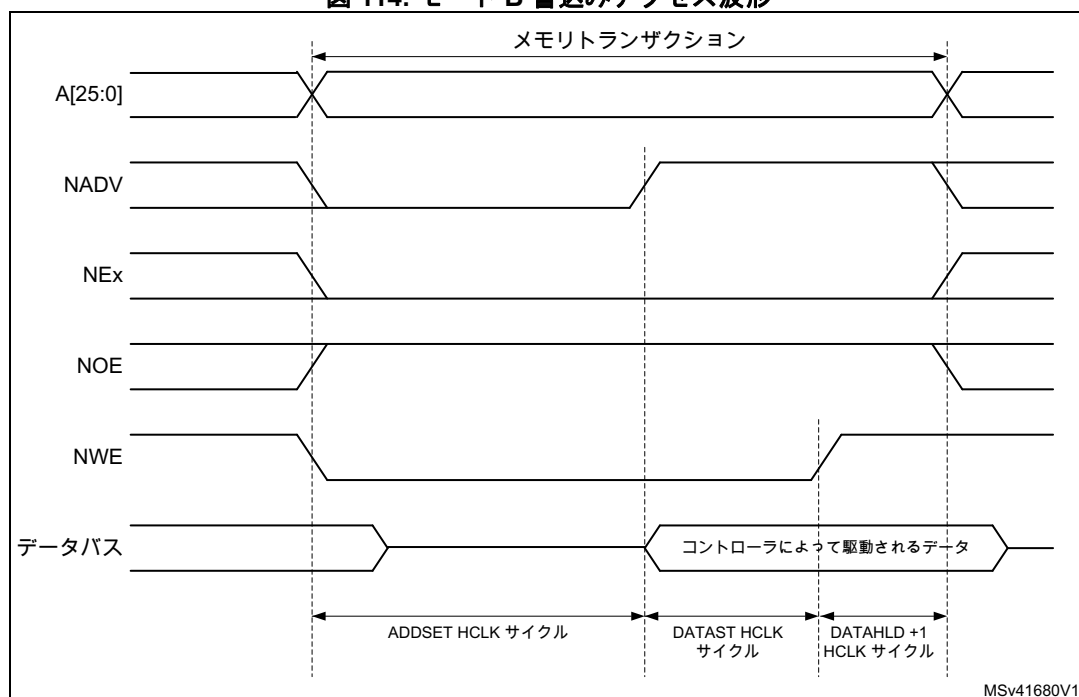


図 114. モード B 書込みアクセス波形



モード 1 と比較した場合の相違点は、NWE のトグルリングと拡張モードがセットされたとき（モード B）の独立した読出しおよび書込みのタイミングです。

表 181. FMC_BCRx ビットフィールド（モード 2/B）

ビット番号	ビット名	設定値
31	FMCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	無視
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0（非同期モードで無効）
18:16	CPSIZE	0x0（非同期モードで無効）
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	モード B では 0x1、モード 2 では 0x0
13	WAITEN	0x0（非同期モードで無効）
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5:4	MWID	必要に応じて設定します。

表 181. FMC_BCRx ビットフィールド (モード 2/B) (続き)

ビット番号	ビット名	設定値
3:2	MTYP	0x2 (NOR 型 Flash メモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 182. FMC_BTRx ビットフィールド (モード 2/B)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (拡張モードが無効のとき、読出しアクセスの場合は DATAHLD HCLK サイクル、書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	拡張モードがセットされている場合は 0x1
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	読出しアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	読出しアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 183. FMC_BWTRx ビットフィールド (モード 2/B)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	拡張モードがセットされている場合は 0x1
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	書込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	書込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

注 : FMC_BWTRx レジスタは、拡張モードがセットされている場合 (モード B) のみ有効です。それ以外の場合には、その内容は意味を持ちません。

モード C - NOR Flash - OE トグリング

図 115. モード C 読出しアクセス波形

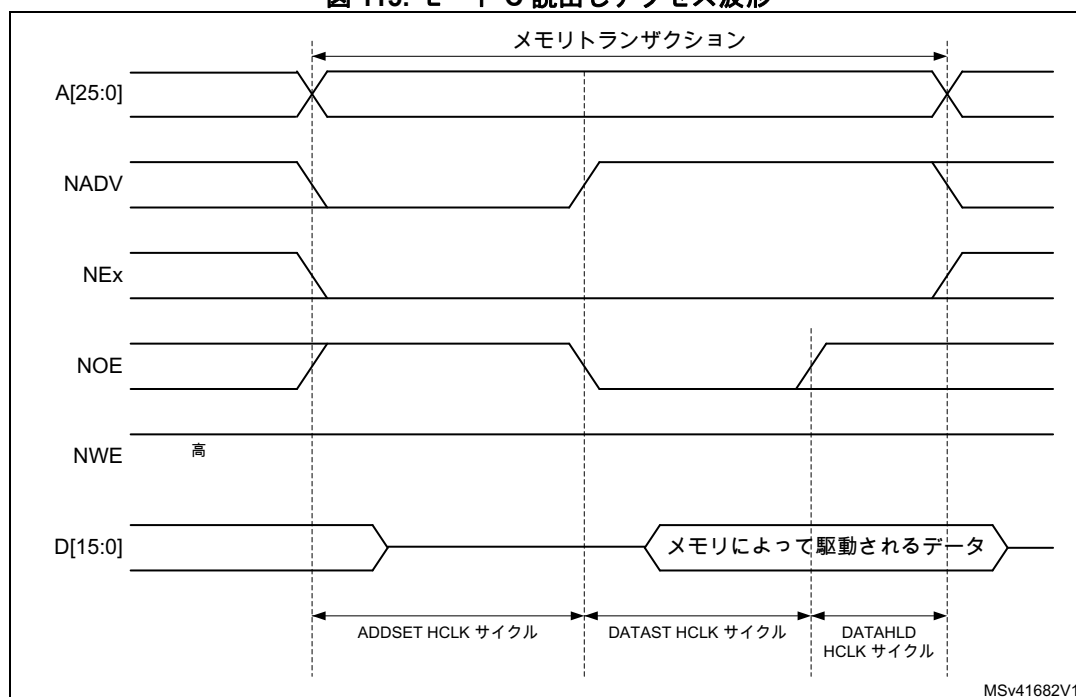
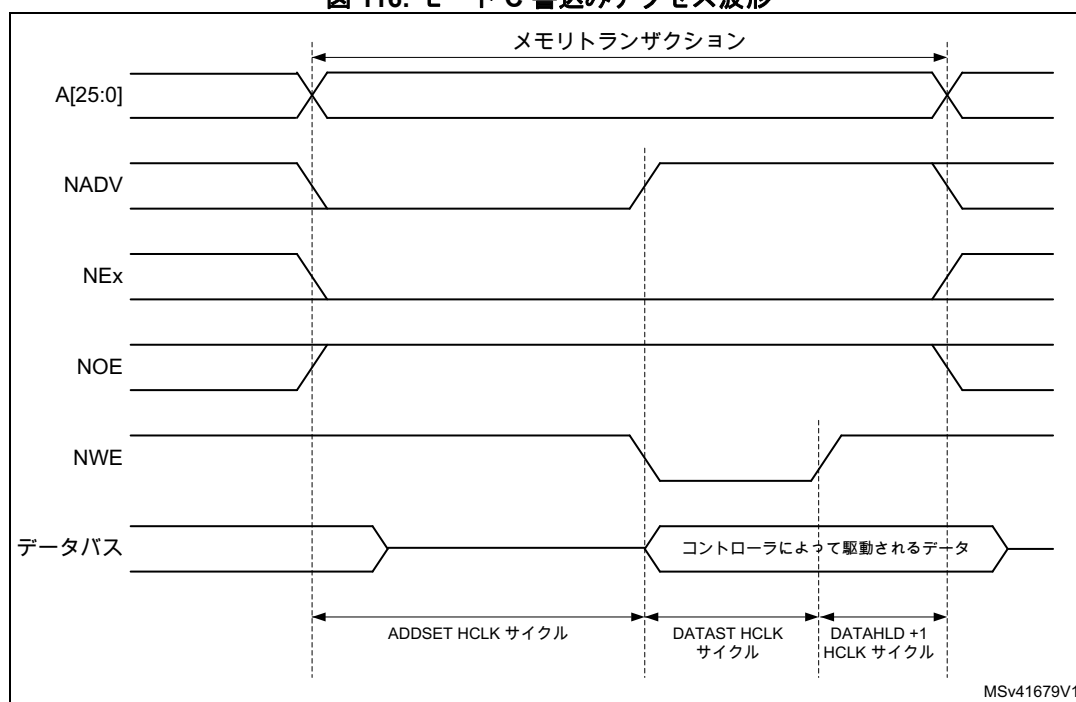


図 116. モード C 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグリングと、独立した読出しおよび書き込みのタイミングです。

表 184. FMC_BCRx ビットフィールド (モード C)

ビット番号	ビット名	設定値
31	FMCEEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	無視
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCAWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5:4	MWID	必要に応じて設定します。
3:2	MTYP	0x02 (NOR 型 Flash メモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 185. FMC_BTRx ビットフィールド (モード C)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (読出しアクセスの場合は DATAHLD HCLK サイクル)。
29:28	ACCMOD	0x2
27:24	DATLAT	0x0
23:20	CLKDIV	0x0
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	読出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	読出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 186. FMC_BWTRx ビットフィールド (モード C)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	0x2
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	書込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	書込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

モード D - 拡張アドレスによる非同期アクセス

図 117. モード D 読出しアクセス波形

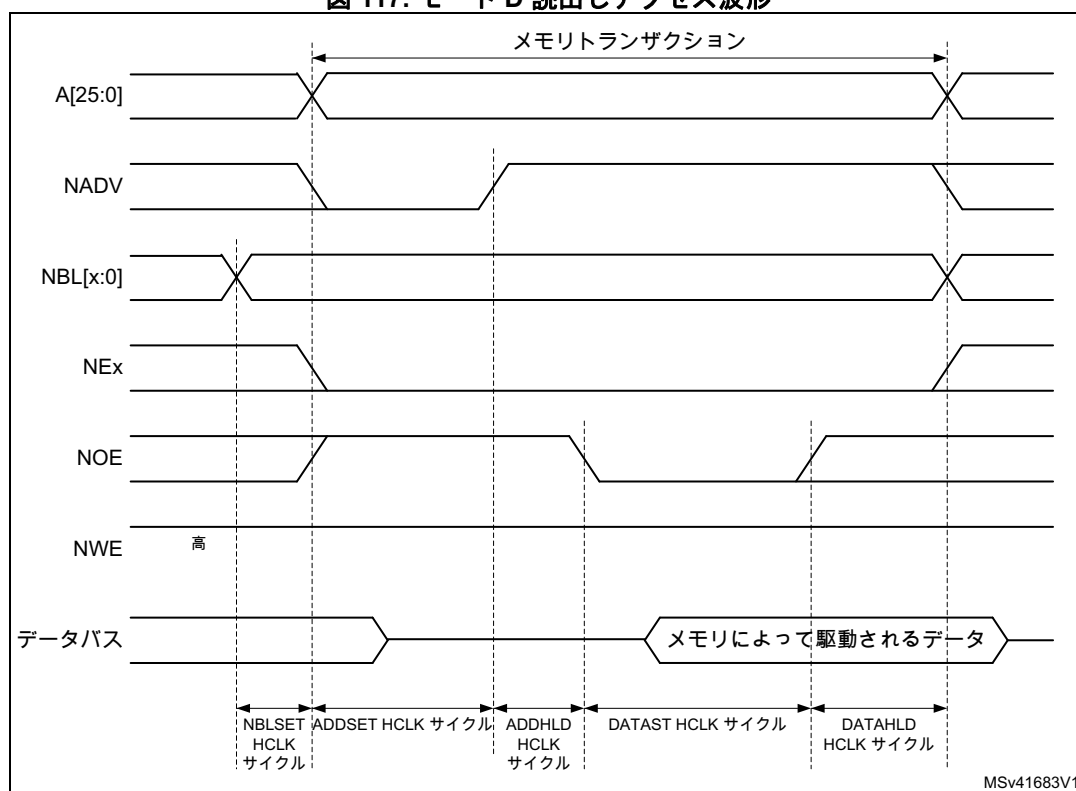
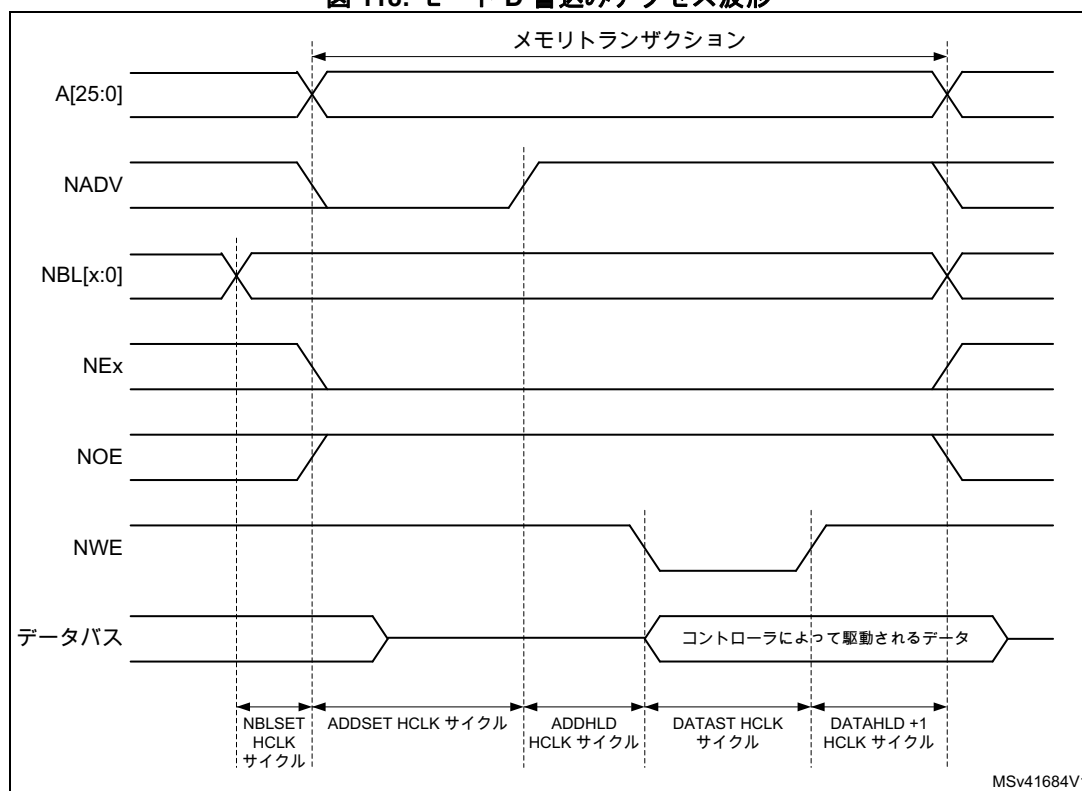


図 118. モード D 書込みアクセス波形



モード 1 と比較した場合の相違点は、NADV が変化した後にはトグルを続ける NOE のトグル と、独立した読出し書込みタイミングです。

表 187. FMC_BCRx ビットフィールド (モード D)

ビット番号	ビット名	設定値
31	FMCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1

表 187. FMC_BCRx ビットフィールド (モード D) (続き)

ビット番号	ビット名	設定値
6	FACCEN	メモリのサポート状況に応じて設定します。
5:4	MWID	必要に応じて設定します。
3:2	MTYP	必要に応じて設定します。
1	MUXEN	0x0
0	MBKEN	0x1

表 188. FMC_BTRx ビットフィールド (モード D)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (読出しアクセスの場合は DATAHLD HCLK サイクル)。
29:28	ACCMOD	0x3
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	読出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	読出しにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3:0	ADDSET	読出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 1。

表 189. FMC_BWTRx ビットフィールド (モード D)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	0x3
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	書込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3:0	ADDSET	書込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 1。

マルチプレクス・モード - NOR 型 Flash メモリに対するマルチプレクス非同期アクセス

図 119. マルチプレクス読出しアクセス波形

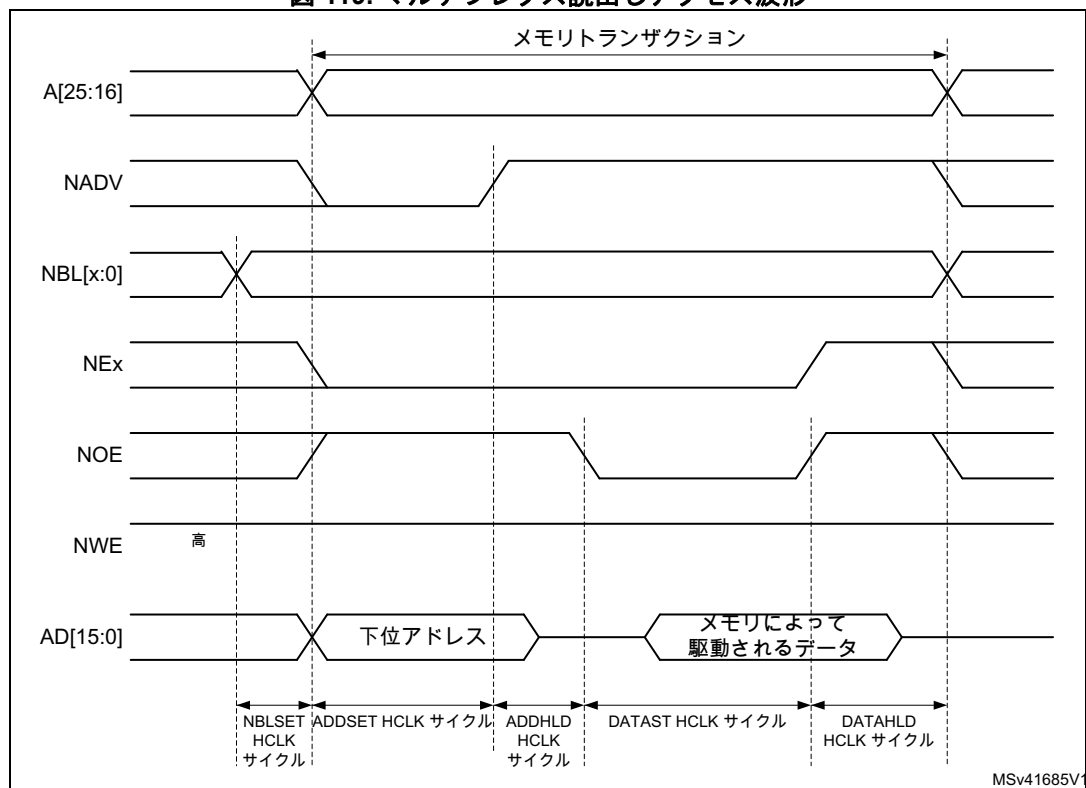
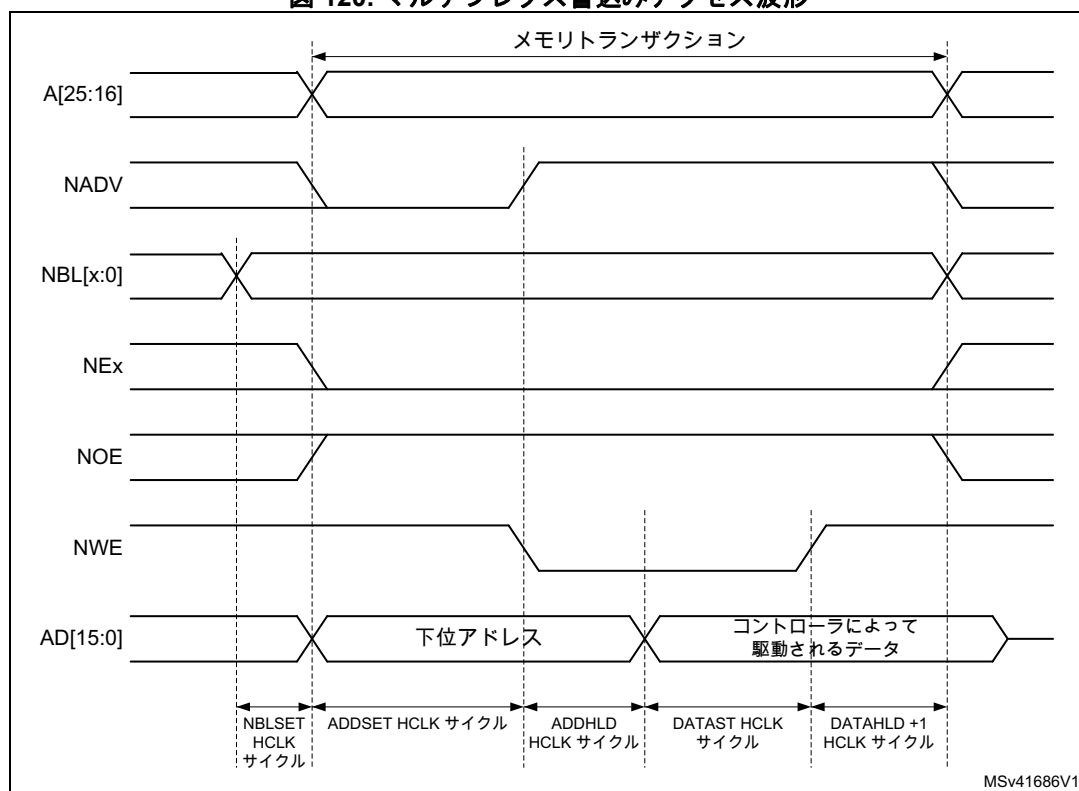


図 120. マルチプレクス書込みアクセス波形



モード D との相違点は、データバス上の下位アドレスバイトのドライブです。

表 190. FMC_BCRx ビットフィールド (マルチプレクス・モード)

ビット番号	ビット名	設定値
31	FMCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1

表 190. FMC_BCRx ビットフィールド (マルチプレクス・モード) (続き)

ビット番号	ビット名	設定値
6	FACCEN	0x1
5:4	MWID	必要に応じて設定します。
3:2	MTYP	0x2 (NOR 型 Flash メモリ) または 0x1 (PSRAM)
1	MUXEN	0x1
0	MBKEN	0x1

表 191. FMC_BTRx ビットフィールド (マルチプレクス・モード)

ビット番号	ビット名	設定値
31:30	DATAHLD	データホールドフェーズの時間 (読出しアクセスの場合は DATAHLD HCLK サイクル、書込みアクセスの場合は DATAHLD+1 HCLK サイクル)。
29:28	ACCMOD	0x0
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15:8	DATAST	2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	アクセスの中間フェーズの時間 (ADDHLD HCLK サイクル)。
3:0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

非同期アクセスにおけるウェイト管理

非同期メモリが、データの受け取りまたは提供の準備がまだできていないことを知らせるために、WAIT 信号をアサートした場合、FMC_BCRx レジスタの ASYNCWAIT ビットをセットする必要があります。

WAIT 信号がアクティブな場合 (WAITPOL ビットに依存して、ハイまたはロー)、DATAST ビットによってプログラムされる 2 番目のアクセスフェーズ (データセットアップフェーズ) は、WAIT ビットが非アクティブになるまで延長されます。データセットアップフェーズとは異なり、ADDSET および ADDHLD ビットによってプログラムされる最初のアクセスフェーズ (アドレスセットアップおよびアドレスホールドフェーズ) は、WAIT に対応しないので、延長されません。

データセットアップフェーズは、メモリトランザクションが終了する 4 HCLK サイクル前に WAIT が検出されるようにプログラムする必要があります。次のようなケースを考慮する必要があります。

1. メモリが、次のようにトグルする NOE/NWE に合わせた WAIT 信号をアサートする場合。

$$\text{DATA_ST} \geq (4 \times \text{HCLK}) + \text{max wait assertion time}$$

2. メモリが、NEX（またはトグルしない NOE/NWE）に合わせた WAIT 信号をアサートする場合。
次の条件が成立するならば、

$$\text{max_wait_assertion_time} > \text{address_phase} + \text{hold_phase}$$

以下の通りです。

$$\text{DATA_ST} \geq (4 \times \text{HCLK}) + (\text{max_wait_assertion_time} - \text{address_phase} - \text{hold_phase})$$

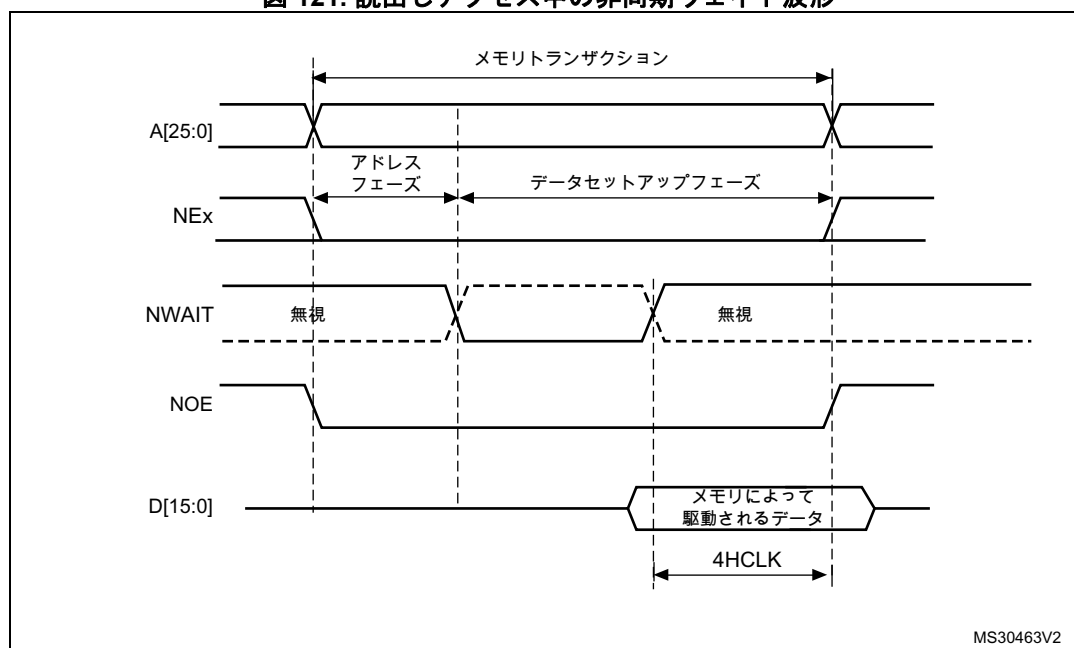
それ以外の場合は、

$$\text{DATA_ST} \geq 4 \times \text{HCLK}$$

ここで、max_wait_assertion_time は、NEX/NOE/NWE がローになったときにメモリが WAIT 信号をアサートするために必要な最大時間です。

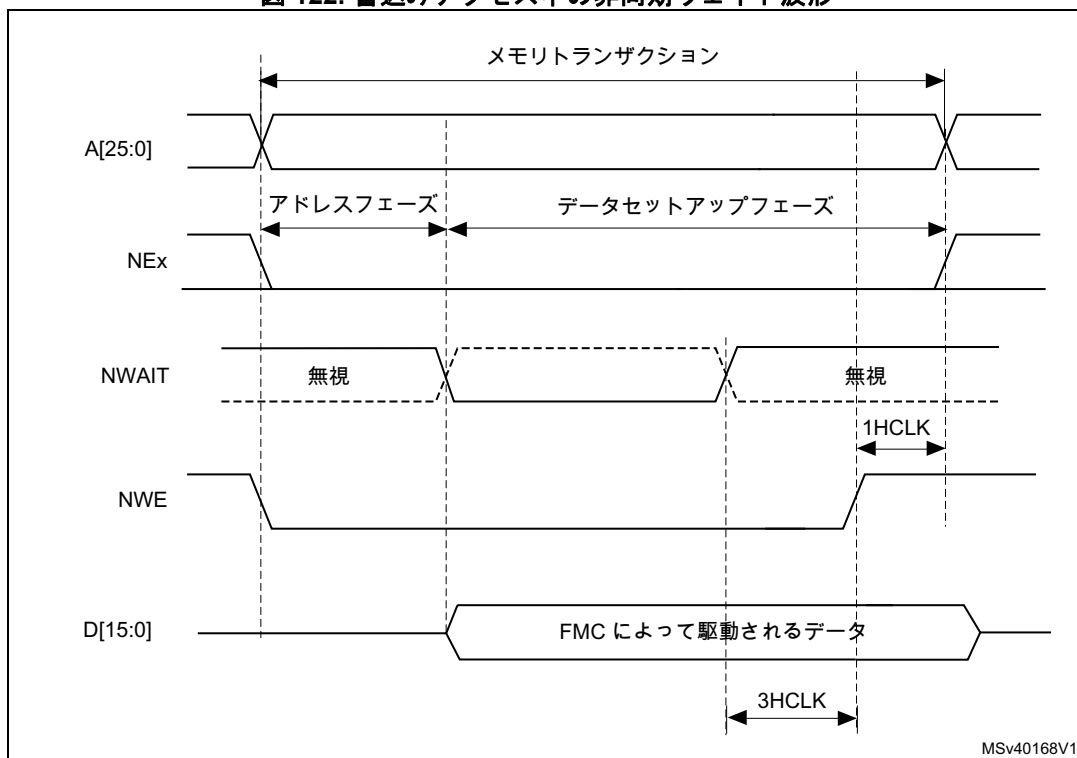
図 121 と 図 122 に、非同期メモリによって WAIT が解除された後、メモリ・アクセス・フェーズに追加される HCLK クロックのサイクル数を示します（上記のケースに無関係）。

図 121. 読出しアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC BCRx レジスタの WAITPOL ビット設定値に依存します。

図 122. 書き込みアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC_BCRx レジスタの WAITPOL ビット設定値に依存します。

CellularRAM™ (PSRAM) のリフレッシュ管理

CellularRAM™ では、チップ・セレクト信号 (NE) をメモリ・デバイスに指定されている t_{CEM} タイミングよりも長くローに維持することはできません。このタイミングは FMC_PCSCNTR レジスタでプログラムすることができます。そこには、非同期アクセスの場合は HCLK サイクルで、同期アクセスの場合は FMC_CLK サイクルで、NE のローパルスの最大期間を定義します。

22.6.5 同期トランザクション

メモリクロック FMC_CLK は、HCLK の約数です。この値は CLKDIV 値と MWID/AHB データサイズに依存し、次式で与えられます。

MWID サイズが 16 ビットまたは 8 ビットの場合、FMC_CLK の分周比は常に CLKDIV のプログラムされた値で決まります。

例：

- CLKDIV = 1、MWID = 16ビット、AHB データサイズ = 8 ビットの場合、 $FMC_CLK = HCLK / 2$

NOR 型 Flash メモリは、NADV のアサーションから CLK がハイになるまでの最小時間を指定します。この制約を満たすために、FMC は、同期アクセスの最初の内部クロックサイクルでは (NADV アサーションの前)、クロックをメモリに供給しません。これにより、メモリクロックの立ち上がりエッジは、NADV ローパルスの中間で発生します。

データ遅延と NOR メモリ遅延

データ遅延は、データをサンプリングする前のウェイトサイクル数です。DATLAT の値は、NOR 型 Flash 設定レジスタで指定された遅延の値と一致しなければなりません。FMC は、NADV がローのときのクロックサイクルをデータ遅延カウントに含めません。

注意： 一部の NOR Flash メモリは、データ遅延カウントに NADV ローサイクルを含めるので、NOR Flash の遅延と FMC DATLAT パラメータの厳密な関係は、次のいずれかです。

- NOR 型 Flash 遅延 = (DATLAT + 2) CLK クロック・サイクル、または
- NOR 型 Flash 遅延 = (DATLAT + 3) CLK クロック・サイクル

最近のメモリの中には、遅延フェーズにおいて NWAIT をアサートするものがあります。そのような場合には、DATLAT を最小値に設定することができます。結果として、FMC はデータをサンプリングして、データが有効であるかどうか評価するのに十分な時間、ウェイトします。このように、FMC は、メモリが遅延から抜けたことを検出し、実際のデータが処理されます。

その他のメモリは、遅延中には NWAIT をアサートしません。この場合、FMC とメモリの両方について、遅延が正しく設定されなければなりません。そうしないと、正しいデータの代わりに無効なデータが取られたり、有効なデータがメモリサイクルの初期フェーズで失われることになります。

シングルバースト転送

選択されたバンクが同期アクセスのバーストモードに設定されている場合、例えば、16 ビットメモリで AHB シングルバーストトランザクションがリクエストされると、FMC は長さ 1 (AHB 転送が 16 ビットの場合) または 2 (AHB 転送が 32 ビットの場合) のバーストトランザクションを実行して、最後のデータがストローブされると、チップセレクト信号をデアサートします。

このような転送はサイクル数の点で、非同期読出し動作に比較して最も効率的ではありません。しかし、時間がかかるメモリアクセスモードの再プログラムには、まず、ランダム非同期アクセスが必要です。

CellularRAM™ 1.5 のページ境界またぎ

CellularRAM™ 1.5 では、ページ境界をまたぐバーストアクセスは許可されていません。メモリ・ページ・サイズに従って FMC_BCR1 レジスタの CPSIZE ビットを設定することで、メモリ・ページ・サイズに達した時点で自動的にバースト・アクセスを分割するために FMC コントローラを使用します。

ウェイト管理

同期 NOR Flash メモリでは、NWAIT がプログラムされた遅延期間 ((DATLAT+2) CLK サイクルに対応) 後に評価されます。

NWAIT がアクティブの場合 (WAITPOL = 0 ではローレベル、WAITPOL = 1 ではハイレベル)、ウェイトステートは NWAIT が非アクティブ (WAITPOL = 0 ではハイレベル、WAITPOL = 1 ではローレベル) になるまで挿入されます。

NWAIT が非アクティブのときには、データはただちに (ビット WAITCFG=1) または次のクロックエッジで (ビット WAITCFG=0)、有効とみなされます。

NWAIT 信号によるウェイトステートの挿入時には、コントローラはメモリへのクロックパルスの送信を続け、チップセレクトと出力イネーブル信号を有効に維持します。データは有効と見なしません。

バースト・モードでの NOR Flash の NWAIT 信号には、以下の 2 つのタイミング設定があります。

- Flash メモリは、ウェイト・ステートの 1 データサイクル前に NWAIT 信号をアサートします (リセット後のデフォルト)。
- Flash メモリは、ウェイト・ステートの間に NWAIT 信号をアサートします。

FMC は FMC_BCRx レジスタ (x = 0..3) の WAITCFG ビットを使って、各チップセレクトに対する両 NOR Flash ウェイト・ステート設定をサポートします。

図 123. 待ち設定波形

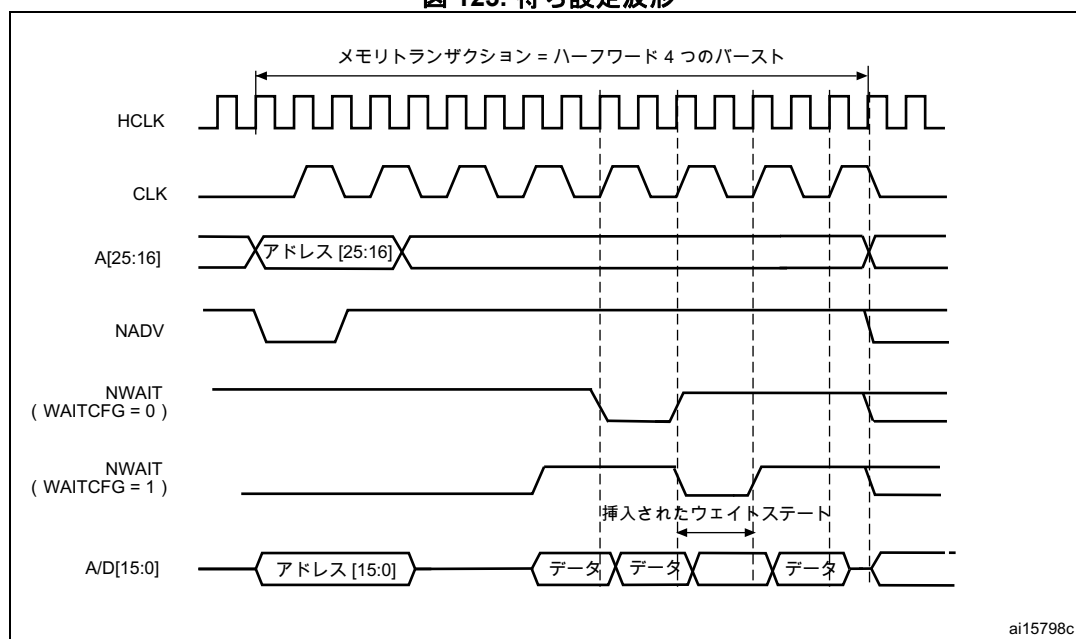
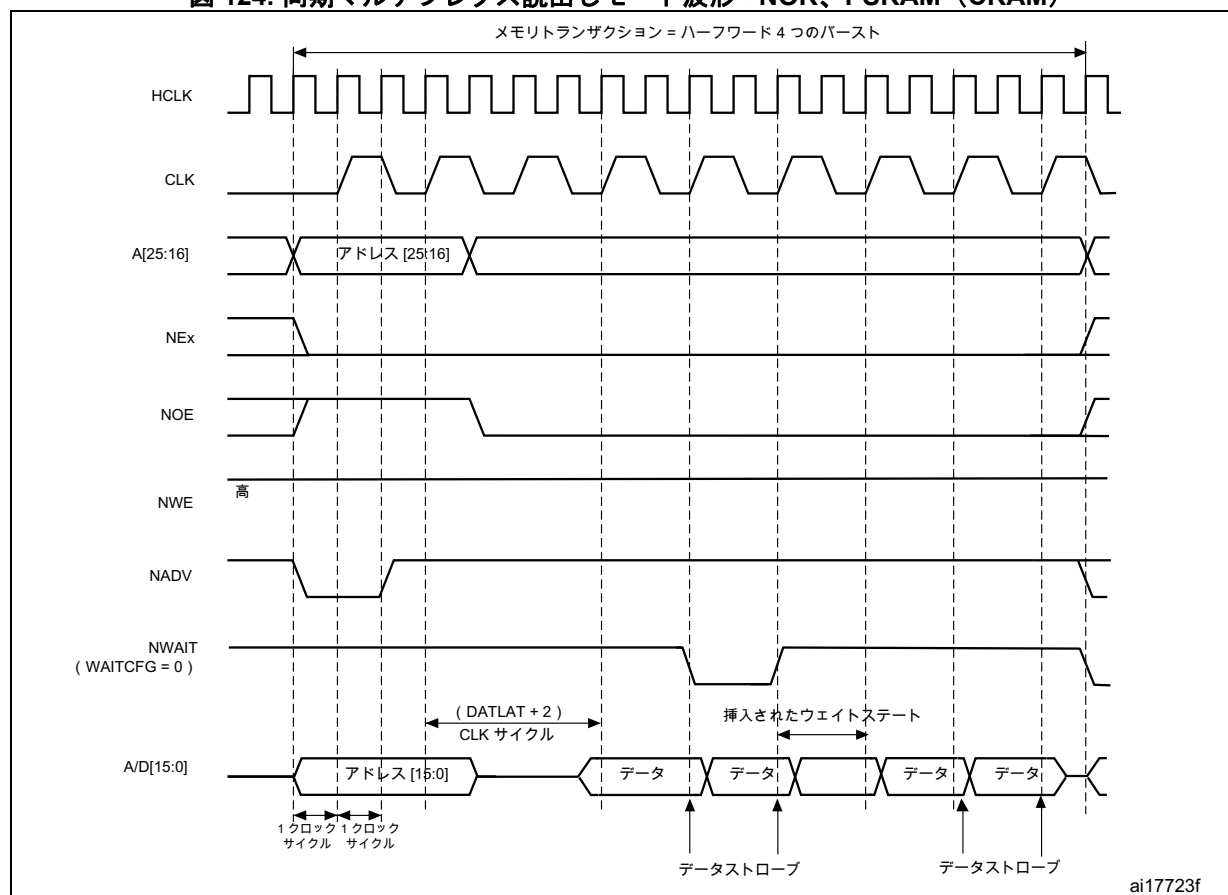


図 124. 同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)



1. バイトレーン出力 (NBL は示されていません。NOR アクセス時にはハイに保たれ、PSRAM (CRAM) アクセス時にはローに保たれます。)

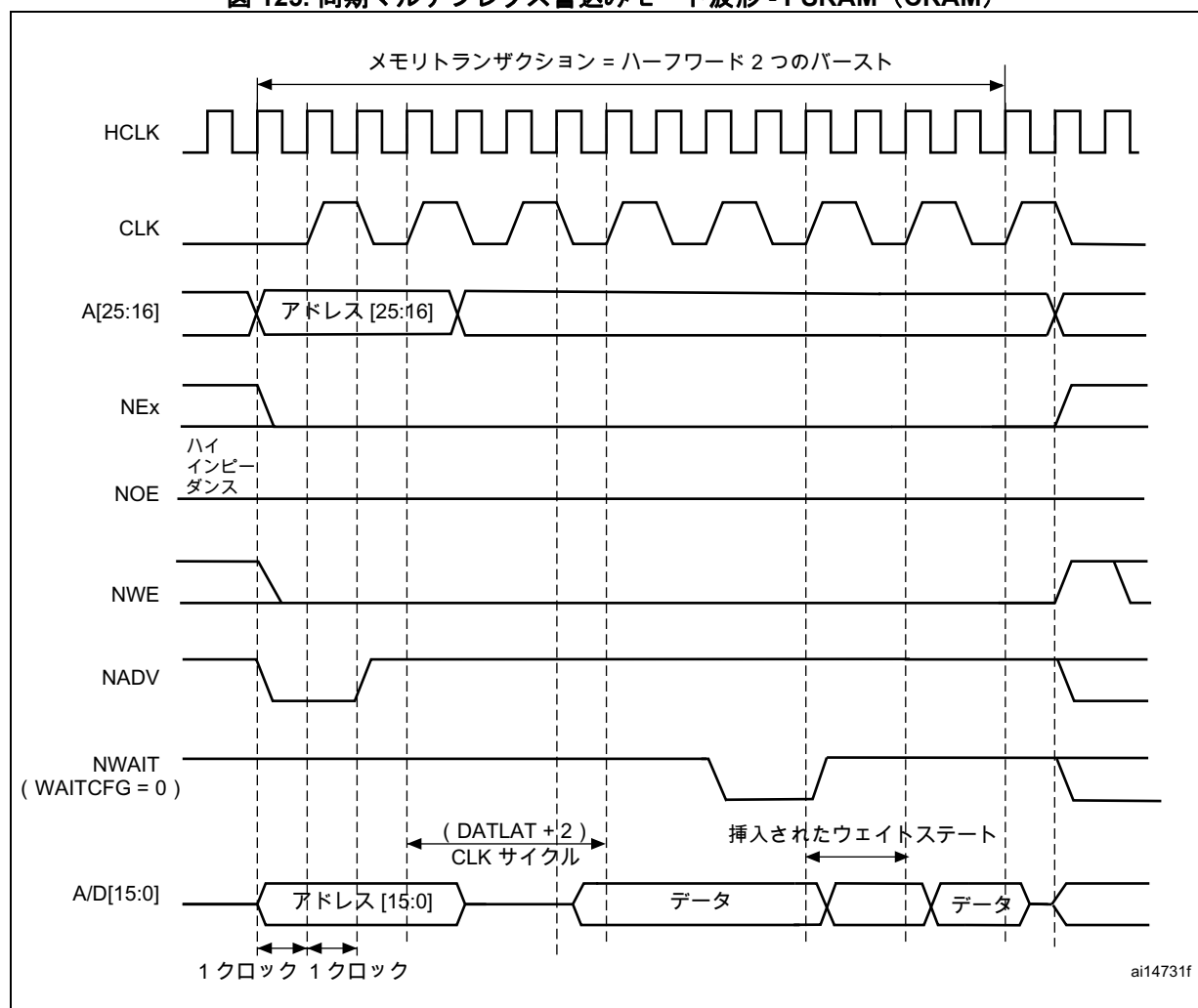
表 192. FMC_BCRx ビットフィールド (同期マルチプレクス読出しモード)

ビット番号	ビット名	設定値
31	FMCCEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	無視
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	同期読出し時は影響しません。
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	同期読出し時は影響しません。
11	WAITCFG	メモリに応じて設定します。
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	0x1
7	予約済み	0x1
6	FACCEN	メモリ・サポートに従って設定します (NOR 型 Flash メモリ)
5~4	MWID	必要に応じて設定します。
3~2	MTYP	0x1 または 0x2
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 193. FMC_BTRx ビットフィールド (同期マルチプレクス読出しモード)

ビット番号	ビット名	設定値
31:30	DATAHLD	無視
29:28	ACCMOD	0x0
27~24	DATLAT	データ遅延
27~24	DATLAT	データ遅延
23~20	CLKDIV	0x0 の場合、CLK=HCLK となります。 0x1 の場合、CLK=2×HCLK となります。 ..
19~16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15~8	DATAST	無視
7~4	ADDHLD	無視
3~0	ADDSET	無視

図 125. 同期マルチプレクス書込みモード波形 - PSRAM (CRAM)



1. メモリは、NWAIT 信号を 1 サイクル前に発行しなければならないので、WAITCFG を 0 にプログラムする必要があります。
2. バイトレーン (NBL) 出力は、図に示されており、NEx がアクティブな間はローに保たれます。

表 194. FMC BCRx ビットフィールド（同期マルチプレクス書込みモード）

ビット番号	ビット名	設定値
31	FMCEEN	0x1
30 : 24	予約済み	0x000
23:22	NBLSET[1:0]	無視
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x1
18:16	CPSIZE	必要に応じて (CRAM 1.5 は 0x1)
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。

表 194. FMC_BCRx ビットフィールド (同期マルチプレクス書込みモード) (続き)

ビット番号	ビット名	設定値
12	WREN	0x1
11	WAITCFG	0x0
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	同期書込み時は影響しません。
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5~4	MWID	必要に応じて設定します。
3~2	MTYP	0x1
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 195. FMC_BTRx ビットフィールド (同期マルチプレクス書込みモード)

ビット番号	ビット名	設定値
31~30	DATAHLD	無視
29:28	ACCMOD	0x0
27~24	DATLAT	データ遅延
23~20	CLKDIV	0x0 の場合、CLK=HCLK となります。 0x1 の場合、CLK=2×HCLK となります。
19~16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)。
15~8	DATAST	無視
7~4	ADDHLD	無視
3~0	ADDSET	無視

22.6.6 NOR/PSRAM コントローラレジスタ

SRAM/NOR 型 Flash のバンク x のチップセレクト制御レジスタ (FMC_BCRx) (x = 1~4)

アドレスオフセット : $0x00 + 0x8 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 30DB, 0x0000 30D2, 0x0000 30D2, 0x0000 30D2

このレジスタは、SRAM、PSRAM、FRAM、および NOR 型 Flash メモリで使用される各メモリ・バンクの制御情報を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		WFDIS	CCLK EN	CBURST RW	CPSIZE[2:0]		
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCW AIT	EXT MOD	ウェイト EN	WREN	ウェイト CFG	Res.	ウェイト POL	BURSTE N	Res.	FACC EN	MWID[1:0]		MTYP[1:0]		マルチ プレク ス EN	MBK EN
rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31 **FMCEN** : FMC コントローラ有効化

このビットは、FMC コントローラを有効または無効にします。

0 : FMC コントローラを無効にします。

1 : FMC コントローラを有効にします。

注 : **FMC_BCR2..4 レジスタの FMCEN ビットは無視されます。このビットは FMC_BCR1 レジスタからのみ有効化できます。**

ビット 30:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **NBLSET[1:0]** : バイト・レーン (NBL) セットアップ

これらのビットは、NBLx ローからチップセレクト NEx ローまでの NBL セットアップ時間を設定します。

00 : NBL セットアップ時間は、0 AHB クロックサイクルです。

01 : NBL セットアップ時間は、1 AHB クロックサイクルです。

10 : NBL セットアップ時間は、2 AHB クロックサイクルです。

11 : NBL セットアップ時間は、3 AHB クロックサイクルです。

ビット 21 **WFDIS** : 書込み FIFO 無効化

このビットは、FMC コントローラで使用される書込み FIFO を無効にします。

0 : 書込み FIFO 有効 (リセット後のデフォルト)

1 : 書込み FIFO 無効

注 : **FMC_BCR2..4 レジスタの WFDIS ビットは無視されます。このビットは FMC_BCR1 レジスタからのみ有効化できます。**

ビット 20 CCLKEN : 連続クロック有効化

このビットは、外部メモリデバイスに対する FMC_CLK クロック出力を有効化します。

0 : FMC_CLK は同期メモリアクセス（読み出し／書き込みトランザクション）時にのみ発生されます。FMC_CLK クロック比は、FMC_BCRx レジスタに設定した CLKDIV 値により指定されます（リセット後のデフォルト）。

1 : FMC_CLK は、非同期アクセスと同期アクセス時に連続的に発生されます。CCLKEN がセットされると、FMC_CLK クロックがアクティブになります。

注： FMC_BCR2..4 レジスタの CCLKEN ビットは無視されます。このビットは FMC_BCR1 レジスタからのみ有効化できます。FMC_CLK 連続クロックを発生するときは、バンク 1 を同期モードに設定する必要があります。

注： CCLKEN ビットがセットされている場合、FMC_CLK クロック比は FMC_BTR1 レジスタの CLKDIV 値で指定されます。FMC_BWTR1 の CLKDIV は、無視されます。

注： 同期モードを使用し、かつ CCLKEN ビットがセットされる場合、バンク 1 以外のバンクに接続される同期メモリはすべて、同じクロックで駆動されます（他のバンクに対する FMC_BTR2..4 レジスタと FMC_BWTR2..4 レジスタの CLKDIV 値は無効です）。

ビット 19 CBURSTRW : 書き込みバースト有効化

バーストモードで動作する PSRAM (CRAM) の場合、このビットは書き込み動作時に同期アクセスを有効にします。同期読み出しアクセスのイネーブルビットは、FMC_BCRx レジスタの BURSTEN ビットです。

0 : 書き込み動作は、常に非同期モードで実行されます。

1 : 書き込み動作は、同期モードで実行されます。

ビット 18:16 CPSIZE[2:0] : CRAM ページサイズ

これらのビットは、ページ間でアドレス境界をまたぐバーストアクセスができない CellularRAM™ 1.5 に使用されます。これらのビットを設定すると、FMC コントローラは、メモリページサイズに達した時点で自動的にバーストアクセスを分割します（ページサイズについてはメモリのデータシートを参照）。

000 : ページ境界をまたぐ場合、バーストは分割されません（リセット後のデフォルト）

001 : 128 バイト

010 : 256 バイト

011 : 512 バイト

100 : 1024 バイト

その他 : 予約済み

ビット 15 ASYNCWAIT : 非同期転送中のウェイト信号

このビットは、非同期プロトコル時でも、FMC がウェイト信号を有効化／無効化できるようにします。

0 : 非同期プロトコル動作中は、NWAIT 信号は考慮されません（リセット後のデフォルト）。

1 : 非同期プロトコル動作中に NWAIT 信号が考慮されます。

ビット 14 EXTMOD : 拡張モード有効化

このビットを使うと、FMC が FMC_BWTR レジスタで非マルチプレクス非同期アクセスの書き込みタイミングをプログラムできるため、読み出しと書き込みで異なるタイミングが可能になります。

0 : FMC_BWTR レジスタ内部の値は考慮されません（リセット後のデフォルト）。

1 : FMC_BWTR レジスタ内部の値が考慮されます。

注： 拡張モードを無効化した場合、FMC は次のようにモード 1 またはモード 2 で動作することができます。

－ SRAM/PSRAM メモリ・タイプを選択した場合（MTYP = 0x0 または 0x01）、モード 1 はデフォルト・モードです。

－ モード2は、NOR メモリタイプを選択した場合（MTYP = 0x10）のデフォルトモードです。

ビット 13 WAITEN : ウェイト有効ビット

このビットは、同期モードでFlashメモリをアクセスするとき NWAIT 信号を使用したウェイトステートの挿入を有効化／無効化します。

0 : NWAIT 信号は無効です（レベルは考慮されず、プログラムされた Flash 遅延時間後にウェイト・ステートは挿入されません）。

1 : NWAIT 信号は有効です（レベルが考慮され、アサートされた場合、プログラムされた Flash 遅延時間後にウェイト・ステートが挿入されます）（リセット後のデフォルト）。

ビット 12 WREN : 書き込み有効ビット

このビットは、FMC によるバンクへの書き込み動作の有効／無効を示します。

0 : FMC によるバンクへの書き込み動作は無効にされています。AHB エラーが報告されます。

1 : FMC によるバンクへの書き込み動作は有効にされています (リセット後のデフォルト)。

ビット 11 WAITCFG : ウェイトタイミング設定

NWAIT 信号は、メモリからのデータが有効か否か、または同期モードでメモリをアクセスするときウェイトステートを挿入すべきかを表示します。この設定ビットは、ウェイトステートの 1 クロック前、またはウェイトステート中に、メモリによって NWAIT がアサートされるかどうかを定義します。

0 : NWAIT 信号は、ウェイト・ステートの 1 データ・サイクル前にアクティブです (リセット後のデフォルト)。

1 : NWAIT 信号は、ウェイトステートの間アクティブです (PSRAM には不使用)。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 WAITPOL : ウェイト信号極性ビット

同期モードまたは非同期モードで使用されるメモリからのウェイト信号の極性を定義します。

0 : NWAIT アクティブ・ロー (リセット後のデフォルト)

1 : NWAIT アクティブ・ハイ

ビット 8 BURSTEN : バースト有効ビット

このビットは、読出し動作での同期アクセスを有効化/無効化します。これは、バースト・モードでの同期メモリ動作に対してのみ有効です。

0 : バーストアクセスモードは無効です (リセット後のデフォルト)。読出しアクセスは非同期モードで実行されます。

1 : バーストモードを有効化。読出しアクセスは同期モードで実行されます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 FACCEN : Flashアクセス有効化

NOR 型 Flash メモリ・アクセス動作を有効にします。

0 : 対応する NOR 型 Flash メモリアccessは無効です。

1 : 対応する NOR 型 Flash メモリ・アクセスは有効です (リセット後のデフォルト)。

ビット 5:4 MWID[1:0] : メモリのデータバス幅

外部メモリデバイスの幅を定義します。すべてのタイプのメモリに対して有効です。

00 : 8 ビット

01 : 16 ビット (リセット後のデフォルト)

10 : 予約済み

11 : 予約済み

ビット 3:2 MTYP[1:0] : メモリタイプ

対応するメモリ・バンクに接続される外部メモリのタイプを定義します。

00 : SRAM/FRAM (バンク 2...4 のリセット後のデフォルト)

01 : PSRAM (CRAM) / FRAM

10 : NOR 型 Flash/OneNAND Flash (バンク 1 のリセット後のデフォルト)

11 : 予約済み

ビット 1 MUXEN : アドレス / データマルチプレクス有効ビット

このビットがセットされているときには、データバス上でアドレスとデータがマルチプレクスされます。NOR および PSRAM メモリでのみ有効です。

0 : アドレス/データはマルチプレクスされません。

1 : アドレス/データはデータ・バス上でマルチプレクスされます (リセット後のデフォルト)。

ビット 0 MBKEN : メモリバンク有効ビット

メモリバンクを有効にします。リセット後は、バンク 1 は有効であり、その他はすべて無効です。無効なバンクにアクセスすると、AHB バスでエラーが発生します。

0 : 対応するメモリ・バンクは無効です。

1 : 対応するメモリ・バンクは有効です。

SRAM/NOR 型 Flash のバンク x のチップセレクト・タイミング・レジスタ (FMC_BTRx)

アドレスオフセット : $0x04 + 0x8 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0FFF FFFF

このレジスタは、SRAM、PSRAM、および NOR 型 Flash メモリで使用される各メモリ・バンクの制御情報を含みます。FMC_BCRx レジスタの EXTMOD ビットがセットされている場合、このレジスタは、書込みと読出しアクセスに分割されます。すなわち、読出しアクセスを設定するためのレジスタ（このレジスタ）と、書込みアクセスを設定するためのレジスタ（FMC_BWTRx レジスタ）の 2 つのレジスタが使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAS[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 DATAHLD[1:0] : データホールドフェーズ時間

これらのビットは、データホールドフェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ（図 108 ~ 図 120 を参照）、非同期アクセスで使用されます。

読出しアクセスの場合

00 : DATAHLD フェーズ時間 = 0 * HCLK クロックサイクル（デフォルト）

01 : DATAHLD フェーズ時間 = 1 * HCLK クロックサイクル

10 : DATAHLD フェーズ時間 = 2 * HCLK クロックサイクル

11 : DATAHLD フェーズ時間 = 3 * HCLK クロックサイクル

書込みアクセスの場合

00 : DATAHLD フェーズ時間 = 1 * HCLK クロックサイクル（デフォルト）

01 : DATAHLD フェーズ時間 = 2 * HCLK クロックサイクル

10 : DATAHLD フェーズ時間 = 3 * HCLK クロックサイクル

11 : DATAHLD フェーズ時間 = 4 * HCLK クロックサイクル

ビット 29:28 ACCMOD[1:0] : アクセスモード

タイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC_BCRx レジスタの EXTMOD ビットが 1 のときのみ考慮されます。

00 : アクセスモード A

01 : アクセスモード B

10 : アクセスモード C

11 : アクセスモード D

ビット 27:24 **DATLAT[3:0]** : (下のビット説明参照) 同期メモリのデータ遅延

読出し／書込みバーストモードを有効にした同期アクセスの場合 (BURSTEN / CBURSTRW ビットをセット)、最初のデータの読み書きの前にメモリに発行するメモリクロックサイクル数 (+2) を定義します。

このタイミングパラメータは、HCLK 周期ではなく、FMC_CLK 周期で表されます。

非同期アクセスでは、この値は無視されます。

0000 : 最初のバーストアクセスでは、2 CLK クロックサイクルのデータ遅延

1111 : 最初のバーストアクセスでは、17 CLK クロックサイクルのデータ遅延 (リセット後のデフォルト)

ビット 23:20 **CLKDIV[3:0]** : クロック分周比 (FMC_CLK 信号)

FMC_CLK クロック出力信号の周期を定義します。HCLK サイクル数で表されます。

0000 : FMC_CLK 周期 = 1 × HCLK 周期

0001 : FMC_CLK 周期 = 2 × HCLK 周期

0010 : FMC_CLK 周期 = 3 × HCLK 周期

1111 : FMC_CLK 周期 = 16 × HCLK 周期 (リセット後のデフォルト値)

非同期 NOR 型 Flash、SRAM、または PSRAM アクセスでは、この値は無視されます。

注 : FMC_CLK 分周比の式については、[セクション 22.6.5 : 同期トランザクション](#)を参照してください。

ビット 19:16 **BUSTURN[3:0]** : バスターンアラウンドフェーズ時間

これらのビットはソフトウェアから書き込まれて、同じバンクの次のトランザクションまでの現在の読出しまたは書込みトランザクションの終わりに遅延を追加します。

この遅延を使うと、連続トランザクション間の最小時間 (NEx ハイから NEx ローまでの t_{EHEL}) と、読出しアクセス後にメモリがデータ・バスを開放するために要する最大時間 (t_{EHQZ} 、チップ・イネーブル・ハイから出力ハイインピーダンスまで) を次のように一致させることができます。モード D およびマルチプレクスモードでは、この遅延が推奨されます。非マルチプレクスメモリでは、バスターンアラウンド遅延を最小値にセットできます。

$(BUSTURN + 1) \text{ HCLK 周期} \geq \max(t_{EHEL} \text{ min}, t_{EHQZ} \text{ max})$

FRAM メモリでは、バス・ターン・アラウンド遅延を最小 t_{PC} (プリチャージ時間) タイミングに合わせて設定する必要があります。バスターンアラウンド遅延は、 t_{PC} メモリタイミングと合わせて、同じバンクでの連続トランザクション間 (読出し／読出し、書込み／書込み、読出し／書込み、および書込み／読出し) に挿入されます。チップセレクトは、連続アクセス間でトグルします。

$(BUSTURN + 1) \text{ HCLK 周期} \geq t_{PC} \text{ min}$

0000 : BUSTURN フェーズ時間 = 1 HCLK クロックサイクル追加

.....

1111 : BUSTURN フェーズ時間 = 16 × HCLK クロックサイクル追加 (リセット後のデフォルト値)

ビット 15:8 **DATAST[7:0]** : データフェーズ時間

これらのビットは、非同期アクセスで使用するデータフェーズの時間を定義するためにソフトウェアにより書き込まれます (図 108 ~ 図 120 を参照)。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 × HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 × HCLK クロックサイクル

.....

1111 1111 : DATAST フェーズ時間 = 255 × HCLK クロックサイクル (リセット後のデフォルト値)

各メモリタイプおよびアクセスモードでのデータフェーズ時間については、それぞれの図 (図 108 ~ 図 120) を参照してください。

例 : モード 1、書込みアクセス、DATAST=1 : データフェーズ時間 = DATAST+1 = 2 HCLK クロックサイクル

注 : 同期アクセスでは、この値は無視されます。

ビット 7:4 **ADDHLD[3:0]** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアから書き込まれ (図 108 ~ 図 120 を参照)、モード D またはマルチプレクサアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 * HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 * HCLK クロックサイクル

.....

1111 : ADDHLD フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスホールドフェーズ時間については、それぞれの図 (図 108 ~ 図 120) を参照してください。

注 : 同期アクセスでは、この値は使用されず、アドレスホールドフェーズは、常に 1 メモリクロック周期の長さです。

ビット 3:0 **ADDSET[3:0]** : アドレスセットアップフェーズ時間

これらのビットは、アドレス・セットアップ・フェーズの時間を定義するために、ソフトウェアによって書き込まれ (図 108 ~ 図 120 を参照)、SRAM、ROM、非同期 NOR Flash、および PSRAM で使用されます。

0000 : ADDSET フェーズ時間 = 0 * HCLK クロックサイクル

.....

1111 : ADDSET フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスセットアップフェーズ時間については、それぞれの図 (図 108 ~ 図 120) を参照してください。

注 : 同期アクセスでは、この値は無視されます。

マルチプレクスモードとモード D では、ADDSET の最小値は 1 です。

モード 1 と PSRAM メモリでは、ADDSET の最小値は 1 です。

注 : PSRAM (CRAM) は内部リフレッシュがあるため遅延 (レイテンシ) が可変になっています。したがって、これらのメモリは必要に応じてレイテンシ期間を延長するために、レイテンシフェーズ全体に NWAIT 信号を発行します。

PSRAM (CRAM) では、フィールド DATLAT を 0 に設定しなければなりません。こうすると、FSMC は、すぐにレイテンシフェーズを終了して、メモリからの NWAIT 信号のサンプリングを開始し、メモリがレディ状態になると、読出しまたは書き込みを開始します。

この方法は、古い世代の Flash メモリと異なり、NWAIT 信号を発行する最新世代の同期 Flash メモリでも使用できます (使用中の特定の Flash メモリのデータシートを参照してください)。

SRAM/NOR 型 Flash の書き込みタイミング・レジスタ x (FMC_BWTRx)

アドレスオフセット : $0x104 + 0x8 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0FFF FFFF

このレジスタには各メモリバンクの制御情報が格納されます。SRAM、PSRAM、NOR Flash メモリに使用されます。FMC_BCRx レジスタの EXTMOD ビットがセットされているときには、このレジスタは書き込みアクセスについてアクティブになります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]			
rw	rw	rw	rw									rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 **DATAHLD[1:0]** : データホールドフェーズ時間

これらのビットは、データホールドフェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ（図 108 ~ 図 120 を参照）、非同期書き込みアクセスで使用されます。

00 : DATAHLD フェーズ時間 = 1 * HCLK クロックサイクル（デフォルト）

01 : DATAHLD フェーズ時間 = 2 * HCLK クロックサイクル

10 : DATAHLD フェーズ時間 = 3 * HCLK クロックサイクル

11 : DATAHLD フェーズ時間 = 4 * HCLK クロックサイクル

ビット 29:28 **ACCMOD[1:0]** : アクセスモード。

次のタイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC_BCRx レジスタの EXTMOD ビットが 1 にセットされている場合のみ考慮されます。

00 : アクセスモード A

01 : アクセスモード B

10 : アクセスモード C

11 : アクセスモード D

ビット 27:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **BUSTURN[3:0]** : パスターンアラウンドフェーズ時間

これらのビットはソフトウェアから書き込まれて、同じバンクの次のトランザクションまでの現在の書き込みトランザクションの終わりに遅延を追加します。

FRAM メモリでは、パス・ターン・アラウンド遅延を最小 t_{PC} （プリチャージ時間）タイミングに合わせて設定する必要があります。パスターンアラウンド遅延は、同じバンクでの連続トランザクション間（読出し／読出し、書き込み／書き込み、読出し／書き込み、および書き込み／読出し）に挿入されます。チップセレクトは、連続アクセス間でトグルします。

(BUSTURN + 1) HCLK 周期 $\geq t_{PC}$ min

0000 : BUSTURN フェーズ時間 = 1 HCLK クロックサイクル追加

.....

1111 : BUSTURN フェーズ時間 = 16 * HCLK クロックサイクル追加（リセット後のデフォルト値）

ビット 15:8 **DATAST[7:0]** : データフェーズ時間

これらのビットは、非同期 SRAM、PSRAM、NOR Flash メモリ・アクセスで使用するデータ・フェーズの時間を定義するために、ソフトウェアによって書き込まれます（図 108 ~ 図 120 を参照）。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 * HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 * HCLK クロックサイクル

.....

1111 1111 : DATAST フェーズ時間 = 255 * HCLK クロックサイクル（リセット後のデフォルト値）

ビット 7:4 **ADDHLD[3:0]** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアによって書き込まれ (図 117 ~ 図 120 を参照)、非同期マルチプレクスアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 * HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 * HCLK クロックサイクル

.....

1111 : ADDHLD フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

注 : 同期 NOR 型 Flash アクセスでは、この値は使用されず、アドレス・ホールド・フェーズは常に 1 Flash クロック周期の長さです。

ビット 3:0 **ADDSET[3:0]** : アドレスセットアップフェーズ時間。

これらのビットは、アドレスセットアップフェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ (図 108 ~ 図 120 を参照)、非同期アクセスで使用されます。

0000 : ADDSET フェーズ時間 = 0 * HCLK クロックサイクル

.....

1111 : ADDSET フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

注 : 同期アクセスでは、この値は使用されず、アドレス・セットアップ・フェーズは常に 1 Flash クロック周期の長さです。マルチプレクスモードでは、ADDSET の最小値は 1 です。

PSRAM チップセレクトカウンタレジスタ (FMC_PCSCNTR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

このレジスタは、同期および非同期モードの PSRAM チップセレクトカウンタの値を格納します。チップセレクトカウンタはすべてのバンクに共通で、バンクごとに個別に有効にできます。PSRAM の読みまたは書き込みアクセス時に、この値はタイマにロードされ、タイマは NE 信号がローに保持されている間、デクリメントされます。タイマが 0 に達すると、PSRAM コントローラは現在のアクセスを分割し、NE をトグルして PSRAM デバイスのリフレッシュを可能にし、そして新しいアクセスを再開します。プログラムされたカウンタ値によって、PSRAM デバイスに指定された最大 NE パルス幅 (t_{CEM}) が保証されます。カウンタは、NE のハイからローへの遷移によって新しいアクセスが開始されるたびに、リロードされてデクリメントが開始されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTB4EN	CNTB3EN	CNTB2EN	CNTB1EN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSCOUNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CNTB4EN** : カウンタバンク 4 有効化

このビットは、PSRAM/NOR バンク 4 に対するチップセレクトカウンタを有効にします。

0 : バンク 4 に対するカウンタは無効です。

1 : バンク 4 に対するカウンタは有効です。

ビット 18 **CNTB3EN** : カウンタバンク 3 有効化

このビットは、PSRAM/NOR バンク 3 に対するチップセレクトカウンタを有効にします。

0 : バンク 3 に対するカウンタは無効です。

1 : バンク 3 に対するカウンタは有効です。

ビット 17 **CNTB2EN** : カウンタバンク 2 有効化

このビットは、PSRAM/NOR バンク 2 に対するチップセレクトカウンタを有効にします。

0 : バンク 2 に対するカウンタは無効です。

1 : バンク 2 に対するカウンタは有効です。

ビット 16 **CNTB1EN** : カウンタバンク 1 有効化

このビットは、PSRAM/NOR バンク 1 に対するチップセレクトカウンタを有効にします。

0 : バンク 1 に対するカウンタは無効です。

1 : バンク 1 に対するカウンタは有効です。

ビット 15:0 **CSCOUNT[15:0]** : チップセレクトカウンタ

このビットフィールドはチップがローを選択する期間の最大値を定義するために使用されます。これは次の式によって求められます。

$CSCOUNT[15:0] \times T_{AHB}$ 、ここで T_{AHB} は AHB クロック周期です。

リフレッシュを考慮し、PSRAM チップは $t_{CEM} \sim 4 \mu s$ を超えてローを選択してはなりません。

CSCOUNT[15:0] は非同期および同期モードの両方に適用されます。

CSCOUNT[15:0] = 0x0000 の場合、この機能は無効です。

22.7 NAND 型 Flash コントローラ

FMC は、以下のタイプのデバイスを駆動するのに適した信号タイミングを発生します。

- 8 ビットと 16 ビットの NAND Flash メモリ

NAND バンクは、専用レジスタによって設定されます ([セクション 22.7.7](#) を参照)。プログラム可能なメモリパラメータは、アクセスタイミング ([表 196](#) を参照) と ECC 構成を含みます。

表 196. プログラム可能な NAND Flash のアクセス・パラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
メモリ セットアップ時間	コマンドアサーション前に アドレスをセットアップする ために必要なクロックサイクル (HCLK) 数	読出し／書込み	AHB クロック サイクル (HCLK)	1	255
メモリウェイト	コマンドアサーションの最小時間 (HCLK クロックサイクル数)	読出し／書込み	AHB クロック サイクル (HCLK)	2	255
メモリホールド	コマンドアサーション解除後に、 アドレスを (書込みアクセス時は データも) 保持する必要がある クロックサイクル数 (HCLK)	読出し／書込み	AHB クロック サイクル (HCLK)	1	254
メモリデータバス ハイ インピーダンス	書込みアクセス開始後に データバスがハイインピーダンス 状態に保たれるクロックサイクル 数 (HCLK)	書込み	AHB クロック サイクル (HCLK)	1	255

22.7.1 外部メモリインタフェース信号

次の表に、通常、NAND 型 Flash メモリとのインタフェースに使用される信号を示します。

注： 接頭辞「N」は、アクティブラーの信号を表します。

8 ビット NAND Flash メモリ

表 197. 8 ビット NAND 型 Flash

FMC 信号名	I/O	機能
A[17]	O	NAND 型 Flash アドレス・ラッチ・イネーブル (ALE) 信号
A[16]	O	NAND 型 Flash コマンド・ラッチ・イネーブル (CLE) 信号
D[7:0]	I/O	8 ビットマルチプレクス、双方向アドレス/データバス
NCE	O	チップセレクト
NOE(= NRE)	O	出力イネーブル (メモリ信号名：読出しイネーブル、NRE)
NWE	O	書込みイネーブル
NWAIT/INT	I	FMC への NAND 型 Flash レディ / ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

16 ビット NAND Flash メモリ

表 198. 16 ビット NAND 型 Flash

FMC 信号名	I/O	機能
A[17]	O	NAND 型 Flash アドレス・ラッチ・イネーブル (ALE) 信号
A[16]	O	NAND 型 Flash コマンド・ラッチ・イネーブル (CLE) 信号
D[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス
NCE	O	チップセレクト
NOE(= NRE)	O	出力イネーブル (メモリ信号名：読出しイネーブル、NRE)
NWE	O	書込みイネーブル
NWAIT/INT	I	FMC への NAND 型 Flash レディ / ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

22.7.2 NAND 型 Flash によってサポートされるメモリとトランザクション

表 199 に、サポートされるデバイス、アクセスモード、およびトランザクションを示します。NAND Flash コントローラによって許可されない（またはサポートされない）トランザクションは、グレイで示されています。

表 199. サポートされるメモリおよびトランザクション

デバイス	モード	読出し/ 書込み	AHB データ サイズ	メモリ データ サイズ	許可/禁止	コメント
NAND 8 ビット	非同期	R	8	8	Y	-
	非同期	W	8	8	Y	-
	非同期	R	16	8	Y	2 つの FMC アクセスに分割
	非同期	W	16	8	Y	2 つの FMC アクセスに分割
	非同期	R	32	8	Y	4 つの FMC アクセスに分割
	非同期	W	32	8	Y	4 つの FMC アクセスに分割
NAND 16 ビット	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割

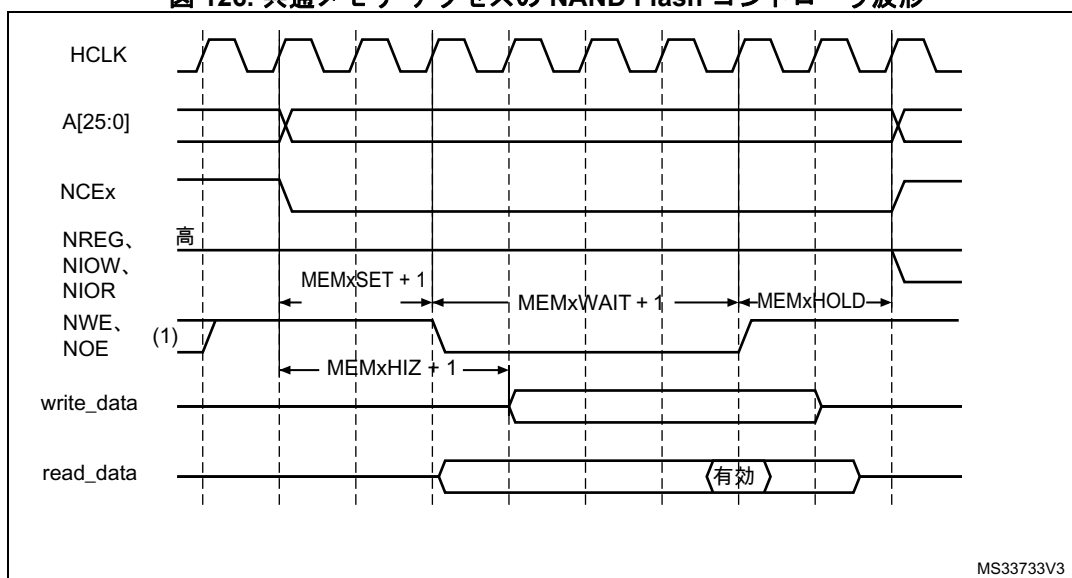
22.7.3 NAND Flash メモリのタイミング図

NAND 型 Flash メモリのバンクは、以下のレジスタを通じて管理されます。

- 制御レジスタ : FMC_PCR
- 割込みステータスレジスタ : FMC_SR
- ECC レジスタ : FMC_ECCR
- 共通メモリ空間用タイミングレジスタ : FMC_PMEM
- 属性メモリ空間用タイミングレジスタ : FMC_PATT

各タイミング設定レジスタは、NAND Flash アクセスの 3 つのフェーズの HCLK サイクル数を定義するための 3 つのパラメータと、書込みアクセス時にデータ・バスの駆動を開始するタイミングを定義する 1 つのパラメータを格納します。図 126 に共通メモリアクセス用のタイミングパラメータの定義を示します。属性メモリ空間のアクセスタイミングが同様であることがわかります。

図 126. 共通メモリ・アクセスの NAND Flash コントローラ波形



1. 書き込みアクセス時、NOE はハイ（非アクティブ）に保たれます。読出しアクセス時、NWE はハイ（非アクティブ）に保たれます。
2. 書き込みアクセスの場合、ホールド・フェーズ遅延は (MEMHOLD) HCLK サイクルとなり、読出しアクセスの場合は (MEMHOLD + 2) HCLK サイクルとなります。

22.7.4 NAND Flash 動作

NAND Flash デバイスのコマンド・ラッチ・イネーブル (CLE) およびアドレス・ラッチ・イネーブル (ALE) 信号は、FMC コントローラのアドレス信号によって駆動されます。これは、NAND Flash メモリにコマンドまたはアドレスを送信するには、CPU は、CPU のメモリ空間内の特定のアドレスに書き込みを行う必要があることを意味します。

NAND Flash デバイスからの一般的なページ読出し動作には、次のステップが必要です。

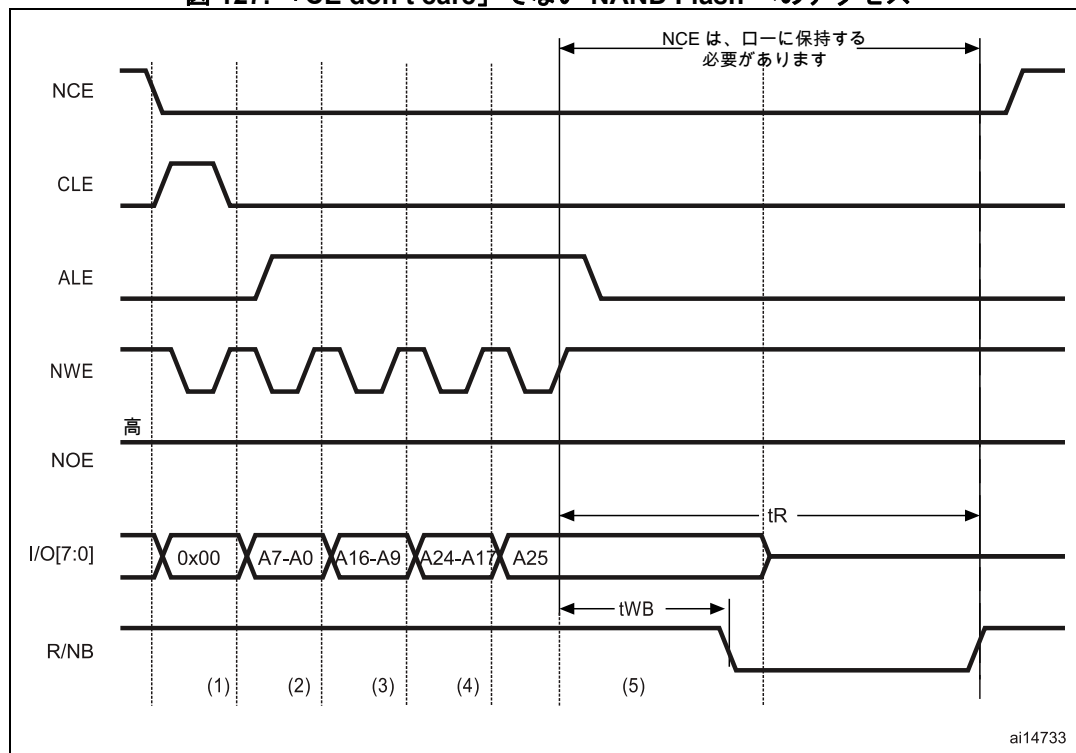
1. FMC_PCR レジスタと FMC_PMEM レジスタ（デバイスによっては、FMC_PATT も。[セクション 22.7.5 : NAND Flash のプリウェイト機能](#)を参照）を、NAND Flash メモリの特性（NAND Flash のデータ・バス幅に対する PWID ビット、PTYP = 1、必要に応じて PWAITEN = 0 または 1、タイミング設定については[セクション 22.5.2 : NAND Flash メモリ・アドレス・マッピング](#)を参照）に従って設定することで、対応するメモリ・バンクのプログラムおよび有効化を行います。
2. CPU は、1 つの Flash コマンド・バイトと等しいデータ・バイト（たとえば、Samsung の NAND Flash デバイスの場合は 0x00）で、共通メモリ空間にバイト書き込みを行います。NAND Flash メモリの LE 入力は、書き込みストローブ（NWE 上のローパルス）の間はアクティブであり、書き込まれたバイトは、NAND Flash メモリによってコマンドとして認識されます。コマンドがメモリデバイスにラッチされたら、次のページ読出し動作のために書き込む必要はありません。
3. CPU は、4 バイト（より小容量のデバイスの場合は 3 バイト）を共通メモリまたは属性空間に書き込むことによって、読出し操作の開始アドレス (STARTAD) を送信できます (STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17]、および 64 M ビット x 8 ビット NAND Flash メモリの場合は最後に STARTAD[25])。NAND 型 Flash デバイスの ALE 入力は、書き込みストローブの間アクティブであり（NWE 上のローパルス）、書き込まれたバイトは、読出し動作の開始アドレスとして解釈されます。属性メモリ空間を使用すると、FMC の異なるタイミング設定を使用でき、一部の NAND Flash メモリで必要とされるプリウェイト機能を実装できます（詳細については、[セクション 22.7.5 : NAND Flash のプリウェイト機能](#)を参照してください）。

4. コントローラは、NAND Flash メモリがレディ (R/NB 信号がハイ) になるのを待ってから、同じメモリ・バンクまたは別のメモリ・バンクへの新しいアクセスを開始します。待っている間、コントローラは NCE 信号をアクティブ (ロー) に保ちます。
5. CPU は、NAND Flash ページ (データ・フィールド+スペア・フィールド) をバイト単位で読み出すために、共通メモリ空間からバイト読み出し動作を行います。
6. 次の NAND Flash ページは、CPU コマンドやアドレス書き込み動作なしに読み出すことができます。これは次の 3 つの方法で行うことができます。
 - ステップ 5 に記述された動作を実行します。
 - ステップ 3 から動作をリスタートすることによって、新しいランダムアドレスにアクセスできます。
 - ステップ 2 からリスタートすることによって、NAND 型 Flash デバイスに新しいコマンドを送信できます。

22.7.5 NAND Flash のプリウエイト機能

一部の NAND Flash デバイスでは、アドレスの最後の部分を書き込んだ後、コントローラは R/NB 信号を待つ必要があります。(図 127 を参照。)

図 127. 「CE don't care」でない NAND Flash へのアクセス



1. CPU はアドレス 0x7001 0000 にバイト 0x00 を書き込みました。
2. CPU はアドレス 0x7002 0000 にバイト A7~A0 を書き込みました。
3. CPU はアドレス 0x7002 0000 にバイト A16~A9 を書き込みました。
4. CPU はアドレス 0x7002 0000 にバイト A24~A17 を書き込みました。
5. CPU はアドレス 0x7802 0000 にバイト A25 を書き込みました : FMC は FMC_PATT のタイミング定義を使用して書き込みアクセスを実行します。ここで $ATTHOLD \geq 7$ (ただし、 $(7+1) * HCLK = 112 \text{ ns} > t_{WB \text{ max}}$)。これにより、NCEは R/NB がローから再びハイになるまでローを維持します (NCE が無視されない NAND 型 Flash メモリの場合のみ必要)。

この機能が必要な場合、MEMHOLD 値をプログラムすることにより t_{WB} タイミングを確実に満たすことができます。ただし、NAND Flash メモリに対するすべての CPU 読出しアクセスでは、(MEMHOLD+ 2) HCLK サイクル、CPU 書込みアクセスでは、(MEMHOLD) HCLK サイクルのホールド遅延が NWE 信号の立上がりエッジと次のアクセスとの間に挿入されます。

このタイミング制約に対処するためには、 t_{WB} タイミングに応じた ATTHOLD 値でタイミングレジスタをプログラムし、MEMHOLD 値を最小値のままにしておくことで、属性メモリ空間を使用できます。この場合、CPU は NAND Flash のすべての読出しおよび書込みアクセスには共通メモリ空間を使用する必要があります。ただし、NAND Flash デバイスに最後のアドレス・バイトを書き込むときには、属性メモリ空間に書き込む必要があります。

22.7.6 NAND 型 Flash メモリ内のエラー訂正コード (ECC) の計算

FMC NAND カードコントローラには、メモリバンクごとに 1 つずつ、2 つのエラー訂正コード計算ハードウェアブロックがあります。これらのハードウェアブロックは、ソフトウェアで ECC を処理する場合、ホスト CPU の負荷を削減します。

この 2 つの ECC ブロックは同一であり、それぞれバンク 2 とバンク 3 に接続されます。結果として、バンク 4 に接続されたメモリでは、ハードウェア ECC 計算を使用できません。

FMC に実装されたエラー訂正コード (ECC) のアルゴリズムでは、NAND Flash メモリに対する 256、512、1024、2048、4096、または 8192 バイトの読出しまたは書込みごとに、1 ビットのエラー訂正と 2 ビットのエラー検出を実行できます。ハミングコーディングアルゴリズムを採用し、行と列のパリティ計算を行います。

ECC モジュールは、NAND Flash メモリバンクがアクティブになるたびに、NAND Flash データバスおよび読出し／書込み信号 (NCE と NWE) を監視します。

ECC は次のように動作します。

- NAND Flash メモリのバンク 2 またはバンク 3 へアクセスするときは、D[15:0] バス上のデータがラッチされて、ECC 計算に使用されます。
- NAND Flash メモリの他のアドレスをアクセスするときは、ECC ロジックはアイドル状態となり、動作しません。このため、NAND Flash メモリに対するコマンドまたはアドレスを定義する書込み動作は、ECC 計算で考慮されません。

目的のバイト数がホスト CPU によって NAND Flash メモリから読み出されるか、NAND Flash メモリに書き込まれたら、FMC_ECCR レジスタを読み出して、計算値を取得する必要があります。読出し後は、ECCEN ビットを"0"にリセットすることによってクリアしなくてはなりません。新しいデータブロックを計算するには、FMC_PCR レジスタの ECCEN ビットを 1 にセットする必要があります。

ECC 計算を行うときは：

1. FMC_PCR レジスタで ECCEN ビットを有効にします。
2. NAND 型 Flash メモリ・ページヘータを書き込みます。NAND ページへ書き込む際、ECC ブロックは ECC 値を計算します。
3. FMC_ECCR レジスタにある ECC 値を読み出して、変数内に格納します。
4. ECCEN ビットをクリアし、FMC_PCR レジスタで有効にした後、書き込んだデータを NAND ページから読み出します。NAND ページを読み出す際、ECC ブロックは ECC 値を計算します。
5. FMC_ECCR レジスタにある新しい ECC 値を読み出します。
6. 2 つの ECC 値が同じ場合、訂正は不要です。その他の場合、ECC エラーが存在するためソフトウェア訂正ルーチンがエラーが訂正できるか否かの情報を返します。

22.7.7 NAND Flash コントローラ・レジスタ

NAND Flash 制御レジスタ (FMC_PCR)

アドレスオフセット : 0x80

リセット値 : 0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS[2:0]			TAR3
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR[2:0]			TCLR[3:0]				Res.	Res.	ECCEN	PWID[1:0]		PTYP	PBKEN	PWAITEN	Res.
rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 **ECCPS[2:0]** : ECC ページ・サイズ

外部 ECC のページサイズを定義します :

000 : 256 バイト

001 : 512 バイト

010 : 1024 バイト

011 : 2048 バイト

100 : 4096 バイト

101 : 8192 バイト

ビット 16:13 **TAR[3:0]** : ALE から RE 間の遅延

ALE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。

時間は、 $t_{ar} = (TAR + SET + 2) \times THCLK$ です。THCLK は HCLK クロック周期です。

0000 : 1 HCLK サイクル (デフォルト)

1111 : 16 HCLK サイクル

注 : SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 12:9 **TCLR[3:0]** : CLE から RE 間の遅延

CLE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。

時間は、 $t_{clr} = (TCLR + SET + 2) \times THCLK$ です。THCLK は HCLK クロック周期です。

0000 : 1 HCLK サイクル (デフォルト)

1111 : 16 HCLK サイクル

注 : SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **ECCEN** : ECC 計算ロジックイネーブルビット

0 : ECC ロジックは無効であり、リセットされています (リセット後のデフォルト)。

1 : ECC ロジックは有効です。

ビット 5:4 **PWID[1:0]** : データ・バス幅

外部メモリデバイスの幅を定義します。

00 : 8 ビット

01 : 16 ビット (リセット後のデフォルト)

10 : 予約済み

11 : 予約済み

ビット 3 **PTYP** : メモリタイプ

対応するメモリバンクに接続されたデバイスのタイプを定義します。

0 : 予約済みであり、リセット値に保持する必要があります。

1 : NAND 型 Flash (リセット後のデフォルト)

ビット 2 **PBKEN** : NAND Flash メモリ・バンク・イネーブル・ビット

メモリバンクを有効にします。無効なメモリバンクにアクセスすると、AHB バスでエラーが発生します。

0 : 対応するメモリバンクは無効です (リセット後のデフォルト)。

1 : 対応するメモリバンクは有効です。

ビット 1 **PWAITEN** : ウェイト機能イネーブル・ビット

NAND 型 Flash メモリ・バンクのウェイト機能を有効にします。

0 : 無効

1 : 有効

ビット 0 予約済みであり、リセット値に保持する必要があります。

FIFO ステータスおよび割込みレジスタ (FMC_SR)

アドレスオフセット : 0x84

リセット値 : 0x0000 0040

このレジスタは、FIFO ステータスと割込みに関する情報を格納します。FMC には、AHB の 16 ワードまでのデータをメモリに書き込んで格納するときに使用される FIFO があります。

これは、FMC が FIFO をメモリへ転送すると同時に、FIFO に迅速に書き込んで、FMC 以外のペリフェラルへのトランザクションに AHB を解放するために使用されます。これらのレジスタビットの 1 つは、ECC のために、FIFO のステータスを示します。

ECC は、データのメモリへの書き込み中に計算されます。正しい ECC を読み出すため、ソフトウェアは FIFO がエンプティになるまで待つ必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
									r	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FEMPT** : FIFO エンプティ

FIFO のステータスを表示する読取り専用ビット

0 : FIFO は空ではありません。

1 : FIFO エンプティ

ビット 5 **IFEN** : 割込み立下がりエッジ検出イネーブルビット

0 : 割込み立下がりエッジ検出リクエストは無効

1 : 割込み立下がりエッジ検出リクエストは有効

ビット 4 **ILEN** : 割込みハイレベル検出イネーブルビット

0 : 割込みハイレベル検出リクエストは無効

1 : 割込みハイレベル検出リクエストは有効

ビット 3 **IREN** : 割込み立上がりエッジ検出イネーブルビット

0 : 割込み立上がりエッジ検出リクエストは無効

1 : 割込み立上がりエッジ検出リクエストは有効

ビット 2 IFS : 割込み立下がリエッジステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

0 : 割込み立下がリエッジは発生しませんでした。

1 : 割込み立下がリエッジが発生しました。

注 : ソフトウェアがこのビットに 1 を書き込むとセットされます。

ビット 1 ILS : 割込みハイレベルステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

0 : 割込みハイレベルは発生しませんでした。

1 : 割込みハイレベルが発生しました。

ビット 0 IRS : 割込み立上がりエッジステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

0 : 割込み立上がりエッジは発生しませんでした。

1 : 割込み立上がりエッジが発生しました。

注 : ソフトウェアがこのビットに 1 を書き込むとセットされます。

共通メモリ空間タイミング・レジスタ (FMC_PMEM)

アドレスオフセット : アドレス : 0x88

リセット値 : 0xFCFC FCFC

FMC_PMEM 読出し／書込みレジスタは、NAND Flash メモリ・バンクのタイミング情報を格納します。この情報は、NAND Flash の共通メモリ空間に、コマンド、アドレスの書込みアクセス、およびデータ読出し／書込みアクセスするために使われます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMHIZ[7:0]								MEMHOLD[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMWAIT[7:0]								MEMSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 MEMHIZ[7:0] : 共通メモリ x データバスハイインピーダンス時間

ソケット上の共通メモリ空間への NAND Flash による書込みアクセスの開始後に、データ・バスをハイインピーダンスに保つ HCLK クロック・サイクル数を定義します。これは、書込み トランザクションに対してのみ有効です。

0000 0000 : 1 HCLK サイクル

1111 1110 : 255 HCLK サイクル

1111 1111 : 予約済み

ビット 23:16 MEMHOLD[7:0] : 共通メモリホールド時間

ソケット x 上の共通メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンドのデアサーション (NWE、NOE) 後にアドレスを (書込みアクセス時はデータも) 保持する書込みアクセス時の HCLK クロック・サイクル数と読出しアクセス時の HCLK (+2) クロック・サイクル数を定義します。

0000 0000 : 予約済み

0000 0001 : 書込みアクセスの場合 1 HCLK サイクル／読出しアクセスの場合 3 HCLK サイクル

1111 1110 : 書込みアクセスの場合 254 HCLK サイクル／読出しアクセスの場合 256 HCLK サイクル

1111 1111 : 予約済み

ビット 15:8 **MEMWAIT[7:0]** : 共通メモリウェイト時間

ソケット上の共通メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンド (NWE、NOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されます。

0000 0000 : 予約済み

0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)

1111 1110 : 255 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)

1111 1111 : 予約済み

ビット 7:0 **MEMSET[7:0]** : 共通メモリ x セットアップ時間

ソケット x 上の共通メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンド・アサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロック・サイクル数を定義します。

0000 0000 : 1 HCLK サイクル

1111 1110 : 255 HCLK サイクル

1111 1111 : 予約済み

属性メモリ空間タイミング・レジスタ (FMC_PATT)

アドレスオフセット : 0x8C

リセット値 : 0xFCFC FCFC

FMC_PATT 読出し／書込みレジスタは、NAND Flash メモリバンクのタイミング情報を格納します。この情報は、最後のアドレス書込みアクセスの際に、タイミングが以前のアクセスのタイミングと異なる必要がある場合に、NAND Flash の属性メモリ空間へ 8 ビット・アクセスするために使用されます (レディ／ビジー管理については、[セクション 22.7.5 : NAND Flash のプリウェイト機能](#)を参照)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZ[7:0]								ATTHOLD[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTWAIT[7:0]								ATTSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **ATTHIZ[7:0]** : 属性メモリデータバスハイインピーダンス時間

ソケット上の属性メモリ空間への NAND Flash による書込みアクセスの開始後に、データ・バスをハイインピーダンスに保つ HCLK クロック・サイクル数を定義します。これは、書込みトランザクションに限り有効です。

0000 0000 : 0 HCLK サイクル

1111 1110 : 255 HCLK サイクル

1111 1111 : 予約済み

ビット 23:16 **ATTHOLD[7:0]** : 属性メモリホールド時間

ソケット上の属性メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンドのデアサーション (NWE、NOE) 後にアドレスを (書込みアクセス時はデータも) 保持する書込みアクセス時の HCLK クロック・サイクル数と読出しアクセス時の HCLK (+2) クロック・サイクル数を定義します。

0000 0000 : 予約済み

0000 0001 : 書込みアクセスの場合 1 HCLK サイクル／読出しアクセスの場合 3 HCLK サイクル

1111 1110 : 書込みしアクセスの場合 254 HCLK サイクル／読出しアクセスの場合 256 HCLK サイクル

1111 1111 : 予約済み

ビット 15:8 **ATTWAIT[7:0]** : 属性メモリウェイト時間

ソケット x 上の属性メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンド (NWE、NOE) をアサートするための HCLK (+1) クロック・サイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されます。

0000 0000 : 予約済み

0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)

1111 1110 : 255 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)

1111 1111 : 予約済み

ビット 7:0 **ATTSET[7:0]** : 属性メモリセットアップ時間

ソケット上の属性メモリ空間への NAND Flash による読出しまたは書込みアクセスの際に、コマンド・アサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロック・サイクル数を定義します。

0000 0000 : 1 HCLK サイクル

1111 1110 : 255 HCLK サイクル

1111 1111 : 予約済み

ECC 結果レジスタ (FMC_ECCR)

アドレスオフセット : 0x94

リセット値 : 0x0000 0000

このレジスタは、FMC NAND コントローラの ECC 計算モジュールによって計算された現在のエラー訂正コード値を格納します。CPU が、NAND Flash メモリ・ページ上の正しいアドレスからデータを読み出すときには ([セクション 22.7.6 : NAND 型 Flash メモリ内のエラー訂正コード \(ECC\) の計算](#)を参照)、NAND Flash メモリに対して読み書きされたデータは、ECC 計算モジュールによって自動的に処理されます。X バイトを読み出した後 (FMC_PCR レジスタの ECCPS フィールドに従って)、CPU は FMC_ECC レジスタから計算された ECC 値を読み出す必要があります。次に、これらの計算されたパリティデータがスピア領域に記録されているパリティ値と一致するか検証して、ページが有効か否か、訂正が必要か否かを確認します。FMC_ECCR レジスタは、読出し後、ECCEN ビットを 0 に設定することによってクリアする必要があります。新しいデータ・ブロックを計算するときは、ECCEN ビットを 1 にセットする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **ECC[31:0]** : ECC 結果

このフィールドは、ECC 計算ロジックによって計算された値を格納します。[表 200](#) に、これらのビットフィールドの値を示します。

表 200. ECC 結果関連ビット

ECCPS[2:0]	ページサイズ (バイト数)	ECC ビット
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]

表 200. ECC 結果関連ビット (続き)

ECCPS[2:0]	ページサイズ (バイト数)	ECC ビット
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

22.8 SDRAM コントローラ

22.8.1 SDRAM コントローラの主な機能

SDRAM コントローラの主な機能は次の通りです。

- 独立した設定を持つ 2 つの SDRAM バンク
- 8 ビット、16 ビットのデータ・バス幅
- 13 ビットのアドレス行、11 ビットのアドレス列、4 つの内部バンク : 4x16Mx16 ビット (128 MB)、4x16Mx8 ビット (64 MB)
- ワード、ハーフワード、バイトアクセス
- Rowおよびバンクの自動境界管理
- マルチバンクピンポンアクセス
- プログラム可能なタイミングパラメータ
- プログラム可能なリフレッシュレートでのオートリフレッシュ動作
- セルフリフレッシュモード
- パワーダウンモード
- ソフトウェアによる SDRAM パワーアップ初期化
- CAS 遅延 = 1、2、3
- 深さ 6 ライン * 32 ビット (6x14 ビットアドレスタグ) のキャッシュ可能読出し FIFO

22.8.2 SDRAM 外部メモリアンタフェース信号

FMC SDRAM コントローラと外部SDRAM デバイスとのインタフェースに使用される SDRAM I/O ピンは、起動時にユーザアプリケーションから設定する必要があります。アプリケーションで使用しない SDRAM コントローラ I/O ピンは、他の用途に使用することができます。

表 201. SDRAM 信号

SDRAM 信号	I/O タイプ	説明	オルタネート機能
SDCLK	O	SDRAM クロック	-
SDCKE[1:0]	O	SDCKE0 : SDRAM バンク 1 クロック有効化 SDCKE1 : SDRAM バンク 2 クロック有効化	-
SDNE[1:0]	O	SDNE0 : SDRAM バンク 1 チップ有効化 SDNE1 : SDRAM バンク 2 チップ有効化	-
A[12:0]	O	アドレス	FMC_A[12:0]
D[15:0]	I/O	双方向データバス	FMC_D[15:0]
BA[1:0]	O	バンクアドレス	FMC_A[15:14]
NRAS	O	Rowアドレスストローブ	-

表 201. SDRAM 信号 (続き)

NCAS	O	Columnアドレスストロープ	-
SDNWE	O	書込みイネーブル	-
NBL[1:0]	O	書込みアクセス用の出力バイトマスク (メモリ信号名: DQM[1:0])	FMC_NBL[1:0]

22.8.3 SDRAM コントローラの機能説明

すべての SDRAM コントローラ出力 (信号、アドレス、データ) は、メモリクロック (FMC_SDCCLK) の立下がりエッジで変化します。

SDRAM 初期化

初期化シーケンスはソフトウェアから管理されます。2 つのバンクを使用する場合、FMC_SDCMR レジスタの対象バンクビット CTB1 と CTB2 をセットして、初期化シーケンスをバンク 1 とバンク 2 に対して同時に発生させる必要があります。

1. FMC_SDCRx レジスタにメモリデバイス機能を設定します。SDRAM クロック周波数、RBURST、RPIPEは、FMC_SDCR1 レジスタに設定する必要があります。
2. メモリデバイスタイミングは FMC_SDTRx レジスタに設定します。TRP タイミングと TRC タイミングは、FMC_SDTR1 レジスタに設定します。
3. MODE ビットに"001"を書き込み、FMC_SDCMR レジスタの対象バンクビット (CTB1 および／または CTB2) を設定して、メモリに対するクロックの供給を開始させます (SDCKE はハイに駆動されます)。
4. 所定の遅延時間を待ちます。代表的遅延時間は約 100 μ s です (パワーアップ後に必要とされる遅延については、SDRAM データシート参照)。
5. MODE ビットに"010"を書き込み、FMC_SDCMR レジスタの対象バンクビット (CTB1 および／または CTB2) を設定して、「Precharge All」コマンドを発行させます。
6. MODE ビットに"011"を書き込み、FMC_SDCMR レジスタの対象バンクビット (CTB1 および／または CTB2) 、さらに連続オートリフレッシュコマンド数 (NRFS) を設定します。発行が必要なオートリフレッシュコマンド数については SDRAM データシートを参照してください。代表値は 8 です。
7. SDRAM デバイスに応じて MRD フィールドを設定し、MODE ビットに "100" を書き込み、FMC_SDCMR レジスタの対象バンクビット (CTB1 および／または CTB2) を設定して、「Load Mode Register」コマンドを発行させ、SDRAM デバイスに書き込みを行います。
特に、

- a) CAS 遅延は、FMC_SDCR1/2 レジスタの設定値に従って選択する必要があります。
- b) バースト長 (BL) の 1 は、モードレジスタの M[2:0] ビットに"000"を設定することによって選択する必要があります。SDRAM デバイスのデータシートを参照してください。

モードレジスタが両 SDRAM バンクに対して一致しない場合、このステップを各バンクに 1 回ずつ 2 回繰り返し、目標バンクビットをセットする必要があります。

8. リフレッシュレートを FMC_SDRTR レジスタに設定します。
リフレッシュレートは、リフレッシュサイクル間の遅延に対応します。この値を SDRAM デバイスに対して使用する必要があります。
9. モバイル SDRAM デバイスの場合、拡張モードレジスタのプログラムは SDRAM デバイスが初期化されてから行ってください。まず、BA1 = 1 かつ BA = 0 (BA[1:0] アドレスマッピングについては、SDRAM アドレスマッピングのセクションを参照) であるときに、ダミーの読出しアクセスを 1 回行う必要があります。これは、ロードモードレジスタではなく拡張モードレジスタを選択して、必要な値を書き込むためです。

この段階で、SDRAM デバイスはコマンドを受け付ける準備が整います。SDRAM アクセスの実行中にシステムリセットが発生した場合、データバスはそのまま SDRAM デバイスによって駆動される可能性があります。従って、SDRAM デバイスはリセット後、NOR Flash、PSRAM、SRAM、NAND Flash コントローラのいずれかで新しいアクセスを発行する前に、再初期化をする必要があります。

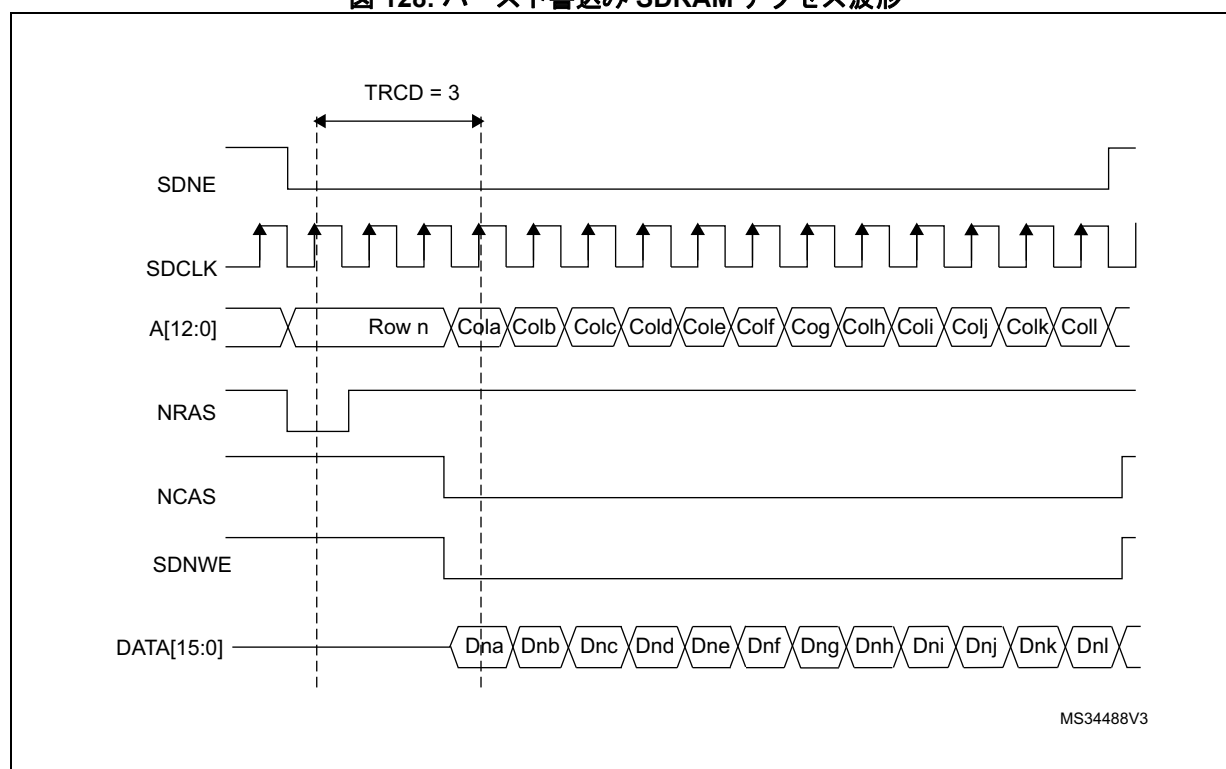
注： 2 つの SDRAM デバイスを FMC に接続する場合は、コマンドモードレジスタにより両デバイスに同時に実行されるすべてのアクセス（ロードモードレジスタコマンド）は、FMC_SDTR1 レジスタで SDRAM バンク 1 に対して設定されたタイミングパラメータ（TMRD タイミング、TRAS タイミング）を使って発行されます。

SDRAM コントローラ書き込みサイクル

SDRAM コントローラはシングルおよびバースト書き込みリクエストを受け付けて、これらをシングルメモリアccessに変換します。両ケースとも、SDRAM コントローラは各バンクのアクティブ行を記録して、異なるバンクへ連続書き込みアクセスを行えるようにします（マルチバンクピンポンアクセス）。

書き込みアクセスを行う前に、FMC_SDCRx レジスタの WP ビットをクリアして、SDRAM バンク書き込み保護機能を無効化しておく必要があります。

図 128. バースト書き込み SDRAM アクセス波形



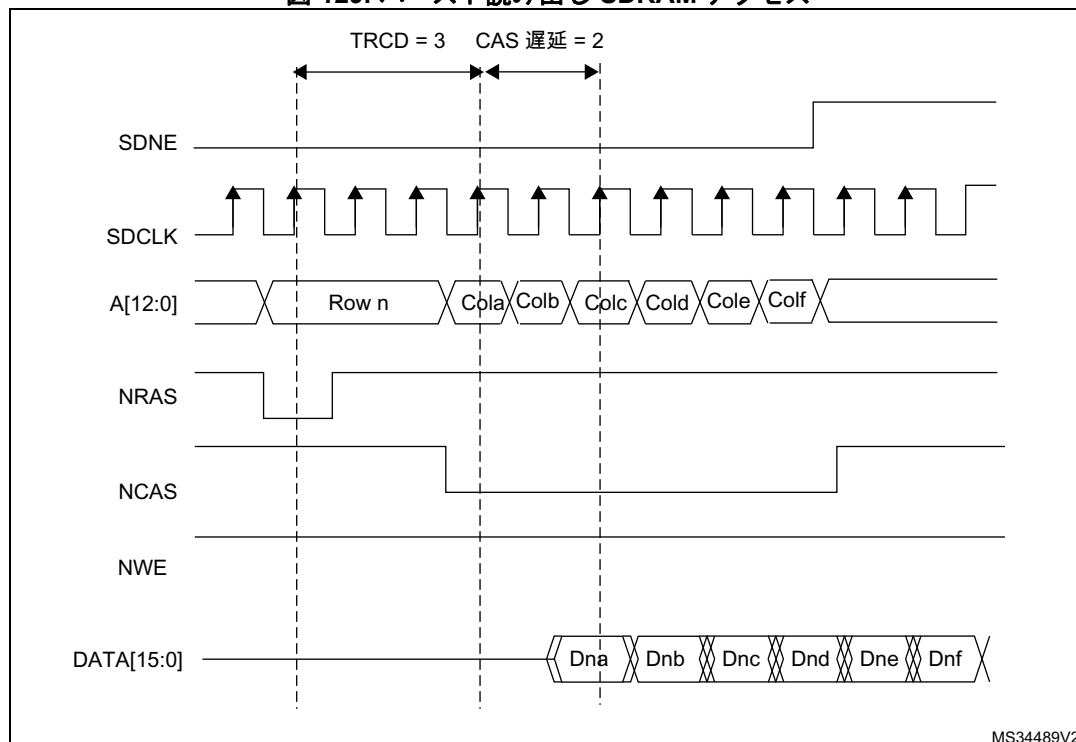
SDRAM コントローラは常に次のアクセスをチェックしています。

- 次のアクセスが同じRowまたは別のアクティブなRowである場合、書き込み動作が実行されます。
- 次のアクセスが別のRow（非アクティブ）を対象とする場合、SDRAM コントローラはプリチャージコマンドを発生し、新しいRowをアクティブにして、書き込みコマンドを開始します。

SDRAM コントローラ読み出しサイクル

SDRAM コントローラはシングルおよびバースト読み出しリクエストを受け付けて、これらをシングルメモリアクセスに変換します。両ケースとも、SDRAM コントローラは各バンクのアクティブ行を記録して、異なるバンクへ連続読み出しアクセスを行えるようにします（マルチバンクピンポンアクセス）。

図 129. バースト読み出し SDRAM アクセス



MS34489V2

FMC SDRAM コントローラは、キャッシュ可能読み出し FIFO（6 ライン x 32 ビット）を内蔵しています。これは、CAS 遅延中および RPIPE 遅延中に、予め読み出しておいたデータを保存するために、次の式に従って使用されます。次の読み出しアクセスを予測するときは、FMC_SDCR1 レジスタの RBURST ビットをセットする必要があります。

予測データの番号 = CAS 遅延 + 1 + (RPIPE 遅延)/2

例：

- CAS 遅延 = 3、RPIPE 遅延 = 0 の場合：4 つのデータ（使途未定）が FIFO に保存されます。
- CAS 遅延 = 3、RPIPE 遅延 = 2 の場合：5 つのデータ（使途未定）が FIFO に保存されます。

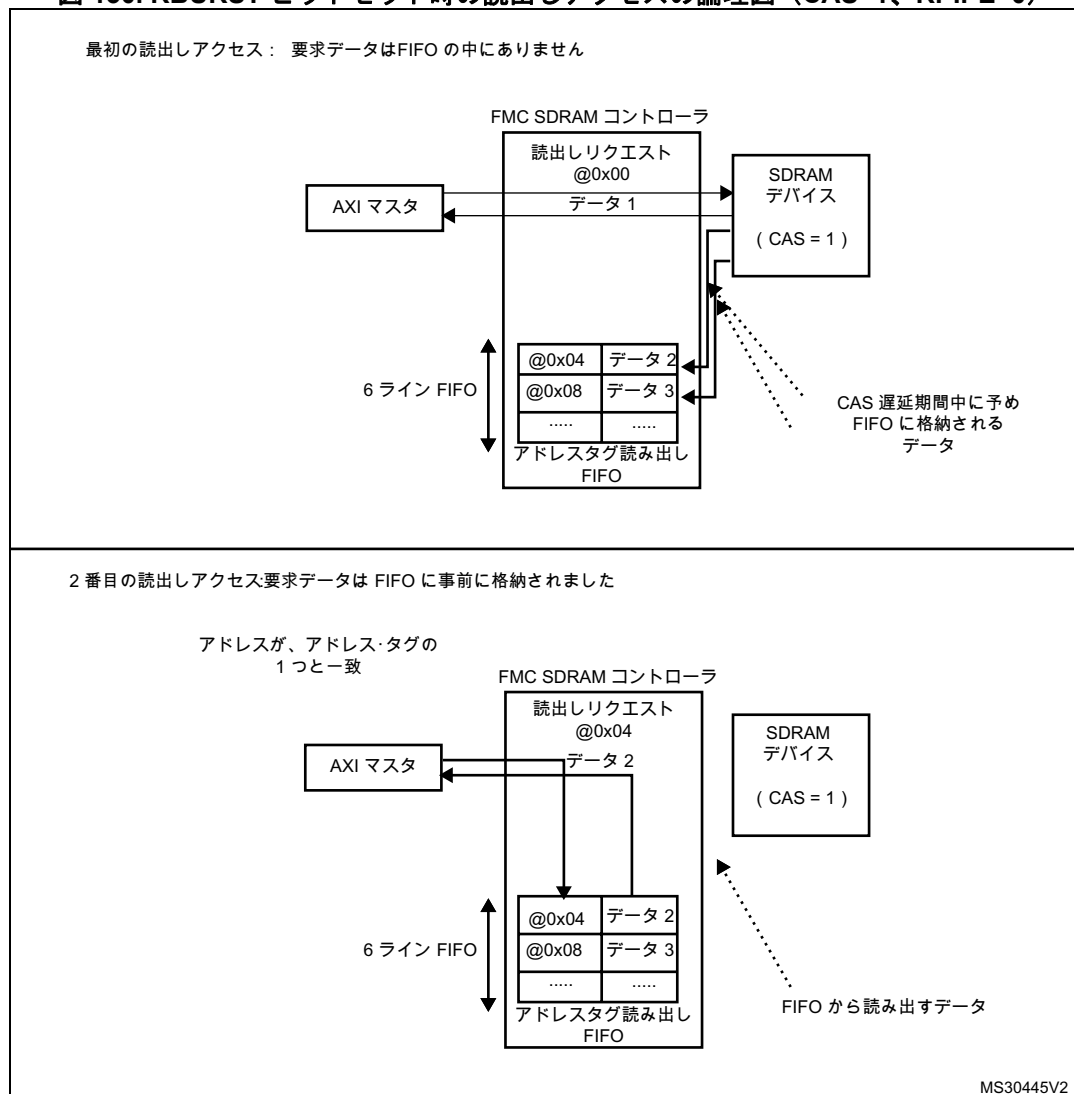
読み出し FIFO は、各ラインに対して値を識別する 14 ビットのアдресタグを持っています。11 ビットは Column アドレスに、2 ビットは内部バンクとアクティブ Row の選択に、1 ビットは SDRAM デバイスの選択に、それぞれ使用します。

AHB バースト読み出し時に予め行の終わりに到達している場合、予め読み出されたデータ（使途未定）は読み出し FIFO に保存されません。シングル読み出しアクセスの場合、データは正常に FIFO に保存されます。

読出しリクエストが発生するたびに、SDRAM コントローラは次をチェックします。

- アドレスがアドレスタグの 1 つに一致すると、データが直接 FIFO から読み出され、対応するアドレスタグ／ライン値がクリアされ、FIFO 内の残りのデータはエンプティラインを回避するためコンパクト化されます。
- その他の場合、新しい読み出しコマンドがメモリに対して発行され、FIFO は新しいデータで更新されます。FIFO がフルの場合、古いデータは失われます。

図 130. RBURST ビットセット時の読出しアクセスの論理図 (CAS=1、RPIPE=0)



書き込みアクセス時またはプリチャージコマンド時、読出し FIFO が一掃されて、新しいデータ書き込みの準備が整います。

最初の読出しリクエストの後、現在のアクセスが行境界に対して実行されなかった場合、SDRAM コントローラは、CAS 遅延中および RPIPE 遅延中（設定した場合）に次の読出しアクセスがあるものと見なします。これは、メモリアドレスをインクリメントして行います。次の条件を満たす必要があります。

- FMC SDCR1 レジスタの RBURST 制御ビットを"1"にセットする必要があります。

アドレスは以下に示す AHB リクエストに応じて管理されます。

- 次の AHB リクエストはシーケンシャル (AHB パースト) である。
この場合、SDRAM コントローラはアドレスをインクリメントさせます。
- 次の AHB リクエストはシーケンシャルでない。
 - 新しい読出しリクエストが同じ行または別のアクティブな行を対象とする場合、新しいアドレスがメモリへ渡され、マスタは CAS 遅延の間停止され、メモリからの新しいデータを待ちます。
 - 新しい読出しリクエストがアクティブな行を対象としない場合、SDRAM コントローラはプリチャージコマンドを発生し、新しい行をアクティブにして、読出しコマンドを開始します。

RURST がリセットされると、読み出し FIFO は使用されません。

Rowおよびバンク境界管理

読出しアクセスまたは書込みアクセスがRow境界をまたぐとき、次の読出しアクセスまたは書込みアクセスがシーケンシャルで、かつ現在のアクセスがRow境界に対して実行された場合、SDRAM コントローラは次の動作を行います。

1. アクティブRowのプリチャージ
2. 新しいRowのアクティブ化
3. 読出し／書込みコマンドの開始

行境界で、すべての列とデータバス幅構成に対して次の行の自動アクティブ化をサポートしています。

必要に応じて、SDRAM コントローラは次のようにコマンドの間に追加クロックサイクルを挿入します。

- プリチャージコマンドとアクティブコマンドの間に挿入して TRP パラメータを一致させます (次のアクセスが同じバンクの異なるRowの場合のみ)
- アクティブコマンドと読出しコマンドの間に挿入して TRCD パラメータを一致させます。

これらのパラメータは FMC_SDTRx レジスタで定義されます。

行境界をまたぐ読出しアクセスとバースト書込みアクセスについては、[図 128](#) と [図 129](#) を参照してください。

図 131. Row境界をまたぐ読出しアクセス

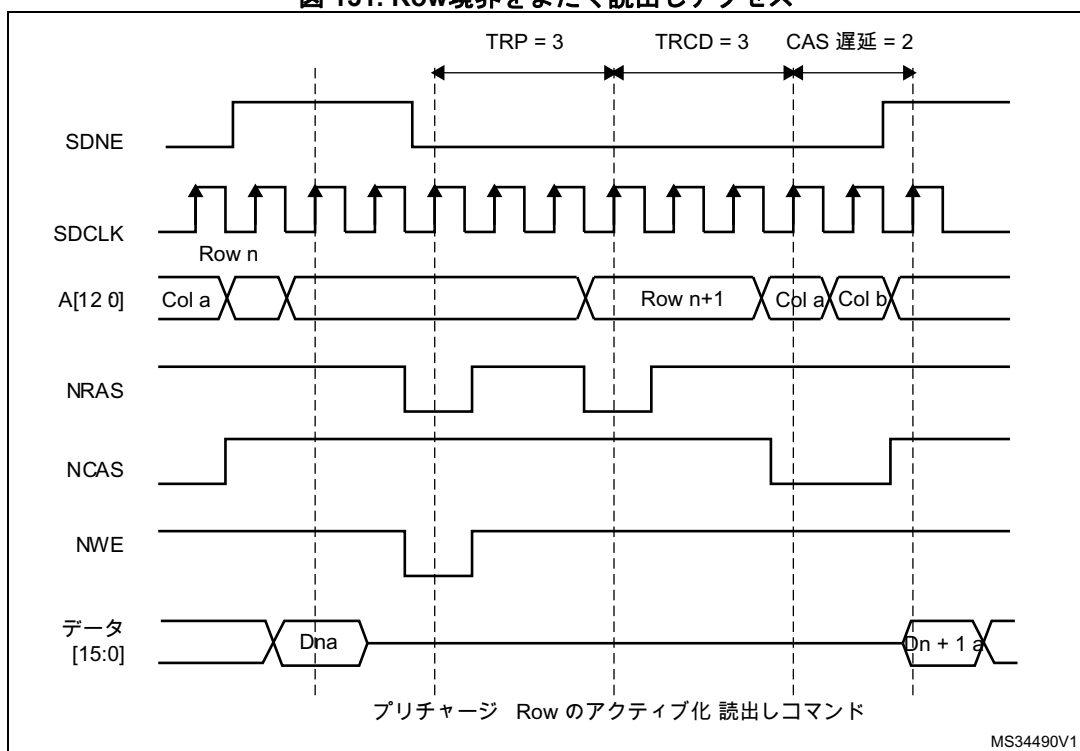
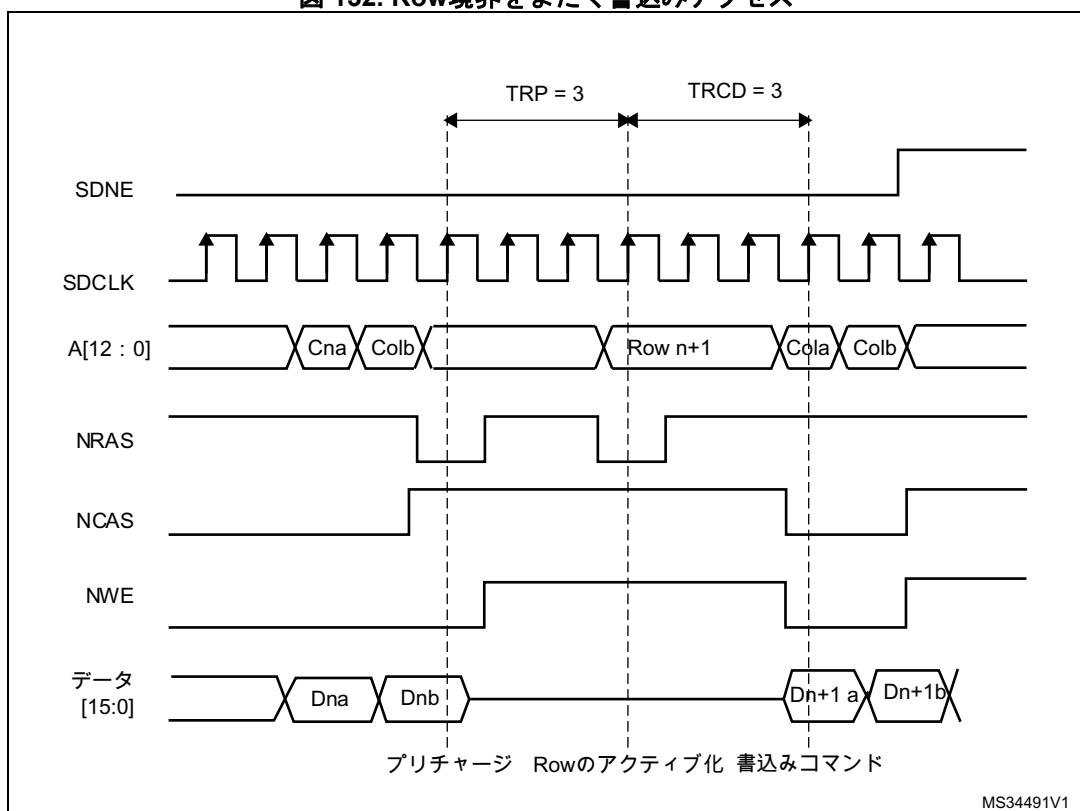


図 132. Row境界をまたぐ書き込みアクセス



次のアクセスがシーケンシャルで、かつ現在のアクセスがバンク境界をまたぐ場合、SDRAM コントローラは次のバンクの最初のRowをアクティブ化して、新しい読出し／書込みコマンドを開始します。次の2つのケースが可能です。

- 現在のバンクが直前のバンクでない場合、新しいバンク内のアクティブなRowをプリチャージする必要があります。バンク境界で、すべてのRow／Columnとデータバス幅設定に対して次のRowの自動アクティブ化をサポートしています。
- 現在のバンクが最後で、選択された SDRAM がバンク 1 に接続されている場合、SDRAM バンク 2 に接続されているデバイスの次の row の自動有効化はサポートされていません。バンク 2 へのアクセス前に必ず、バンク 1 の PALL ソフトウェア・コマンドを発行しなくてはなりません。

SDRAM コントローラリフレッシュサイクル

オートリフレッシュコマンドは、SDRAM デバイスの値をリフレッシュするときに使います。SDRAM コントローラは、オートリフレッシュコマンドを周期的に発行します。内部カウンタに、FMC_SDRTR レジスタの COUNT 値がロードされます。この値は、リフレッシュサイクル間のメモリクロックサイクル数を決定します（リフレッシュレート）。このカウンタがゼロになると、内部パルスが発生されます。

メモリアクセスが進行中である場合、オートリフレッシュリクエストは遅延されます。ただし、メモリアクセスとオートリフレッシュリクエストが同時に発生した場合は、オートリフレッシュリクエストが優先されます。

メモリアクセスがオートリフレッシュ動作中に発生すると、リクエストはバッファされて、オートリフレッシュが完了した後に処理されます。

前のオートリフレッシュリクエストの処理中に、新しいオートリフレッシュリクエストが発生すると、ステータスレジスタの RE（リフレッシュエラー）ビットがセットされます。有効化されている場合（REIE = "1"）、割込みが発生します。

SDRAM ラインがアイドル状態でない（すべての行が終了していない）場合、SDRAM コントローラはオートリフレッシュの前に PALL（プリチャージ ALL）コマンドを発行します。

FMC_SDCMR コマンドモードレジスタからオートリフレッシュコマンドが発生されると（MODE ビット = "011"）、最初に PALL コマンド（MODE ビット = "010"）を発行する必要があります。

22.8.4 低電力モード

次の2つの低消費電力モードがあります。

- セルフリフレッシュモード
SDRAM デバイス自体によりオートリフレッシュサイクルが実行されて、外部クロックなしでデータを保持します。
- パワーダウンモード
オートリフレッシュサイクルは SDRAM コントローラにより実行されます。

セルフリフレッシュモード

MODE ビットに"101"を設定し、かつ FMC_SDCMR レジスタの目標バンクビット（CTB1 および／または CTB2）を設定すると、このモードが選択されます。

SDRAM クロックは TRAS 遅延後に停止し、内部リフレッシュタイマは次の条件をみたすときにカウントを停止します。

- セルフリフレッシュコマンドが両デバイスに発行される。
- 一方のデバイスは非アクティブである（SDRAM バンクが初期化されていない）。

セルフリフレッシュモードになる前に、SDRAM コントローラは PALL コマンドを自動的に発行します。

書き込みデータ FIFO がノットエンプティの場合、すべてのデータがメモリに送られた後にセルフリフレッシュモードが開始され、BUSY ステータスフラグはセットされたままになります。

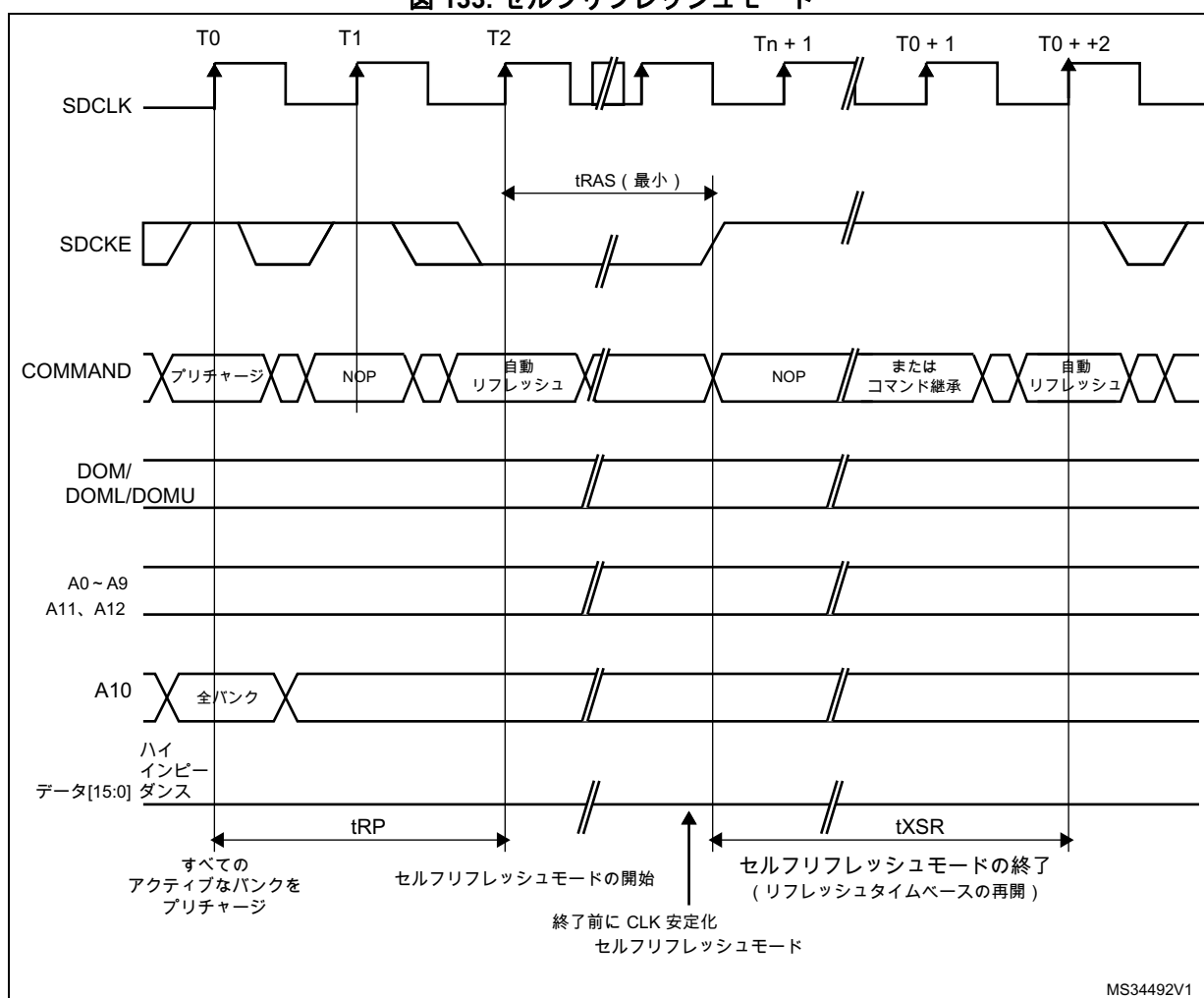
セルフリフレッシュモードでは、ローを維持する SDCKE 以外のすべての SDRAM デバイス入力が無視されます。

SDRAM デバイスは最小時間 t_{RAS} の間セルフリフレッシュモードに留まる必要があります、無限時間セルフリフレッシュモードを維持することもできます。この最小時間を保証するため、 t_{RAS} 遅延中にセルフリフレッシュを開始した後、BUSY ステータスフラグはハイを維持します。

SDRAM デバイスを選択すると直ちに、SDRAM コントローラはセルフリフレッシュモードを終わらせる一連のコマンドを発生します。メモリアクセスの後、選択されたデバイスは通常モードに留まります。

セルフリフレッシュを終了するときは、MODE ビットに"000" (通常モード) を設定し、FMC_SDCMR レジスタの目標バンクビット (CTB1 および/または CTB2) を設定する必要があります。

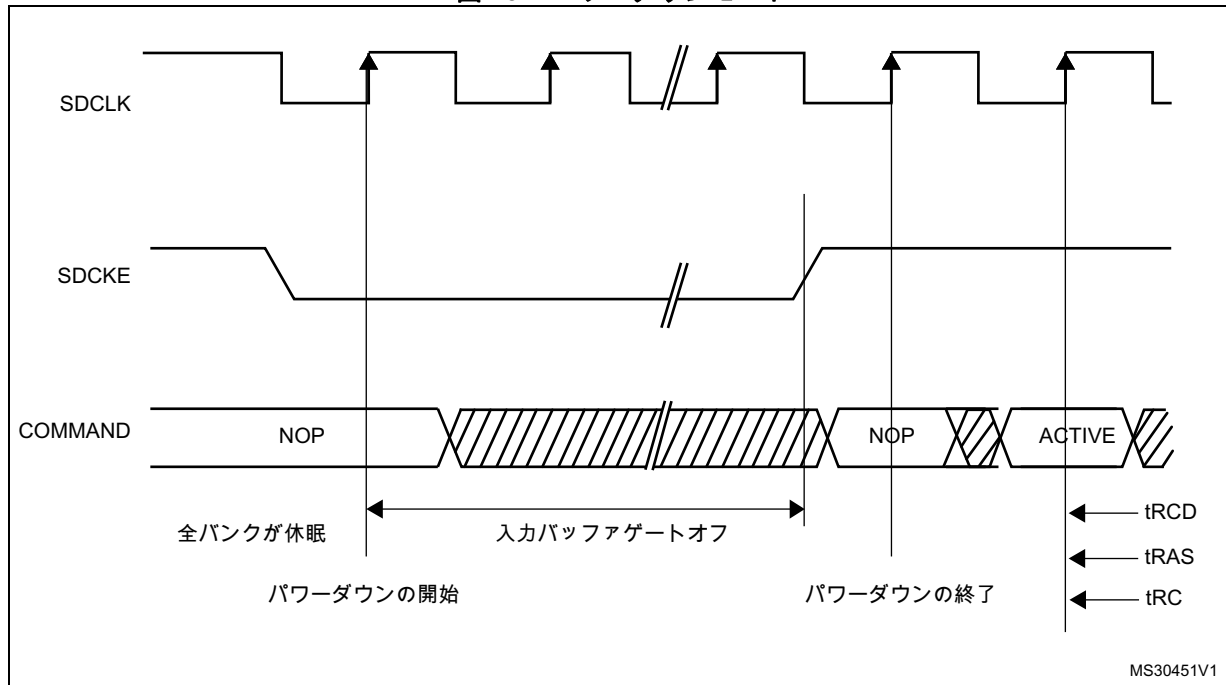
図 133. セルフリフレッシュモード



パワーダウンモード

MODE ビットに"110"を設定し、かつ FMC_SDCMR レジスタの目標バンクビット (CTB1 および／または CTB2) を設定すると、このモードが選択されます。

図 134. パワーダウンモード



書き込みデータ FIFO がノットエンプティの場合、すべてのデータがメモリに送られた後にパワーダウンモードが開始されます。

SDRAM デバイスが選択されると直ちに、SDRAM コントローラはパワーダウンモードを終了します。メモリアクセスの後、選択されたデバイスは通常モードに留まります。

パワーダウンモード時、ローを維持する SDCKE 以外のすべての SDRAM デバイス入力と出力バッファは非アクティブになります。

SDRAM デバイスはリフレッシュ周期より長い間パワーダウンモードに留まることができないため、自分自身でオートリフレッシュサイクルを実行できません。したがって、SDRAM コントローラは次の動作によりリフレッシュ動作を実行します。

1. パワーダウンモードを終了し、SDCKE をハイに駆動します。
2. パワーダウンモード中に Row がアクティブであった場合にのみ PALL コマンドを発生します。
3. オートリフレッシュコマンドを発生します。
4. 再度 SDCKE をローに駆動し、パワーダウンモードに戻ります。

パワーダウンモードを終了するときは、MODE ビットに"000" (通常モード) を設定し、FMC_SDCMR レジスタの目標バンクビット (CTB1 および／または CTB2) を設定する必要があります。

22.8.5 SDRAM コントローラレジスタ

SDRAM 制御レジスタ 1、2 (FMC_SDCR1、2)

アドレスオフセット : $0x140 + 0x4 \times (x - 1)$, ($x = 1, 2$)

リセット値 : $0x0000 : 02D0$

このレジスタは各 SDRAM メモリバンクの制御パラメータを格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC[1:0]	
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:13 RPIPE[1:0] : 読出しパイプ

これらのビットは、CAS遅延後のデータ読出しに対して、クロック・サイクル数 で遅延を指定します。

00 : クロック・サイクル遅延 なし

01 : 1サイクルの クロック遅延

10 : 2サイクルの クロック遅延

11 : 予約済み

注 : FMC_SDCR2 レジスタの対応するビットは読出し専用です。

ビット 12 RBURST : バースト読出し

このビットはバースト読出しモードを有効化します。SDRAM コントローラは CAS遅延中に次の読出しコマンドを予測し、データを読み出し FIFO に保存します。

0 : シングル読出しリクエストをバーストとして管理しません。

1 : シングル読出しリクエストを常にバーストとして管理します。

注 : FMC_SDCR2 レジスタの対応するビットは 無視されます。

ビット 11:10 SDCLK[1:0] : SDRAM クロック設定

これらのビットは両 SDRAM バンクの SDRAM クロック周期を指定し、周波数の変更前にクロックの無効化を可能にします。この場合、SDRAM の再初期化が必要です。

00 : SDCLK クロックを無効化

10 : SDCLK 周期 = $2 \times \text{HCLK}$ 周期

11 : SDCLK 周期 = $3 \times \text{HCLK}$ 周期

注 : FMC_SDCR2 レジスタの対応するビットは 無視されます。

ビット 9 WP : 書込み保護機能

このビットは、SDRAM バンクに対する書込みモードアクセスを有効化します。

0 : 書込みアクセスを許可

1 : 書込みアクセスを無視

ビット 8:7 CAS[1:0] : CAS 遅延

このビットは、メモリクロックサイクル数で SDRAM CAS遅延を設定します。

00 : 予約済み

01 : 1 サイクル

10 : 2 サイクル

11 : 3 サイクル

ビット 6 NB : 内部バンク数

このビットは内部バンク数を設定します。

0 : 2 内部バンク

1 : 4 内部バンク

ビット 5:4 MWID[1:0] : メモリデータバス幅。

これらのビットはメモリデバイス幅を指定します。

00 : 8 ビット

01 : 16 ビット

10 : 予約済み

11 : 予約済み

ビット 3:2 NR[1:0] : Row アドレスビット数

これらのビットは、Rowアドレスのビット数を指定します。

00 : 11 ビット

01 : 12 ビット

10 : 13 ビット

11 : 予約済み

ビット 1:0 NC[1:0] : Column アドレスビット数

これらのビットは、Columnアドレスのビット数を指定します。

00 : 8 ビット

01 : 9 ビット

10 : 10 ビット

11 : 11 ビット。

注 : RBURST または RPIPE の設定値を変更する前、または SDCLK クロックを無効化する前に、先に PALL コマンドを送信して実行中の動作の完了を確認する必要があります。

SDRAM タイミングレジスタ 1、2 (FMC_SDTR1、2)

アドレスオフセット : $0x148 + 0x4 \times (x - 1)$, ($x = 1, 2$)

リセット値 : 0x0FFF FFFF

このレジスタは各 SDRAM バンクのタイミングパラメータを格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	TRCD[3:0]				TRP[3:0]				TWR[3:0]			
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC[3:0]				TRAS[3:0]				TXSR[3:0]				TMRD[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 TRCD[3:0] : Row から Column までの遅延

これらのビットは、アクティブコマンドと読出し／書込みコマンドとの間の遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

ビット 23:20 TRP[3:0] : Row プリチャージ遅延

これらのビットは、プリチャージコマンドと別のコマンドとの間の遅延をメモリクロックサイクル数で指定します。TRP タイミングは FMC_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRP には低速側デバイスのタイミングを設定する必要があります。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

注： FMC_SDTR2 レジスタの対応するビットは 無視されます。

ビット 19:16 TWR[3:0] : 復帰遅延

これらのビットは、書込みコマンドとプリチャージコマンドとの間の遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

注： TWR は SDRAM のデータシートで規定される書込み復帰時間(t_{WR})に一致するようにプログラムし、次を保証する必要があります。

$TWR \geq TRAS - TRCD$ かつ $TWR \geq TRC - TRCD - TRP$

例：TRAS = 4 サイクル、TRCD = 2 サイクル。したがって、 $TWR \geq 2$ サイクル。TWR には 0x1 を設定する必要があります。

2 つの SDRAM デバイスを使用する場合、FMC_SDTR1 および FMC_SDTR2 には、より低速な SDRAM デバイスと同じ TWR タイミングを設定する必要があります。

1 つの SDRAM デバイスしか使用しない場合、TWR タイミングは未使用バンクではリセット値 (0xF) に保持する必要があります。

ビット 15:12 TRC[3:0] : Row サイクル遅延

これらのビットは、リフレッシュコマンドとアクティブコマンドとの間の遅延、および連続する 2 つのリフレッシュコマンドの間の遅延をそれぞれ定義します。メモリクロックサイクル数で表されます。TRC タイミングは FMC_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRC には低速側デバイスのタイミングを設定する必要があります。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

注： TRC は SDRAM デバイスデータシートで規定される TRC と TRFC (オートリフレッシュ周期) タイミングに一致する必要があります。

注： FMC_SDTR2 レジスタの対応するビットは 無視されます。

ビット 11:8 TRAS[3:0] : セルフリフレッシュ時間

これらのビットは、メモリクロックサイクルで最小セルフリフレッシュ周期を指定します。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

ビット 7:4 TXSR[3:0] : セルフリフレッシュ遅延の終了

これらのビットは、セルフリフレッシュコマンドの解除からアクティブコマンド発行までの遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

注 : 2 つの SDRAM デバイスを使用する場合、FMC_SDTR1 および FMC_SDTR2 には、より低速な SDRAM デバイスと同じ TXSR タイミングを設定する必要があります。

ビット 3:0 TMRD[3:0] : ロードモードレジスタのアクティブまでの時間

これらのビットは、ロードモードレジスタコマンドとアクティブまたはリフレッシュコマンドとの間の遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

...

1111 : 16 サイクル

注 : 2 つの SDRAM デバイスを接続する場合は、コマンドモードレジスタにより両デバイスに同時に実行されるすべてのアクセス（ロードモードレジスタコマンド）は、FMC_SDTR1 レジスタでバンク 1 に対して設定されたタイミングパラメータ（TMRD タイミング、TRAS タイミング）を使って発行されます。

TRP タイミングと TRC タイミングは FMC_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRP と TRC には低速側デバイスのタイミングを設定する必要があります。

SDRAM コマンドモードレジスタ (FMC_SDCMR)

アドレスオフセット : 0x150

リセット値 : 0x0000 0000

このレジスタは、SDRAM デバイスをアクセスするときに発行されたコマンドを格納します。このレジスタを使って、SDRAM デバイスの初期化、セルフリフレッシュの開始、パワーダウンモードの開始を行います。MODE フィールドに書き込みを行うと直ちに、CTB1 コマンドビットと CTB2 コマンドビットに応じて一方または両方の SDRAM バンクへコマンドが発行されます。このレジスタは、両 SDRAM バンクに対して同じです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRD[12:7]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRD[6:0]						NRFS[3:0]				CTB1	CTB2	MODE[2:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:9 MRD[12:0] : モードレジスタの定義

この 13 ビットフィールドは、SDRAM モードレジスタ値を指定します。モードレジスタは、ロードモードレジスタコマンドを使って設定します。

ビット 8:5 NRFS[3:0] : オートリフレッシュ数

これらのビットは、MODE = "011"のときに発行される連続オートリフレッシュコマンド数を指定します。

0000 : 1 オートリフレッシュサイクル

0001 : 2 オートリフレッシュサイクル

...

1110 : 15 オートリフレッシュサイクル

1111 : 16 オートリフレッシュサイクル

ビット 4 CTB1 : コマンドターゲットバンク 1

このビットは、SDRAM バンク 1 へコマンドを発行するか否かを指定します。

0 : SDRAM バンク 1 へコマンドを発行しない

1 : SDRAM バンク 1 へコマンドを発行する

ビット 3 CTB2 : コマンドターゲットバンク 2

このビットは、SDRAM バンク 2 へコマンドを発行するか否かを指定します。

0 : SDRAM バンク 2 へコマンドを発行しない

1 : SDRAM バンク 2 へコマンドを発行する

ビット 2:0 MODE[2:0] : コマンドモード

これらのビットは、SDRAM デバイスへ発行するコマンドを指定します。

000 : 通常モード

001 : クロック設定イネーブル

010 : PALL (「全バンクプリチャージ」) コマンド

011 : オートリフレッシュコマンド

100 : ロードモードレジスタ

101 : セルフリフレッシュコマンド

110 : パワーダウンコマンド

111 : 予約済み

注 : コマンドを発行する場合、少なくとも 1 つのコマンドターゲットバンクビット (CTB1 または CTB2) をセットする必要があります。そうしないと、そのコマンドは無視されます。

注 : 2 つの SDRAM バンクを使用する場合、オートリフレッシュコマンドおよび PALL コマンドを CTB1 および CTB2 ビットがセットされた 2 つにデバイスに対して同時に発行する必要があります。そうしないと、そのコマンドは無視されます。

注 : 1 つの SDRAM バンクのみを使用し、該当する CTB ビットをセットした状態でコマンドを発行する場合、もう一方の使用されていないバンクの CTB ビットを 0 に保持する必要があります。

SDRAM リフレッシュ・タイマ・レジスタ (FMC_SDRTR)

アドレスオフセット : 0x154

リセット値 : 0x0000 0000

このレジスタは、リフレッシュタイマカウント値を設定することにより、リフレッシュサイクル間の SDCLK クロックサイクル数でリフレッシュレートを設定します。

$$\text{Refresh rate} = (\text{COUNT} + 1) \times \text{SDRAM clock frequency}$$

$$\text{COUNT} = (\text{SDRAM refresh period} / \text{Number of rows}) - 20$$

例

$$\text{Refresh rate} = 64 \text{ ms} / (8196 \text{ rows}) = 7.81 \mu\text{s}$$

ここで、64 ms は SDRAM リフレッシュ周期です。

$$7.81\mu\text{s} \times 60\text{MHz} = 468.6$$

読出しリクエストが受理されているときに内部リフレッシュリクエストが発生した場合には、安全マージンを確保するためリフレッシュレートを 20 SDRAM クロックサイクルだけ大きくする必要があります（上の例の場合）。これは、COUNT 値"0000111000000"（448）に対応します。

この 13 ビットフィールドは、SDRAM クロックを使ってデクリメントされるタイマにロードされます。このタイマはゼロになると、リフレッシュパルスが発生します。COUNT 値は、少なくとも 41 SDRAM クロックサイクルである必要があります。

FMC_SDRTR レジスタが設定されると直ちに、タイマはカウントを開始します。レジスタに設定された値が "0" の場合、リフレッシュは実行されません。初期化の後にこのレジスタを再設定して、リフレッシュレートが変更されるのを回避する必要があります。

リフレッシュパルスが発生されるたびに、この 13 ビット COUNT フィールドがカウンタへ再ロードされます。

メモリアクセスが進行中の場合、オートリフレッシュリクエストは遅延させられます。ただし、メモリアクセスとオートリフレッシュリクエストが同時に発生した場合は、オートリフレッシュリクエストが優先されます。メモリアクセスがリフレッシュ動作中に発生すると、リクエストはバッファされて、リフレッシュが完了した後に処理されます。

このレジスタは、SDRAM バンク 1 と SDRAM バンク 2 に共通です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	REIE	COUNT[12:0]													CRE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **REIE** : RES 割込み有効

0 : 割込みは無効です。

1 : RE = 1 の場合に割込みが発生します

ビット 13:1 **COUNT[12:0]** : リフレッシュタイマカウント

この 13 ビットフィールドは、SDRAM デバイスのリフレッシュレートを指定します。メモリクロックサイクル数で表されます。この値は、少なくとも 41 SDRAM クロックサイクル（0x29）である必要があります。

リフレッシュレート = (COUNT + 1) × SDRAM クロック周波数

COUNT = (SDRAM リフレッシュ周期 / Row数) - 20

ビット 0 **CRE** : リフレッシュエラーフラグのクリア

このビットを使って、ステータスレジスタのリフレッシュエラーフラグ（RE）をクリアします。

0 : 影響なし。

1 : リフレッシュエラーフラグをクリア

注 : プログラムされた COUNT 値は、以下で求めるタイミングの合計とイコールであってはなりません。
TWR + TRP + TRC + TRCD + 4 メモリ・クロック・サイクル

SDRAM ステータス・レジスタ (FMC_SDSR)

アドレスオフセット : 0x158

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	MODES2 [1:0]	MODES1 [1:0]			RE
										r	r	r	r	r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 BUSY : ビジーステータス

このビットは、コマンドモードリクエスト後の SDRAM コントローラのステータスを指定します。

0 : SDRAM コントローラは新しいリクエスト受付の準備完了

1 : SDRAM コントローラは新しいリクエスト受付の準備未完了

ビット 4:3 MODES2[1:0] : バンク 2 のステータスモード

このビットは、SDRAM バンク 2 のステータスモードを指定します。

00 : 通常モード

01 : セルフリフレッシュモード

10 : パワーダウンモード

ビット 2:1 MODES1[1:0] : バンク 1 のステータスモード

このビットは、SDRAM バンク 1 のステータスモードを指定します。

00 : 通常モード

01 : セルフリフレッシュモード

10 : パワーダウンモード

ビット 0 RE : リフレッシュエラーフラグ

0 : リフレッシュエラーは検出されない

1 : リフレッシュエラーが検出された

REIE = 1 かつ RE = 1 の場合割込みが発生します。

22.8.6 FMC レジスタマップ

表 202. FMCレジスタマップとリセット値

オフセット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	FMC_BCR1	FMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		WFDIS	CCLKEN	CBURSTW	CPSIZE [2:0]			ASYNWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN		
	リセット値	0								0 0	0	0	0	0	0	0	0	0	0	0	1	1	0	Res.	0	0	1	1	0	1	1	0	1	1	
0x08	FMC_BCR2	FMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN		
	リセット値	0								0 0				0	0	0	0	0	0	1	1	0	Res.	0	0	0	1	1	0	1	0	0	1	0	
0x10	FMC_BCR3	FMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN		
	リセット値	0								0 0	0			0	0	0	0	0	0	1	1	0	Res.	0	0	0	1	1	0	1	0	0	1	0	
0x18	FMC_BCR4	FMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN		
	リセット値	0								0 0				0	0	0	0	0	0	1	1	0	Res.	0	0	0	1	1	0	1	0	0	1	0	
0x04	FMC_BTR1	DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]							
	リセット値	0 0	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1		
0x0C	FMC_BTR2	DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]							
	リセット値	0 0	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1		
0x14	FMC_BTR3	DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]							
	リセット値	0 0	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1		
0x1C	FMC_BTR4	DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]							
	リセット値	0 0	0 0	0 0	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1	1 1		
0x20	FMC_PCSCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTB4EN	CNTB3EN	CNTB2EN	CNTB1EN	CSCCOUNT[15:0]																	
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x104	FMC_BWTR1	DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]						
	リセット値	0 0	0 0	0 0										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x10C	FMC_BWTR2	DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		BUSTURN [3:0]			DATAST[7:0]								ADDHLD[3:0]			ADDSET[3:0]						
	リセット値	0 0	0 0	0 0										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

表 202. FMCレジスタマップとリセット値 (続き)

オフセット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x114	FMC_BWTR3	DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN [3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]						
	リセット値	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x11C	FMC_BWTR4	DATAHLD[1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN [3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]						
	リセット値	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x80	FMC_PCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS [2:0]		TAR[3:0]			TCLR[3:0]			Res.	Res.	ECCEN	PWID [1:0]	PTYP	PBKEN	PWAITEN	Res.				
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
0x84	FMC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
	リセット値																										1	0	0	0	0	0	0
0x88	FMC_PMEM	MEMHIZx[7:0]					MEMHOLDx[7:0]					MEMWAITx[7:0]					MEMSETx[7:0]																
	リセット値	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x8C	FMC_PATT	ATTHIZ[7:0]					ATTHOLD[7:0]					ATTWAIT[7:0]					ATTSET[7:0]																
	リセット値	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x94	FMC_ECCR	ECCx[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	FMC_SDCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIPE [1:0]	RBURST	SDCLK [1:0]	WP	CAS [1:0]	NB	MWID [1:0]	NR [1:0]	NC							
	リセット値																	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0
0x144	FMC_SDCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBURST	SDCLK [1:0]	WP	CAS [1:0]	NB	MWID [1:0]	NR [1:0]	NC						
	リセット値																			0	1	1	0	1	0	0	1	0	0	0	0	0	0
0x148	FMC_SDTR1	Res.	Res.	Res.	Res.	TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14C	FMC_SDTR2	Res.	Res.	Res.	Res.	TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x150	FMC_SDCMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRD[12:0]										NRFS[3:0]			CTB1	CTB2	MODE[2:0]						
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x154	FMC_SDRTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REIE	COUNT[12:0]										CRE			
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x158	FMC_SDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	MODES2 [1:0]		MODES1 [1:0]		Res.
	リセット値																											0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

23 Octo-SPI インタフェース (OCTOSPI)

23.1 概要

OCTOSPI は、シリアル PSRAM、シリアル NAND およびシリアル NOR Flash メモリ、HyperRAM™、HyperFlash™ メモリなど、ほとんどの外部シリアルメモリを以下の機能モードでサポートしています。

- インダイレクトモード：すべての動作は、コマンド、アドレス、データ、および転送パラメータをプリセットする OCTOSPI レジスタを使用して実行されます。
- 自動ステータスポーリングモード：外部メモリステータスレジスタが定期的に読み出され、フラグのセット時に割込みを生成することができます。この機能は、レギュラコマンドプロトコルでのみ使用できます。
- メモリマップドモード：外部メモリはメモリ空間に配置され、システムからは内部メモリであるかのように見え、読出しと書込みの両方の動作がサポートされます。

OCTOSPI は、以下の関連するフレームフォーマットを備えたプロトコルをサポートしています。

- コマンド、アドレス、オルタネートバイト、ダミーサイクル、およびデータフェーズから成るレギュラコマンドフレームフォーマット
- HyperBus™ フレームフォーマット

23.2 OCTOSPI の主な機能

- 機能モード：インダイレクト、自動ステータスポーリング、およびメモリマップドモード
- メモリマップドモードでの読出しおよび書込みのサポート
- 外部 (P) SRAM メモリのサポート
- シングル、デュアル、クワッド、およびオクタル通信に対応
- 2 つのクワッドメモリに並列にアクセスすることで 8 ビットの同時送受信を可能にするデュアルメモリ構成
- SDR (シングルデータレート) および DTR (ダブル転送レート) のサポート
- データストローブのサポート
- 完全にプログラム可能な OP コード
- 完全にプログラム可能なフレームフォーマット
- 読出し方向のメモリへのラップタイプのアクセスをサポート
- HyperBus のサポート
- 受信および送信用の内蔵 FIFO
- 非同期バスクロックとカーネルクロックのサポート
- 8、16、32 ビットのデータアクセスを許可
- DMA のプロトコルに対応
- インダイレクトモードで動作するための DMA チャンネル
- FIFO 閾値、タイムアウト、動作完了、アクセスエラーによる割込み生成
- トランザクションの受け入れが 1 つに制限された AHB インタフェース：メモリ側で前の転送が完了した後でのみ、AHB バス上で次の転送を受け入れるインタフェース

23.3 OCTOSPI の実装

表 203. OCTOSPI の実装

OCTOSPI の機能	OCTOSPI1/2
HyperBus 規格準拠	X
Xcella 規格準拠	X
XSPI (JEDEC251ES) 規格準拠	X
AMBA® AHB 準拠のデータインタフェース	X
デュアル AHB インタフェース	X
非同期 AHB クロックとカーネルクロック	X
機能モード: インダイレクト、自動ステータスポーリング、およびメモリマップドモード	X
メモリマップドモードでの読出しおよび書込みのサポート	X
デュアルクワッド構成	X
SDR (シングルデータレート) および DTR (ダブル転送レート)	X
データストローブ (DS、DQS)	X
完全にプログラム可能な OP コード	X
完全にプログラム可能なフレームフォーマット	X
受信および送信用の内蔵 FIFO	X
8、16、32 ビットのデータアクセス	X
FIFO 閾値、タイムアウト、動作完了、アクセスエラーによる割込み	X
拡張 CSHT タイムアウト	X
メモリマップド書込み	X
リフレッシュカウンタ	X
GPDMA インタフェース	X
プリフェッチ無効化	-
プリフェッチ用ハードウェアおよびソフトウェア	-

23.4 OCTOSPI の機能詳細

23.4.1 OCTOSPI ブロック図

図 135. OCTOSPI ブロック図 (オクタル構成)

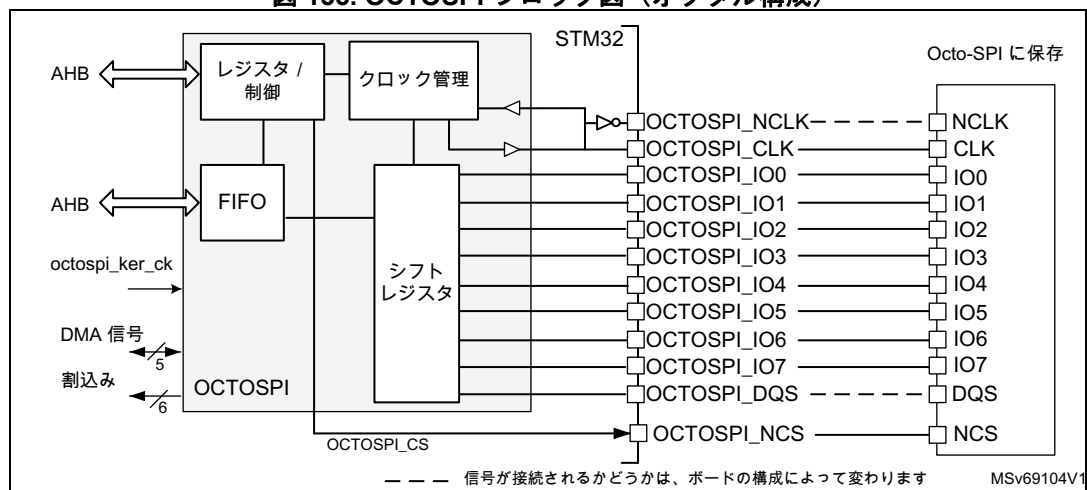


図 136. OCTOSPI ブロック図 (クワッド構成)

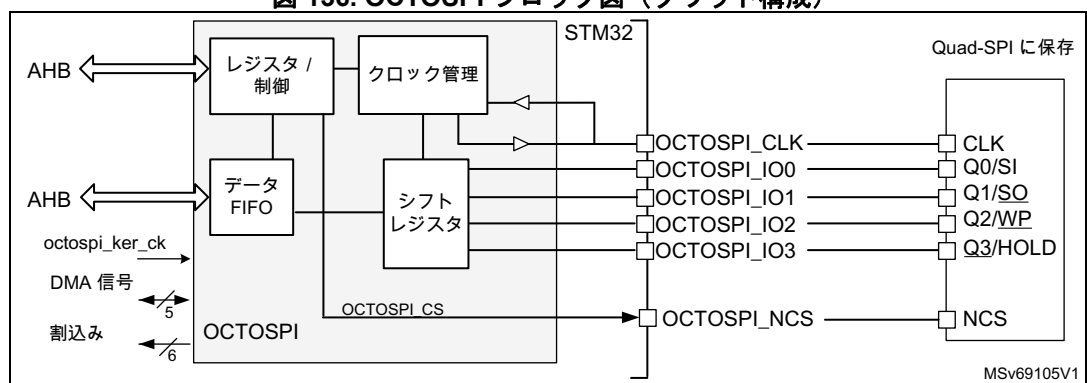
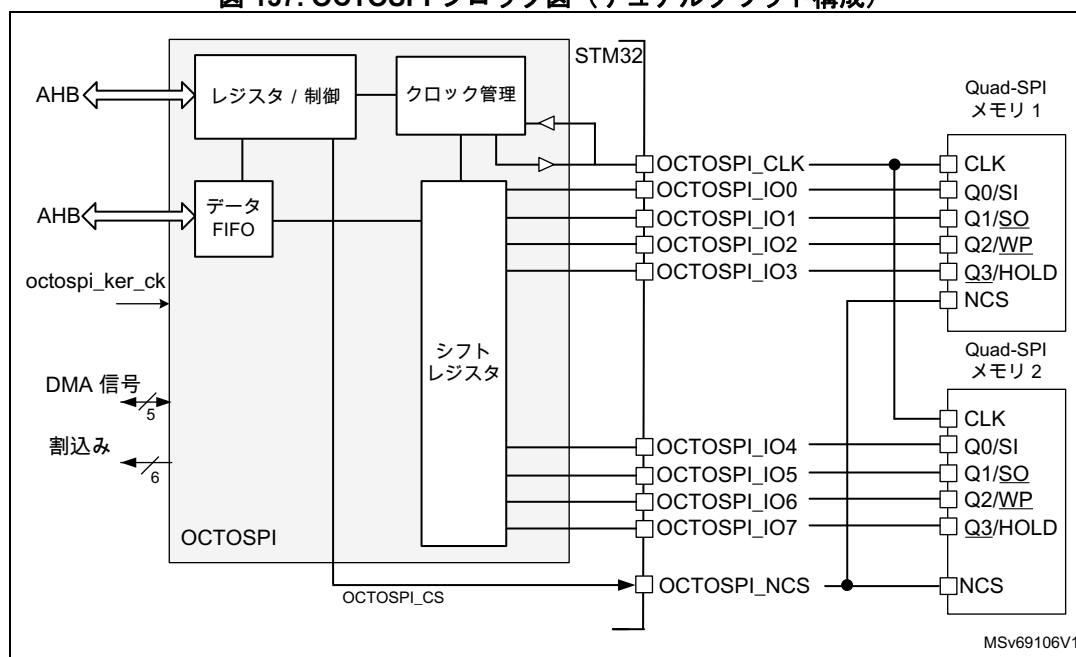


図 137. OCTOSPI ブロック図 (デュアルクワッド構成)



23.4.2 OCTOSPI ピンおよび内部信号

表 204. OCTOSPI の入出力ピン

ピン名	タイプ	説明
OCTOSPI_NCLK	出力	1.8 V HyperBus プロトコルに対応した OCTOSPI 反転クロック
OCTOSPI_CLK		OCTOSPI クロック
OCTOSPI_IOn (n = 0~7)	入力／出力	OCTOSPI データピン
OCTOSPI_NCS	出力	メモリのチップセレクト
OCTOSPI_DQS	入力／出力	メモリとの間のデータストローブ/書込みマスク信号

注意： データの破損を避けるために、すべての OCTOSPI 入出力ピンに同じ設定（出力速度、HSLV）を使用してください。

表 205. OCTOSPI 内部信号

信号名	タイプ	説明
octospi_hclk	入力	OCTOSPI 用 AHB クロック
octospi_ker_ck	入力	OCTOSPI カーネルクロック
octospi_dma	N/A	DMA リクエスト信号
octospi_it	出力	グローバル割込みライン（割込みの複数ソースについては表 208 を参照）

23.4.3 OCTOSPI インタフェースを用いたメモリモード

OCTOSPI は次のプロトコルをサポートしています。

- レギュラコマンドプロトコル
- HyperBus プロトコル

OCTOSPI は機能モードに応じて、下記の 6 から 12 本の信号を使ってメモリとインタフェースを取ります。

- NCS : チップセレクト
- CLK : 通信クロック
- NCLK : 1.8 V HyperBus プロトコルでのみ使用される反転クロック
- DQS : レギュラコマンドプロトコルでのみ、入力としてのみ使用されるデータストローブ
- IO[3:0] : LSB 用データバス
- IO[7:4] :
 - デュアルクワッド構成およびオクタル構成で使用される MSB 用データバス
 - データバスは、Quad-SPI モードのリマップとして使用できます

23.4.4 OCTOSPI レギュラコマンドプロトコル

レギュラコマンドプロトコルでは、OCTOSPI はコマンドを使って外部デバイスと通信します。各コマンドには以下の 5 つのフェーズを含めることができます。

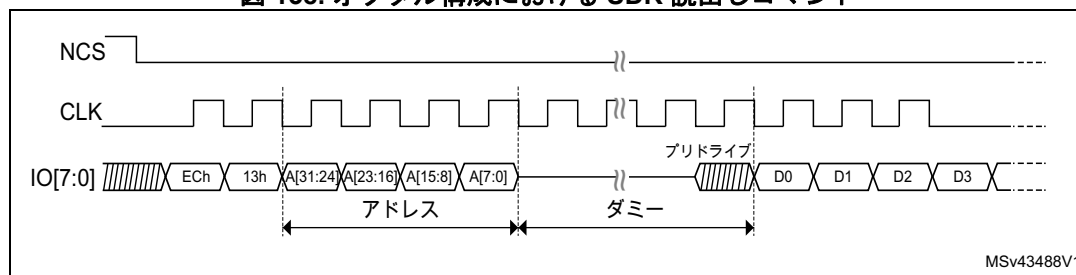
- 命令フェーズ
- アドレスフェーズ
- オルタネートバイトフェーズ
- ダミーサイクルフェーズ
- データフェーズ

これらのフェーズはいずれもスキップするように構成できますが、単一フェーズコマンドの場合、サポートされる使用例は命令フェーズのみです。

NCS は各コマンドの開始前に立下がり、各コマンドの終了後に再び立ち上がります。

メモリマップドモードでは、読出しと書込みの両方の動作がサポートされており、その結果、いくつかの設定レジスタは書込み動作を指定するために複製されます（読出し動作は通常のレジスタを使用して設定されます）。

図 138. オクタル構成における SDR 読出しコマンド



レギュラコマンドプロトコルの具体的な機能は 0x0100～0x01FC のオフセット範囲のレジスタによって設定されます。

命令フェーズ

このフェーズでは、1~4バイトの命令が外部デバイスに送信され、実行すべき動作のタイプが指定されます。送信される命令のサイズは OCTOSPI_CCR の ISIZE[1:0] で設定され、命令は OCTOSPI_IR の INSTRUCTION[31:0] にプログラムされます。

命令フェーズでは、オプションで以下を送信できます。

- 一度に 1 ビット (Single-SPI モードで IO0、SO 信号を経由して)
- 一度に 2 ビット (Dual-SPI モードで IO0/IO1 を経由して)
- 一度に 4 ビット (Quad-SPI モードで IO0~IO3 を経由して)
- 一度に 8 ビット (Octal-SPI モードで IO0~IO7 を経由して)。

これは、OCTOSPI_CCR の IMODE[2:0] を使用して設定できます。

命令は、OCTOSPI_CCR の IDTR をセットすることによって、DTR モードでクロックの立上がりおよび立下がりの各エッジで送信することができます。

OCTOSPI_CCR の IMODE[2:0] = 000 の場合、命令フェーズはスキップされますので、コマンドシーケンスはアドレスフェーズが存在する場合はそこから開始されます。

メモリマップドモードでは、書込み動作に使用する命令は OCTOSPI_WIR にて指定され、命令フォーマットは OCTOSPI_WCCR にて指定されます。読出し動作に使用する命令と命令フォーマットは、OCTOSPI_IR と OCTOSPI_CCR にて指定されます。

アドレスフェーズ

アドレスフェーズでは、動作のアドレスを示す 1~4バイトが外部デバイスに送信されます。送信されるアドレスバイト数は、OCTOSPI_CCR の ADSIZE[1:0] で設定されます。

インダイレクトモードと自動ステータスポーリングモードでは、送信されるアドレスバイトは OCTOSPI_AR の ADDRESS[31:0] によって指定されます。メモリマップドモードでは、アドレスは AHB を経由して直接与えられます (システム内のマスタから)。

アドレスフェーズでは、以下のものを送信できます。

- 一度に 1 ビット (Single-SPI モードで IO0、SO 信号を経由して)
- 一度に 2 ビット (Dual-SPI モードで IO0/IO1 を経由して)
- 一度に 4 ビット (Quad-SPI モードで IO0~IO3 を経由して)
- 一度に 8 ビット (Octal-SPI モードで IO0~IO7 を経由して)

これは、OCTOSPI_CCR の ADMODE[2:0] を使用して設定できます。

アドレスは、OCTOSPI_CCR の ADDTR をセットすることで DTR モードで (クロックの立上がりおよび立下がりの各エッジで) 送信することができます。

ADMODE[2:0] = 000 の場合、アドレスフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

メモリマップドモードでは、書込み動作のアドレスフォーマットは OCTOSPI_WCCR にて指定されます。読出し動作のアドレスフォーマットは OCTOSPI_CCR にて指定されます。

オルタネートバイトフェーズ

オルタネートバイトフェーズでは、一般に動作のモードを制御するために、1~4 バイトが外部デバイスに送信されます。送信されるオルタネートバイト数は、OCTOSPI_CCR の ABSIZE[1:0] で設定されます。送信されるバイト内容は、OCTOSPI_ABR にて指定されます。

オルタネートバイトフェーズでは、以下のものを送信できます。

- 一度に 1 ビット (Single-SPI モードで IO0、SO 信号を経由して)
- 一度に 2 ビット (Dual-SPI モードで IO0/IO1 を経由して)
- 一度に 4 ビット (Quad-SPI モードで IO0~IO3 を経由して)
- 一度に 8 ビット (Octal-SPI モードで IO0~IO7 を経由して)

これは、OCTOSPI_CCR の ABMODE[2:0] を使用して設定できます。

オルタネートバイトは、OCTOSPI_CCR の ABDTR をセットすることで DTR モードで (クロックの立上がりおよび立下がりの各エッジで) 送信することができます。

ABMODE[2:0] = 000 の場合、オルタネートバイトフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

場合によっては、オルタネートバイトフェーズ中にフルバイトではなく 2 分の 1 バイト (4 ビット) だけを送信する必要があるかもしれません。たとえば、Dual-SPI モード使用時で、オルタネートバイトに 2 サイクルだけが使われる場合などです。

この場合、ファームウェアによって Quad-SPI モード (ABMODE[2:0] = 011) を使用し、ALTERNATE[31:0] のビット 7 および 3 を 1 (IO3 ラインをハイに保持) に、OCTOSPI_IR のビット 6 および 2 を 0 (IO2 ラインをローに保持) に設定して 1 バイトを送信できます。

送信される 4 ビットの上位 2 ビットは ALTERNATE[31:0] のビット 5:4 にセットされ、下位 2 ビットはビット 1:0 にセットされます。たとえば、4bit の値 2 (0010) が IO0/IO1 で送信される場合、ALTERNATE[31:0] は 0x8A (1000_1010) にセットされている必要があります。

メモリマップドモードでは、書込み動作に使用するオルタネートバイトは OCTOSPI_WABR にて指定され、オルタネートバイトフォーマットは OCTOSPI_WCCR にて指定されます。読出し動作に使用するオルタネートバイトとオルタネートバイトフォーマットは OCTOSPI_ABR と OCTOSPI_CCR にて指定されます。

ダミーサイクルフェーズ (メモリ待ち時間)

ダミーサイクルフェーズでは、データを送受信せずに 1~31 回の空サイクルが設けられます。これは、より高いクロック周波数が使われている場合に、外部デバイスにデータフェーズの準備をする時間を与えるためです。このフェーズでのサイクル数は、OCTOSPI_TCR の DCYC[4:0] で指定されます。SDR モードと DTR モードのどちらでも、この持続時間はフル CLK サイクル数として指定されます。

DCYC[4:0] = 00000 の場合、ダミーサイクルフェーズはスキップされますので、コマンドシーケンスはデータフェーズが存在する場合は直接そこに進みます。

データ信号を出力モードから入力モードに変更するのに十分な「ターンアラウンド」時間を確保するために、Dual-SPI、Quad-SPI、または Octal-SPI モードを使用して外部デバイスからデータを受信する場合、少なくとも 1 ダミーサイクルが必要です。

メモリマップドモードでは、書込み動作のダミーサイクルは OCTOSPI_WTCR にて指定されます。読出し動作のダミーサイクルは OCTOSPI_TCR にて指定されます。

データフェーズ

データフェーズでは、外部デバイスとの間で任意のバイト数を送受信できます。

インダイレクトモードでは、送受信するバイト数は OCTOSPI_DLR で指定されます。このモードでは、外部デバイスに送信するデータを OCTOSPI_DR に書き込む必要があります。一方、インダイレクト読出しモードでは、外部デバイスから受信するデータは OCTOSPI_DR を読み出して取得します。

自動ステータスポーリングモードでは、受信するバイト数は OCTOSPI_DLR にて指定され、外部デバイスから受信するデータは OCTOSPI_DR を読み出して取得できます。

メモリマップドモードでは、読み書きされるデータは、直接 AHB を経由して Cortex コアまたは DMA との間で送受信されます。

データフェーズでは、以下のものを送受信できます。

- 一度に 1 ビット (Single-SPI モードで IO0/IO1 (SO/SI それぞれ) を経由して)
- 一度に 2 ビット (Dual-SPI モードで IO0/IO1 を経由して)
- 一度に 4 ビット (Quad-SPI モードで IO0~IO3 を経由して)
- 一度に 8 ビット (Octal-SPI モードで IO0~IO7 を経由して)

これは、OCTOSPI_CCR の DMODE[2:0] を使用して設定できます。

データは、OCTOSPI_CCR の DDTR をセットすることで DTR モードで (クロックの立上がりおよび立下りの各エッジで) 送受信することができます。

DMODE[2:0] = 000 の場合、データフェーズはスキップされるので、コマンドシーケンスは NCS を立ち上げることですぐに終了します。この設定はインダイレクト書き込みモードでのみ使用してください。

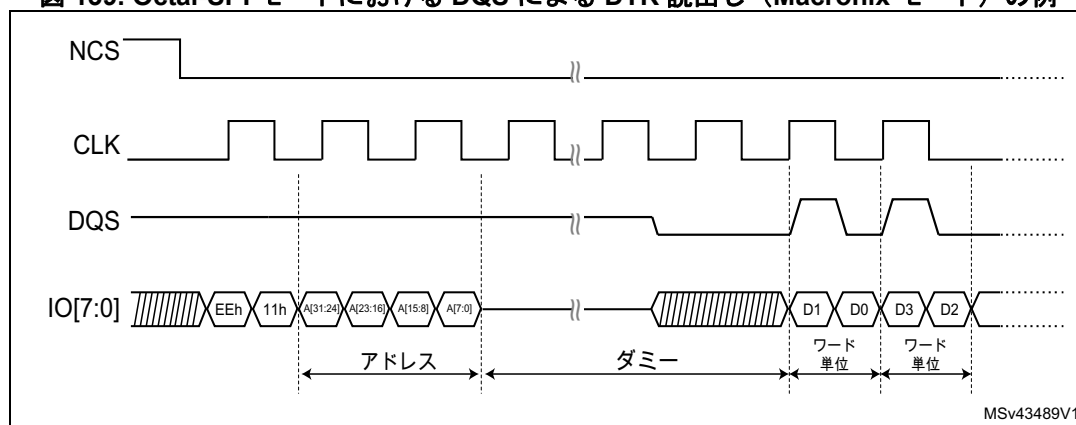
メモリマップドモードでは、書き込み動作のデータフォーマットは OCTOSPI_WCCR にて指定されます。読出し動作のデータフォーマットは OCTOSPI_CCR にて指定されます。

DQS の使用

DQS 信号は、デバイスがデータに合わせて DQS をトグルしているときの読出しトランザクション中のデータストロブに使用できます。

DQS 管理は OCTOSPI_CCR の DQSE を設定することによって有効にできます。

図 139. Octal-SPI モードにおける DQS による DTR 読出し (Macronix モード) の例



23.4.5 OCTOSPI レギュラコマンドプロトコル信号インタフェース

Single-SPI モード

レガシー SPI モードでは、1 ビットの順次送受信のみが可能です。このモードでは、データは SO 信号 (その I/O は IO0 と共有される) を経由して外部デバイスに送信されます。外部デバイスから受信するデータは、SI (その I/O は IO1 と共有される) を経由して到着します。

OCTOSPI_CCR および OCTOSPI_WCCR の IMODE、ADMODE、ABMODE、および DMODE フィールドをそれぞれ 001 にセットすることによって、この Single-SPI モードを使用できるように、フェーズごとに個別に設定することができます。

各フェーズを Single-SPI モードに設定する場合：

- IO0 (SO) は出力モードです。
- IO1 (SI) は入力モードです (ハイインピーダンス)。
- IO2は出力モードで、強制的に 0に設定されます (書き込み保護機能を無効にするため)。
- IO3は出力モードで、強制的に 1に設定されます (ホールド機能を無効にするため)。
- IO4 から IO7 は出力モードで、強制的に 0 に設定されます。

これは、DMODE[2:0] = 001であれば、ダミーフェーズにも該当します。

Dual-SPI モード

Dual-SPI モードでは、IO0/IO1 信号を経由して 2 ビットが同時に送受信されます。

OCTOSPI_CCR および OCTOSPI_WCCR の IMODE、ADMODE、ABMODE、および DMODE フィールドをそれぞれ 010 にセットすることによって、DUAL-SPI モードを使用できるように、フェーズごとに個別に設定することができます。

各フェーズを Dual-SPI モードに設定する場合：

- IO0/IO1は、データフェーズの読出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。
- IO2は出力モードで、強制的に 0に設定されます。
- IO3は出力モードで、強制的に 1に設定されます。
- IO4 から IO7 は出力モードで、強制的に 0 に設定されます。

ダミーフェーズでは、DMODE[2:0] = 010の場合、IO0/IO1は常にハイインピーダンスです。

Quad-SPI モード

Quad-SPI モードでは、IO0/IO1/IO2/IO3 信号を経由して 4 ビットが同時に送受信されます。

OCTOSPI_CCR および OCTOSPI_WCCR の IMODE、ADMODE、ABMODE、および DMODE フィールドをそれぞれ 011 にセットすることによって、Quad-SPI モードを使用できるように、フェーズごとに個別に設定することができます。

各フェーズを Quad-SPI モードに設定する場合：

- IO0~IO3 はすべて、データフェーズの読出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。
- IO4 から IO7 は出力モードで、強制的に 0 に設定されます。

ダミーフェーズでは、DMODE[2:0] = 011 の場合、IO0~IO3 はすべてハイインピーダンスです。

Octal-SPI モード

通常の Octal-SPI モードでは、IO[0:7] 信号を経由して 8 ビットが同時に送受信されます。

OCTOSPI_CCR および OCTOSPI_WCCR の IMODE、ADMODE、ABMODE、および DMODE フィールドをそれぞれ 100 にセットすることによって、Octal-SPI モードを使用できるように、フェーズごとに個別に設定することができます。

Octal-SPI モードに設定されている各フェーズでは、IO[0:7] はすべてデータフェーズの読出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。

ダミーフェーズでは、DMODE[2:0] = 100の場合、IO[0:7] はすべてハイインピーダンスです。

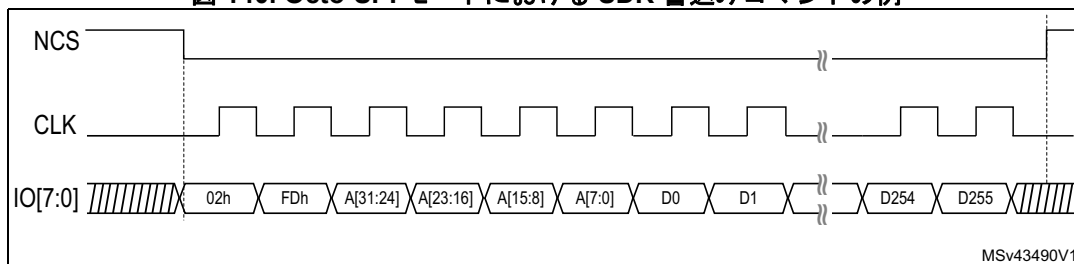
シングルデータレート (SDR) モード

すべてのフェーズは、デフォルトで SDR モードで動作します。

このモードでは、QUADSPI が IO0/SO、IO1~IO7 信号を駆動している場合、これらの信号は CLK の立下がりエッジでのみ遷移します。

SDR モードでデータを受信する場合、OCTOSPI は外部デバイスも CLK の立下がりエッジを使用してデータを送信するとみなします。デフォルトでは (OCTOSPI_TCR において SSHIFT = 0 の場合)、信号は次の CLK の (立上がり) エッジを使用してサンプリングされます。

図 140. Octo-SPI モードにおける SDR 書き込みコマンドの例



注：最後のデータが読み取られた後、内部同期のために、Octo-SPI インタフェースによって最大 6 つの追加のダミークロックサイクルが生成される場合があります。

ダブル転送レート (DTR) モード

OCTOSPI_CCR の IDTR、ADDTR、ABDTR、および DDTR をセットすることによって、命令、アドレス、オルタネートバイト、およびデータの各フェーズを DTR モードで動作するように設定できます。

メモリマップドモードでは、書き込み動作の各フェーズの DTR モードは OCTOSPI_WCCR にて指定されます。読出し動作の各フェーズの DTR モードは OCTOSPI_CCR にて指定されます。

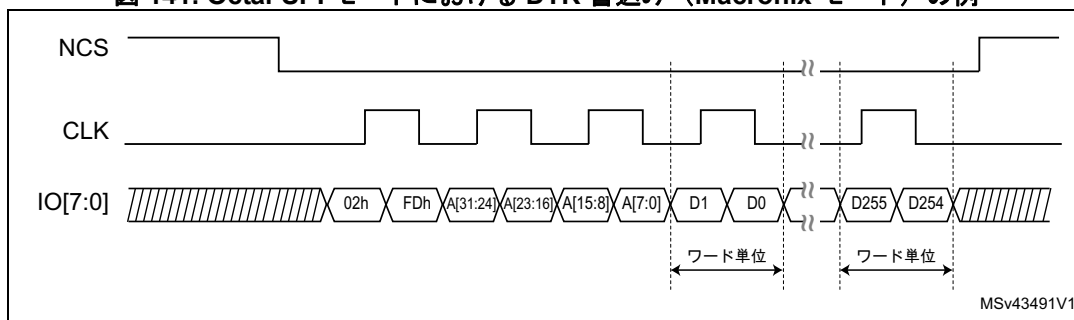
DTR モードでは、OCTOSPI が命令、アドレス、オルタネートバイトの各フェーズで IO0/SO、IO1~IO7 信号を駆動している場合、CLK の立下がりエッジおよび立上がりエッジのそれぞれでビットを送受信します。

DTR モードでデータを受信する場合、OCTOSPI は外部デバイスも CLK の立上がりおよび立下りの両方のエッジを使用してデータを送信するとみなします。OCTOSPI_CCR において DDTR = 1 の場合、ソフトウェアは OCTOSPI_TCR の SSHIFT をクリアする必要があります。したがって、信号は 1/2 CLK サイクル後にサンプリングされます (下図の逆エッジで)。

DTR モードでは、OCTOSPI_TCR の DHQC をセットして、出力をクォータサイクルだけシフトし、メモリ側のホールド問題を避けることを推奨します。

注：予期せぬ動作を引き起こす場合があるため、プリスケール値が 0 の場合は DHQC をセットしてはなりません。

図 141. Octal-SPI モードにおける DTR 書き込み (Macronix モード) の例



注：最後のデータが読み取られた後、内部同期のために、Octo-SPI インタフェースによって最大 6 つの追加のダミークロックサイクルが生成される場合があります。

デュアルクワッド構成

OCTOSPI_CR において DMM = 1 の場合、OCTOSPI はデュアルメモリ構成です。すなわち、DMODE = 011 の場合、2 つの外部 Quad-SPI デバイス（デバイス A とデバイス B）が使用され、サイクルごとに 8 ビットの送受信（DTR モードでは 16 ビット）を行うことで、スループットを効果的に 2 倍にします。

どちらのデバイス（A または B）も同じ CLK と NCS 信号を使用しますが、それぞれが個別の IO0～IO3 信号を有しています。

デュアルクワッド構成は、SDR モードまたは DTR モードと組み合わせて使用できるだけでなく、Single-SPI、Dual-SPI、および Quad-SPI モードとも組み合わせて使用できます。

OCTOSPI_DCR1 の DEVSZ[4:0] で指定されているデバイスのサイズは、外部デバイスの総容量（コンポーネント 1 個のサイズの 2 倍）を反映していなければなりません。

アドレス X が偶数の場合、OCTOSPI がアドレス X に対して与えるバイトはデバイス A のアドレス X/2にあるバイトであり、アドレス X+1に対して与えるバイトはデバイス B のアドレス X/2にあるバイトです。言い換えれば、偶数アドレスにあるバイトはすべてデバイス A に、奇数アドレスにあるバイトはすべてデバイス B に格納されます。

デュアルクワッド構成でデバイスのステータスレジスタを読み出す場合、レギュラコマンドプロトコルで読み出すバイト数の 2 倍のバイトを読み出す必要があります。各デバイスがステータスレジスタのフェッチ命令後に有効な 8 ビットを提供する場合、OCTOSPI はデータ長 2 バイト（16 ビット）で設定する必要があります、それにより OCTOSPI は各デバイスから 1 バイトを受信します。

各デバイスが 16 ビットのステータスを提供する場合、OCTOSPI は、4 バイトを読み出して、デュアルクワッド構成の両方のデバイスのステータスビットをすべて取得できるように設定する必要があります。結果（データレジスタ内）の最下位バイトはデバイス A のステータスレジスタの最下位バイトです。次のバイトはデバイス B のステータスレジスタの最下位バイトです。さらに、データレジスタの 3 番目のバイトはデバイス A の 2 番目のバイトです。4 番目のバイトはデバイス B の 2 番目のバイトになります（デバイスに 16 ビットステータスレジスタがある場合）。

デュアルクワッド構成では、常に偶数バイトにアクセスする必要があります。このため、DMM = 1 のとき、OCTOSPI_DLR の DL[31:0] のビット 0 は 1 のままとなります。

デュアルクワッド構成では、デバイス A のインタフェース信号の動作は基本的に通常モードの動作と同じです。デバイス B のインタフェース信号の波形は、命令、アドレス、オルタネートバイト、およびダミーサイクルフェーズの間は、デバイス A のものとまったく同じです。つまり、どちらのデバイスも常に同じ命令や同じアドレスを受信するということです。

データフェーズ中は、2つのバス AIOx と BIOx は並行してデータ転送を行います。デバイス A に対する送信（または受信）データはデバイス B のものとは異なります。

23.4.6 HyperBus プロトコル

OCTOSPI は HyperBus プロトコルを使って外部デバイスと通信することができます。

HyperBus は、動作電圧に応じて 11 から 12 ピンを使用します。

- 双方向データバスとして IO[7:0]
- データ読み書きストローブおよび遅延挿入のための RWDS (DQS ピンに配置)
- NCS
- CLK
- 1.8 V 動作の NCLK (このモードをサポートするには、デバイスに 1.8 V が供給されていなければなりません)

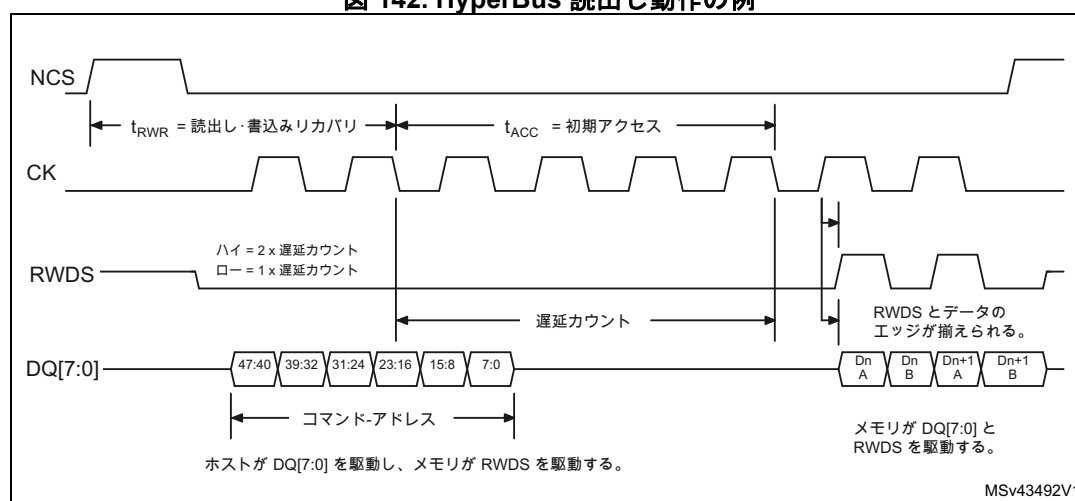
HyperBus では、コマンド指定もオルタネートバイトも必要としません。結果として、トランザクションのタイミングを定義するために個別のレジスタセットが使用されます。

HyperBus のフレームは次の 2 つのフェーズで構成されます。

- コマンド／アドレスフェーズ
- データフェーズ

NCS はトランザクションの開始前に立ち下がり、各トランザクションの終了後に再び立ち上がります。

図 142. HyperBus 読出し動作の例



注： 最後のデータが読み取られた後、内部同期のために、Octo-SPI インタフェースによって最大 6 つの追加のダミークロックサイクルが生成される場合があります。

HyperBus の具体的な機能は 0x0200～0x02FC のオフセット範囲のレジスタによって設定されます。

コマンド／アドレスフェーズ

この初期フェーズでは、OCTOSPI は外部デバイスとの間で実施する動作を指定するために、IO[7:0] を経由して 48 ビットを送信します。

表 206. コマンド/アドレスフェーズの説明

CA ビット	ビット名	説明
47	R/W#	トランザクションが読出しか書込みかを識別します。
46	アドレス空間	トランザクションがメモリか、またはレジスタ空間にアクセスしているかを示します。
45	バーストタイプ	バーストがリニアか、またはラップトであるかを示します。
44 ~ 16	行アドレスおよび上位列アドレス	行アドレスおよび上位列アドレスを選択します。
15 ~ 3	予約済み	-
2 ~ 0	下位列アドレス	ハーフページ内の開始 16 ビットワードを選択します。

アドレス空間は OCTOSPI_DCR1 のメモリタイプを示す MTYP[2:0] に設定します。

デバイスの合計サイズは OCTOSPI_DCR1 の DEVSIZE[4:0] に設定します。マルチチップ製品 (MCP) の場合、デバイスサイズはすべての MCP ダイのすべてのサイズの合計です。

初期遅延を設けた読出し/書込み動作

HyperBus の読出しおよび書込み動作では、次の 2 つのタイミングを考慮する必要があります。

- t_{RWR} : デバイスの最小限の読出し／書込みリカバリ時間 (OCTOSPI_HLCR の TRWR[7:0] で定義)
- t_{ACC} : メモリ待ち時間に応じたデバイスへのアクセス時間 (OCTOSPI_HLCR の TACC[7:0] で定義)

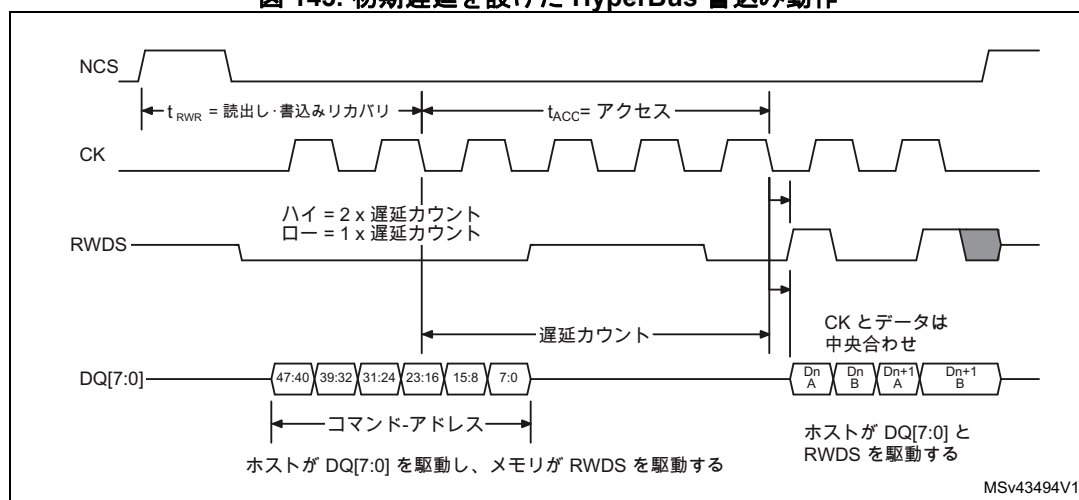
読出し動作中、デバイスは RWDS を次の 2 通りの方法で使します (図 142 を参照)。

- コマンド／アドレスフェーズ時は、追加の遅延を要求するため
- データフェーズ時は、データストローブのため

書込み動作中、RWDS は次のように使用されます。

- デバイスによって、コマンド／アドレスフェーズ時に追加の遅延を要求するために。
- OCTOSPI によって、データフェーズ時にデータ書込みマスキングのために。

図 143. 初期遅延を設けた HyperBus 書き込み動作



追加の遅延を設けた読み出し/書き込み動作

デバイスに追加の遅延が必要な場合（たとえば SDRAM のリフレッシュ期間中）、コマンド/アドレスフェーズ中に RWDS 信号の 1 つの期間中に RWDS を 1 につなげる必要があります。

デバイスの要求を満たすために、OCTOSPI によって追加の t_{ACC} 期間が追加されます。

図 144. 追加の遅延を設けた HyperBus 読み出し動作

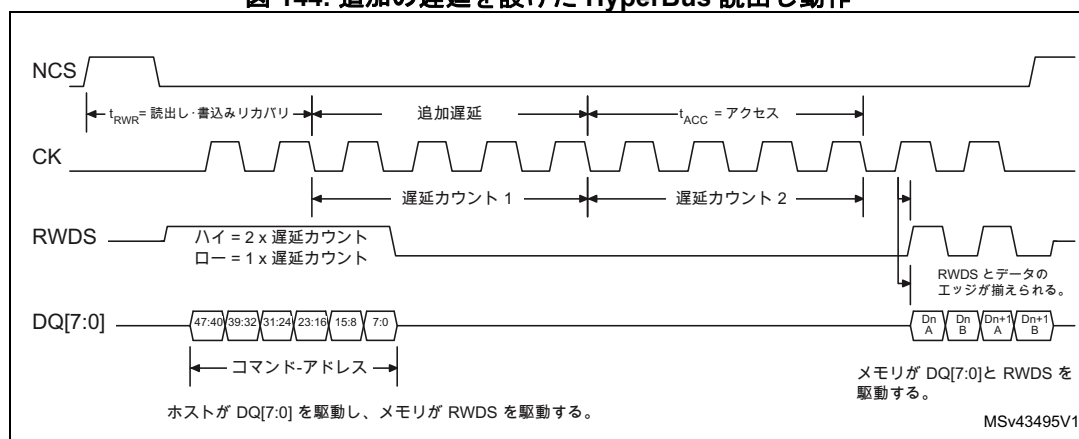
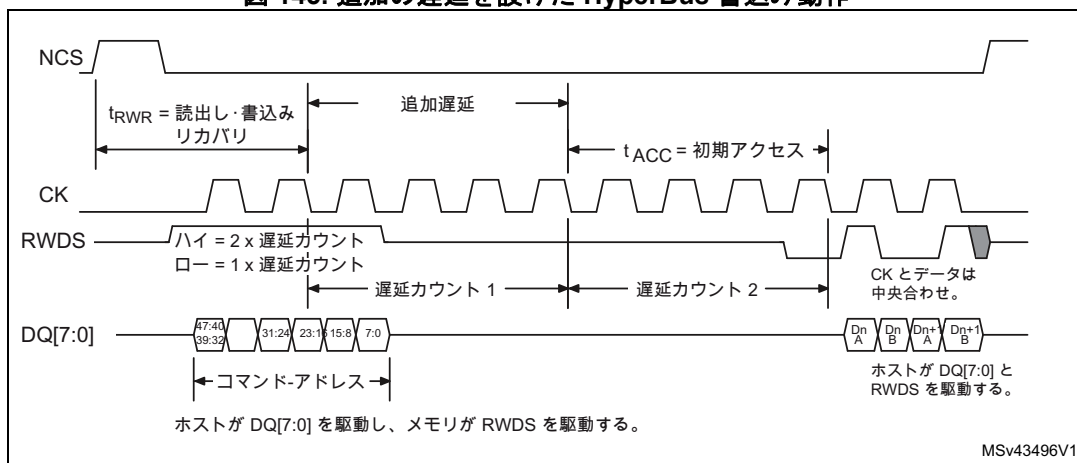


図 145. 追加の遅延を設けた HyperBus 書き込み動作



固定-遅延モード

デバイスまたはアプリケーションによっては、上に述べたような可変の遅延を設けた動作を望まないものもあります。

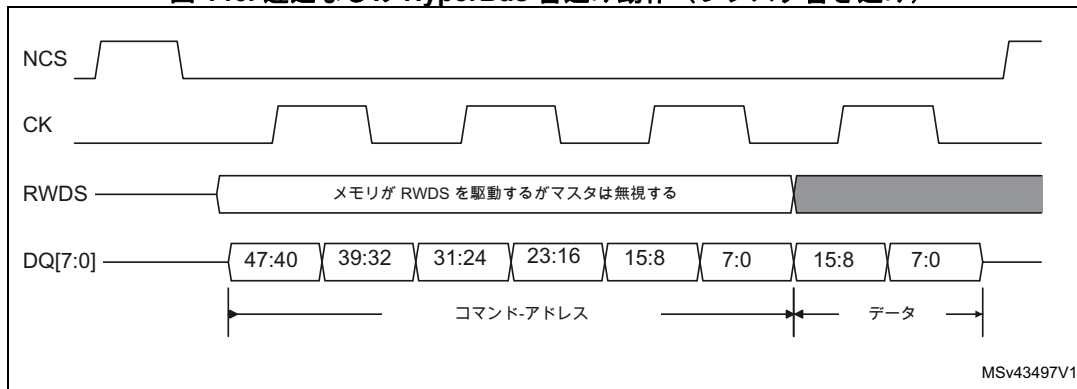
OCTOSPI_HLCR の LM をセットすることで、遅延を強制的に $2 \times t_{ACC}$ にすることができます。

この OCTOSPI 遅延モードでは、RWDS 信号の状態が OCTOSPI によって考慮されず、常に追加の遅延が追加されて、固定の $2 \times t_{ACC}$ の遅延時間が発生します。

遅延なしの書き込み動作

また、デバイスによっては書き込み動作にゼロ遅延を要求する可能性もあります。この書き込みゼロ遅延は、OCTOSPI_HLCR の WZL を設定することによって強制することができます。

図 146. 遅延なしの HyperBus 書き込み動作（レジスタ書き込み）

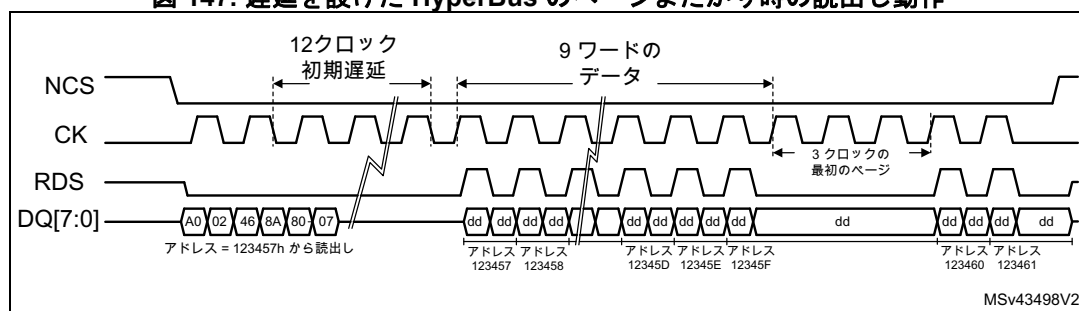


読出し動作中のページまたがり時の遅延

デバイスによっては、ページをまたがって読出し動作をする際に追加の遅延が必要になることがあります。

ページアクセスには初期遅延を考慮する必要があるので、その結果、最初のアクセスがページ境界に近いと、 t_{ACC} 時間を考慮するためにページまたがり時に自動的に遅延が追加されます。

図 147. 遅延を設けた HyperBus のページまたがり時の読出し動作



23.4.7 特有の機能

OCTOSPI は次のようないくつかの特有の機能をサポートしています。

- ラップのサポート
- NCS 境界とリフレッシュ

ラップのサポート

OCTOSPI では、HyperBus プロトコルで定義されているハイブリッドラップがサポートされています。ハイブリッドラップは、レギュラコマンドプロトコルでもサポートされます。

ハイブリッドラップでは、トランザクションは最初のラップの後もインクリメントアクセスで続行できます。

対象のメモリがサポートするラップサイズは、OCTOSPI_DCR2 の WRAPSIZE で設定します。

ラップはメモリの読出し方向でのみサポートされます。また、データサイズ = 4 バイトの場合のみサポートされます。ラップド読出しは、HyperBus とレギュラコマンドプロトコルの両方でサポートされます。ラップド読出しアクセスを有効にするには、ラップド読出しアクセス特性に応じた専用の OCTOSPI_WPxxx レジスタをプログラムする必要があります。これらのレジスタは、HyperBus とレギュラコマンドプロトコルの両方に適用されます。

対象のメモリがハイブリッドラップをサポートしていない場合は、WRAPSIZE は 0 にセットする必要があります。

注： ラップ動作はリフレッシュによって中断することはできません。リフレッシュイベントはラップの完了後にのみ考慮されます。

NCS 境界とリフレッシュ

OCTOSPI トランザクションを調整するために次の 2 つのプロセスをアクティブにすることができます。

- NCS 境界
- リフレッシュ

NCS 境界機能によって、トランザクションは整列されたアドレスの境界に制限されます。整列させるアドレスのサイズは、OCTOSPI_DCR3 の CSBOUND[4:0] で設定され、 $2^{CSBOUND}$ に等しくなります。

たとえば、CSBOUND[4:0] = 0x4 の場合、境界は $2^4 = 16$ バイトにセットされます。そうすると、LSB アドレスが 0xF に等しくなるたびに、および新しいトランザクションが発行されて次のデータに取りかかるたびに、NCS が解除されます。

CSBOUND[4:0] = 0 の場合、この機能は無効になります。最小値の 3 が推奨されます。

書込みモードでは、Flash メモリの他のページをプログラムするためにコマンドが必要なので、NCS 境界機能は書込みモードの Flash メモリデバイスには使用できません。

リフレッシュ機能では、トランザクションの期間を OCTOSPI_DCR4 の REFRESH[31:0] にプログラムされた値に制限します。期間はサイクル数で表されます。これにより、外部 RAM は定期的に内部リフレッシュ動作を実行することができます。

リフレッシュ値は、コマンド／アドレス／オルタネート／ダミーフェーズを含む最小トランザクションサイズ（サイクル数の単位で）より大きくなければなりません。

NCS 境界とリフレッシュが同時に有効にされた場合、NCS は最初の条件が合致した時に解除されます。

中断された転送後の再開始

読取りまたは書き込み動作がタイムアウトまたは通信調整機能によって中断された場合、Octo-SPI インタフェースでは、ポートの所有権を取り戻した後、できるだけ早く、中断前に実際にアクセスした最後のアドレスに続くアドレスとともに最初のコマンドシーケンスを再発行します。最初に設定された転送は継目なく進行し、終了します。

23.4.8 OCTOSPI の動作モードの概要

OCTOSPI には、使用されるローレベルのプロトコル（レギュラコマンドまたは HyperBus のいずれか）に関係なく、次の動作モードがあります。

- インダイレクトモード（読出しまたは書き込み）
- 自動ステータスポーリングモード（レギュラコマンドプロトコルでのみ）
- メモリマップドモード

23.4.9 OCTOSPI インダイレクトモード

インダイレクトモードでは、OCTOSPI レジスタに書き込むことでコマンドが開始され、データレジスタへの書き込み/読出しによってデータが転送されます。その方法はその他の通信ペリフェラルの場合と同様です。

OCTOSPI_CR において FMODE[1:0] = 00 の場合、OCTOSPI はインダイレクト書き込みモードです。このモードでは、データフェーズ中に外部デバイスへバイトが送信されます。データは、OCTOSPI_DR への書き込みによって提供されます。

FMODE[1:0] = 01 の場合、OCTOSPI はインダイレクト読出しモードです。このモードでは、データフェーズ中に外部デバイスからバイトが受信されます。データは、OCTOSPI_DR の読出しによって復元されます。

インダイレクトモードでは、DQS は無効の状態で、OCTOSPI が 8 つのレーンにわたって DTR モードに設定されている場合、指定する開始アドレスとデータ長は偶数でなければなりません。

注： 開始アドレスが前回のインダイレクトアクセスの開始アドレスと同じであっても、OCTOSPI_AR レジスタを更新する必要があります。

読出し/書き込みするバイト数は OCTOSPI_DLR で指定されます。

- DL[31:0] = 0xFFFF FFFF の場合、データ長は不定長とみなされ、OCTOSPI は単純に外部デバイスの最後（DEVSIZE によって定義される）に達するまでデータを転送し続けます。転送されるバイトが無い場合は、OCTOSPI_CCR の DMODE[2:0] を 0 にセットする必要があります。
- DL[31:0] = 0xFFFF FFFF で DEVSIZE[4:0] = 0x1F（最大値は 4GB のデバイスを示す）の場合、転送は無期限に続けられ、アポトリクエスト後または OCTOSPI の無効化後にやっと停止します。最後のメモリアドレスが（アドレス 0xFFFF FFFF で）読み出された後も、アドレス = 0x0000 0000 で読出しが続けられます。

プログラムされた送信または受信バイト数に達すると、OCTOSPI_SR に TCF ビットがセットされ、OCTOSPI_CR で TCIE = 1 になっていれば割込みが生成されます。データ数が定義されていない場

合、外部 SPI メモリの限界に達した時点で、OCTOSPI_DCR1に定義されているデバイスのサイズに応じて TCF がセットされます。

レギュラコマンドプロトコルでの転送開始のトリガ

レギュラコマンドプロトコルを使用している場合、インダイレクトモードで転送を開始する方法は、OCTOSPI の設定に応じて 3 通りあります。一般に、転送の開始はソフトウェアがコマンドに必要な最後の情報を提供するとすぐにトリガされます。もっと具体的に言えば、インダイレクトモードでは、以下のいずれかのイベントシーケンスが発生した場合に転送が開始されます。

- アドレスが不要 (ADMODE[2:0] = 000) で、ソフトウェアからのデータ提供が不要 (FMODE[1:0] = 01 または DMODE[2:0] = 000) な場合で、OCTOSPI_IR の INSTRUCTION[31:0] に書き込みが実行された時点
- アドレスが必要 (ADMODE[2:0] ≠ 000) で、ソフトウェアからのデータ提供が不要 (FMODE[1:0] = 01 または DMODE[2:0] = 000) な場合で、OCTOSPI_AR の ADDRESS[31:0] に書き込みが実行された時点
- ソフトウェアからのデータ提供が必要 (FMODE[1:0] = 00 で DMODE[2:0] ≠ 000) な場合で、OCTOSPI_DR の DATA[31:0] に書き込みが実行された時点

OCTOSPI_ABR への書き込みが通信開始をトリガすることはありません。オルタネートバイトが必要な場合、事前にプログラムしておく必要があります。

コマンドが開始されたらすぐに、BUSY ビットが自動的に OCTOSPI_SR にセットされます。

HyperBus プロトコルでの転送開始のトリガ

OCTOSPI の設定に応じて、インダイレクトモードでコマンドを開始するさまざまな方法があります。一般に、ファームウェアが転送を始めるのに必要な最後の情報を提供するとすぐに転送は開始されます。もっと具体的に言えば、以下のレジスタ設定のいずれかが実行する最後のものであった場合、それによってインダイレクトモードでの通信が開始されます。

- インダイレクト読出しモード (FMODE[1:0] = 01) で、ADMODE[2:0] ≠ 000 で ADDRESS[31:0] (OCTOSPI_AR) への書き込みを行ったとき。
- インダイレクト書き込みモード (FMODE = 00 の場合) で、DATA[31:0] (OCTOSPI_DR) への書き込みを行ったとき。
- インダイレクト読出しモード (ADMODE[2:0] = 000 かつ FMODE = 01 の状態) に対して、INSTRUCTION[31:0] (OCTOSPI_IR) への (ダミー) 書き込みを行ったとき。

転送が開始されたらすぐに、BUSY ビット (OCTOSPI_SR[5]) が自動的にセットされます。

FIFO およびデータの管理

インダイレクトモードでは、データは OCTOSPI に内蔵されている 32 バイトの FIFO を通ります。OCTOSPI_SR の FLEVEL は、現在 FIFO に保持されているバイト数を示します。

AHB バーストランザクションがサポートされています。バーストのデータは OCTOSPI_DR に連続して書き込まれ、すぐに内蔵 FIFO に転送されます。

インダイレクト書き込みモード (FMODE[1:0] = 00) では、ソフトウェアは OCTOSPI_DR に書き込む時点で FIFO にデータを追加します。書き込みにより FIFO に追加されるバイト数は、ワード書き込みで 4 バイト、ハーフワード書き込みで 2 バイト、バイト書き込みで 1 バイトです。ソフトウェアが FIFO に追加するバイト数が多すぎる (DL[31:0] で示される数より多い) 場合、余分なバイトは、書き込み動作の終了時 (TCF がセットされるとき) に FIFO から一掃されます。

OCTOSPI_DR へのバイト/ハーフワードアクセスは、32 ビットレジスタの最下位バイト/ハーフワードにのみ実行する必要があります。

FTHRES を使用して FIFO 閾値を定義します。そのポイントに達すると FIFO 閾値フラグ FTF がセットされます。インダイレクト読み出しモードでは、FIFO から読み出される有効バイト数が閾値を超えると FTF がセットされます。また、FTHRES の設定にかかわらず、外部デバイスから最後のバイトが読み出された後に FIFO 内にデータが残っている場合も FTF がセットされます。インダイレクト書き込みモードでは、FIFO 内の空のバイト数が閾値を超えると FTF がセットされます。

FTIE = 1 の場合、FTF がセットされた時点で割込みが発生します。DMAEN = 1 の場合、FTF がセットされた時点で DMA 転送が開始されます。閾値条件が真でなくなると（CPU または DMA によって十分なデータが転送された後）すぐに、ハードウェアによって FTF がクリアされます。

次のラインに対する要求がない限り、RX FIFO に最後に読み込まれたデータは有効なままです。つまり、アプリケーションが同じ場所で連続して何回か読み出す場合、データは RX FIFO から提供され、離れているメモリから再度読み出されることはありません。

23.4.10 OCTOSPI 自動ステータスポーリングモード

自動ステータスポーリングモードでは、OCTOSPI は定期的にコマンドを開始させて、定義された数のステータスバイト（最大 4）を読み出します。受信バイトをマスクして、ステータスビットの一部を分離することができます。また、選択されたビットに値が定義されている場合は割込みを生成することができます。自動ステータスポーリングモードはレギュラコマンドプロトコルでのみ使用する必要があります。HyperBus プロトコルの場合、HyperFlash メモリへの読み出しステータスレジスタは 2 つのステップ（書き込み操作とそれに続く読み出し操作）で行う必要があるため、悪用することができません。

デバイスへのアクセスは、インダイレクト読み出しモードと同じ方法で開始されます。この時点で OCTOSPI_SR の BUSY がハイになり、周期的なアクセスが実行される間もハイのままです。

OCTOSPI_PSMAR の MASK[31:0] の内容は、自動ステータスポーリングモードで外部デバイスからのデータをマスクするために使用されます。

- MASK[n] = 0 の場合、リザルトのビット n はマスクされ、考慮されません。
- MASK[n] = 1 かつ ビット[n] の値が OCTOSPI_PSMAR の MATCH[n] の値と同じであれば、ビット n での一致が検出されます。

OCTOSPI_CR において PMM = 0 の場合、AND 一致モードがアクティブになります。SMF は、マスクされていないすべてのビットで一致が検出された場合のみ OCTOSPI_SR にセットされます。

OCTOSPI_CR において PMM = 1 の場合、OR 一致モードがアクティブになります。SMF は、マスクされていない任意のビットで一致が検出された場合にセットされます。

割込みは、SMIE = 1 の場合に SMF = 1 になるとき呼び出されます。

APMS が OCTOSPI_CR でセットされている場合、一致が検出されるとすぐに動作は停止し、BUSY が 0 になります。そうでない場合、BUSY は 1 のままで、アボートが検出されるまで、または OCTOSPI が無効化される（EN = 0）まで周期的なアクセスが継続します。

OCTOSPI_DR には、最後に受信したステータスバイトが含まれます（FIFO は無効化されます）。レジスタの内容は、一致ロジックで使用するマスキングの影響を受けません。OCTOSPI_SR の FTF は新たなステータスの読み出しが完了するとすぐにセットされます。FTF は、データが読み出された直後にクリアされます。

自動ステータスポーリングモードでは、可変の遅延はサポートされていません。したがって、メモリは固定遅延に設定する必要があります。

23.4.11 OCTOSPI メモリマップドモード

メモリマップドモードに設定された場合、外部 SPI デバイスは内部メモリとみなされます。

注： 外部デバイスの容量がもっと大きい場合でも、アドレス指定できるのは 256 M バイトまでです。

DEVSIZE[4:0] で規定されている範囲外のアドレスにアクセスすると、256MB の範囲内であっても、AHB エラーが発生します。このエラーの影響は、アクセスを試みた AHB マスタに依存します。

- Cortex CPU の場合は、ハードフォールト割込みが発生します。
- DMA の場合は、DMA 転送エラーが発生し、対応する DMA チャンネルが自動的に無効になります。

バイト、ハーフワード、およびワードアクセスはすべてサポートされます。

XIP (Execute In Place) 動作のサポートが実装されています。これにより、OCTOSPI は最も最近にアクセスしたアドレスに続くアドレスに継続してバイトをロードします。続くバイトに次のアクセスが連続している場合、それらの動作結果はプリフェッチされているので、その動作は迅速に完了します。

デフォルトでは、OCTOSPI はプリフェッチ動作を停止することはありません。外部デバイスへのアクセスが長時間なくても、NCS をローに保持したまま前の読出し動作をアクティブな状態に保つか、または新たな転送を再起動します。

NCS がローのときに外部デバイスはより多く電力を消費する傾向があるため、アプリケーションがタイムアウトカウンタの起動 (OCTOSPI_CR において TCEN = 1) を要求する場合があります。クロックが非アクティブになってからアクセスのない状態で x サイクルが経過すると、OCTOSPI_LPTR の TIMEOUT[15:0] で定義された期間の後、NCS が解除されます。

最初のメモリマップドアクセスが発生するとすぐに、BUSY はハイになります。プリフェッチ動作の影響で、BUSY の立下がりには、アバートが検出される、またはペリフェラルが無効化されるまで起こりません。

メモリマップド書込みを使用して Flash メモリをプログラムすることは、お勧めできません。インダイレクト書込みモードがこの動作を実現します。

23.4.12 OCTOSPI の設定の概要

OCTOSPI の設定は次の 3 つのステップで行われます。

1. OCTOSPI のシステム設定
2. OCTOSPI のデバイス設定
3. OCTOSPI のモード設定

23.4.13 OCTOSPI のシステム設定

OCTOSPI は OCTOSPI_CR を使用して設定します。ユーザは下記をプログラムする必要があります。

- FMODE[1:0] による機能モード
- 自動ステータスポーリングモードの動作 (PMM および APMS で必要な場合)
- FTHRES による FIFO レベル
- DMAEN による DMA の使用
- TCEN によるタイムアウトカウンタの使用
- デュアルメモリ構成 (必要な場合、DMM によって)

割込みを使用する場合、それぞれのイネーブルビットもこのフェーズ中にセットすることができます。

タイムアウトカウンタを使用する場合、タイムアウト値を OCTOSPI_LPTR にプログラムします。

DMA チャンネルは OCTOSPI の設定中に有効にしてはなりません。予期しないリクエストの生成を避けるため、動作を完全に設定した後にのみ有効にする必要があります。

DMA と OCTOSPI はデータ長に関して首尾一貫して設定する必要があります。FTHRES の値は DMA のバーストサイズを反映する必要があります。

23.4.14 OCTOSPI のデバイス設定

対象の外部デバイスに関連するパラメータは OCTOSPI_DCR1 と OCTOSPI_DCR2 によって設定されます。ユーザは下記をプログラムする必要があります。

- DEVSIZ[4:0] によるデバイスサイズ
- CSHT[5:0] によるチップセレクト最小ハイ時間
- FRCK と CKMODE によるクロックモード
- PRESCALER[7:0] によるデバイス周波数

MTYP[2:0] では、8ラインモードで使用されるメモリタイプを指定します。

- 8データビットモード (DMODE[2:0] = 100) で D0/D1 順序の Micron モード
- 8 データビットモード (DMODE[2:0] = 100) で D1/D0 順序の Macronix モード。
MTYP[2:0] = 001 は OctaFlash メモリを対象とし、MTYP[2:0] = 011 は特定のアドレスフェーズを持つ OctaRAM™ メモリを対象とします (アドレスは Macronix 要件に合うように行と列で構築します)。
- HyperBus メモリモード: プロトコルは HyperBus 仕様に準拠します。
- レジスタ空間をアドレス指定する HyperBus レジスタモード: このモードでのメモリマップドアクセスをキャッシュ不可能にするか、またはインダイレクト読み出し/書き込みモードを使用する必要があります。

DEVSIZ[4:0] は、次の式を使用して外部メモリのサイズを定義します。

$$\text{デバイス内のバイト数} = 2^{\text{DEVSIZ}+1}$$

ここで、DEVSIZ+1は外部デバイスのアドレスを指定するために必要なアドレスビット数です。外部デバイスの容量はインダイレクトモードでは最大 4 GB (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 MB に制限されます。

DMM = 1 の場合、DEVSIZ[4:0] は 2つのデバイスの総容量を示します。

OCTOSPI が 2 つのコマンドを実行する場合、一方のコマンドはもう一方のコマンドの直後に実行され、デフォルトでは 2 つのコマンドの間に 1 CLK サイクルの間だけハイレベルのチップセレクト信号 (NCS) を立ち上げます。

外部デバイスがコマンド間により長い時間を必要とする場合、チップセレクトハイ時間を示す CSHT[5:0] を使用して、NCS がハイに保たなければならない最小の CLK サイクル数を指定することができます。

CKMODE は、コマンド間での CLK のレベルを示します (NCS = 1 のとき)。

HyperBus プロトコルでは、デバイスのタイミング (t_{ACC} と t_{RWR}) および遅延モードを OCTOSPI_HLCR に設定する必要があります。

23.4.15 OCTOSPI レギュラコマンドモードの設定

インダイレクトモードの設定

FMODE[1:0] = 00 の場合、インダイレクト書込みモードが選択され、データを外部デバイスに送信することができます。FMODE[1:0] = 01 の場合、インダイレクト読出しモードが選択され、データを外部デバイスから読み出すことができます。

OCTOSPI がインダイレクトモードで使用される場合、フレームは次のように構成されます。

1. OCTOSPI_DLR に読み出される／書き込まれるデータのバイト数を指定します。
2. OCTOSPI_TCR に、フレームのタイミングを指定します。
3. OCTOSPI_CCR に、フレームフォーマットを指定します。
4. OCTOSPI_IR に、命令を指定します。
5. OCTOSPI_ABR に、アドレスフェーズの直後に送信されるオプションのオルタネートバイトを指定します。
6. OCTOSPI_AR に、ターゲットアドレスを指定します。
7. 必要に応じて DMA チャンネルを有効にします。
8. OCTOSPI_DR により FIFO に対してデータの読出し／書込みを行います (DMA を使用しない場合)。

アドレスレジスタ (OCTOSPI_AR) とデータレジスタ (OCTOSPI_DR) のどちらも特定のコマンドに関して更新する必要がない場合、OCTOSPI_IR が書き込まれるとすぐにコマンドシーケンスが開始されます。これは、ADMODE[2:0] と DMODE[2:0] がどちらも 000 である場合、またはインダイレクト読出しモード (FMODE[1:0] = 01) で ADMODE[2:0] = 000 の場合に該当します。

アドレスを必要とする場合 (ADMODE[2:0] ≠ 000) で、データレジスタに書き込む必要がない場合 (FMODE[1:0] = 01 または DMODE[2:0] = 000)、OCTOSPI_AR が書き込まれてアドレスが更新されるとすぐにコマンドシーケンスが開始されます。

データ転送時 (FMODE[1:0] = 00 かつ DMODE[2:0] ≠ 000)、OCTOSPI_DR による FIFO への書込みによって通信の開始がトリガされます。

自動ステータスポーリングモードの設定

自動ステータスポーリングモードは、FMODE[1:0] = 10 にセットすることで有効になります。このモードでは、プログラムされたフレームが送信され、定期的にデータが取得されます。

各フレームに読み込まれるデータの最大量は 4 バイトです。OCTOSPI_DLR でもっと大量のデータが要求されても、そのリクエストは無視され、4 バイトだけが読み出されます。OCTOSPI_PIR では周期が指定されます。

ステータスデータが取得されると、以下を処理できます。

- SMF をセットする (有効になっていれば割込みを生成する)。
- ステータスバイトの周期的な取得を自動的に停止する。

受信した値は OCTOSPI_PSMKR に格納されている値でマスクし、OCTOSPI_PSMAR に格納されている値で論理和または論理積をとることができます。

一致する場合、SMF がセットされ、有効になっていれば割込みが生成されます。AMPS がセットされている場合、OCTOSPI を自動的に停止することができます。いずれにしろ、一番最後に取得した値は OCTOSPI_DR で使用可能です。

OCTOSPI が自動ステータスポーリングモードで使用される場合、フレームは次のように構成されます。

1. OCTOSPI_PSMKR に、入力マスクを指定します。
2. OCTOSPI_PSMAR に、比較値を指定します。

3. OCTOSPI_PIR に、読出し周期を指定します。
4. OCTOSPI_DLR に、読み出されるデータのバイト数を指定します。
5. OCTOSPI_TCR に、フレームのタイミングを指定します。
6. OCTOSPI_CCR に、フレームフォーマットを指定します。
7. OCTOSPI_IR に、命令を指定します。
8. OCTOSPI_ABR に、アドレスフェーズの直後に送信されるオプションのオルタネートバイトを指定します。
9. OCTOSPI_AR に、オプションのターゲットアドレスを指定します。

アドレスレジスタ (OCTOSPI_AR) が特定のコマンドに関して更新する必要がない場合、OCTOSPI_CCR が書き込まれるとすぐにコマンドシーケンスが開始されます。これは ADMODE[2:0] = 000 の場合に該当します。

アドレスを必要とする場合 (ADMODE[2:0] ≠ 000)、OCTOSPI_AR が書き込まれてアドレスが更新されるとすぐにコマンドシーケンスが開始されます。

メモリマップドモードの設定

メモリマップドモードでは、外部デバイスは内部メモリとみなされますが、アクセス中にいくらかの遅延があります。このモードでは外部デバイスに対して読出しおよび書込み動作が許可されます。

メモリマップド書込みを使用して Flash メモリをプログラムすることは、消去やプログラム・ステータスの内部フラグをポーリングする必要があるため、お勧めできません。インダイレクト書込みモードでは、場合によっては自動ステータスポーリングモードと組み合わせて、この動作を実現します。

メモリマップドモードに移行するには、OCTOSPI_CR において FMODE[1:0] = 11 にセットします。

プログラムされた命令およびフレームは、AHB マスタがメモリマップド空間にアクセスするときに送信されます。

FIFO は、任意のリニア読出しを予測するプリフェッチバッファとして使用されます。このモードでの OCTOSPI_DR へのアクセスはすべてゼロで返されます。

メモリマップドモードでは、データ長レジスタ (OCTOSPI_DLR) は意味を持ちません。

OCTOSPI がメモリマップドモードで使用される場合、フレームは次のように構成されます。

1. 読出し動作に対して、OCTOSPI_TCR にフレームのタイミングを指定します。
2. 読出し動作に対して、OCTOSPI_CCR にフレームフォーマットを指定します。
3. OCTOSPI_IR に、命令を指定します。
4. 読出し動作に対して、OCTOSPI_ABR にアドレスフェーズの直後に送信されるオルタネートバイト (オプション) を指定します。
5. 書込み動作に対して、OCTOSPI_WTCR にフレームのタイミングを指定します。
6. 書込み動作に対して、OCTOSPI_WCCR にフレームフォーマットを指定します。
7. OCTOSPI_WIR に、命令を指定します。
8. 書込み動作に対して、OCTOSPI_WABR にアドレスフェーズの直後に送信されるオルタネートバイト (オプション) を指定します。

すべての設定操作は、メモリ領域への最初のアクセスの前に完了する必要があります (BUSY = 0 をチェックすることで保証されます)。BUSY = 1 のときのレジスタ書込み操作は効果がなく、エラー応答は通知されません。最初のアクセスで OCTOSPI はビジーになり、それ以上の設定は許可されません。この場合、BUSY をローにする唯一の方法は、ENABLE ビットをクリアするか、ABORT ビットをセットして中止することです。

DQS を使用しない場合の OCTOSPI 遅延データサンプリング

デフォルトでは、DQS を使用しない場合、外部デバイスに駆動されているデータを OCTOSPI がサンプリングするタイミングは、外部デバイスが信号を駆動してから 1/2 CLK サイクル後です。

外部信号の遅延が発生した場合、それは後でデータをサンプリングする際に有用となるかもしれませんが、OCTOSPI_TCR の SSHIFT を使用することで、データのサンプリングを 1/2 CLK サイクルだけシフトすることができます。

データフェーズが DTR モード (DDTR = 1) に設定されている場合、ファームウェアは SSHIFT をクリアする必要があります。

DQS を使用する場合の OCTOSPI 遅延データサンプリング

外部 DQS をサンプリングクロックとして使用する場合、データ伝播遅延を補正するためにその時間をシフトすることができます。このシフトは、OCTOSPI の外側に配置された外部遅延ブロックによって行われます。この機能の制御は、デバイスの実装により異なります (詳細については製品リファレンスマニュアルを参照)。

遅延を補償する必要がない構成では、OCTOSPI_DCR1 で DLYBYP をセットすることで外部遅延ブロックをバイパスできます。

23.4.16 OCTOSPI HyperBus プロトコルの設定

インダイレクトモードの設定 (HyperBus)

FMODE[1:0] = 00 の場合、インダイレクト書き込みモードが選択され、データを外部デバイスに送信することができます。FMODE[1:0] = 01 の場合、データを外部デバイスから読み出すことができるインダイレクト読み出しモードが選択されます。ADMODE は 000 と異なる値 (たとえば、ADMODE = 100 など) で設定する必要があります。

OCTOSPI がインダイレクトモードで使用される場合、フレームは次のように構成されます。

1. OCTOSPI_DLR に読み出される／書き込まれるデータのバイト数を指定します。
2. OCTOSPI_AR に、ターゲットアドレスを指定します。
3. 必要に応じて DMA チャンネルを有効にします。
4. OCTOSPI_DR により FIFO に対してデータの読み出し／書き込みを行います (DMA を使用しない場合)。

インダイレクト読み出しモードでは、OCTOSPI_AR が書き込まれてアドレスが更新されるとすぐにコマンドシーケンスが開始されます。

インダイレクト書き込みモードでは、OCTOSPI_DR による FIFO への書き込みによって通信の開始がトリガされます。

メモリマップドモードの設定 (HyperBus)

メモリマップドモードでは、外部デバイスは内部メモリとみなされますが、アクセス中にいくらかの遅延があります。このモードでは外部デバイスに対して読み出しおよび書き込み動作が許可されます。

メモリマップド書き込みを使用して Flash メモリをプログラムすることは、お勧めできません。インダイレクト書き込みモードがこの動作を実現します。

メモリマップドモードに移行するには、FMODE[1:0] = 11 にセットします。プログラムされた命令およびフレームは、AHB マスタがメモリマップド空間にアクセスするときに送信されます。

FIFO は、任意のリニア読み出しを予測するプリフェッチバッファとして使用されます。このモードでの OCTOSPI_DR へのアクセスはすべてゼロで返されます。

メモリマップドモードでは、データ長レジスタ (OCTOSPI_DLR) は意味を持ちません。

メモリ領域へ最初にアクセスする前に、すべての設定操作を完了する必要があります。最初のアクセスで OCTOSPI はビジーになり、いかなる設定も許可されません。この場合、BUSY をローにする唯一の方法は、ENABLE ビットをクリアするか、ABORT ビットをセットして中止することです。

23.4.17 OCTOSPI エラー管理

次の場合に、エラーが生成される可能性があります。

- インダイレクトまたは自動ステータスポーリングモードでは、QUADSPI_AR に間違ったアドレスがプログラムされている場合 (DEVSIZE[4:0] で定義されているデバイスのサイズに照らし)、TEF がセットされ、有効になっていれば割込みが生成されます。
- インダイレクトモードでは、アドレスにデータ長を加えた値がデバイスのサイズを超えている場合、アクセスがトリガされるとすぐに TEF がセットされます。
- メモリマップドモードでは、AHB マスタによる範囲外のアクセスが行われた場合、誤った AHB リクエストへの応答として AHB エラーが生成されます。
- メモリマップドモードが無効になっているときに、メモリマップド領域にアクセスすると、誤った AHB リクエストへの応答として AHB エラーが生成されます。

OCTOSPI は、次の状況で AHB スレーブエラーを発生します。

- メモリマップドモードが無効になっていて、AHB 読出しリクエストが発生した場合。
- 読出しまたは書込みアドレスが、外部メモリのサイズを超過した場合。
- 読出しまたは書込みバーストの進行中に、アボートを受信した場合。
- 読出しまたは書込みバーストの進行中に、OCTOSPI が無効になった場合。
- 書込みラップバーストを受信した場合。
- オクタル DTR モードまたはデュアルメモリ構成の OCTOSPI_WCCR で DQSE = 0 の間に書込みリクエストが受信された場合。
- MTYPE[2:0] が HyperBus の場合を除き、DMODE[2:0] = 000 (データフェーズなし) のときに書込みリクエストを受信した場合。
- ラップ読出しバースト時にアクセスサイズが不正である場合。これは、HSIZE が 4 バイトとは異なることを意味します (メモリマップドモードの場合のみ)。
- 4 バイトと異なるサイズの読出しラップバーストを受信した時のラップサイズが不正である場合 (メモリマップドモードの場合のみ)。

23.4.18 OCTOSPI の BUSY およびアボート

OCTOSPI が外部デバイスでの動作を開始すると、BUSY が OCTOSPI_SR に自動的にセットされます。

インダイレクトモードでは、OCTOSPI が要求されたコマンドシーケンスを完了し、FIFO が空になると、BUSY がリセットされます。

自動ステータスポーリングモードでは、一致 (APMS = 1 の場合) またはアボートの検出により、最後の周期的なアクセスが完了して初めて BUSY ビットはローになります。

メモリマップドモードでの最初のアクセス後、アボートの検出でのみ、BUSY はローになります。

OCTOSPI_CR の ABORT をセットすることによって、いかなる動作も中止することができます。アボートが完了すると、BUSY および ABORT は自動的にリセットされ、FIFO が一掃されます。

ABORT を設定する前に、ソフトウェアは同期バリアを使用して現在のすべてのトランザクションが終了していることを確認する必要があります。OCTOSPI_DR でデータの読取りまたは書込み動作を

処理するために DMA が有効になっている場合は、OCTOSPI を中止する前に DMA チャンルを無効にすることをお勧めします。

注： デバイスには、ステータスレジスタへの書き込み動作が中止された場合に誤動作するものもあります。

23.4.19 OCTOSPI の再設定または無効化

OCTOSPI の再設定の前に、ソフトウェアはすべてのトランザクションが完了していることを確認する必要があります。

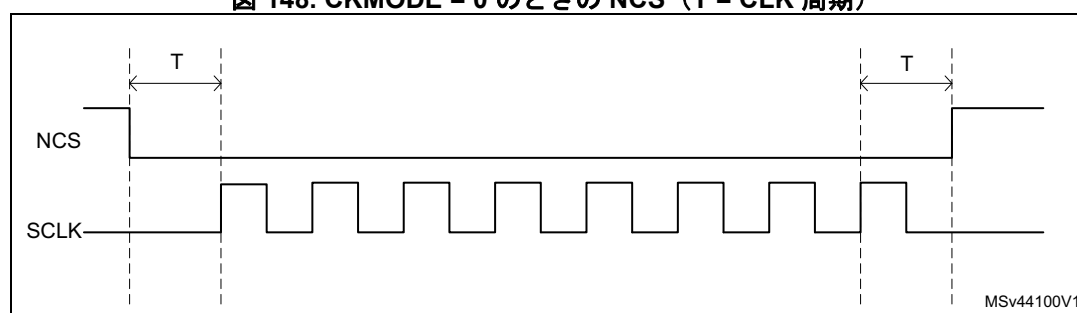
- メモリマップド書き込みの後、ソフトウェアはダミーリードに続いて同期バリア、その後に ABORT を実行する必要があります。
- メモリマップド読出しの後、ソフトウェアは同期バリア、その後に ABORT を実行する必要があります。

23.4.20 NCS の動作

デフォルトでは NCS ハイで、外部デバイスは選択解除されています。NCS の立下がり動作が始まる前に、立上がりは動作終了の直後に起こります。

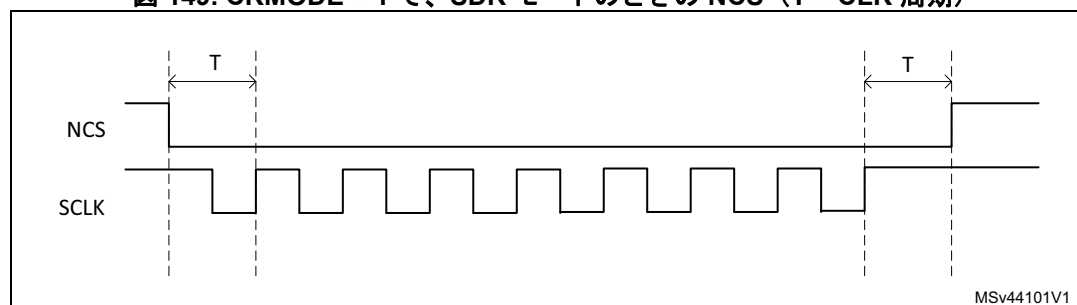
CKMODE = 0（クロックモード 0：進行中の動作がない場合は CLK がローに保持される）のとき、NCS の立下がり動作の最初の CLK エッジ立上がりの 1 CLK サイクル前に、NCS の立上がりは動作の最後の CLK エッジ立上がりから 1 CLK サイクル後に起こります（下図参照）。

図 148. CKMODE = 0 のときの NCS (T = CLK 周期)



CKMODE = 1（クロックモード 3：進行中の動作がない場合は CLK がハイになる）、かつ SDR モードのとき、NCS の立下がり動作の最初の CLK エッジ立上がりの 1 CLK サイクル前に、NCS の立上がりは動作の最後の CLK エッジ立上がりから 1 CLK サイクル後に起こります（下図参照）。

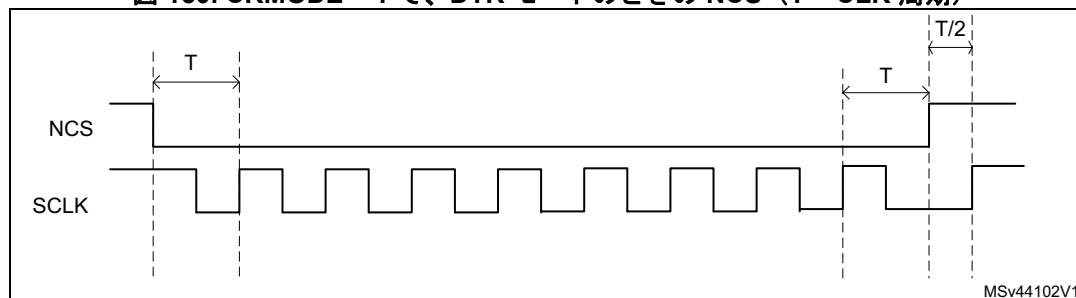
図 149. CKMODE = 1 で、SDR モードのときの NCS (T = CLK 周期)



CKMODE = 1（クロックモード 3）、かつ DDTR = 1（データ DTR モード）のとき、NCS の立下がり動作の最初の CLK エッジ立上がりの 1 CLK サイクル前に、NCS の立上がりは動作の最後のアク

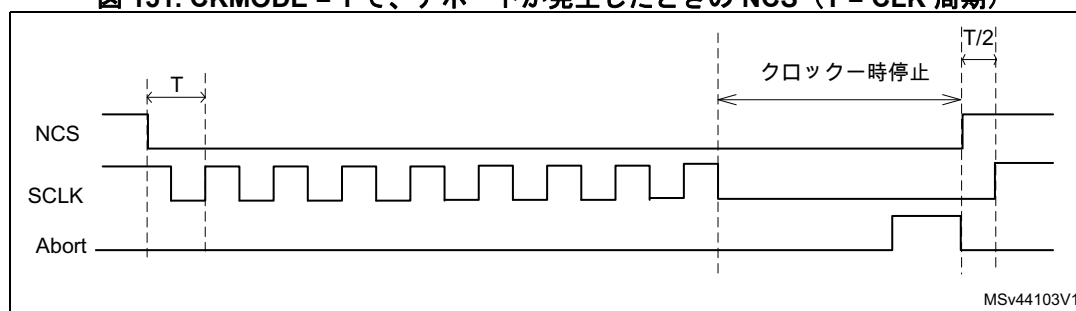
ティブ CLK エッジ立ち上がりから 1 CLK サイクル後に起こります（下図参照）。DTR 動作は立下がりエッジで終わらなければならないため、NCS が立ち上がる時 CLK はローであり、その後 CLK は 1/2 CLK サイクル後に立ち上がり直します。

図 150. CKMODE = 1 で、DTR モードのときの NCS (T = CLK 周期)



読出し動作中で FIFO がフルのまま、または書込み動作中で FIFO が空のままであれば、ソフトウェアが FIFO を処理するまでその動作は停止し、CLK はローのままとなります。動作が停止した時点でアボートが発生した場合、アボートが要求された直後に、NCS の立ち上がりが起こり、CLK は 1/2 CLK サイクル後に立ち上がり直します（下図参照）。

図 151. CKMODE = 1 で、アボートが発生したときの NCS (T = CLK 周期)



23.5 アドレスのアライメントとデータ数

以下の表は、使用例に応じたアドレスアラインメントとプログラムされたデータ数への影響をまとめたものです。

表 207. アドレスアライメントのケース

メモリアイプ	トランザクションタイプ	アドレスの制約 ⁽¹⁾	アドレスの制約不遵守の場合の影響	バイト数の制約 ⁽¹⁾	バイト数の制約不遵守の場合の影響
シングル、デュアル、クワッド Flash または SRAM (DMM = 0)	IND ⁽²⁾ 読出し	なし	なし	なし	なし
	MM ⁽³⁾ 読出し				
	IND 書込み				
	MM 書込み				

表 207. アドレスアライメントのケース (続き)

メモリタイプ	トランザクションタイプ	アドレスの制約 ⁽¹⁾	アドレスの制約不遵守の場合の影響	バイト数の制約 ⁽¹⁾	バイト数の制約不遵守の場合の影響
シングル、デュアル、クワッド Flash または SRAM (DMM = 1)	INDの読出し	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MMの読出し	なし	なし	なし	なし
	IND 書込み	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MM 書込み	偶数	スレーブエラー	偶数	最後のバイトが失われます。
SDR モード の オクタル Flash	INDの読出し	なし	なし	なし	なし
	MMの読出し				
	IND 書込み				
	MM 書込み				
DTR モードで WDM なしのオクタルメモリ ⁽⁶⁾	INDの読出し	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MMの読出し	なし	なし	なし	なし
	IND 書込み	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MM 書込み	偶数	スレーブエラー	偶数	最後のバイトが失われます。
DTR モードで WDM ありのオクタル Flash または RAM	INDの読出し	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MMの読出し	なし	なし	なし	なし
	IND 書込み				
	MM 書込み				
HyperBus	INDの読出し	偶数	ADDR[0] は 0 にセットされます。 ⁽⁴⁾	偶数	DLR[0] は 1 にセットされます。 ⁽⁵⁾
	MMの読出し	なし	なし	なし	なし
	IND 書込み				
	MM 書込み				

1. ソフトウェアで遵守すること。
2. IND = インダイレクトモード
3. MM = メモリマップドモード
4. 転送開始時の追加データ
5. 転送終了時の追加データ
6. WDM = 書込みデータマスク

23.6 OCTOSPI 割込み

割込みは次のイベントによって生成できます。

- タイムアウト
- ステータス一致
- FIFO 閾値
- 転送完了
- 転送エラー

より高い柔軟性を実現するため、個別の割込みイネーブルビットを使用できます。

表 208. OCTOSPI の割込みリクエスト

割込みイベント	イベントフラグ	有効制御ビット
タイムアウト	TOF	TOIE
ステータス一致	SMF	SMIE
FIFO 閾値	FTF	FTIE
転送完了	TCF	TCIE
転送エラー	TEF	TEIE

23.7 OCTOSPI レジスタ

23.7.1 OCTOSPI 制御レジスタ (OCTOSPI_CR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FMODE[1:0]		Res.	Res.	Res.	Res.	PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
		rw	rw					rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTHRES[4:0]					MSEL	DMM	Res.	Res.	TCEN	DMAEN	ABORT	EN
			rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **FMODE[1:0]** : 機能モード

このビットフィールドは、OCTOSPI 機能の動作モードを定義します。

00 : インダイレクト書込みモード

01 : インダイレクト読出しモード

10 : 自動ステータスポーリングモード (レギュラコマンドプロトコルにのみ該当)

11 : メモリマップドモード

すでに DMAEN = 1 の場合、対応するチャネル用の DMA コントローラは FMODE[1:0] 値を変更する前に無効にする必要があります。DMAEN = 1 のときに FMODE[1:0] と FTHRES[4:0] が誤って更新されると、DMA リクエスト信号は自動的に非アクティブ状態になります。

注 : このビットフィールドは、**BUSY が 0 のときにのみ修正できます。**

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PMM** : ポーリング一致モード

このビットは、自動ステータスポーリングモードの時に「一致」を判断するために使用する必要のある方法を示します。

0 : AND 一致モード。SMF は、デバイスから受信したマスクされていないすべてのビットが一致レジスタの対応するビットと一致している場合に、セットされます。

1 : OR 一致モード。SMF は、デバイスから受信したマスクされていないビットのいずれかが一致レジスタの対応するビットと一致している場合に、セットされます。

注： このビットは、**BUSY** が 0 のときにのみ修正できます。

ビット 22 **APMS** : 自動ステータスポーリングモード停止

このビットは、一致後に自動ステータスポーリングモードを停止するかどうかを決定します。

0 : 自動ステータスポーリングモードは、アボートまたは OCTOSPI の無効化によってのみ停止されます。

1 : 自動ステータスポーリングモードは、一致が検出されるとすぐに停止されます。

注： このビットは、**BUSY** が 0 のときにのみ修正できます。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **TOIE** : タイムアウト割込み有効化

このビットは、タイムアウト割込みを有効にします。

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 19 **SMIE** : ステータス一致割込み有効化

このビットは、ステータス一致割込みを有効にします。

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 18 **FTIE** : FIFO 閾値割込み有効

このビットは、FIFO 閾値割込みを有効にします。

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 17 **TCIE** : 転送完了割込み有効化

このビットは、転送完了割込みを有効にします。

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 16 **TEIE** : 転送エラー割込み有効化

このビットは、転送エラー割込みを有効にします。

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **FTHRES[4:0]** : FIFO 閾値レベル

このビットフィールドでは、インダイレクトモードで、OCTOSPI_SR の FIFO 閾値フラグを示す FTF のセットを引き起こす FIFO 内のバイトの閾値数を定義します。

00000 : FTF は、インダイレクト書込みモードで、FIFO 内に書き込める空きバイトが 1 バイト以上ある場合、またはインダイレクト読出しモードで、FIFO から読み出せる有効なバイトが 1 バイト以上ある場合にセットされます。

00001 : FTF は、インダイレクト書込みモードで、FIFO 内に書き込める空きバイトが 2 バイト以上ある場合、またはインダイレクト読出しモードで、FIFO から読み出せる有効なバイトが 2 バイト以上ある場合にセットされます。

.....

11111 : FTF は、インダイレクト書込みモードで、FIFO 内に書き込める空きバイトが 32 バイトある場合、またはインダイレクト読出しモードで、FIFO から読み出せる有効なバイトが 32 バイトある場合にセットされます。

注： **DMAEN = 1** の場合、対応するチャネル用の DMA コントローラは FTHRES[4:0] 値を変更する前に無効にする必要があります。

ビット 7 **MSEL** : 外部メモリ選択

このビットは、シングルメモリ構成 (DMM = 0) において Single-、Dual-、Quad-SPI モードでアドレス指定する外部メモリを選択します。

0 : 外部メモリ 1 が選択されます (データは IO[3:0] を経由して交換される)。

1 : 外部メモリ 2 が選択されます (データは IO[7:4] を経由して交換される)。

このビットは、DMM = 1 のとき、または Octal-SPI モードが選択されたときは、無視されます。

ビット 6 **DMM** : デュアルメモリ構成

このビットはデュアルメモリ構成をアクティブにします。このモードでは、2 つの外部デバイスを同時に使用して、スループットおよび容量を 2 倍にします。

0 : デュアルメモリ構成は無効です。

1 : デュアルメモリ構成は有効です。

注： このビットは、BUSY が 0 のときにのみ修正できます。

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TCEN** : タイムアウトカウンタ有効化

このビットは、メモリマップドモードが選択されている (FMODE[1:0] = 11) 場合にのみ有効です。このビットは、タイムアウトカウンタを有効にします。

0 : タイムアウトカウンタは無効です。したがって、チップセレクト (NCS) はメモリマップドモードにおける 1 回のアクセス後は無限にアクティブなままです。

1 : タイムアウトカウンタは有効です。したがって、チップセレクトは、メモリマップドモードで外部デバイスが非アクティブになってから TIMEOUT[15:0] サイクル後に解除されます。

注： このビットは、BUSY が 0 のときにのみ修正できます。

ビット 2 **DMAEN** : DMA 有効化

インダイレクトモードで、DMA を使用して、OCTOSPI_DR 経由でデータを入出力することができます。FTF がセットされた時点で DMA 転送が開始されます。

0 : インダイレクトモードに対して DMA は無効です。

1 : インダイレクトモードに対して DMA は有効です。

注： DMA 転送が進行中に DMAEN ビットをリセットすると、DMA とのハンドシェイクが切れます。DMA の動作中に、このビットを書き込まないでください。

ビット 1 **ABORT** : アボートリクエスト

このビットは進行中のコマンドシーケンスを中止します。アボートが完了すると自動的にリセットされます。このビットは現在の転送を中止させます。

0 : アボートは要求されていません。

1 : アボートが要求されています。

注： このビットは常に 0 として読み出されます。

ビット 0 **EN** : イネーブル

このビットは、OCTOSPI を有効にします。

0 : OCTOSPI は無効です。

1 : OCTOSPI は有効です。

注 : 動作中にこの EN ビットがクリアされた場合、ACK を受信せずに DMA リクエストを中止できます。
DMA 転送中にこのビットが 0 にセットされた場合、DMA への REQ 信号は、DMA からの ACK 信号がアクティブになるのを待たずに非アクティブ状態に戻ります。

23.7.2 OCTOSPI デバイス設定レジスタ 1 (OCTOSPI_DCR1)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	MTYP[2:0]			Res.	Res.	Res.	DEVSIZ[4:0]				
					rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CSHT[5:0]						Res.	Res.	Res.	Res.	DLY BYP	Res.	FRCK	CKMO DE
		rw	rw	rw	rw	rw	rw					rw		rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **MTYP[2:0]** : メモリタイプ

このビットフィールドは、サポートするメモリのタイプを示します。

000 : DTR 8 データビットモードで D0/D1 順序の Micron モード。Single-、Dual-、Quad-、および Octal-SPI モードでのレギュラコマンドプロトコル。

注 : このモードでは、DQS 信号極性がメモリクロック信号に対して反転されます。これはデフォルト値であり、Micron 以外のメモリの MTYP[2:0] を変更する際は注意する必要があります。

001 : DTR 8 データビットモードで D1/D0 順序の Macronix モード。Single-、Dual-、Quad-、および Octal-SPI モードでのレギュラコマンドプロトコル。

010 : 標準モード

011 : DTR 8 データビットモードで D1/D0 順序の Macronix RAM モード。アドレスマッピング専用の Single-、Dual-、Quad-、および Octal-SPI モードでのレギュラコマンドプロトコル。

100 : HyperBus メモリモード。プロトコルは HyperBus 仕様に準拠。

101 : HyperBus レジスタモード。レジスタ空間にアドレス指定。このモードでメモリマップドアクセスを行う場合、キャッシュ不可能にするか、またはインダイレクト読み出し/書き込みモードを使用する必要があります。

その他 : 予約済み

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **DEVSIZ[4:0]** : デバイスサイズ

このビットフィールドは、次の式を使用して外部デバイスのサイズを定義します。

デバイス内のバイト数 = $2^{[DEVSIZ+1]}$ 。

DEVSIZ + 1 は外部デバイスのアドレスを指定するために必要とされる有効なアドレスビット数です。デバイスの容量はインダイレクトモードでは最大 4 G バイト (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 M バイト に制限されます。

レギュラコマンドプロトコルでは、DMM = 1 の場合、DEVSIZ[4:0] は 2 つの外部デバイスのうちの 2 つの容量を示します。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **CSHT[5:0]** : チップセレクトハイ時間

CSHT + 1 は、外部デバイスに対して発行されるコマンド間でチップセレクト (NCS) がハイに保たれなければならない CLK サイクルの最小数を定義します。

0x0 : NCS は、外部デバイスのコマンド間で少なくとも 1 サイクルの間はハイに保持されます。

0x1 : NCS は、外部デバイスのコマンド間で少なくとも 2 サイクルの間はハイに保持されます。

.....

0x3F : NCS は、外部デバイスのコマンド間で少なくとも 64 サイクルの間はハイに保持されます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **DLYBYP** : 遅延ブロックバイパス

0 : 内部サンプリングクロック (フィードバッククロック) または DQS 外部データストローブ信号は遅延ブロックによって遅延されます (このブロックは OCTOSPI ペリフェラルの一部ではないため、詳細については、リファレンスマニュアルの該当するセクションを参照してください)。

1 : 遅延ブロックはバイパスされるので、内部サンプリングクロックまたは DQS 外部データストローブ信号は遅延ブロックに影響されません。遅延は、遅延値を遅延ブロックで最小値に設定しても、遅延ブロックをバイパスしなかった場合より短くなります。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **FRCK** : フリーランニングクロック

このビットによってフリーランニングクロックを設定します。

0 : CLK はフリーランニングではありません。

1 : CLK はフリーランニングです (常に提供される)。

注 : フリーランニングクロックモードは遅延の校正のみを目的としています。FRCK がセットされているとき、メモリやその他のデバイスへのアクセスはできません。

ビット 0 **CKMODE** : クロックモード 0/モード 3

このビットは、コマンド間 (NCS = 1 のとき) に CLK が取るレベルを示します。

0 : CLK は NCS がハイ (チップセレクト解除) である間はローに保持される必要があります。これは、クロックモード 0 と呼ばれます。

1 : CLK は NCS がハイ (チップセレクト解除) である間はハイに保持される必要があります。これは、クロックモード 3 と呼ばれます。

23.7.3 OCTOSPI デバイス設定レジスタ 2 (OCTOSPI_DCR2)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRAPSIZE[2:0]		
													rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **WRAPSIZE[2:0]** : ラップサイズ

このビットフィールドは、メモリが設定される際のラップサイズを示します。ラップされた命令用の個別のコマンドを備えるメモリに対して、このビットフィールドは、OCTOSPI_WPIR レジスタに保持されたコマンドに関連付けられたラップサイズを示します。

000 : ラップド読み出しはこのメモリでサポートされていません。

010 : 外部メモリは 16 バイトのラップサイズをサポートしています。

011 : 外部メモリは 32 バイトのラップサイズをサポートしています。

100 : 外部メモリは 64 バイトのラップサイズをサポートしています。

101 : 外部メモリは 128 バイトのラップサイズをサポートしています。

その他 : 予約済み

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PRESCALER[7:0]** : クロックプリスケアラ

このビットフィールドは、カーネルクロックに基づいて CLK を生成するスケアラ係数 (値 + 1) を定義します。

0 : $F_{CLK} = F_{KERNEL}$ 、カーネルクロックが直接 OCTOSPI CLK (プリスケアラはバイパス) として使用されます。この場合、DTR モードを使用する場合は、デューティサイクル 50% のカーネルクロックを OCTOSPI に提供することが必須です。

1 : $F_{CLK} = F_{KERNEL}/2$

2 : $F_{CLK} = F_{KERNEL}/3$

.....

255 : $F_{CLK} = F_{KERNEL}/256$

クロック分周比が奇数の場合、CLK のデューティサイクルは 50% ではありません。クロック信号は、ハイに保持された時間より 1 サイクル分長く、ローに保持されます。

23.7.4 OCTOSPI デバイス設定レジスタ 3 (OCTOSPI_DCR3)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBOUND[4:0]				
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **CSBOUND[4:0]** : NCS 境界

このビットフィールドは、トランザクション境界機能を有効にします。有効な場合、最小値の 3 が推奨されます。 $2^{CSBOUND}$ バイトの各境界で NCS が解除されます。

0 : NCS 境界は無効です。

その他 : NCS 境界は、 $2^{CSBOUND}$ バイトに設定されます。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

23.7.5 OCTOSPI デバイス設定レジスタ 4 (OCTOSPI_DCR4)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFRESH[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFRESH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **REFRESH[31:0]** : リフレッシュレート

このビットフィールドは、リフレッシュレート機能を有効にします。NCS は、書込みの REFRESH + 1 クロックサイクルごと、および読出しの REFRESH + 4 クロックサイクルごとに解除されます。Single-、Dual-、または Quad-SPI モードで 1 つのバイト送信中にリフレッシュが発生した場合、そのバイト送信を完了させる必要があるため、これらの 2 つの値は数クロックサイクル拡張される可能性があります。

0 : リフレッシュは無効です。

その他 : 最大通信長は REFRESH + 1 クロックサイクルに設定されます。

注 : REFRESH カウントは分周されたクロック周期に基づいています。したがって、OCTOSPI_DCR2 PRESCALER ビットフィールドが変更された場合、それに応じて REFRESH フィールドを更新する必要があります。

23.7.6 OCTOSPI ステータスレジスタ (OCTOSPI_SR)

アドレスオフセット : 0x0020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
		r	r	r	r	r	r			r	r	r	r	r	r

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **FLEVEL[5:0]** : FIFO レベル

このビットフィールドは FIFO 内に保持されている有効バイト数を示します。FIFO が空の場合、FLEVEL = 0、フルになっている場合、32 です。

自動ステータスポーリングモードでは、FLEVEL はゼロです。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **BUSY** : ビジー

このビットは動作が進行中のときにセットされます。これは、外部デバイスの動作が終わり、FIFO が空になると自動的にクリアされます。

ビット 4 **TOF** : タイムアウトフラグ

このビットは、タイムアウトが発生するとセットされます。CTOF に 1 を書き込むことによってクリアされます。

ビット 3 **SMF** : ステータス一致フラグ

このビットは、マスクされていない受信データが一致レジスタ (OCTOSPI_PSMAR) の対応するビットと一致している場合に、自動ステータスポーリングモードでセットされます。
CSMF に 1 を書き込むことによってクリアされます。

ビット 2 **FTF** : FIFO 閾値フラグ

インダイレクトモードでは、このビットは FIFO 閾値に達した時点で、または外部デバイスからの読出し完了後に FIFO 内にデータが残っている場合にセットされます。
閾値条件が真でなくなるとすぐに自動的にクリアされます。
自動ステータスポーリングモードでは、このビットはステータスレジスタが読み出されるたびにセットされ、データレジスタが読み出された時点でクリアされます。

ビット 1 **TCF** : 転送完了フラグ

このビットは、プログラムされたデータ数が転送された時点でインダイレクトモードにセットされるか、または転送が中止された時点でいずれかのモードにセットされます。また、CTCF に 1 を書き込むことによってクリアされます。

ビット 0 **TEF** : 転送エラーフラグ

このビットは、インダイレクトモードで無効なアドレスにアクセスされているときに、インダイレクトモードでセットされます。
CTEF に 1 を書き込むことによってクリアされます。

23.7.7 OCTOSPI フラグクリアレジスタ (OCTOSPI_FCR)

アドレスオフセット : 0x0024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
											w	w		w	w

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CTOF** : タイムアウトフラグのクリア

このビットに 1 を書き込むと、OCTOSPI_SR レジスタの TOF フラグがクリアされます。

ビット 3 **CSMF** : ステータス一致フラグのクリア

このビットに 1 を書き込むと、OCTOSPI_SR レジスタの SMF フラグがクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CTCF** : 転送完了フラグのクリア

このビットに 1 を書き込むと、OCTOSPI_SR レジスタの TCF フラグがクリアされます。

ビット 0 **CTEF** : 転送エラーフラグのクリア

このビットに 1 を書き込むと、OCTOSPI_SR レジスタの TEF フラグがクリアされます。

23.7.8 OCTOSPI データ長レジスタ (OCTOSPI_DLR)

アドレスオフセット : 0x0040

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DL[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **DL[31:0]** : データ長

インダイレクトモードおよび自動ステータスポーリングモードで取得されるデータ数（値 + 1）自動ステータスポーリングモードの場合、3 より大きい値（4 バイトを示す）を使用する必要があります。

インダイレクトモードでのすべて 1 は、不定長を意味します。この場合、DEVSZ によって定義されているメモリの最後まで OCTOSPI が継続します。

0x0000_0000 : 1 バイトが転送されます。

0x0000_0001 : 2 バイトが転送されます。

0x0000_0002 : 3 バイトが転送されます。

0x0000_0003 : 4 バイトが転送されます。

.....

0xFFFF_FFFD : 4,294,967,294 (4G-2) バイトが転送されます。

0xFFFF_FFFE : 4,294,967,295 (4G-1) バイトが転送されます。

0xFFFF_FFFF : 不定長 -- すべてのバイトが外部デバイスの終了（DEVSZ によって定義）まで転送されます。DEVSZ = 0x1F の場合、読み出しが無期限に続けられます。

デュアルメモリ構成（DMM = 1）では、DL[0] は、たとえこのビットに 0 が書き込まれても 1 のままとなります。したがって、アクセスのたびに必ず偶数バイトが転送されます。

このビットフィールドはメモリマップドモードでは効果がありません。

23.7.9 OCTOSPI アドレスレジスタ (OCTOSPI_AR)

アドレスオフセット : 0x0048

リセット値 : 0x0000 0000

このレジスタは、BUSY = 0、かつ FMODE ≠ 11 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **ADDRESS[31:0]** : アドレス

外部デバイスに送信されるアドレス HyperBus プロトコルでは、16 ビットワード指向なので、このフィールドは偶数でなければなりません。デュアルメモリ構成では、AR[0] は 0 に強制されます。

23.7.10 OCTOSPI データレジスタ (OCTOSPI_DR)

アドレスオフセット : 0x0050

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 DATA[31:0] : データ

外部 SPI デバイスに対して送受信されるデータ

インダイレクト書込みモードでは、このレジスタに書き込まれたデータはいったん FIFO に格納されてから、データフェーズ時に外部デバイスに送信されます。FIFO がフルの場合、書き込まれるデータ量を受け入れるだけの十分な空間が FIFO にできるまで、書込み動作は停止されます。

インダイレクト読出しモードでは、このレジスタを読み出すことで (FIFO 経由)、外部デバイスから受信したデータを取得できます。FIFO 内のバイト数が読出し動作で要求したバイト数ほど多くない場合で、BUSY = 1 のとき、データ量が十分になるか、転送が完了するかのいずれか早く起こるほうまで、読出し動作は停止されます。

自動ステータスポーリングモードでは、このレジスタは外部デバイスから読み出された最後のデータを格納しています (マスキングなし)。

このレジスタへのワード、ハーフワード、バイトアクセスはサポートされます。インダイレクト書込みモードでの書込みにより FIFO に追加されるバイト数は、バイト書込みで 1 バイト、ハーフワード書込みで 2 バイト、ワード書込みで 4 バイトです。

同様に、インダイレクト読出しモードでの読出しにより FIFO から削除されるバイト数は、バイト読出しで 1 バイト、ハーフワード読出しで 2 バイト、ワード読出しで 4 バイトです。インダイレクトモードでのアクセスは、このレジスタの下位に合わせなければなりません。つまり、バイト読出しでは DATA[7:0] が、ハーフワード読出しでは DATA[15:0] が読み出されます。

23.7.11 OCTOSPI ポーリングステータスマスクレジスタ (OCTOSPI_PSMKR)

アドレスオフセット : 0x0080

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASK[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **MASK[31:0]** : ステータスマスク

自動ステータスポーリングモードで受信したステータスバイトに適用されるマスク

ビット n の場合

0 : 自動ステータスポーリングモードで受信したデータのビット n はマスクされ、その値は一致ロジックでは考慮されません。

1 : 自動ステータスポーリングモードで受信したデータのビット n はマスクされず、その値は一致ロジックで考慮されます。

23.7.12 OCTOSPI ポーリングステータス一致レジスタ (OCTOSPI_PSMAR)

アドレスオフセット : 0x0088

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MATCH[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **MATCH[31:0]** : ステータス一致

一致を得るためにマスクされたステータスレジスタと比較される値

23.7.13 OCTOSPI ポーリングインターバルレジスタ (OCTOSPI_PIR)

アドレスオフセット : 0x0090

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **INTERVAL[15:0]** : ポーリングインターバル

自動ステータスポーリングフェーズ時の読出し間の CLK サイクル数

23.7.14 OCTOSPI 通信設定レジスタ (OCTOSPI_CCR)

アドレスオフセット : 0x0100

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DQSE	Res.	DDTR	DMODE[2:0]			Res.	Res.	ABSIZE[1:0]		ABDTR	ABMODE[2:0]		
		rW		rW	rW	rW	rW			rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADSIZE[1:0]		AD DTR	ADMODE[2:0]			Res.	Res.	ISIZE[1:0]		IDTR	IMODE[2:0]		
		rW	rW	rW	rW	rW	rW			rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DQSE** : DQS 有効化

このビットは、データストロープ管理を有効にします。

0 : DQS は無効です。

1 : DQS は有効です。

ビット 28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **DDTR** : データダブル転送レート

このビットは、データフェーズの DTR モードをセットします。

0 : データフェーズの DTR モードは無効です。

1 : データフェーズの DTR モードは有効です。

ビット 26:24 **DMODE[2:0]** : データモード

このビットフィールドは動作のデータフェーズのモードを定義します。

000 : データはありません。

001 : データは 1 本のライン上にあります。

010 : データは 2 本のライン上にあります。

011 : データは 4 本のライン上にあります。

100 : データは 8 本のライン上にあります。

その他 : 予約済み

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **ABSIZE[1:0]** : オルタネートバイトサイズ

このビットフィールドはオルタネートバイトのサイズを定義します。

00 : 8 ビットオルタネートバイト

01 : 16 ビットオルタネートバイト

10 : 24 ビットオルタネートバイト

11 : 32 ビットオルタネートバイト

ビット 19 **ABDTR** : オルタネートバイトダブル転送レート

このビットは、オルタネートバイトフェーズの DTR モードを設定します。

0 : オルタネートバイトフェーズの DTR モードは無効です。

1 : オルタネートバイトフェーズの DTR モードは有効です。

ビット 18:16 **ABMODE[2:0]** : オルタネートバイトモード

このビットフィールドは、動作のオルタネートバイトフェーズのモードを定義します。

000 : オルタネートバイトはありません。

001 : オルタネートバイトは 1 本のライン上にあります。

010 : オルタネートバイトは 2 本のライン上にあります。

011 : オルタネートバイトは 4 本のライン上にあります。

100 : オルタネートバイトは 8 本のライン上にあります。

その他 : 予約済み

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 **ADSIZE[1:0]** : アドレスサイズ

このビットフィールドはアドレスサイズを定義します。

00 : 8 ビットアドレス

01 : 16 ビットアドレス

10 : 24 ビットアドレス

11 : 32 ビットアドレス

ビット 11 **ADDTR** : アドレスダブル転送レート

このビットは、アドレスフェーズの DTR モードをセットします。

0 : アドレスフェーズの DTR モードは無効です。

1 : アドレスフェーズの DTR モードは有効です。

ビット 10:8 **ADMODE[2:0]** : アドレスモード

このビットフィールドは動作のアドレスフェーズのモードを定義します。

000 : アドレスはありません。

001 : アドレスは 1 本のライン上にあります。

010 : アドレスは 2 本のライン上にあります。

011 : アドレスは 4 本のライン上にあります。

100 : アドレスは 8 本のライン上にあります。

その他 : 予約済み

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ISIZE[1:0]** : 命令サイズ

このビットフィールドは命令のサイズを定義します。

- 00 : 8 ビット命令
- 01 : 16 ビット命令
- 10 : 24 ビット命令
- 11 : 32 ビット命令

ビット 3 **IDTR** : 命令ダブル転送レート

このビットは、命令フェーズの DTR モードをセットします。

- 0 : 命令フェーズの DTR モードは無効です。
- 1 : 命令フェーズの DTR モードは有効です。

ビット 2:0 **IMODE[2:0]** : 命令モード

このビットフィールドは動作の命令フェーズのモードを定義します。

- 000 : 命令はありません。
- 001 : 命令は 1 本のライン上にあります。
- 010 : 命令は 2 本のライン上にあります。
- 011 : 命令は 4 本のライン上にあります。
- 100 : 命令は 8 本のライン上にあります。
- その他 : 予約済み

23.7.15 OCTOSPI タイミング設定レジスタ (OCTOSPI_TCR)

アドレスオフセット : 0x0108

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SSHIFT	Res.	DHQC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCYC[4:0]				
											rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SSHIFT** : サンプリングシフト

デフォルトでは、OCTOSPI は外部デバイスによるデータ駆動の 1/2 CLK サイクル後にデータをサンプリングします。

このビットを使用することで、外部信号遅延を考慮して、後でデータをサンプリングすることができます。

0 : シフトなし

1 : 1/2 サイクルのシフト

データフェーズが DTR モード (DDTR = 1) に設定されている場合、ソフトウェアは SSHIFT = 0 であることを確認する必要があります。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **DHQC** : 遅延ホールドクォータサイクル

0 : 遅延ホールドなし

1 : 1/4 サイクルのホールド

ビット 27:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DCYC[4:0]** : ダミーサイクル数

このビットフィールドは、メモリ待ち時間に応じたダミーフェーズの時間の長さを定義します。
SDR モードと DTR モードの両方で、CLK サイクルの数を指定します (0~31)。

23.7.16 OCTOSPI 命令レジスタ (OCTOSPI_IR)

アドレスオフセット : 0x0110

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **INSTRUCTION[31:0]** : 命令

外部 SPI デバイスに送信される命令

23.7.17 OCTOSPI オルタネートバイトレジスタ (OCTOSPI_ABR)

アドレスオフセット : 0x0120

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTERNATE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **ALTERNATE[31:0]** : オルタネートバイト

アドレスの直後に外部 SPI デバイスに送信されるオプションのデータ。

23.7.18 OCTOSPI 低電力タイムアウトレジスタ (OCTOSPI_LPTR)

アドレスオフセット : 0x00130

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TIMEOUT[15:0]** : タイムアウト期間

メモリマップドモードでのアクセス終了のたびに、OCTOSPI はその後のバイトをプリフェッチし、それらを FIFO に保持します。

このビットフィールドは、クロックが非アクティブになってから NCS を立ち上げるまでに OCTOSPI が待つ CLK サイクルの数を示し、外部デバイスを消費電力がより少ない状態にします。

23.7.19 OCTOSPI ラップ通信設定レジスタ (OCTOSPI_WPCCR)

アドレスオフセット : 0x0140

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DQSE	Res.	DDTR	DMODE[2:0]			Res.	Res.	ABSIZE[1:0]		ABDTR	ABMODE[2:0]		
		rW		rW	rW	rW	rW			rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADSIZE[1:0]		AD DTR	ADMODE[2:0]			Res.	Res.	ISIZE[1:0]		IDTR	IMODE[2:0]		
		rW	rW	rW	rW	rW	rW			rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DQSE** : DQS 有効化

このビットは、データストロブ管理を有効にします。

0 : DQS は無効です。

1 : DQS は有効です。

ビット 28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **DDTR** : データダブル転送レート

このビットは、データフェーズの DTR モードをセットします。

0 : データフェーズの DTR モードは無効です。

1 : データフェーズの DTR モードは有効です。

ビット 26:24 **DMODE[2:0]** : データモード

このビットフィールドは動作のデータフェーズのモードを定義します。

- 000 : データはありません。
- 001 : データは 1 本のライン上にあります。
- 010 : データは 2 本のライン上にあります。
- 011 : データは 4 本のライン上にあります。
- 100 : データは 8 本のライン上にあります。
- その他 : 予約済み

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **ABSIZE[1:0]** : オルタネートバイトサイズ

このビットフィールドはオルタネートバイトのサイズを定義します。

- 00 : 8 ビットオルタネートバイト
- 01 : 16 ビットオルタネートバイト
- 10 : 24 ビットオルタネートバイト
- 11 : 32 ビットオルタネートバイト

ビット 19 **ABDTR** : オルタネートバイトダブル転送レート

このビットは、オルタネートバイトフェーズの DTR モードを設定します。

- 0 : オルタネートバイトフェーズの DTR モードは無効です。
- 1 : オルタネートバイトフェーズの DTR モードは有効です。

ビット 18:16 **ABMODE[2:0]** : オルタネートバイトモード

このビットフィールドは、動作のオルタネートバイトフェーズのモードを定義します。

- 000 : オルタネートバイトなし
- 001 : オルタネートバイトは 1 本のライン上にあります。
- 010 : オルタネートバイトは 2 本のライン上にあります。
- 011 : オルタネートバイトは 4 本のライン上にあります。
- 100 : オルタネートバイトは 8 本のライン上にあります。
- その他 : 予約済み

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 **ADSIZE[1:0]** : アドレスサイズ

このビットフィールドはアドレスサイズを定義します。

- 00 : 8 ビットアドレス
- 01 : 16 ビットアドレス
- 10 : 24 ビットアドレス
- 11 : 32 ビットアドレス

ビット 11 **ADDTR** : アドレスダブル転送レート

このビットは、アドレスフェーズの DTR モードをセットします。

- 0 : アドレスフェーズの DTR モードは無効です。
- 1 : アドレスフェーズの DTR モードは有効です。

ビット 10:8 **ADMODE[2:0]** : アドレスモード

このビットフィールドは動作のアドレスフェーズのモードを定義します。

- 000 : アドレスはありません。
- 001 : アドレスは 1 本のライン上にあります。
- 010 : アドレスは 2 本のライン上にあります。
- 011 : アドレスは 4 本のライン上にあります。
- 100 : アドレスは 8 本のライン上にあります。
- その他 : 予約済み

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ISIZE[1:0]** : 命令サイズ

このビットフィールドは命令のサイズを定義します。

- 00 : 8 ビット命令
- 01 : 16 ビット命令
- 10 : 24 ビット命令
- 11 : 32 ビット命令

ビット 3 **IDTR** : 命令ダブル転送レート

このビットは、命令フェーズの DTR モードをセットします。

- 0 : 命令フェーズの DTR モードは無効です。
- 1 : 命令フェーズの DTR モードは有効です。

ビット 2:0 **IMODE[2:0]** : 命令モード

このビットフィールドは動作の命令フェーズのモードを定義します。

- 000 : 命令はありません。
- 001 : 命令は 1 本のライン上にあります。
- 010 : 命令は 2 本のライン上にあります。
- 011 : 命令は 4 本のライン上にあります。
- 100 : 命令は 8 本のライン上にあります。
- その他 : 予約済み

23.7.20 OCTOSPI ラップタイミング設定レジスタ (OCTOSPI_WPTCR)

アドレスオフセット : 0x0148

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	S SHIFT	Res.	DHQC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCYC[4:0]				
											rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SSHIFT** : サンプリングシフト

デフォルトでは、OCTOSPI は外部デバイスによるデータ駆動の 1/2 CLK サイクル後にデータをサンプリングします。

このビットを使用することで、外部信号遅延を考慮して、後でデータをサンプリングすることができます。

0 : シフトなし

1 : 1/2 サイクルのシフト

データフェーズが DTR モード (DDTR = 1) に設定されている場合、ファームウェアは SSHIFT = 0 であることを確実にする必要があります。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **DHQC** : 遅延ホールドクォータサイクル

ホールド要件を満たすために、DTR 通信の出力に 1/4 サイクルの遅延を追加します。

0 : 1/4 サイクル遅延なし

1 : 1/4 サイクルの遅延を挿入

ビット 27:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DCYC[4:0]** : ダミーサイクル数

このビットフィールドは、メモリ待ち時間に応じたダミーフェーズの時間の長さを定義します。
SDR モードと DTR モードの両方で、CLK サイクルの数を指定します (0~31)。

23.7.21 OCTOSPI ラップ命令レジスタ (OCTOSPI_WPIR)

アドレスオフセット : 0x0150

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **INSTRUCTION[31:0]** : 命令

外部 SPI デバイスに送信される命令

23.7.22 OCTOSPI ラップオルタネートバイトレジスタ (OCTOSPI_WPABR)

アドレスオフセット : 0x0160

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTERNATE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **ALTERNATE[31:0]** : オルタネートバイト

アドレスの直後に外部 SPI デバイスに送信されるオプションのデータ。

23.7.23 OCTOSPI 書込み通信設定レジスタ (OCTOSPI_WCCR)

アドレスオフセット : 0x0180

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。その内容は、メモリマップトモードで書き込み動作をリクエストする場合にのみ意味を持ちます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DQSE	Res.	DDTR	DMODE[2:0]			Res.	Res.	ABSIZE[1:0]		ABDTR	ABMODE[2:0]		
		rw		rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADSIZE[1:0]		ADDTR	ADMODE[2:0]			Res.	Res.	ISIZE[1:0]		IDTR	IMODE[2:0]		
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DQSE** : DQS 有効化

このビットは、データストロブ管理を有効にします。

0 : DQS は無効です。

1 : DQS は有効です。

ビット 28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **DDTR** : データダブル転送レート

このビットは、データフェーズの DTR モードをセットします。

0 : データフェーズの DTR モードは無効です。

1 : データフェーズの DTR モードは有効です。

ビット 26:24 **DMODE[2:0]** : データモード

このビットフィールドは動作のデータフェーズのモードを定義します。

000 : データはありません。

001 : データは 1 本のライン上にあります。

010 : データは 2 本のライン上にあります。

011 : データは 4 本のライン上にあります。

100 : データは 8 本のライン上にあります。

その他 : 予約済み

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **ABSIZE[1:0]** : オルタネートバイトサイズ

このビットフィールドはオルタネートバイトのサイズを定義します。

00 : 8 ビットオルタネートバイト

01 : 16 ビットオルタネートバイト

10 : 24 ビットオルタネートバイト

11 : 32 ビットオルタネートバイト

ビット 19 **ABDTR** : オルタネートバイトダブル転送レート

このビットは、オルタネートバイトフェーズの DTR モードを設定します。

0 : オルタネートバイトフェーズの DTR モードは無効です。

1 : オルタネートバイトフェーズの DTR モードは有効です。

ビット 18:16 ABMODE[2:0] : オルタネートバイトモード

このビットフィールドは、動作のオルタネートバイトフェーズのモードを定義します。

000 : オルタネートバイトはありません。

001 : オルタネートバイトは 1 本のライン上にあります。

010 : オルタネートバイトは 2 本のライン上にあります。

011 : オルタネートバイトは 4 本のライン上にあります。

100 : オルタネートバイトは 8 本のライン上にあります。

その他 : 予約済み

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 ADSIZE[1:0] : アドレスサイズ

このビットフィールドはアドレスサイズを定義します。

00 : 8 ビットアドレス

01 : 16 ビットアドレス

10 : 24 ビットアドレス

11 : 32 ビットアドレス

ビット 11 ADDTR : アドレスダブル転送レート

このビットは、アドレスフェーズの DTR モードをセットします。

0 : アドレスフェーズの DTR モードは無効です。

1 : アドレスフェーズの DTR モードは有効です。

ビット 10:8 ADMODE[2:0] : アドレスモード

このビットフィールドは動作のアドレスフェーズのモードを定義します。

000 : アドレスはありません。

001 : アドレスは 1 本のライン上にあります。

010 : アドレスは 2 本のライン上にあります。

011 : アドレスは 4 本のライン上にあります。

100 : アドレスは 8 本のライン上にあります。

その他 : 予約済み

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 ISIZE[1:0] : 命令サイズ

このビットフィールドは命令のサイズを定義します。

00 : 8 ビット命令

01 : 16 ビット命令

10 : 24 ビット命令

11 : 32 ビット命令

ビット 3 IDTR : 命令ダブル転送レート

このビットは、命令フェーズの DTR モードをセットします。

0 : 命令フェーズの DTR モードは無効です。

1 : 命令フェーズの DTR モードは有効です。

ビット 2:0 IMODE[2:0] : 命令モード

このビットフィールドは動作の命令フェーズのモードを定義します。

000 : 命令はありません。

001 : 命令は 1 本のライン上にあります。

010 : 命令は 2 本のライン上にあります。

011 : 命令は 4 本のライン上にあります。

100 : 命令は 8 本のライン上にあります。

その他 : 予約済み

23.7.24 OCTOSPI 書込みタイミング設定レジスタ (OCTOSPI_WTCR)

アドレスオフセット : 0x0188

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。その内容は、メモリマップトモードで書き込み動作をリクエストする場合にのみ意味を持ちます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCYC[4:0]				
											rW	rW	rW	rW	rW

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DCYC[4:0]** : ダミーサイクル数

このビットフィールドは、メモリ待ち時間に応じたダミーフェーズの時間の長さを定義します。

SDR モードと DTR モードの両方で、CLK サイクルの数を指定します (0~31)。

23.7.25 OCTOSPI 書込み命令レジスタ (OCTOSPI_WIR)

アドレスオフセット : 0x0190

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。その内容は、メモリマップトモードで書き込み動作をリクエストする場合にのみ意味を持ちます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INSTRUCTION[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSTRUCTION[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **INSTRUCTION[31:0]** : 命令

外部 SPI デバイスに送信される命令

23.7.26 OCTOSPI 書込みオルタネートバイトレジスタ (OCTOSPI_WABR)

アドレスオフセット : 0x01A0

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。その内容は、メモリマップトモードで書き込み動作をリクエストする場合にのみ意味を持ちます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTERNATE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **ALTERNATE[31:0]** : オルタネートバイト

アドレスの直後に外部 SPI デバイスに送信されるオプションのデータ。

23.7.27 OCTOSPI HyperBus 遅延設定レジスタ (OCTOSPI_HLCR)

アドレスオフセット : 0x0200

リセット値 : 0x0000 0000

このレジスタは、BUSY が 0 のときにのみ修正できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRWR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TACC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	WZL	LM
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **TRWR[7:0]** : 読出し・書込み最小リカバリ時間

通信クロックサイクル数で表されるデバイスの読出しから書込み/書込みから読出しへの最小リカバリ時間

ビット 15:8 **TACC[7:0]** : アクセス時間

通信クロックサイクル数で表される、メモリ待ち時間に応じたデバイスアクセス時間

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **WZL** : 書込みゼロ遅延

このビットは、書込み動作でのゼロ遅延を有効にします。

0 : 書込みアクセスでの遅延あり

1 : 書込みアクセスでの遅延なし

ビット 0 **LM** : 遅延モード

このビットは、遅延モードを選択します。

0 : 可変の初期遅延

1 : 固定遅延

表 209. OCTOSPI レジスタマップとリセット値


916/3086 RM0481 Rev 1 [English Rev 1] 

表 209. OCTOSPI レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0088	OCTOSPI_PSMAR	MATCH[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x008C	予約済み	予約済み																																		
0x0090	OCTOSPI_PIR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	INTERVAL[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0094 - 0x00FC	予約済み	予約済み																																		
0x0100	OCTOSPI_CCR	Res	Res	DQSE	Res	DDTR	DMODE [2:0]		Res	Res	Res	Res	Res	Res	ABDTR	ABMODE [2:0]		Res	Res	Res	Res	ADSIZE [1:0]	ADDTR	ADMODE [2:0]		Res	Res	ISIZE[1:0]		IDTR	IMODE [2:0]					
	リセット値			0		0	0	0	0				0	0	0	0	0	0				0	0	0	0	0			0	0	0	0	0			
0x0104	予約済み	予約済み																																		
0x0108	OCTOSPI_TCR	Res	SSHIFT	Res	DHQC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DCYC[4:0]							
	リセット値		0		0																								0	0	0	0	0			
0x010C	予約済み	予約済み																																		
0x0110	OCTOSPI_IR	INSTRUCTION[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0114- 0x011C	予約済み	予約済み																																		
0x0120	OCTOSPI_ABR	ALTERNATE[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0124 - 0x012C	予約済み	予約済み																																		
0x0130	OCTOSPI_LPTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMEOUT[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0134 - 0x013C	予約済み	予約済み																																		
0x0140	OCTOSPI_WPCCR	Res	Res	DQSE	Res	DDTR	DMODE [2:0]		Res	Res	Res	Res	Res	Res	ABDTR	ABMODE [2:0]		Res	Res	Res	Res	ADSIZE [1:0]	ADDTR	ADMODE [2:0]		Res	Res	ISIZE [1:0]		IDTR	IMODE [2:0]					
	リセット値	0		0		0	0	0	0			0	0	0	0	0	0	0				0	0	0	0	0	0			0	0	0	0			
0x0144	予約済み	予約済み																																		
0x0148	OCTOSPI_WPTCR	Res	SSHIFT	Res	DHQC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DCYC[4:0]							
	リセット値		0		0																								0	0	0	0	0			
0x014C	予約済み	予約済み																																		
0x0150	OCTOSPI_WPIR	INSTRUCTION[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0154 - 0x015C	予約済み	予約済み																																		

表 209. OCTOSPI レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0160	OCTOSPI_ WPABR	ALTERNATE[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0164 - 0x017C	予約済み	予約済み																															
0x0180	OCTOSPI_WCCR	Res	Res	DQSE	Res	DDTR	DMODE [2:0]		Res	Res	Res	Res	ABSIZE [1:0]	ABDTR	ABMODE [2:0]		Res	Res	Res	Res	ADSIZE [1:0]	ADDTR	ADMODE [2:0]		Res	Res	Res	Res	ISIZE [1:0]	IDTR	IMODE [2:0]		
	リセット値	0		0		0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0
0x0184	予約済み	予約済み																															
0x0188	OCTOSPI_WTCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DCYC[4:0]				
	リセット値																												0	0	0	0	0
0x018C	予約済み	予約済み																															
0x0190	OCTOSPI_WIR	INSTRUCTION[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0194 - 0x019C	予約済み	予約済み																															
0x01A0	OCTOSPI_WABR	ALTERNATE[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01A4 - 0x01FC	予約済み	予約済み																															
0x0200	OCTOSPI_HLCR	Res	Res	Res	Res	Res	Res	Res	Res	TRWR[7:0]					TACC[7:0]							Res	Res	Res	Res	Res	Res	WZL	LM				
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

24 セキュアデジタル入力/出力 MultiMediaCard インタフェース (SDMMC)

24.1 SDMMC の主な機能

SD/SDIO、embedded MultiMediaCard (eMMC) ホストインタフェース (SDMMC) は、AHB バスと SD メモリカード、SDIO カード、および eMMC デバイス間のインタフェースを提供します。

MMCA 技術委員会から発行された MultiMediaCard システム仕様書は、MultiMediaCard Association のウェブサイト www.jedec.org から入手できます。

SD メモリカードと SD I/O カードシステム仕様書は、SD card Association のウェブサイト www.sdcard.org から入手できます。

SDMMC には以下の機能があります。

- Embedded MultiMediaCard System Specification Version 5.1 に準拠。
3 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）、4 ビットおよび 8 ビット
(HS200 SDMMC_CK 速度は最大許容 I/O 速度に制限されています) (HS400 はサポートされていません)
- 以前のバージョンの MultiMediaCard との完全互換（下位互換性）
- SD memory card specifications version 6.0 に完全準拠。
(SDR104 SDMMC_CK 速度は最大許容 I/O 速度に制限され、SPI モードおよび UHS-II モードはサポートされません)
- SDIO card specification version 4.0 に完全準拠。
2 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）および 4 ビット
(SDR104 SDMMC_CK 速度は最大許容 I/O 速度に制限され、SPI モードおよび UHS-II モードはサポートされません)
- 8 ビットモードで最大 208 MB/秒のデータ転送
(最大許容 I/O 速度に基づきます)
- 外部双方向ドライバを制御するデータおよびコマンド出力有効信号
- IDMA のリンクされたリストのサポート

MultiMediaCard/SD バスは、カードをホストに接続します。

現行バージョンの SDMMC は、1 度に 1 枚の SD/SDIO/eMMC カードと、eMMC のスタックだけをサポートします。

24.2 SDMMC の実装

表 210. SDMMC の機能

SDMMC のモード／機能 ⁽¹⁾	SDMMC1	SDMMC2
可変遅延 (SRD104、HS200)	X	X
SDMMC_CKIN	X	X
SDMMC_CDIN、SDMMC_D0DIR	X	-
SDMMC_D123DIR	X	-

1. X：サポートされています。

24.3 SDMMC バストポロジ

バス上の通信は、コマンド／レスポンスとデータ転送に基づいて行われます。

SD/SDIO/eMMC バス上の基本的なトランザクションは、コマンド／レスポンストランザクションです。これらの種類のバストランザクションでは、コマンドやレスポンスの決められた通信フォーマットでカードの情報を直接転送します。また、データトークンを持つ操作もあります。

データ転送は次の方法に従って行います。

- ブロックモード：N の範囲 0~14 でブロックサイズ 2^N バイトのデータブロック
- SDIO マルチバイトモード：ブロックサイズ範囲 1~512 バイトの単一データブロック
- eMMC ストリームモード：連続的なデータストリーム

eMMC カードに対するデータ転送は、データブロックまたはストリーム単位で行われます。

図 152. SDMMC 「レスポンス無し」と「データなし」操作

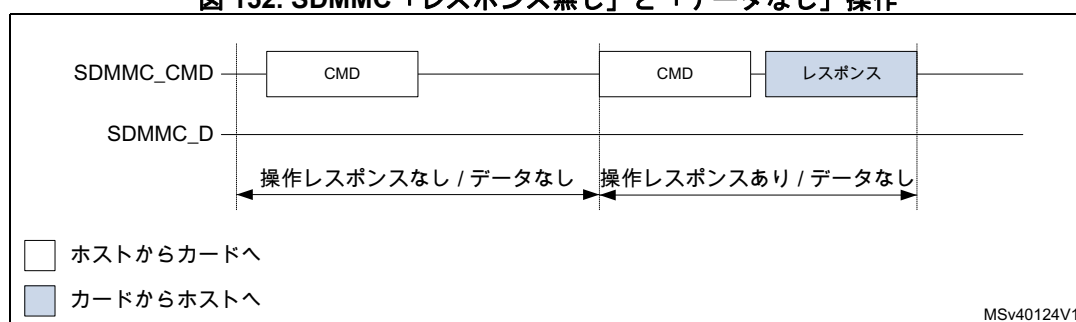
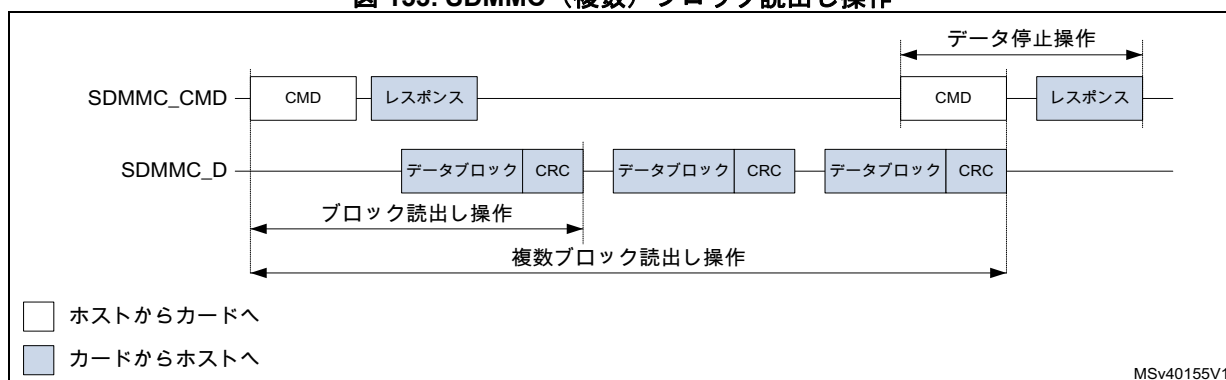
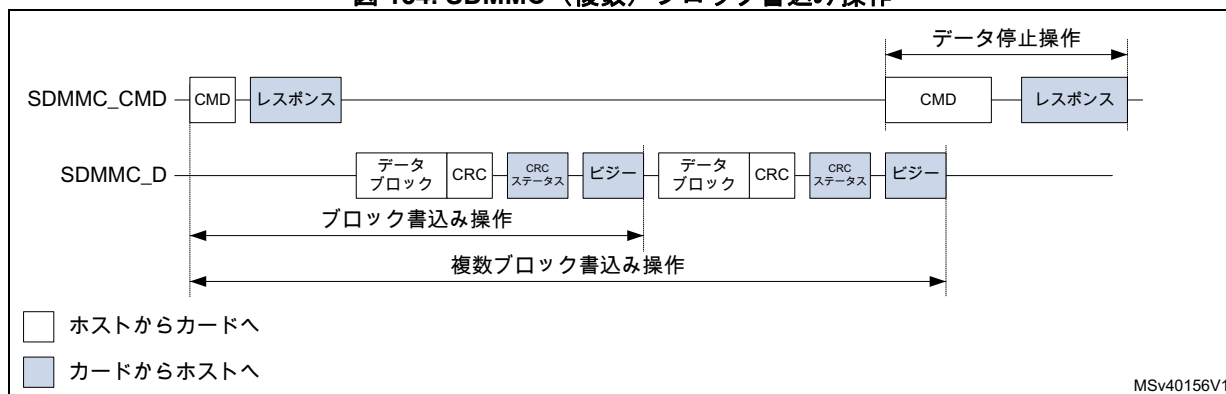


図 153. SDMMC (複数) ブロック読出し操作



注： 事前定義されたブロック数での eMMC 複数ブロック読出し終了時に、送信停止コマンドは必要ありません。

図 154. SDMMC (複数) ブロック書き込み操作



注： 事前定義されたブロック数での eMMC 複数ブロック書き込み終了時に、送信停止コマンドは必要ありません。

注： SDMMC は、ビジー信号がアサートされている限り (SDMMC_D0 をローへ固定)、データを一切送信しません。

図 155. SDMMC (シーケンシャル) ストリーム読み出し操作

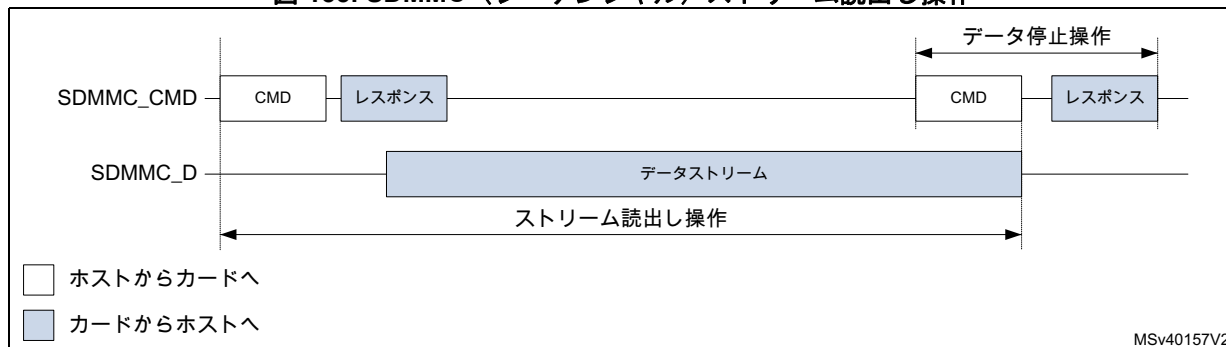
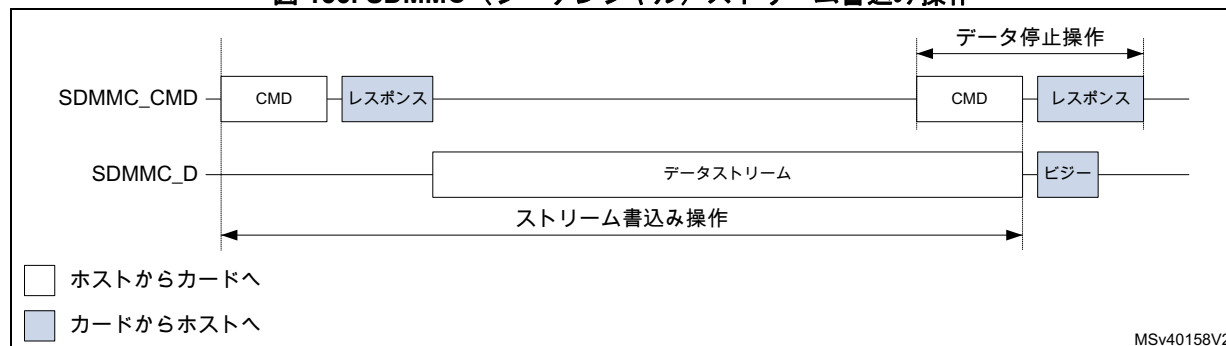


図 156. SDMMC (シーケンシャル) ストリーム書き込み操作



ストリームデータ転送は、シングルデータレートモード (DS、ハイスピード および SDR) の SDMMC_D0 で 1 ビット幅のビットバス設定でのみ動作します。

24.4 SDMMC 動作モード

表 211. SDMMC 動作モード SD および SDIO

SDIO バス速度モード ⁽¹⁾⁽²⁾	最大バス速度 ⁽³⁾ [MB/秒]	最大クロック周波数 [MHz] ⁽⁴⁾	信号電圧 [V]
DS (デフォルトスピード)	12.5	25	3.3
HS (ハイスピード)	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8
SDR104	104	208	1.8

1. SDR シングルデータレートシグナリング
2. DDR ダブルデータレートシグナリング (データが両方の SDMMC_CLK クロックエッジでサンプリングされます)
3. 4 ビットバス幅での SDIO バス速度
4. 最大許容 I/O 速度に基づく最大周波数

SDR104 モードでは、サンプリングポイントチューニングを使用した可変遅延サポートが必要です。
SDR50 モードでは、可変遅延の使用はオプションです。

表 212. SDMMC 動作モード eMMC

eMMC バス速度モード ⁽¹⁾⁽²⁾	最大バス速度 ⁽³⁾ [MB/秒]	最大クロック周波数 [MHz] ⁽⁴⁾	信号電圧 [V] ⁽⁵⁾
レガシー互換	26	26	3/1.8/1.2V
ハイスピード SDR	52	52	3/1.8/1.2V
ハイスピード DDR	104	52	3/1.8/1.2V
ハイスピード HS200	200	200	1.8/1.2V

1. SDR シングルデータレートシグナリング
2. DDR ダブルデータレートシグナリング (データが両方の SDMMC_CLK クロックエッジでサンプリングされます)
3. 8 ビットバス幅での eMMC バス速度
4. 最大許容 I/O 速度に基づく最大周波数
5. サポートされる信号電圧レベルは I/O ポートの特性によって異なります。デバイスのデータシートを参照してください。

HS200 モードでは、サンプリングポイントチューニングを使用した可変遅延サポートが必要です。

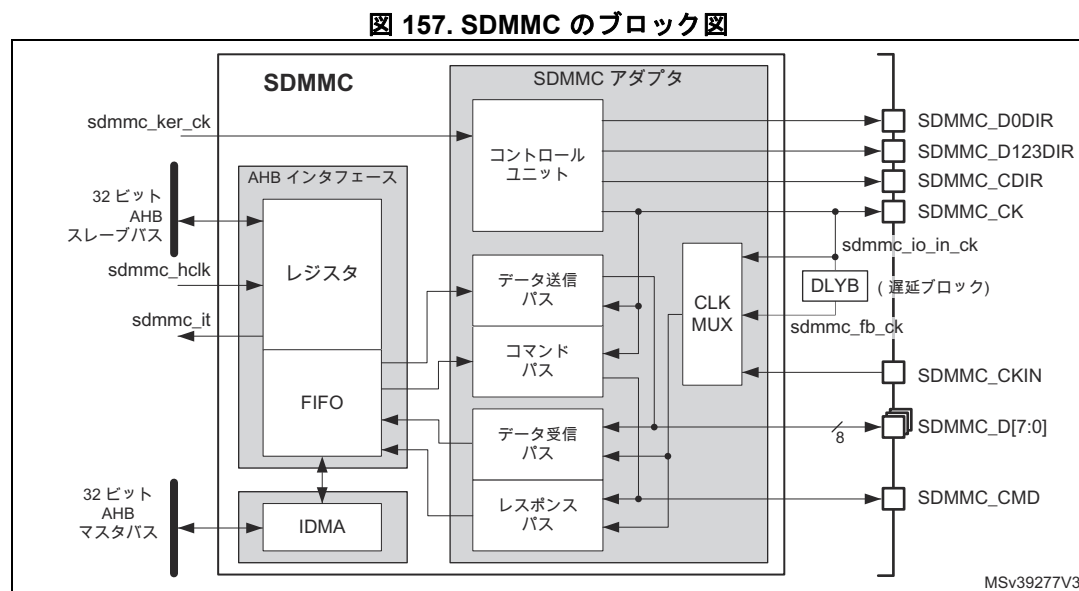
24.5 SDMMC の機能説明

SDMMC は、4つの部分で構成されています。

- AHB スレーブインタフェースは、SDMMC アダプタレジスタにアクセスして、割込み信号および IDMA 制御信号を生成します。
- SDMMC アダプタブロックは、クロック生成ユニット、コマンドおよびデータ転送など、eMMC/SD/SD I/O カードに固有のすべての機能を提供します。
- 内部 DMA (IDMA) ブロックは AHB マスタインタフェースを搭載しています。
- 遅延ブロック (DLYB) は、受信データのサンプル・クロック配置を処理します。遅延ブロックは、SDMMC の一部ではありません。遅延ブロックは、SDR104 または HS200 をサポートする場合に必須です。

24.5.1 SDMMC ブロック図

図 157 に SDMMC ブロック図を示します。



24.5.2 SDMMC ピンおよび内部信号

表 213 では SDMMC 内部入力/出力信号を、表 214 では SDMMC ピン (オルタネート機能) を一覧化しています。

表 213. SDMMC 内部入力/出力信号

信号名	信号タイプ	説明
sdmmc_ker_ck	デジタル入力	SDMMC カーネルクロック
sdmmc_hclk	デジタル入力	AHB クロック
sdmmc_it	デジタル出力	SDMMC グローバル割込み

表 213. SDMMC 内部入力/出力信号 (続き)

信号名	信号タイプ	説明
sdmmc_io_in_ck	デジタル入力	SD/SDIO/eMMC カードフィードバッククロック。この信号は、SDMMC_CK ピンに内部接続されます (DS および ハイスピードモードの場合)。
sdmmc_fb_ck	デジタル入力	DLYB 遅延ブロック後の SD/SDIO/eMMC カード調整フィードバッククロック (SDR50、DDR50、SDR104、HS200 の場合)

表 214. SDMMC ピン

信号名	信号タイプ	説明
SDMMC_CK	デジタル出力	SD/SDIO/eMMC カードへのクロック
SDMMC_CKIN	デジタル入力	SD/SDIO/eMMC カードの外部ドライバからのクロックフィードバック (SDR12、SDR25、SDR50、DDR50 の場合)
SDMMC_CMD	デジタル入出力	SD/SDIO/eMMC カードの双方向コマンド/レスポンス信号
SDMMC_CD1R	デジタル出力	SD/SDIO/eMMC カードの SDMMC_CMD 信号の I/O 方向指示
SDMMC_D[7:0]	デジタル入出力	SD/SDIO/eMMC カードの双方向データライン
SDMMC_D0DIR	デジタル出力	SD/SDIO/eMMC カードの SDMMC_D0 データラインの I/O 方向指示
SDMMC_D123DIR	デジタル出力	SD/SDIO/eMMC カードのデータライン SDMMC_D[3:1] の I/O 方向指示

24.5.3 概要

SDMMC_D[7:0] ラインには異なる動作モードがあります。

- デフォルトでは、SDMMC_D0 ラインはデータ転送に使用されます。初期化後、ホストはデータバス幅を変更することができます。
- eMMC では、1 ビット (SDMMC_D0)、4 ビット (SDMMC_D[3:0])、または 8 ビット (SDMMC_D[7:0]) のデータバス幅を使用できます。
- SD カードまたは SDIO カードでは、1 ビット (SDMMC_D0) または 4 ビット (SDMMC_D[3:0]) を使用できます。すべてのデータラインはプッシュプルモードで動作します。

外部ドライバ (電圧スイッチトランシーバ) の接続を許可するために、データラインでのデータフローの方向を I/O 方向信号で示します。SDMMC_D0DIR 信号は SDMMC_D0 データライン、SDMMC_D123DIR は SDMMC_D[3:1] データラインの I/O 方向を示します。

SDMMC_CMD はプッシュプルモードでのみ動作します。

外部ドライバ (電圧スイッチトランシーバ) の接続を許可するために、SDMMC_CMD ラインでのデータフローの方向を I/O 方向信号 SDMMC_CD1R で示します。

カードへの SDMMC_CK クロックは sdmmc_ker_ck から生成されます。

- sdmmc_ker_ck クロックが 50% デューティサイクルの場合、バイパスモードでも使用できます (CLKDIV = 0)。
- sdmmc_ker_ck クロックのデューティサイクルが 50% ではない場合、2 以上で分周するために CLKDIV を使用する必要があります (CLKDIV > 0)。
- SDMMC_CMD / SDMMC_D[7:0] 出力と SDMMC_CK の間の位相関係は、NEGEDGE ビットで選択できます。この位相関係は、CLKDIV、NEGEDGE および DDR の設定に依存します。[図 158](#) を参照してください。

図 158. SDMMC コマンドおよびデータの位相関係

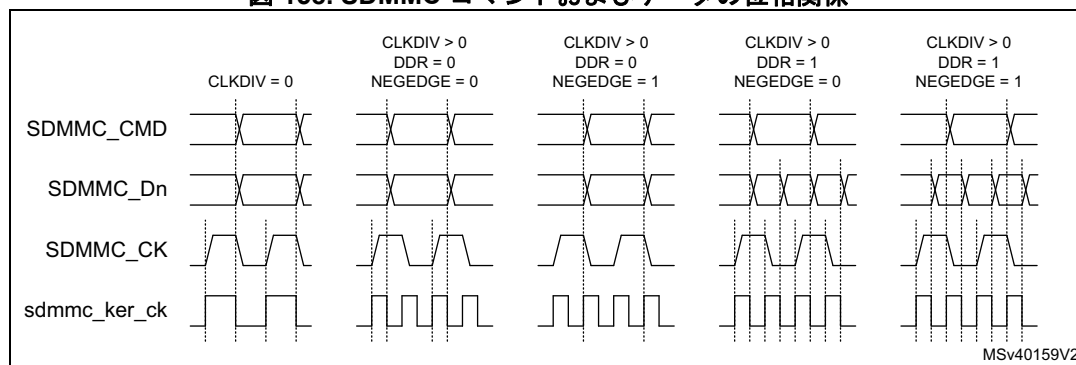


表 215. SDMMC コマンドおよびデータの位相選択

CLKDIV	DDR	NEGEDGE	SDMMC_CK	コマンド出力	データ出力
0	x	x	= sdmmc_ker_ck	sdmmc_ker_ck 立下がりエッジで生成	
>0	0	0	sdmmc_ker_ck 立上がり エッジで生成	SDMMC_CK 立上がりエッジに続く sdmmc_ker_ck 立下がりエッジで生成	
		1		SDMMC_CK 立下がりエッジを生成する同じ sdmmc_ker_ck 立上がりエッジで生成	
	1	0		SDMMC_CK 立上がりエッジに続く sdmmc_ker_ck 立下がりエッジで生成	SDMMC_CK エッジに続く sdmmc_ker_ck 立下がり エッジで生成
		1		SDMMC_CK 立下がりエッジを生成する同じ sdmmc_ker_ck 立上がりエッジで生成	

デフォルトでは、**sdmmc_io_in_ck** フィードバッククロック入力、SDMMC 受信パスで受信データをサンプリングするために選択されます。これは SDMMC_CK ピンから生成されます。

サンプリングクロックの位相をチューニングして受信データタイミングを調整するために、デバイスで利用できる DLYB 遅延ブロックを **sdmmc_io_in_ck** 信号 (DLYB 入力 dlyb_in_ck) と SDMMC の **sdmmc_fb_ck** クロック入力 (DLYB 出力 dlyb_out_ck) の間に接続できます。受信パスで **sdmmc_fb_ck** クロック入力を選択すると、受信データの位相調整サンプリングクロックを使用できるようになります。これは、SDMMC で SDR104 動作モードおよび HS200 動作モードをサポートするために必要であり、SDR50 モードおよび DDR50 モードではオプションです。

外部ドライバ (電圧スイッチトランシーバ) を使用する場合、SDMMC_CKIN フィードバッククロック入力を選択して受信データをサンプリングできます。

SD/SDIO/eMMC カードの場合、クロック周波数は 0~208 MHz の範囲 (最大 I/O 速度により制限) で可変です。

選択したバスモード (SDR または DDR) に応じて、各クロックサイクルで SDMMC_D[7:0] ラインに 1 ビットまたは 2 ビットが転送されます。SDMMC_CMD ラインは各クロックサイクルで 1 ビットだけ転送します。

24.5.4 SDMMC アダプタ

SDMMC アダプタ (図 157 : SDMMC のブロック図を参照) は、MultiMediaCard スタックやセキュアデジタルメモリカードとのインタフェースを提供するマルチメディア/セキュアデジタルメモリカードバスマスタです。これは次のサブユニットで構成されています。

- 制御ユニット
- データ送信パス
- コマンドパス
- データ受信パス
- レスポンスパス
- 受信データバスクロックマルチプレクサ
- SDMMC の外部の遅延ブロック (DLYB)
- アダプタレジスタブロック
- データ FIFO
- 内部 DMA (IDMA)

注 : アダプタレジスタと FIFO は、AHBクロックドメイン (sdmmc_hclk) を使用します。制御ユニット、コマンドパス、およびデータ送信パスは、SDMMC アダプタクロックドメイン (sdmmc_ker_ck) を使用します。レスポンスパスとデータ受信パスは、sdmmc_io_in_ck、SDMMC_CKIN、または DLYB によって生成された sdmmc_fb_ck からの SDMMC アダプタフィードバッククロックドメインを使用します。

デバイスの DLYB 遅延ブロックは、SDMMC アダプタと連携して、SDMMC 受信モードでサンプリングクロックの位相を調整してデータを受信するために使用できます。これは、SDMMC で SDR104 動作モードおよび HS200 動作モードをサポートするために必要であり、SDR50 モードおよび DDR50 モードではオプションです。

アダプタレジスタブロック

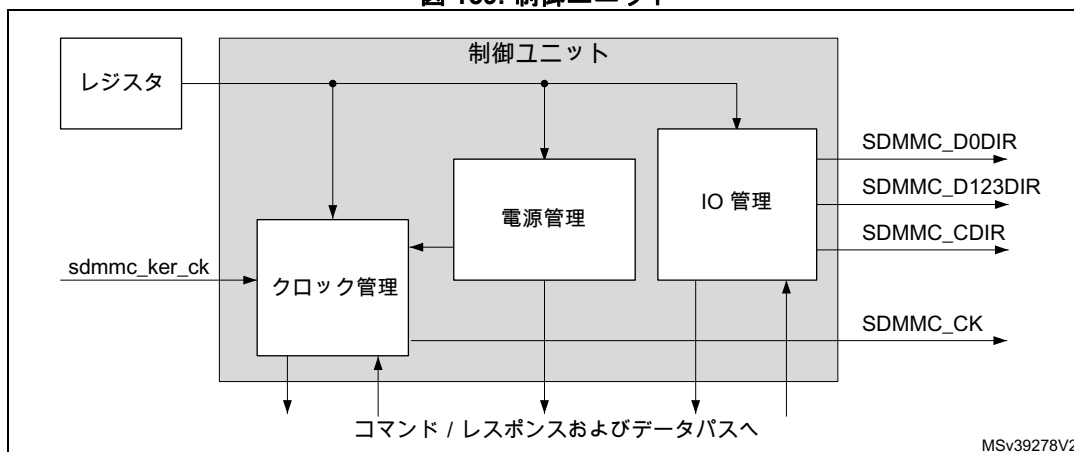
アダプタレジスタブロックには、すべてのシステム制御レジスタ、SDMMC コマンドおよびレスポンスレジスタ、データ FIFO が含まれています。

このブロックは、SDMMC アダプタのスタティックフラグをクリアする SDMMC クリアレジスタの対応するビット位置からの信号も生成します。

制御ユニット

図 159 に示された制御ユニットには、電源管理機能、分周回路での SDMMC_CK クロック管理、I/O 方向管理が含まれています。

図 159. 制御ユニット



電源管理サブユニットは、パワーオフおよびパワーアップフェーズの間、カードバス出力信号を無効にします。

3 つの電源フェーズがあります。

- パワーオフ
- パワーアップ
- パワーオン

クロック管理サブユニットは、sdmmc_ker_ck を使用して SDMMC_CK を生成し、分周制御を提供します。また、フロー制御などで SDMMC_CK を停止することもできます。

クロック出力は、次の場合にインアクティブです。

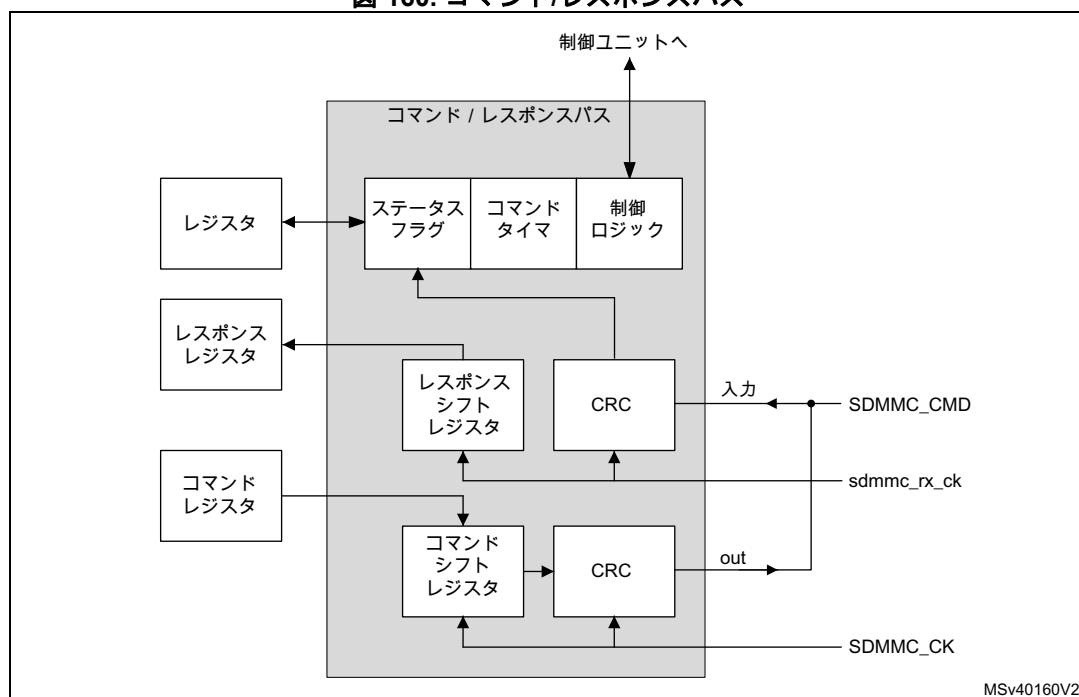
- リセット後
- パワーオフまたはパワーアップフェーズ中
- 節電モード（レジスタビット PWRSV）が有効であり、カードバスが 8 クロック周期の間アイドル状態の場合。クロックは、コマンド/レスポンス CPSM およびデータパス DPSM サブユニットの両方がアイドルフェーズに移行してから 8 サイクル後に停止します。クロックは、コマンド/レスポンス CPSM またはデータパス DPSM がアクティブになった（有効になった）ときに再開します。

I/O 管理サブユニットは、SDMMC_Dn および SDMMC_CMD の I/O 方向信号を管理し、外部電圧トランシーバを制御します。

コマンド/レスポンスパス

コマンド/レスポンスパスサブユニットは、SDMMC_CMD ラインでコマンドとレスポンスを転送します。コマンドパスは SDMMC_CK に基づいてクロック供給され、コマンドをカードに送信します。レスポンスパスは sdmmc_rx_ck に基づいてクロック供給され、カードからレスポンスを受信します。

図 160. コマンド/レスポンスパス



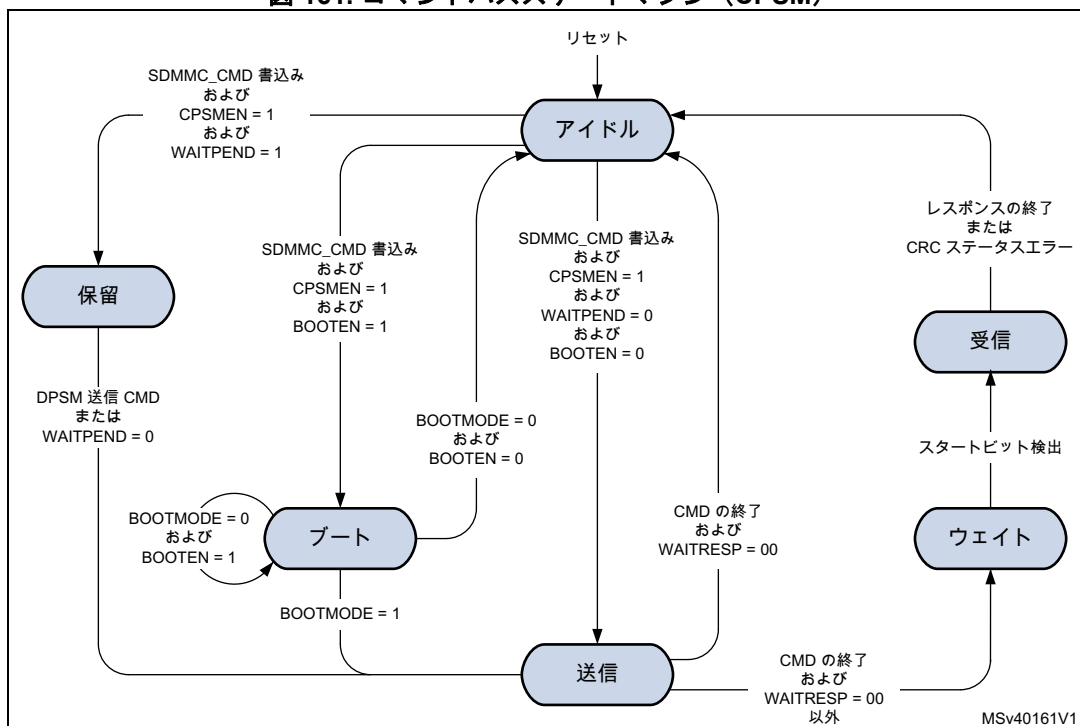
MSv40160V2

コマンド/レスポンスパスステートマシン (CPSM)

- コマンドレジスタが書き込まれて、有効ビットがセットされると、コマンド転送が開始します。コマンドが送信されると、CRC が付加され、コマンドパスステートマシン (CPSM) がステータスフラグをセットします。
 - レスポンスが必要ではない場合は、アイドル状態に入ります。
 - レスポンスが必要な場合は、レスポンスを待ちます。
- レスポンス受信時：
 - CRC 付きのレスポンスの場合、受信した CRC コードと内部生成されたコードを比較し、結果に応じて適切なステータスフラグをセットします。
 - CRC なしのレスポンスの場合、CRC はチェックされず、適切なステータスフラグがセットされません。

アイドル状態ではない場合など CPSM がアクティブであるときは必ず CPSMACT ビットがセットされます。

図 161. コマンドパスステートマシン (CPSM)



- **アイドル**：コマンドパスがインアクティブです。コマンド制御レジスタが書き込まれ、有効ビット (CPSMEN) がセットされると、CPSM は SDMMC_CLK クロック（節電 PWRSAV ビットのために停止している場合）を有効化して次の状態に移行します。
 - WAITPEND = 0 かつ BOOTEN = 0 の場合は送信状態
 - WAITPEND = 1 の場合はペンディング状態
 - BOOTEN = 1 の場合はブート状態
- **送信**：コマンドが送信され、CRC が付加されます。
 - CMDTRANS ビットがセットされている場合、または BOOTEN ビットがセットされていて、かつ BOOTMODE が代替ブートで、DTDIR = 受信の場合、コマンド終了時に CPSM DataEnable 信号が DPSM に発行されます。
 - CMDTRANS ビットがセットされていて、かつ CMDSPEND ビットが 0 の場合、コマンド終了時に割り込み期間が終了します。
 - CMDSTOP ビットがセットされている場合、コマンド終了時に CPSM Abort 信号が DPSM に発行されます。
 - レスポンスが期待されていない (WAITRESP = 00) 場合、CPSM はアイドル状態に移行し、CMDSENT フラグがセットされます。BOOTMODE = 1 かつ BOOTEN = 0 の場合、CMDSENT フラグはコマンド終了ビットの 56 サイクル後に遅延され、それ以外の場合、コマンド終了ビット直後に CMDSENT フラグが生成されます。RESPCMDR レジスタおよび RESPxR レジスタは変更されません。
 - コマンドレスポンスが期待されている (WAITRESP が 00 ではない) 場合、CPSM はウェイト状態に移行し、レスポンスタイムアウトを開始します。

- **ウェイト** : コマンドパスはレスポンスを待ちます。
 - WAITINT ビットが 0 の場合、コマンドタイマが動作を開始し、CPSM はスタートビットを待ちます。
 - a) スタートビットがタイムアウトまでに検出された場合、CPSM は受信状態に移行します。
 - b) CPSM がレスポンススタートビットを検出する前にタイムアウトに達した場合には、タイムアウトフラグ (CTIMEOUT) がセットされ、CPSM がアイドル状態に移行します。RESPCMDR レジスタおよび RESPxR レジスタは変更されません。
 - WAITINT ビットが 1 の場合、タイマは無効になり、CPSM はカードの 1 枚からの割込みリクエスト (レスポンススタートビット) を待ちます。
 - a) スタートビットが検出された場合、CPSM は受信状態に移行します。
 - b) WAITINT に 0 を書き込む (割込みモードを中止する) と、ホストは自身でレスポンスを送信し、スタートビット検出時に CPSM は受信状態に移行します。
- **受信** : コマンドレスポンスが受信されます。コマンド制御レジスタのレスポンスモードビット WAITRESP に応じて、CRC の有無によりレスポンスを短くしたり長くしたりできます。受信した CRC コードがある場合、内部で生成された CRC コードに対して検証されます。
 - CMDSPEND ビットがセットされていて、かつ SDIO レスポンスビット BS = 0 (レスポンスビット [39]) の場合、レスポンスの後に割込み期間が開始されます。CMDSPEND ビットがクリアされている場合、または CMDSPEND ビットが 1 で、かつ SDIO レスポンスビット BS = 1 (レスポンスビット [39]) の場合、割込み期間は開始されません。
 - CMDTRANS ビットと CMDSPEND ビットがセットされていて、かつ SDIO レスポンスビット DF = 1 (レスポンスビット [32]) の場合、レスポンスの後に割込み期間が終了します。
 - CRC ステータスがパスするか、CRC が存在しない場合、CMDREND フラグがセットされ、CPSM がアイドル状態に移行します。
RESPCMDR レジスタおよび RESPxR レジスタは受信したレスポンスで更新されます。
- BOOTMODE = 1 かつ BOOTEN = 0 の場合、CMDREND フラグはレスポンス終了ビットの 56 サイクル後に遅延され、それ以外の場合、レスポンス終了ビット直後に CMDREND フラグが生成されます。
- CMDTRANS ビットがセットされていて、かつ DTDIR = 送信の場合、コマンドレスポンス終了時に CPSM DataEnable 信号が DPSM に発行されます。
 - CRC ステータスが失敗となる場合、CCRCFAIL フラグがセットされ、CPSM がアイドル状態に移行します。
RESPCMDR レジスタおよび RESPxR レジスタは受信したレスポンスで更新されます。
- **ペンディング** : コマンドレジスタのペンディング WAITPEND ビットに応じて、CPSM はペンディング状態に入ります。
 - DATALENGTH ≤ 5 バイトの場合、CPSM は送信状態に入り、DataEnable 信号を生成して CMD12 送信停止コマンドで調整されたデータ転送を開始します。
 - DATALENGTH > 5 バイトの場合、データ転送を開始するために CPSM DataEnable 信号が DPSM に発行されます。CPSM は DPSM からの送信 CMD 信号を待ってから、送信状態に入ります。これにより、CMD12 送信停止コマンドなどをデータで調整して送信できます。
 - WAITPEND に 0 を書き込むと、CPSM は送信状態に移行します。
- **ブート** : コマンドレジスタの BOOTEN ビットがセットされると、CPSM はブート状態に入ります。そして、
 - BOOTMODE = 0 で SDMMC_CMD ラインがローに駆動されている場合、かつ CMDTRANS ビットがセットされ、DTDIR = 受信の場合、CPSM DataEnable 信号が DPSM に発行されます。これで通常のブート動作が可能になります。この状態からは、レジスタビット BOOTEN をクリアすることでブート手順の終了時に離脱し、これによって SDMMC_CMD ラインがハイに駆動され、CPSM Abort 信号が DPSM に発行されてから、アイドル状態に

移行します。SDMMC_CMD ラインがハイになってから 56 サイクル後に CMDSENT フラグが生成されます。

- BOOTMODE = 1 の場合、送信状態へ移行します。これで CMD0（ブート）の送信が可能になります。BOOTEN をクリアしても影響はありません。

注： CPSM は、 N_{CC} および N_{RC} タイミング制約を満たすために、少なくとも 8 SDMMC_CK 周期の間、アイドル状態を保ちます。 N_{CC} は、2 つのホストコマンド間の最小遅延であり、 N_{RC} は、ホストコマンドとカードレスポンスの間の最小遅延です。

注： レスポンスタイムアウトは、64 SDMMC_CK クロック周期の固定値です。

コマンドは、動作を開始するトークンです。コマンドは、ホストから 1 枚のカード（アドレス付きコマンド）または接続されたすべてのカードに送信されます（ブロードキャストコマンドは eMMC V3.31 以前で使用可能）。コマンドは、SDMMC_CMD ライン上で連続的に転送されます。すべてのコマンドは、固定長 48 ビットです。SD メモリカード、SDIO カード、eMMC カードのコマンドトークンの一般的なフォーマットを表 216 に示します。

コマンドトークンデータは 2 つのレジスタから取得され、1 つには 32 ビットの引数が含まれ、他方には 6 ビットのコマンドインデックス（カードに送信される 6 ビット）が含まれます。

表 216. コマンドトークンフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	1	送信ビット
[45:40]	6	x	コマンドインデックス
[39:8]	32	x	引数
[7:1]	7	x	CRC7
0	1	1	終了ビット

コマンドデータの隣にはコマンドタイプ (WAITRESP) ビットがあり、コマンドパスステートマシン (CPSM) を制御します。これらのビットによって、コマンドがレスポンスを必要とするかどうかと、レスポンスが短い (48 ビット) か長い (136 ビット) か、CRC が存在するかどうかも決まります。

レスポンスは、以前に受信したコマンドへの応答として、アドレス指定されたカードまたは接続されたすべてのカードから同時にホストに送信されるトークンです。すべてのレスポンスはコマンドライン SDMMC_CMD 経由で送信されます。レスポンスの送信は、常に、レスポンスコードワードに対応するビット文字列の左のビットから始まります。コード長は、レスポンスのタイプに依存します。レスポンストークン R1、R2、R3、R4、R5、R6 には、コンテンツに応じてさまざまなコード体系があります。SD メモリカード、SDIO カード、eMMC カードのレスポンストークンの一般的なフォーマットを表 217、表 218、表 219 に示します。

レスポンスは、常にスタートビット (常に 0) から始まり、送信方向を示すビットが続きます (card=0)。下の表で x で示されている値は、可変エントリを示します。一部を除くほとんどのレスポンスは、CRC によって保護されます。すべてのコマンドコードワードは、終了ビット (常に 1) で終わります。

レスポンストークンデータは 5 つのレジスタに格納され、4 つは 32 ビットのカードステータス、OCR レジスタ、引数または内部 CRC を含んだ 127 ビット CID や CSD レジスタを含み、あと 1 つのレジスタは 6 ビットのコマンドインデックスを含みます。

表 217. CRC ありのショートレスポンスのトークンフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	x	コマンドインデックス（または予約済み 111111）
[39:8]	32	x	引数
[7:1]	7	x	CRC7
0	1	1	終了ビット

表 218. CRC なしのショートレスポンスのトークンフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	x	コマンドインデックス（または予約済み 111111）
[39:8]	32	x	引数
[7:1]	7	1111111	（予約済み 1111111）
0	1	1	終了ビット

表 219. CRC ありのロングレスポンスのトークンフォーマット

ビット位置	幅	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	111111	予約済み
[127:1]	127:8	x	CID または CSD スライス
	7:1	x	CRC7（CID、CSD レジスタに含まれる）
0	1	1	終了ビット

コマンド/レスポンスパスは、コマンドを送信したりレスポンスを受信したりできるように半二重モードで動作します。CPSM が送信状態でない場合、SDMMC_CMD 出力はハイインピーダンス状態になります。SDMMC_CMD で送信されたデータは、NEGEDGE レジスタビットに応じて SDMMC_CLK と同期します。図 158 を参照してください。

コマンドおよび CRC ありのショートレスポンスでは、CRC 生成回路は、CRC コードの前のすべての 40 ビットについて CRC チェックサムを計算します。これには、スタートビット、送信ビット、コマンドインデックス、およびコマンド引数（またはカードステータス）が含まれます。

ロングレスポンスでは、R2 CID または CSD の 120 ビットに対してのみ CRC チェックサムが計算されます。スタートビット、送信ビット、および 6 つの予約ビットは、CRC 計算では使用されませんので注意してください。

CRC チェックサムは 7 ビットの値です。

$$\text{CRC}[6:0] = \text{剰余} [(M(x) * x^7) / G(x)]$$

$$G(x) = x^7 + x^3 + 1$$

$$M(x) = (\text{最初のビット}) * x^n + (2 \text{ 番目のビット}) * x^{n-1} + \dots + (\text{CRC 前の最後のビット}) * x^0$$

$n = 39$ または 119 の場合。

CPSMEN がセットされている場合、CPSM は、さまざまな動作モードに対応するために数多くの固有コマンドを送信できます。表 220 を参照してください。

表 220. 固有コマンドの概要

VSWITCH	BOOTEN	BOOTMODE	CMDTRANS	WAITPEND	CMDSTOP	WAITINT	説明
1	x	x	x	x	x	x	電圧スイッチシーケンス開始
0	1	x	x	x	x	x	通常ブート開始
0	1	1	x	x	x	x	代替ブート開始
0	0	1	x	x	x	x	代替ブート停止
0	0	0	1	x	x	x	関連付けしたデータ転送でコマンド送信
0	0	0	0	1	1	x	eMMC ストリームデータ転送、データ転送終了までコマンド (STOP_TRANSMISSION) ペンディング
0	0	0	0	1	0	x	eMMC ストリームデータ転送、データ転送終了まで (STOP_TRANSMISSION) ペンディング以外のコマンド
0	0	0	0	0	1	x	コマンド送信 (STOP_TRANSMISSION)、進行中のデータ送信を停止
0	0	0	0	0	0	1	eMMC ウェイト割込み (Wait-IRQ) モードへ移行
0	0	0	0	0	0	0	その他の固有コマンドなし

コマンド/レスポンスパスは、表 221 に示すステータスフラグおよび関連付けられたクリアビットを実装しています。

表 221. コマンドパスステータスフラグ

フラグ	説明
CMDSENT	レスポンスなしのコマンド終了時にセット。(CPSM は送信からアイドルに移行)
CMDREND	CRC が OK のときにコマンドレスポンス終了時にセット。(CPSM は受信からアイドルに移行)
CCRCFAIL	CRC が FAIL のときにコマンドレスポンス終了時にセット。(CPSM は受信からアイドルに移行)
CTIMEOUT	タイムアウト前に受信したレスポンススタートビットがないときにコマンド後にセット。(CPSM はウェイトからアイドルに移行)
CKSTOP	CRC が OK かつ SDMMC_CK が停止したときに電圧スイッチ (VSWITCHEN = 1) コマンドレスポンス後にセット。(CPSM に影響なし)
VSWEND	5ms + 1ms の電圧スイッチ (VSWITCH = 1) タイムアウト後にセット。(CPSM に影響なし)
CPSMACT	コマンド転送中です。(CPSM がアイドル状態ではない)

コマンドパスのエラー処理を、表 222 に示します。

表 222. コマンドパスのエラー処理

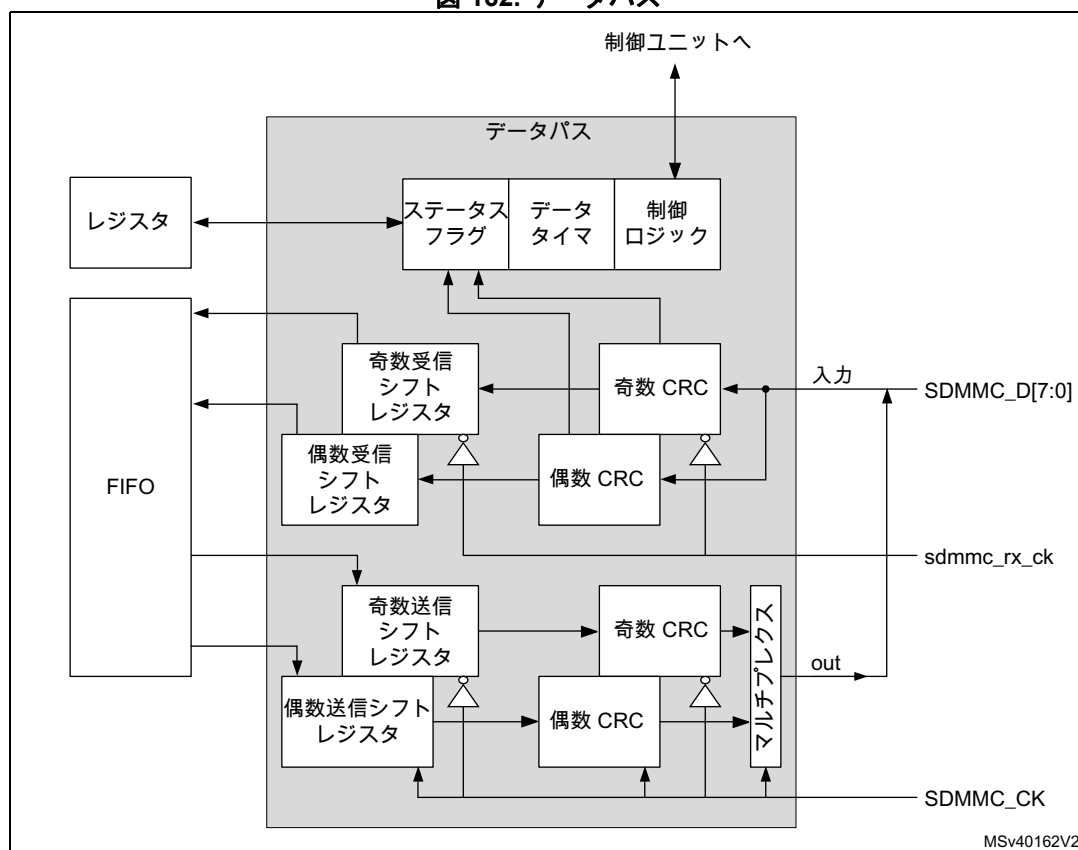
誤差	CPSM の状態	原因	カード動作	ホスト動作	CPSM 動作
タイムアウト	ウェイト	時間内にスタートビットなし	不明	リセットまたはサイクルパワーカード ⁽¹⁾	アイドルに移行
CRC ステータス	受信	否定ステータス	コマンド無視	コマンド再送信 ⁽¹⁾	アイドルに移行
		送信エラー	コマンド許可	コマンド再送信 ⁽¹⁾	

1. CMDTRANS をセットするとき、stop_transmission コマンドも送信して、DPSM をアイドルに移行させる必要があります。

データパス

データパスサブユニットは、カードとの間で SDMMC_D[7:0] ラインでデータ転送を行います。データ送信パスは SDMMC_CK に基づいてクロック供給され、データをカードに送信します。データ受信パスは sdmmc_rx_ck に基づいてクロック供給され、カードからデータを受信します。図 162 にデータパスブロック図を示します。

図 162. データパス



カードデータバス幅は、クロック制御レジスタビット WIDBUS でプログラムできます。サポートされるデータバス幅モードは次のとおりです。

- ワイドバスモードが有効でない場合は、SDMMC_D0 上で 1 ビットだけが転送されます。
- 4 ビット幅バスモードが有効な場合、データは SDMMC_D[3:0] 上で、4 ビットで転送されます。
- 8 ビット幅バスモードが有効な場合、データは SDMMC_D[7:0] 上で、8 ビットで転送されます。

データバス幅の次に、データサンプリングモードはクロック制御レジスタビット DDR でプログラムできます。サポートされるデータサンプリングモードは次のとおりです。

- シングルデータレートシグナリング (SDR) では、データはクロックの立上がりエッジでクロック供給されます。
- ダブルデータレートシグナリング (DDR) では、データはクロックの両方のエッジでクロック供給されます。DDR モードはワイドバスモード (4 ビット幅および 8 ビット幅) でのみサポートされます。

注： データサンプリングモードは、SDMMC_D[7:0] ラインのみに適用されます。(SDMMC_CMD ラインには適用できません)。

DDR モードでは、次のルールに応じて SDMMC_CK の両方のエッジでデータがサンプリングされます。図 163 および図 164 も参照してください。

- クロックの立上がりエッジで奇数バイトがサンプリングされます。
- クロックの立下がりエッジで偶数バイトがサンプリングされます。
- データのペイロードサイズは常に 2 バイトの倍数になります。
- 2 つの CRC16 がデータラインごとに計算されます。
 - クロックの立下がりエッジでクロック供給された奇数ビット CRC16
 - クロックの立上がりエッジでクロック供給された偶数ビット CRC16
- スタートビット、終了ビットおよびアイドル状態がフルサイクルになります。
- CRC ステータス/ブート確認応答およびビジーシグナリングがフルサイクルになり、クロックの立上がりエッジでのみサンプリングされます。

DDR モードでは、SDMMC_CK クロック分周は 2 以上でなければなりません。

図 163. DDR モードのデータパケットクロック供給

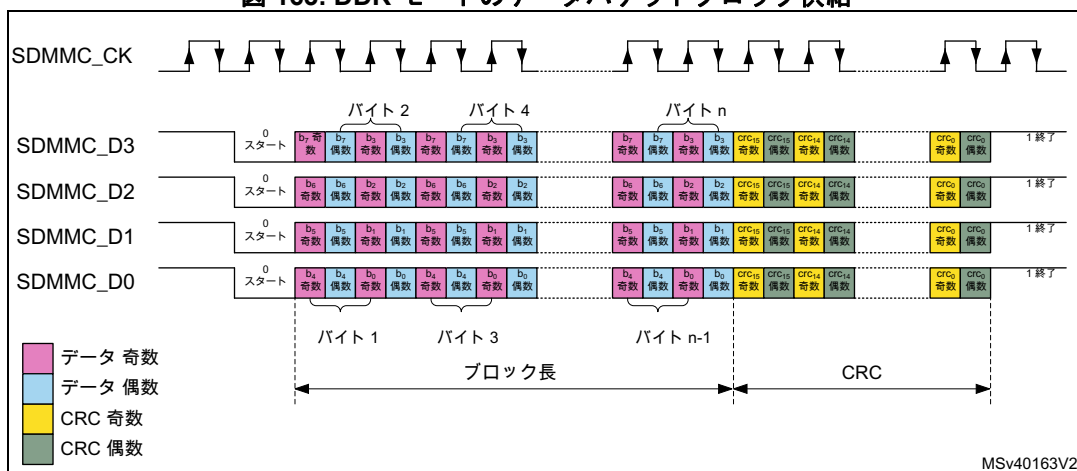
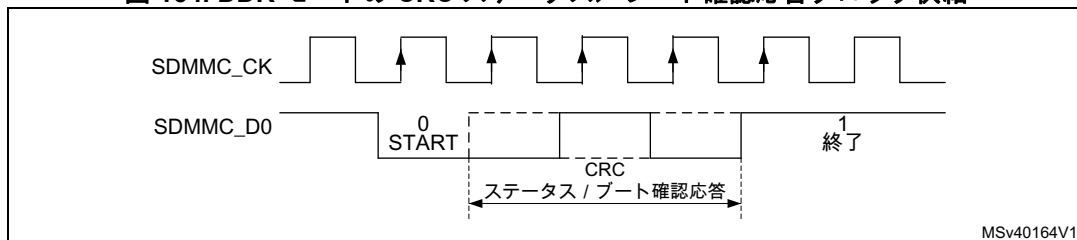


図 164. DDR モードの CRC ステータス/ブート確認応答クロック供給



データパスステートマシン (DPSM)

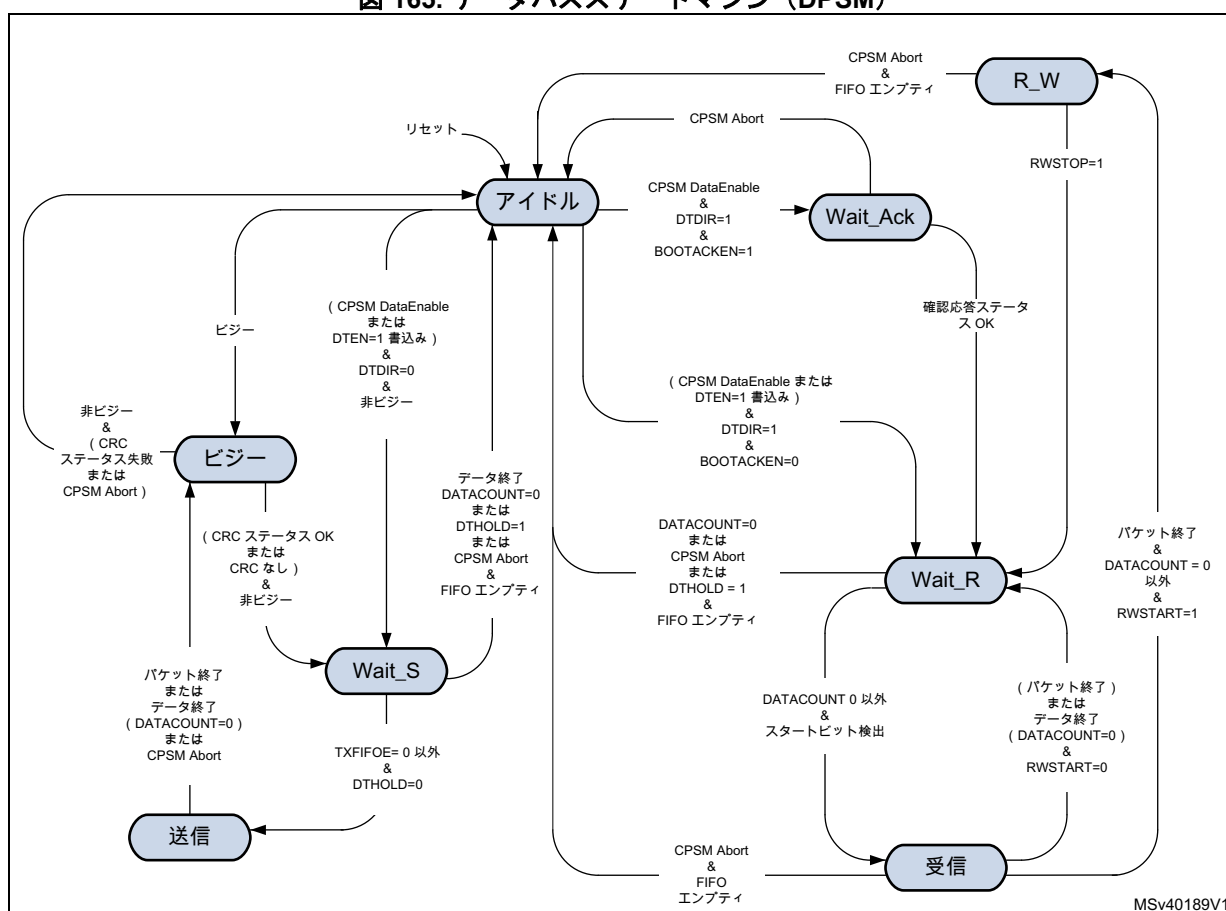
転送の方向（送信または受信）によって、データパスステートマシン（DPSM）は、有効になったときに Wait S または Wait R 状態へ移行します。

- 送信: DPSM は Wait_S 状態へ移行します。送信 FIFO にデータがある場合、DPSM は送信状態へ移行し、データバスサブユニットはカードへのデータ転送を開始します。
- 受信: DPSM は Wait_R 状態へ移行して、スタートビットを待ちます。スタートビットを受信すると、DPSM は受信状態へ移行し、データバスサブユニットはカードからのデータ受信を開始します。

確認応答ありのブート動作の場合、DPSM は Wait_Ack 状態に移行し、ブート確認応答を待つから Wait_R 状態に移行します。

DPSM は、SDMMC_CK で動作します。図 165に示すように、DPSM には、次の状態があります。アイドル状態ではない場合など DPSM がアクティブであるときは必ず DPSMACT ビットがセットされます。

図 165. データパスステートマシン (DPSM)



- **アイドル状態：**データバスはインアクティブであり、SDMMC_D[7:0] 出力は PWRCTRL の設定に従います。CMDTRANS ビットをセットしてコマンドを送信するか、DTEN ビットをセットするか、または SDMMC_D0 でビジーを検出すること（R1b レスポンスありのコマンドの後）のどれかによって DPSM が有効化されます。

ビジーではない場合、DPSM は SDMMC_CLK クロック（節電 PWRSAV ビットのために停止している場合）を有効化し、新しい（DATALENGTH）の値でデータカウンタをロードします。そして、

- データ方向ビット (DTCIR) が送信を示している場合、Wait_S に移行します。
- データ方向ビット (DTCIR) が受信を示している場合、次のように移行します。
 - BOOTACKEN レジスタビットがクリアされている場合は Wait_R になります。
 - BOOTACKEN レジスタビットがセットされている場合は、Wait_Ack となり、確認応答タイムアウトが開始します。

ビジーの場合、DPSM は SDMMC_CLK クロックをアクティブなままとし、ビジー状態に移行します。

注： DTEN は SD、SDIO および eMMC カードでデータ転送の開始に使用してはなりません。

- **Wait_Ack** 状態：データパスがブート確認応答トークンを待ちます。
 - DPSM がタイムアウト前にエラーなし確認応答を受信した場合、Wait_R 状態に移行します。
 - 確認応答とは異なるパターンを受信した場合、確認応答ステータスエラーが生成され、確認応答失敗ステータスフラグ (ACKFAIL) がセットされます。DPSM は、Wait_Ack のままとなります。
 - スタートビットを検出する前にタイムアウト (ACKTIME) に達した場合、タイムアウトステータスフラグ (ACKTIMEOUT) をセットします。DPSM は、Wait_Ack のままとなります。
 - CPSM Abort 信号がセットされている場合、アイドル状態へ移行し、DABORT フラグをセットします。
- **Wait_R** 状態：データカウンタがゼロでなく、データが保持されない場合、データパスは SDMMC_D[n:0] 上のスタートビットを待ちます。データカウンタがゼロで、データを保持している場合、FIFO が空になるのを待ちます。
 - ブロックモードでは、タイムアウトまでにスタートビットを受信した場合、DPSM は受信状態に移行し、DBLOCKSIZE でデータブロックカウンタをロードします。
 - SDIO マルチバイトモードでは、タイムアウトまでにスタートビットを受信した場合、DPSM は受信状態に移行し、DATALENGTH でデータブロックカウンタをロードします。
 - ストリームモードでは、タイムアウトまでにスタートビットを受信した場合、DPSM は受信状態に移行し、DATALENGTH でデータカウンタをロードします。
 - データカウンタ (DATACOUNT) がゼロ (データの終了) に等しい場合、DPSM は受信 FIFO が空になると、アイドル状態へ移行し、DATAEND フラグがセットされます。
 - スタートビットを検出する前にタイムアウト (DATATIME) に達した場合、タイムアウトステータスフラグ (DTIMEOUT) をセットし、DPSM は Wait_R 状態のままとなります。
 - CPSM Abort 信号がセットされている場合：
 - DATACOUNT > 0 の場合、FIFO が空になったとき IDMAEN = 0 が FIFORST でリセットされたときに、DPSM はアイドル状態に移行し、DABORT フラグがセットされます。DATACOUNT がゼロの場合、通常動作を続け、転送が正常に完了するため、DABORT フラグはセットされません。
 - DTHOLD ビットがセットされている場合：
 - DATACOUNT > 0 の場合、受信 FIFO が空になったとき IDMAEN = 0 が FIFORST でリセットされたときに、DPSM はアイドル状態に移行し、DHOLD フラグが発行されます。保留状態ではタイムアウトは無効になります。保留状態で CPSM Abort 信号を受信すると、転送はアボートされます。
 - DATACOUNT = 0 の場合、転送は正常に完了し、DHOLD フラグは発行されません。
 - DPSM が DTEN で開始した場合、エラー (DTIMEOUT) 後、FIFO が空になったとき、IDMAEN = 0 が FIFORST でリセットされたときに、DPSM はアイドル状態に移行します。
- **R_W** 状態：データパスがバスの読出しを待ちます。
 - 読出しウェイト停止ビット (RWSTOP) がセットされると、DPSM は Wait_R 状態に移行し、受信タイムアウトを開始します。
 - CPSM Abort 信号がセットされ、FIFO が空になり、IDMAEN = 0 の場合、FIFORST でリセットされるのを待つ場合、アイドル状態に移行し、DABORT フラグがセットされます。

- **受信状態:** データバスがカードからシリアルデータを受信します。データは数バイトにまとめられ、データ FIFO に書き込まれます。データ制御レジスタで選択された転送モード (DTMODE) に応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
 - ブロックモードでは、データブロックサイズ (DBLOCKSIZE) のデータバイト数を受信すると、DPSM は CRC コードを受信するまで待ちます。
 - SDIO マルチバイトモードでは、データブロックサイズ (DATALENGTH) のデータバイト数を受信すると、DPSM は CRC コードを受信するまで待ちます。
 - a) 受信した CRC コードが内部生成された CRC コードと一致する場合、DPSM は次の状態へ移行します。
 - RWSTART = 1 かつ DATACOUNT > 0 の場合、R_W 状態となり、DBCKEND フラグがセットされます。
 - そうでない場合、Wait_R 状態になります。
 - b) 受信した CRC コードが内部生成された CRC コードと一致しない場合、追加のデータ受信は阻止されます。
 - すべてのデータを受信していない場合 (DATACOUNT > 0)、CRC 失敗ステータスフラグ (DCRCFAIL) をセットして、DPSM は受信状態のままとなります。
 - すべてのデータを受信した場合 (DATACOUNT = 0)、FIFO が空になるのを待ってから、CRC 失敗ステータスフラグ (DCRCFAIL) をセットして、DPSM はアイドル状態に移行します。
 - ストリームモードでは、DPSM はデータカウンタ DATACOUNT > 0 の間、データを受信します。カウンタがゼロになると、シフトレジスタ内の残りのデータがデータ FIFO に書き込まれ、DPSM は Wait_R 状態へ移行します。
 - FIFO オーバーランエラーが発生した場合、DPSM は FIFO オーバーランエラーフラグ (RXOVERR) をセットして、追加のデータ受信は阻止されます。DPSM は受信状態のままとなります。
 - CPSM Abort 信号を受信した場合:
 - DATACOUNT = 0 でデータの最後の 2 ビットまでに CPSM Abort 信号を受信した場合、転送がアボートされます。シフトレジスタに残っているデータがデータ FIFO に書き込まれ、FIFO が空になって、IDMAEN = 0 の場合、FIFORST でリセットされるのを待ってから、DPSM はアイドル状態に移行し、DABORT フラグがセットされます。
 - DATACOUNT = 0 で転送の最後の 2 ビット処理中または以降に CPSM Abort 信号を受信した場合、転送は正常に完了します。DPSM は受信状態のままとなり、DABORT フラグは生成されません。
 - DPSM が DTEN で開始した場合、エラー (DATACOUNT > 0 時の DCRCFAIL または RXOVERR) 後、FIFO が空になったとき、IDMAEN = 0 が FIFORST でリセットされたときに、DPSM はアイドル状態に移行します。
- **Wait_S 状態:** データバスが FIFO からのデータが使用できるようになるのを待ちます。
 - データカウンタ DATACOUNT > 0 の場合、データ FIFO エンプティフラグ (TXFIFOE) がネゲートされるまで待ってから、DTHOLD はセットされず、送信状態へ移行します。
 - データカウンタ (DATACOUNT) が 0 の場合、DPSM はアイドル状態に移行します。
 - DTHOLD が無効である場合、DATAEND フラグがセットされます。
 - DTHOLD が有効である場合、DHOLD フラグがセットされます。
 - DTHOLD がセットされ、DATACOUNT > 0 の場合
 - IDMA が有効である場合、DBCKEND フラグがセットされ、続けて FIFO が一掃され、さらに DPSM はアイドル状態に移行し、DHOLD フラグがセットされます。
 - IDMA が無効である場合、DBCKEND フラグがセットされます。FIFO がソフトウェアによって FIFORST でリセットされるのを待って、DPSM はアイドル状態に移行し、DHOLD フラグを発行します。
 - DTHOLD がセットされ、DATACOUNT = 0 の場合、転送は正常に完了します。

- CPSM Abort 信号を受信した場合
 - CPSM Abort 信号を DATACOUNT = 0 でデータの最後の 2 ビットまでに受信した場合、転送はアボートされ、FIFO が空になるまで、IDMAEN = 0 の場合は FIFORST でリセットされるまで待つてから、DPSM はアイドル状態に移行し、DABORT フラグがセットされます。
 - DATACOUNT = 0 で転送の最後の 2 ビット処理中または以降に CPSM Abort 信号を受信した場合、通常動作を続け、転送が正常に完了するため、DABORT フラグはセットされません。

注： DPSM は、 N_{WR} タイミングの要件を満たすため、少なくとも 2 クロック周期の間、Wait_S 状態を保ちます。ここで、 N_{WR} は、カードレスポンスの受信とホストからのデータ転送の開始との間のクロックサイクル数です。

- **送信状態：** DPSM はカードへのデータの送信を開始します。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモード、SDIO マルチバイトモードまたはストリームモードのいずれかとなります。
 - ブロックモードでは、データブロックサイズ (DBLOCKSIZE) のデータバイト数を送信すると、DPSM は内部生成された CRC コードと終了ビットを送信して、ビジー状態へ移行し、送信タイムアウトを開始します。
 - SDIO マルチバイトモードでは、データブロックサイズ (DATALENGTH) のデータバイト数を送信すると、DPSM は内部生成された CRC コードと終了ビットを送信して、ビジー状態へ移行し、送信タイムアウトを開始します。
 - ストリームモードでは、DPSM はデータカウンタ DATACOUNT > 0 の間、カードにデータを送信します。データカウンタが 0 に達した場合、ビジー状態に移行し、送信タイムアウトを開始します。
DATACOUNT に従って最後のストリームバイトを送信する前に、DPSM は送信 CMD 信号にトリガを発行します。この信号は、CPSM が何らかのペンディングコマンドを送信するために使用します。(例：CMD12 送信停止コマンド)
 - FIFO アンダーランエラーが発生した場合、DPSM は FIFO アンダーランエラーフラグ (TXUNDERR) をセットします。DPSM は送信状態のままとなります。
 - CPSM Abort 信号を受信した場合
 - DATACOUNT = 0 で転送の最後の 2 ビットまでに CPSM Abort 信号を受信した場合、転送はアボートされます。DPSM は終了ビットの前に最後のデータビットを送信します。FIFO は無効化/一掃され、DPSM はビジー状態に移行し、ビジーではなくなるのを待つてから DABORT フラグをセットします。
 - DATACOUNT = 0 で転送の最後の 2 ビット処理中または以降に CPSM Abort 信号を受信した場合、転送は正常に完了し、DABORT フラグはセットされません。
- **ビジー状態：** DPSM は期待されている場合に CRC ステータストークンを待つて、ビジー以外の信号を待ちます。
 - CRC ステータストークンが期待され、「エラーなし送信」を示す場合や期待された CRC が
ない場合：
 - SDMMC_D0 がローでない（カードがビジーでない）場合は Wait_S 状態へ移行します。
 - カードがビジーで、SDMMC_D0 がローの場合、ビジー状態のままとなります。
 - CRC ステータストークンが期待され、「エラーあり送信」を示す場合
 - すべてのデータが送信されていない場合 (DATACOUNT > 0) DPSM は、ビジー以外になるまで待つてから、CRC 失敗ステータスフラグ (DCRCFAIL) がセットされます。FIFO は無効化/一掃され、DPSM はビジー状態のままとなります。
 - すべてのデータが送信された場合 (DATACOUNT = 0) DPSM は、ビジー以外になるまで待つてから、CRC 失敗ステータスフラグ (DCRCFAIL) がセットされ、DPSM はアイドル状態に移行します。

- DPSM がビジー状態の間に CRC ステータス (Ncrc) タイムアウトが発生した場合には、DPSM はデータタイムアウトフラグ (DTIMEOUT) をセットして、ビジー状態のままとなります。
- DPSM がビジー状態の間にビジータイムアウトが発生した場合には、DPSM はデータタイムアウトフラグ (DTIMEOUT) をセットして、ビジー状態のままとなります。
- ビジー状態で CPSM Abort 信号を受信した場合：
 - DATACOUNT > 0 で CRC レスポンスの最後の 2 ビットまでに CPSM Abort 信号を受信した場合、データ転送はアボートされます。DPSM は、ビジー以外になり、FIFO が無効化／一掃されるまで待ってから、アイドル状態に移行して、DABORT フラグがセットされます。
 - DATACOUNT = 0 のとき、または CRC が期待されていなくて、DATACOUNT = 0 で、DTIMEOUT エラーがないときに、CRC レスポンスの最後の 2 ビット処理中または以降に CPSM Abort 信号を受信した場合、転送は正常に完了できるので、DPSM はビジー状態のままで、DABORT フラグは生成されません。
 - DTIMEOUT エラーが発生したときに CPSM Abort 信号を受信した場合、DPSM は、ビジーではなくなるまで、そして FIFO が無効化／一掃されるまで待ってから、アイドル状態に移行し、DABORT フラグがセットされます。
- 送信状態でのアボートによってビジー状態に移行したとき、DPSM は、ビジーではなくなるまで待ってから、アイドル状態に移行し、DABORT フラグがセットされます。
- DPSM が DTEN で開始した場合、エラー (DATACOUNT > 0 時の DCRCFAIL または DTIMEOUT) 後、FIFO がリセットされたときに、DPSM はアイドル状態に移行します。
- SDMMC_D0 でビジーになっているため、DPSM が開始した場合、ビジー以外になるまで待ってから、ビジー終了ステータスフラグ (BUSYD0END) がセットされ、DPSM はアイドル状態に移行します。

データタイム (DATETIME) は、DPSM が Wait_R またはビジー状態のときに、データブロック終了ビットまたはデータ読出しコマンド終了ビット、又は R1b レスポンス後の 2 サイクル有効になり、以下の場合にデータタイムアウトエラー (DTIMEOUT) を生成します。

- データ送信時にタイムアウトが発生します。
 - CRCステータスが期待されており、8 SDMMC_CK サイクル内で受信するスタートビットがない場合、DTIMEOUT フラグがセットされます。
 - ビジー状態がプログラムされたタイムアウト期間により長くなる場合、DTIMEOUT フラグがセットされます。
- データ受信時にタイムアウトが発生します。
 - DATACOUNT > 0 で受信するデータがまだあり、プログラムされたタイムアウト期間の前に受信するスタートビットがない場合、DTIMEOUT フラグがセットされます。
- R1b レスポンス後に、タイムアウトが発生します。
 - ビジー状態がプログラムされたタイムアウト期間により長くなる場合、DTIMEOUT フラグがセットされます。

DATETIME = 0 の場合

- 受信時、スタートビットはデータブロック終了ビットまたはデータ読出しコマンド終了ビットの 2 サイクル後に、存在する必要があります。
- 送信時、CRC トークン終了ビットまたはストリームデータ終了ビットの後 2 サイクルで、ビジーがタイムアウトします。
- R1b レスポンス後、レスポンス終了ビットの後 2 サイクルで、ビジーがタイムアウトします。

データは、カードからホスト (送信) へ、またはホストからカード (受信) へ転送できます。データは、SDMMC_Dn データラインを介して転送され、FIFO に格納されます。

表 223. データトークンフォーマット

説明	スタートビット	データ ⁽¹⁾	CRC16	終了ビット	DTMODE
ブロックデータ	0	(DBLOCKSIZE、DATALENGTH)	はい	1	00
SDIO マルチバイト	0	(DATALENGTH)	はい	1	01
eMMC ストリーム	0	(DATALENGTH)	不可	1	10

1. 転送するデータの総数は DATALENGTH で指定します。ブロックデータの場合、各ブロックのデータの数 DBLOCKSIZE で指定します。

データトークンフォーマットは、適宜レジスタビット DTMODE で選択されます。

データパスは、表 224 に示すようにステータスフラグおよび関連付けられたクリアビットを実装します。

表 224. データパスステータスフラグとクリアビット

フラグ		説明
DATAEND	TX	CRC が OK で、ビジーが終了し、DTHOLD = 0 かつ DATACOUNT = 0 になったときにすべてのデータ転送終了時にセットされます。(DPSM は Wait_S からアイドルに移行)
	RX	CRC が OK で、すべてのデータを読み出したときにすべてのデータ転送終了時にセット。(DATACOUNT = 0 かつ FIFO が空)。(DPSM は Wait_R からアイドルに移行)
	ブート	
DCRCFAIL	TX	FAIL で、ビジーが終了したときに CRC 終了時にセット。(送信するデータがまだあり、CPSM Abort を待っている場合、DPSM はビジーのまま) (すべてのデータが送信された場合、または DPSM が DTEN で開始された場合、DPSM はビジーからアイドル状態に移行)
	RX	FAIL で、FIFO が空のときに CRC 終了時にセット。(受信するデータがまだあり、CPSM Abort を待っている場合、DPSM は受信のまま) (すべてのデータが受信された場合、または DPSM が DTEN で開始された場合、DPSM は受信からアイドル状態に移行)
	ブート	
ACKFAIL	ブート	失敗した場合に、ブート確認応答の終了時にセット。(DPSM は Wait_Ack のままで、CPSM Abort を待機)
DTIMEOUT	CMD R1b	タイムアウト前に受信したビジー終了がないコマンドレスポンスの後にセット。(DPSM はビジーのままで、CPSM Abort を待機)
	TX	Ncrc 内で受信した CRC トークンスタートビットがなかった場合、またはタイムアウト前に受信したビジー終了がなかった場合にセット。(DPSM はビジーのままで、CPSM Abort を待機) (DPSM が DTEN で開始されたときは、アイドル状態に移行) 注：ビジータイムアウト前に CRC コマンド失敗時に DCRCFAIL フラグもセットされる場合があります。
	RX	タイムアウト前に受信したスタートビットがないときにセット。(DPSM は Wait_R のままで、CPSM Abort を待機) (DPSM が DTEN で開始されたときは、アイドル状態に移行)
ACKTIMEOUT	ブート	タイムアウト前に受信したスタートビットがないときにセット。(DPSM は Wait_Ack のままで、CPSM Abort を待機)
DBCKEND	TX	DTHOLD = 1 および IDMAEN = 0 の場合：CRC が OK で、ビジーが終了したとき、データ転送が不完全だったとき (DATACOUNT > 0) にデータブロック転送終了時にセット。(DPSM はビジーから Wait_S に移行)
	RX	RWSTART = 1 の場合：CRC が OK のとき、データ転送が不完全だったとき (DATACOUNT > 0) にデータブロック転送終了時にセット。(DPSM は受信から R_W に移行)
	ブート	

表 224. データパスステータスフラグとクリアビット (続き)

フラグ		説明
DHOLD	TX	DTHOLD = 1 の場合 : CRC が OK で、ビジーが終了したときにデータブロック転送終了時にセット。(DPSM は Wait_S からアイドルに移行)
	RX	DTHOLD = 1 の場合 : CRC が OK で、すべてのデータが読み出された (FIFO が空になった) とき、データ転送が不完全だったとき (DATACOUNT > 0) にデータブロック転送終了時にセット。(DPSM は Wait_R からアイドルに移行)
DABORT	CMD R1b	CPSM によって CPSM Abort イベントが送信され、ビジーが終了したとき。(DPSM はビジーからアイドルに移行)
	TX	転送の最後の 2 ビットまでに CPSM によって CPSM Abort イベントが送信されたとき。(DPSM は任意の状態からアイドルに移行)
	RX	
	ブート	
BUSYD0END	CMD R1b	タイムアウト前にビジー終了になったときのコマンドレスポンスの後にセット。(DPSM はビジーからアイドルに移行)
DPSMACT		データ転送中。(DPSM がアイドル状態ではない)

データパスエラー処理は、表 225 に示します。

表 225. データパスエラー処理

誤差	DPSM の状態	原因	カード動作	ホスト動作	DPSM 動作
タイムアウト	Wait_Ack	時間内に確認応答なし	不明	カードサイクル パワー	Wait_Ack のまま (RCC.SDMMCxRST レジスタビットで SDMMC をリセット)
	Wait_R	時間内にスタート ビットなし	不明	データ受信の停止 送信停止コマンドの 送信	CPSM Abort で アイドルに移行
			不明	ブート手順の停止	
	ビジー	長すぎるビジー (データ転送による)	不明	データ受信の停止 送信停止コマンドの 送信	
		長すぎるビジー (R1b による)	不明	リセットコマンドの 送信	
CRC	受信	送信エラー	追加データの送信	データ受信の停止 送信停止コマンドの 送信	CPSM Abort で アイドルに移行
CRC ステータス	ビジー	否定ステータス	追加データの無視	データ送信の停止	CPSM Abort で アイドルに移行
		送信エラー	追加データを待つ	送信停止コマンドの 送信	
確認応答ステータス	Wait_Ack	送信エラー	ブートデータの送信	ブート手順の停止	CPSM Abort で アイドルに移行

表 225. データパスエラー処理 (続き)

誤差	DPSM の状態	原因	カード動作	ホスト動作	DPSM 動作
オーバーラン	受信	FIFO フル	追加データの送信	データ受信の停止 送信停止コマンドの送信	CPSM Abort で アイドルに移行
アンダーラン	送信	FIFO エンプティ	追加データの受信	データ送信の停止 送信停止コマンドの送信	CPSM Abort で アイドルに移行

データ FIFO

データ FIFO (ファーストインファーストアウト) サブユニットには、送受信データバッファが含まれます。シングル FIFO は、DTDIR ビットで選択されたとおり送信または受信に使用されます。FIFO には、32 ビット幅、16 ワードの深さのデータバッファと、制御ロジックがあります。データ FIFO は AHB クロックドメイン (sdmmc_hclk) で動作するため、SDMMC クロックドメイン (SDMMC_CK/sdmmc_rx_ck) のサブユニットからの信号はすべて再同期されます。

FIFO は次の状態のいずれかになり得ます。

- 送信 FIFO は、カードにデータを送信すると、送信ロジックとデータバッファを参照します。(DTDIR = 0)
- 受信 FIFO は、カードからデータを受信すると、受信ロジックとデータバッファを参照します。(DTDIR = 1)

正常に完了した FIFO からの SDMMC データ転送の終了は、データパスサブユニットによって駆動される DATAEND フラグで示されます。FIFO からの誤った (アボートされた) SDMMC データ転送は、データパスサブユニットによって駆動されるエラーフラグ (DCRCFAIL、DTIMEOUT、DABORT) のいずれかか、FIFO 制御によって駆動される FIFO エラーフラグ (TXUNDERR、RXOVERR) のいずれかで示されます。

データ FIFO には、次の方法でアクセスすることができます。表 226 を参照してください。

表 226. データ FIFO アクセス

データ FIFO アクセス	IDMAEN
AHB スレーブインタフェース経由でファームウェアから	0
AHB マスタインタフェース経由で IDMA から	1

送信 FIFO :

データは、DPSM が有効化されたとき (DPSMACT = 1) に、送信 FIFO に書き込むことができます。

IDMAEN = 1 の場合、FIFO は IDMA によって完全に処理されます。

IDMAEN = 0 の場合、FIFO は AHB スレーブインタフェース経由でファームウェアによって制御されます。送信 FIFO は、連続アドレスを介してアクセス可能です。送信 FIFO には、読出しポイントが示すデータワードを保持するデータ出力レジスタが含まれています。データパスサブユニットがシフトレジスタへロードすると、読出しポイントがインクリメントされ、新しいデータが出力されます。送信 FIFO は、次の方法で処理されます。

1. データ長は DATALENGTH に、ブロック長は DBLOCKSIZE に書き込みます。
 - ブロックデータ転送 (DTMODE = 0) の場合、DATALENGTH は DBLOCKSIZE の整数倍でなければなりません。
2. SDMMC を送信モードにセットします (DTDIR = 0)。

- 送信モードで FIFO を設定します。
- 3. 次のいずれかでデータ転送を有効化します。
 - CMDTRANS ビットをセットして CPSM からコマンドを送信
 - DTEN ビットをセット
- 4. (DPSMACT = 1) の場合、データを FIFO に書き込みます。
 - DPSM は、FIFO がフル (TXFIFO = 1) または DATALENGTH で示された数字になるまで Wait_S 状態のままとなります。
 - SDMMC は、FIFO が空でない限りデータを送信し続けます。データ転送中のハードウェアフロー制御が、FIFO アンダーランを防ぐために使用されます。
- 5. データを FIFO に書き込みます。
 - FIFO をソフトウェアで処理する場合、FIFO がハーフエンプティ (TXFIFOHE フラグ) になるまで待って、FIFO がフル (TXFIFO = 1) になるまで、または最後のデータが書き込まれるまでデータを FIFO に書き込みます。
 - FIFO を IDMA で処理する場合、IDMA が FIFO データを転送します。
- 6. 最後のデータが書き込まれると、データの終了 (DATAEND フラグ) を待ちます。
 - SDMMC がすべてのデータを完全に送信して、DPSM が無効になります (DPSMACT = 0)。

IDMAEN = 0 のとき、データ転送エラーまたは転送保留が発生した場合、ファームウェアは FIFO への書き込みを停止して、FIFORST レジスタビットで FIFO を一掃してリセットする必要があります。

送信 FIFO ステータスフラグを表 227 に示します。

表 227. 送信 FIFO ステータスフラグ

フラグ	説明
TXFIFO	すべての送信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
TXFIFOE	送信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
TXFIFOHE	半分以上の送信 FIFO ワードが空のとき、ハイにセットされます。
TXUNDERR	アンダーランエラーが発生すると、ハイにセットされます。このフラグは、SDMMC クリアレジスタへの書き込みによってクリアされます。

受信 FIFO :

データは、DPSM が有効になっているとき (DPSMACT = 1) に、受信 FIFO から読み出すことができます。

IDMAEN = 1 の場合、FIFO は IDMA によって完全に処理されます。

IDMAEN = 0 の場合、FIFO は AHB スレーブインタフェース経由で ファームウェア によって制御されます。データバスサブユニットがワードデータを受信すると、書き込みデータバス上にデータが出力されます。書き込み操作が完了すると、書き込みポインタがインクリメントされます。読出し側では、読出しポインタの現在値が示す FIFO ワードの内容が、読出しバス上に出力されます。受信 FIFO は、連続アドレスを介してアクセス可能です。

受信 FIFO は、次の方法で処理されます。

1. データ長は DATALENGTH に、ブロック長は DBLOCKSIZE に書き込みます。
 - ブロックデータ転送 (DTMODE = 0) の場合、DATALENGTH は DBLOCKSIZE の整数倍でなければなりません。
2. SDMMC を受信モードにセットします (DTDIR = 1)。
 - 受信モードで FIFO を設定します。

3. 次のいずれかで DPSM 転送を有効化します。
 - CMDTRANS ビットをセットして CPSM からコマンドを送信
 - DTEN ビットをセット
4. DPSMACT = 1 の場合、FIFO はデータを受信する準備ができています。
 - DPSM は受信したデータを FIFO に書き込みます。
 - SDMMC は、FIFO がフルでない限りデータを受信し続けます。データ転送中のハードウェアフロー制御が、FIFO オーバーランを防ぐために使用されます。
5. FIFO からデータを読み出します。
 - FIFO をソフトウェアで処理する場合、FIFO がハーフフル (RXFIFOHF フラグ) になるまで待って、FIFO がエンプティ (RXFIFOE = 1) になるまで FIFO からデータを読み出します。
 - 最後のデータを受信すると、FIFO がエンプティ (DATAEND = 1) になるまで FIFO からデータを読み出します。
 - FIFO を IDMA で処理する場合、IDMA が FIFO データを転送します。
6. SDMMC がすべてのデータを完全に受信して、DPSM が無効になります (DPSMACT = 0)。

IDMAEN = 0 のとき、データ転送保留が発生した場合、ファームウェアは FIFO がエンプティになるまで残りのデータを読み出し、FIFO を FIFORST レジスタビットでリセットする必要があります。これにより、DPSM はアイドル状態 (DPSMACT = 0) に移行します。

IDMAEN = 0 のとき、データ転送エラーが発生した場合、ファームウェアは FIFO の読み出しを停止し、FIFORST レジスタビットで FIFO を一掃してリセットする必要があります。これにより、DPSM はアイドル状態 (DPSMACT = 0) に移行します。

受信 FIFO ステータスフラグを表 228 に示します。

表 228. 受信 FIFO ステータスフラグ

フラグ	説明
RXFIFOE	すべての受信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
RXFIFOE	受信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
RXFIFOHF	半分以上の受信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
RXOVERR	オーバーランエラーが発生すると、ハイにセットされます。このフラグは、SDMMC クリアレジスタへの書き込みによってクリアされます。

CLKMUX ユニット

CLKMUX は、受信したデータおよびコマンドレスポンスと使用するクロックのソース `sdmmc_rx_ck` を選択します。受信データのクロックソースは、次のうちのクロック制御レジスタビット `SELCLKRX` で選択できます。

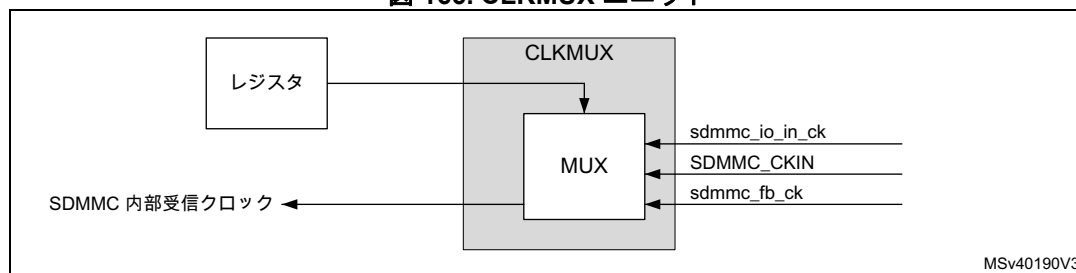
- `sdmmc_io_in_ck` バスマスタメインフィードバッククロック
- `SDMMC_CKIN` 外部バスフィードバッククロック
- `sdmmc_fb_ck` バス調整フィードバッククロック

`sdmmc_io_in_ck` は、外部ドライバがない場合に DS およびハイスピードで選択されます。

`SDMMC_CKIN` は、外部ドライバがある場合に SDR12、SDR25、SDR50 および DDR50 で選択されます。

デバイスの DLYB ブロックが SDR104、HS200 で使用され、オプションで SDR50 モードおよび DDR50 モードで使用されている場合、`sdmmc_fb_ck` クロック入力を選択しなければなりません。

図 166. CLKMUX ユニット



sdmmc_rx_ck ソースの変更は、CPSM および DPSM がアイドル状態のときに行う必要があります。

24.5.5 SDMMC AHB スレーブインタフェース

AHB スレーブインタフェースは、割込みリクエストを生成し、SDMMC アダプタレジスタとデータ FIFO にアクセスします。このインタフェースは、データバス、レジスタデコーダ、および割込みロジックで構成されています。

SDMMC FIFO

FIFO のアクセスは、ワードアクセスのみに制限されます。

- 送信 FIFO モードの場合
 - データは、DATALENGTH に応じたすべてのデータが転送されるまで、ワード (32 ビット) で FIFO に書き込まれます。DATALENGTH が 4 の整数倍ではない場合、最後に残ったデータ (1、2、または 3 バイト) は 1 ワード転送で書き込まれます。
- 受信 FIFO モードの場合
 - データは、DATALENGTH に応じたすべてのデータが転送されるまで、ワード (32 ビット) で FIFO から読み出されます。DATALENGTH が 4 の整数倍ではない場合、最後に残ったデータ (1、2、または 3 バイト) は 0 値バイトでパディングされた 1 ワード転送で書き込まれます。

ハーフワードまたはバイトアクセスで FIFO にアクセスすると、AHB バスフォールトが生成されます。

SDMMC 割込み

割込みロジックは、マスクされていないステータスフラグの少なくとも 1 つがアクティブのときにアサートされる割込みリクエスト信号を生成します。割込みを生成する条件を選択できるように、マスクレジスタが提供されています。ステータスフラグは、対応するマスクフラグがセットされている場合に割込みリクエストを生成します。一部のステータスフラグは、クリアレジスタで間接的なクリアが必要です。

24.5.6 SDMMC AHB マスタインタフェース

AHB マスタインタフェースは、SDMMC IDMA を使用したメモリと FIFO の間でのデータ転送に使用されます。

SDMMC IDMA

DMA (Direct Memory Access) は、SDMMC FIFO とメモリの間でハイスピード転送を行うために使用します。AHB マスタはシステムバスのバンド幅を最適化します。SDMMC 内部 DMA (IDMA) には、送信または受信に使用するチャンネルが 1 つ搭載されています。

IDMA は、IDMAEN ビットで有効になり、8 ビートのバースト転送をサポートします。

- 送信バースト転送モードの場合：
 - データは、DATALENGTH に応じたすべてのデータが転送されるまで、FIFO がバースト転送の数に対してエンプティであればいつでもメモリからバーストでフェッチされます。DATALENGTH がバーストサイズの整数倍ではない場合、残りのバーストサイズより少ないデータはシングル転送モードで転送されます。DATALENGTH が 4 の整数倍ではない場合、最後に残ったデータ (1、2、または 3 バイト) は 1 ワード転送でフェッチされます。
- 受信バースト転送モードの場合：
 - データは、DATALENGTH に応じたすべてのデータが転送されるまで、FIFO にバースト転送の数がある場合に限りメモリにバーストで格納されます。DATALENGTH がバースト転送の整数倍ではない場合、残りのバーストサイズより少ないデータはシングル転送モードで転送されます。DATALENGTH が 4 の整数倍ではない場合、最後に残ったデータ (1、2、または 3 バイト) はハーフワード転送またはバイト転送で格納されます。

さらに、IDMA には、IDMABMODE で選択される次のチャンネル設定があります。

- シングルバッファチャンネル
- リンクされたリストのチャンネル

シングルバッファチャンネル

シングルバッファ設定では、メモリ側のデータにベースアドレス IDMABASE から始まる直線的な方法でアクセスします。IDMA がすべてのデータの転送を終了し、DPSM が転送を完了したときに、DATAEND フラグがセットされます。

リンクされたリストのチャンネル

リンクされたリストの設定 IDMA MODE = 1 では、メモリ側のデータは、ベースアドレス IDMABASE に位置するリンクされたバッファから順次アクセスされます。メモリバッファのサイズは、IDMA SIZE によって定義されます。バッファサイズは、バーストサイズの整数倍でなければなりません。ビット ULA は、リンクされたリストの新しいバッファ設定を、リンクされたリストのテーブルからロードする必要があるかどうかを示すために使用されます。現在のリンクされたリスト項目の ULA ビットがセットされていると、リンクされたリストの新しい設定がロードされます。

最初のリンクされたリスト項目の設定は、ファームウェアで SDMMC レジスタに直接プログラムします。

IDMA では、IDMA SIZE に従って 1 つのリンクされたリストバッファのすべてのデータの転送を終了し、リンクされたリスト項目の ULA ビットが設定されていると、リンクされたリストのテーブルから新たなリンクされたリスト項目をロードし、次のリンクされたリストバッファからデータの転送を継続します。IDMA が IDMA SIZE と ULA に従ってすべてのデータの転送を終了し、DPSM が DATALENGTH に従って転送を完了したときに、DATAEND フラグがセットされます。

以下の場合、リンクされたリストによって、転送するデータより多くのバッファスペースが用意されており、現在のリンクされたリストバッファのデータが完全に転送されていないことになります。

- ULA ビットがセットされていて、DATALENGTH に従ったすべての SDMMC データが転送されてしまっている (DATAEND フラグ)。
- 転送エラー (DATACOUNT > 0 の場合の DCRCFAIL、RXOVERR、TXUNDERR) が発生している。
- 転送が保留されている (DTHOLD)

上記のすべての場合、IDMA のリンクされたリストは停止され、FIFO は一掃/リセットされます。新しい SDMMC 転送を開始または再開する前に、ソフトウェアは正しい IDMABASE および IDMA SIZE を使用して新たなリンクされたリストを初期化する必要があります。

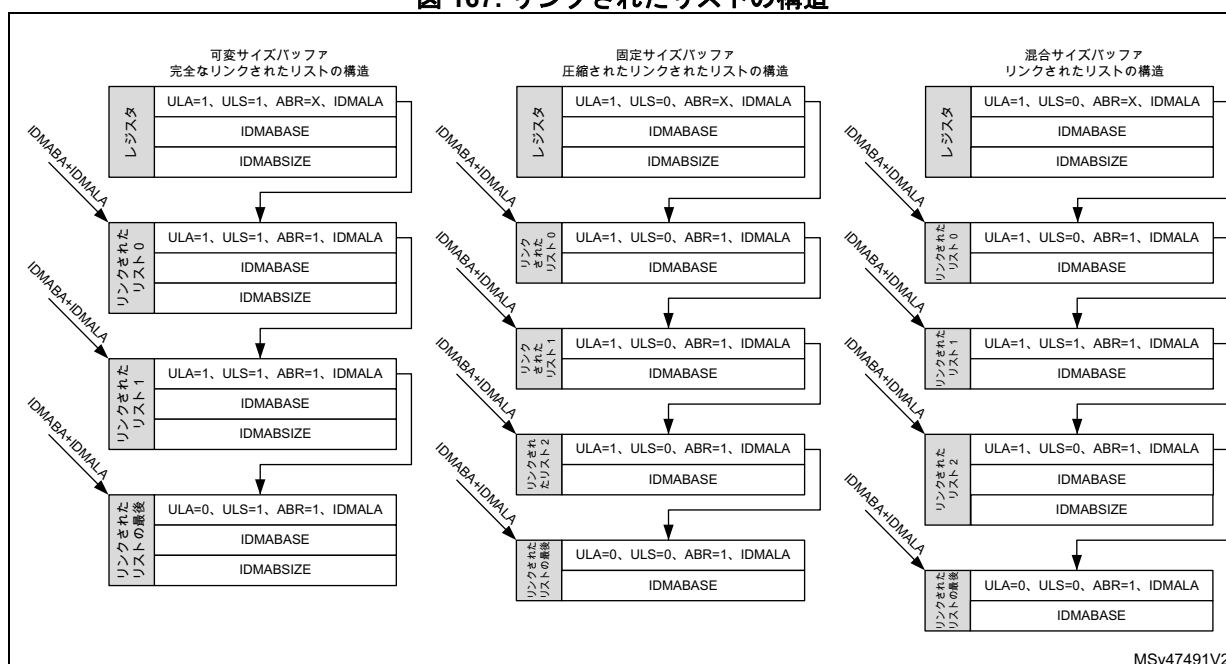
IDMA 転送エラーが発生した場合（[セクション : IDMA 転送エラー管理](#)を参照）、またはリンクされたリストに十分なバッファスペースが用意されていない場合、

- リンクされたリストは ULA = 0 で終了し、最後のリンクされたリストバッファのデータはすべて転送されていますが、DATALENGTH に従ったすべての SDMMC データが転送されている訳ではありません。SDMMC の転送は停止し、IDMA 転送エラーが生成されます（[セクション : IDMA 転送エラー管理](#)を参照）。

特定のリンクされたリスト項目のベースアドレスは、リンクされたリストのベース IDMA 値とリンクされたリストのオフセット IDMALA レジスタ値を加算した値によって与えられます。

各リンクされたリスト項目の内容は ULS ビットで指定でき、これによってオプションで IDMA 転送サイズをロードできるようになり、3 ワードのリンクされたリスト構造になります。IDMA 転送サイズがロードされない（つまり、固定サイズのバッファの場合、IDMA 転送サイズ値と IDMALA 値のみを含む、圧縮された縮小 2 ワードのリンクされたリスト構造を使用することができます。

図 167. リンクされたリストの構造



両方のリンクされたリストの項目の構造を 1 つのリストに混在させることに制限はありません。これにより、必要な場合にのみ IDMA 転送サイズを更新できます。

リンクされたリストのバッファが転送され、現在のバッファの ULA = 1 のときはいつでも、end-of-linked-list-buffer-transfer-complete（リンクされたリスト終端のバッファ転送完了）割り込み（IDMA 転送完了）が生成されることがあります（割り込みが有効な場合）。

リンクされたリストの確認応答

SDMMC 転送中に、ソフトウェアがリンクされたリストを動的に更新する場合、確認応答バッファレディ（ABR）ビットによって新たなリンクされたリストバッファの利用可能性を確認できます。

ABR により新たなリンクされたリストのバッファの準備ができたことが確認応答されると、IDMA は新たなリンクされたリストのバッファからのデータの転送を続行します。

ABR により新たなリンクされたリストのバッファの準備ができていないことが示されると、IDMA 転送エラーが生成されます（[セクション : IDMA 転送エラー管理](#)を参照）。いつ IDMA 転送エラーが発生

するかによって、通常は TXUNDERR または RXOVERR エラーが生成されます。リンクされたリストのバッファが時間内に確認応答されない場合、SDMMC 転送は停止します。

ソフトウェアでプログラムされたレジスタ情報からリンクされたリストを開始する場合、ABR 情報は無視されます。最初のリンクされたリストのバッファは、SDMMC 転送を開始する前に準備完了にしておく必要があります。

IDMA 転送エラー管理

IDMA 転送エラーは次の場合に発生します。

- 予約済みアドレス空間の読取りまたは書込み時（データまたはリンクされたリストの情報に関して）。
- 受信した SDMMC データを保存するためのリンクされたリストのバッファ空間がなくなった場合。
- すべてのリンクされたリストのバッファのデータが転送されたが、さらにまだ SDMMC データを送信する必要がある場合。
- リンクされたリストのバッファの使用可能性が確認応答されない場合。

IDMA 転送エラー時、その後の IDMA 転送は無効になり、IDMATE フラグがセットされ、ハードウェアフロー制御は無効になります。いつ IDMA 転送エラーが発生するかによって、通常は TXUNDERR または RXOVERR エラーが生成されます。

IDMATE フラグの動作は、IDMA 転送エラーが SDMMC 転送中にいつ発生したかに依存します。

- IDMA 転送エラーは、あらゆる SDMMC 転送エラー（TXUNDERR、RXOVERR、DCRCFAIL、または DTIMEOUT）の前に検出されます。
 - IDMATE フラグは、SDMMC 転送エラーフラグと同時にセットされます。
 - TXUNDERR、RXOVERR、DCRCFAIL、または DTIMEOUT 割込みが生成されます。
- IDMA 転送エラーは、STOP_TRANSMISSION コマンド中に検出されます。
 - IDMATE フラグは、DABORT フラグと同時にセットされます。
 - DABORT 割込みが生成されます。
- IDMA 転送エラーは、SDMMC 転送（DHOLD または DATAEND）の終了後に検出されます。
 - IDMATE フラグは、SDMMC 転送の終了時にセットされます。
 - SDMMC 転送終了割込みが生成され、DHOLD または DATAEND フラグがセットされます。

IDMATE は、他の SDMMC 転送割込み時に生成されます（TXUNDERR、RXOVERR、DCRCFAIL、DTIMEOUT、DABORT、DHOLD、または DATAEND）。

24.5.7 AHB と SDMMC_CK のクロック関係

AHB は、少なくとも SDMMC バス帯域幅の 3 倍以上にする必要があります。すなわち、SDR50 4 ビットモード（50 MB/秒）の場合、sdmmc_hclk の最低周波数は 37.5 MHz（150 MB/秒）とします。

表 229. AHB と SDMMC_CK のクロック周波数の関係

SDMMC バスモード	SDMMC バス幅	最大 SDMMC_CK [MHz]	最小 AHB クロック [MHz]
eMMC DS	8	26	19.5
eMMC HS	8	52	39
eMMC DDR52	8	52	78
eMMC HS200	8	200	150
SD DS / SDR12	4	25	9.4

表 229. AHB と SDMMC_CK のクロック周波数の関係 (続き)

SDMMC バスモード	SDMMC バス幅	最大 SDMMC_CK [MHz]	最小 AHB クロック [MHz]
SD ハイスピード / SDR25	4	50	18.8
SD DDR50	4	50	37.5
SD SDR50	4	100	37.5
SD SDR104	4	208	78

24.6 カード機能詳細

24.6.1 SD I/O モード

以下の機能は、SDMMC 固有の操作です。

- SDIO 割込み
- SDIO サスペンド／レジューム操作（書込みおよび読出しサスペンド）
- クロックの停止による SDIO 読出しウェイト操作
- SDMMC_D2 信号による SDIO 読出しウェイト操作

表 230. SDIO 特殊動作制御

動作モード	SDIOEN	RWMOD	RWSTOP	RWSTART	DTDIR
割込み検出	1	X	X	X	X
サスペンド／レジューム操作	X	X	X	X	X
読出しウェイト SDMMC_CK クロック停止 (START)	X	1	0	1	1
読出しウェイト SDMMC_CK クロック停止 (STOP)	X	1	1	1	1
読出しウェイト SDMMC_D2 信号 (START)	X	0	0	1	1
読出しウェイト SDMMC_D2 信号 (STOP)	X	0	1	1	1

SD I/O 割込み

SD インタフェースのピン 8 (4 ビットモードで SDMMC_D1 と共有) には、SD I/O カードからホストへの割込みを可能にする割込み機能が用意されています。割込みの使用は、各カードまたはカード内の各機能でオプションとなっています。SD I/O 割込みはレベル対応であるため、割込みラインはホストによって認識され、割込みが処理されるか、割込み期間の終了によってネゲートされるまで、アクティブ（ロー）に保たれる必要があります。ホストが割込み処理を終えると、SD I/O カードの内部レジスタの該当するビットへの I/O 書込みによって割込みステータスビットがクリアされます。すべての SD I/O カードの割込み出力はアクティブローであり、アプリケーションはすべてのデータライン (SDMMC_D[3:0]) 上に、外部プルアップ抵抗を提供する必要があります。

SD 1 ビットモードでは、ピン 8 が割込み機能 (IRQ) 専用となり、割込みに関するタイミングの制約がありません。

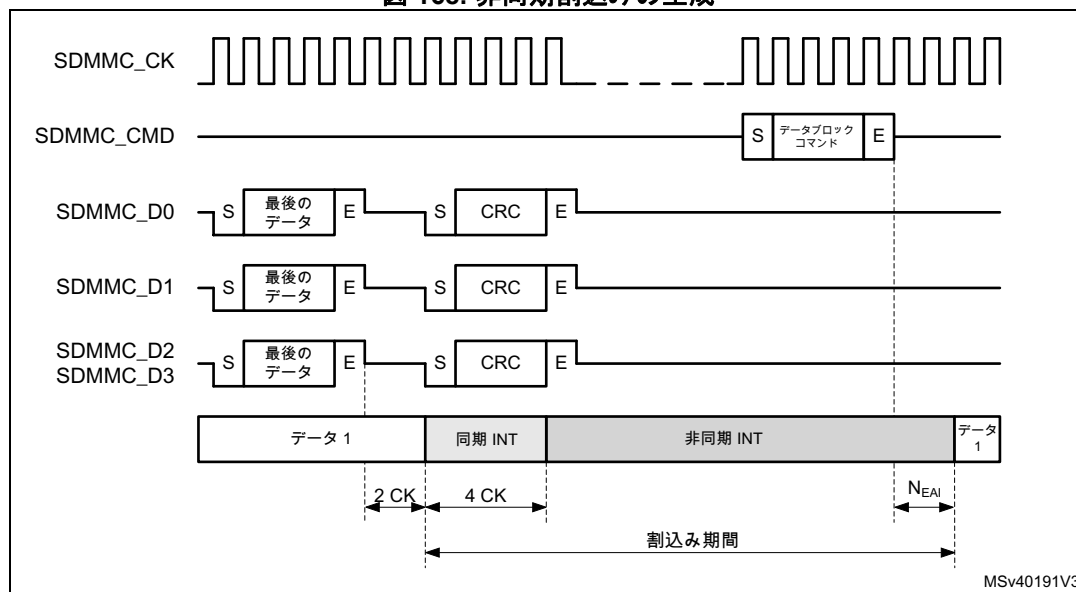
SD 4 ビットモードでは、ホストは、割込み期間に限って、割込み検出器にピン 8 (SDMMC_D1/IRQ) のレベルをサンプリングします。その他の状況では、ホスト割込みはこの値を無視します。割込み期

間は、割込みがカードで有効化され、SDIOEN ビットがセットされた際に開始します。表 230 のレジスタ設定を参照してください。

4 ビットモードでは、カードはカード CCCR レジスタ SAI および EAI ビットで示されたとおりに同期割込みまたは非同期割込みを生成できます。

- 同期割込みは、SDMMC_CK をアクティブにする必要があります。
- 非同期割込みは、SDMMC_CK が最後のデータブロック後のカード割込み期間の開始後 4 サイクル停止したときに生成できます。

図 168. 非同期割込みの生成



割込み期間のタイミングは、バス速度モードに依存します。

レジスタビット BUSPEED で選択する DS、ハイスピード、SDR12、SDR25 のモードでは、割込み期間が SD クロックと同期します。

- 割込み期間は、データブロックを転送するコマンド (CMDTRANS ビットをセットして送信されたコマンド) の終了ビット以降の次のクロック、または DTEN ビットがセットされたときに終了します。
- 割込み期間は、データブロックの完了後に 2 SDMMC_CK で再開します。
- データブロックの差で、割込み期間は 2 SDMMC_CK サイクルに制限されます。

注： DTEN は SD および eMMC カードでデータ転送の開始に使用してはなりません。

図 169. 同期割込み期間のデータ読出し

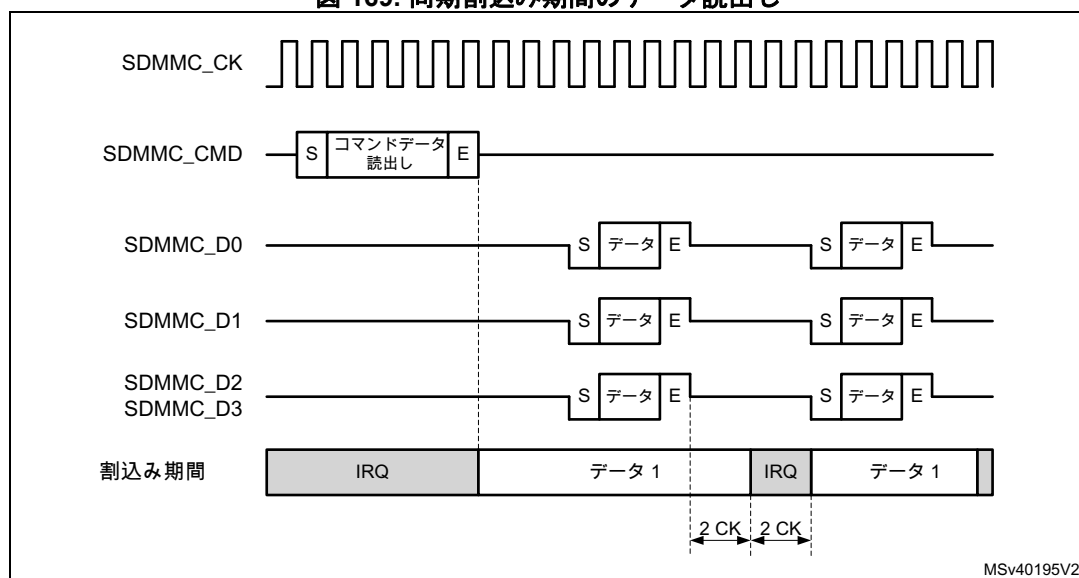
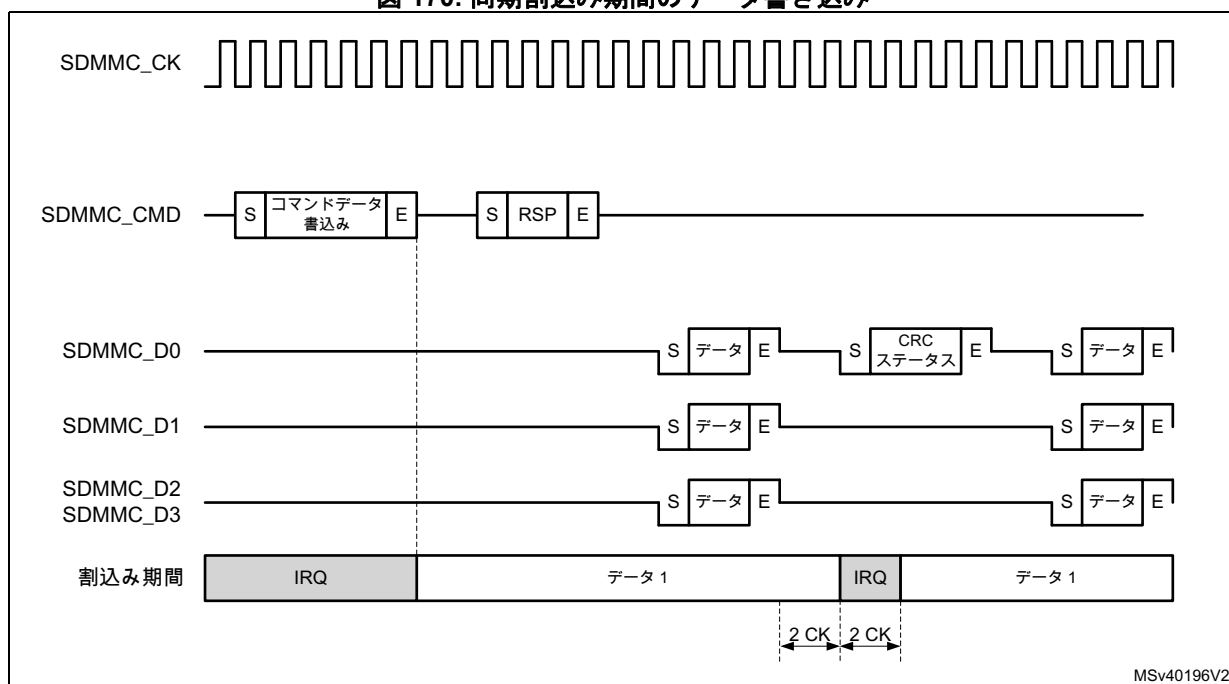


図 170. 同期割込み期間のデータ書き込み



レジスタビット BUSSPEED で選択する SDR50、SDR104、および DDR50 では、カードからホストへの伝搬遅延によって、割込み期間は非同期になります。

- カードの割込み期間は、データブロックを転送するコマンド (CMDTRANS ビットをセットして送信されたコマンド) の終了ビット以降の 0~2 SDMMC_CK サイクル後、または DTEN ビットがセットされたときに終了します。ホストでは、割込み期間はデータブロックを転送するコマンドの終了ビット以降に終了します。コマンド終了ビット後 1~2 サイクルで発行されたカードの割込みは、この割込み期間中ホストによって検出されません。
- カードの割込み期間は、最後のデータブロックの完了後に 2~4 SDMMC_CK で再開します。ホストは、最後のデータブロック後に常に 2 サイクルで割込み期間を再開します。

- データブロックの間で割込み期間はありません。

注: DTEN は SD および eMMC カードでデータ転送の開始に使用してはなりません。

図 171. 非同期割込み期間のデータ読み出し

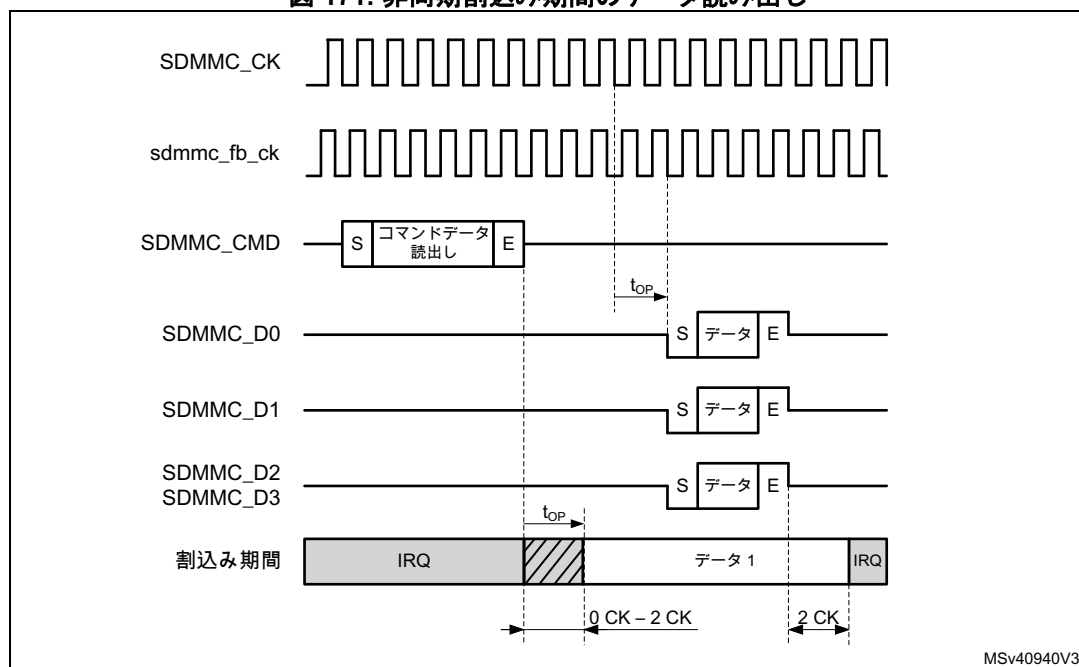
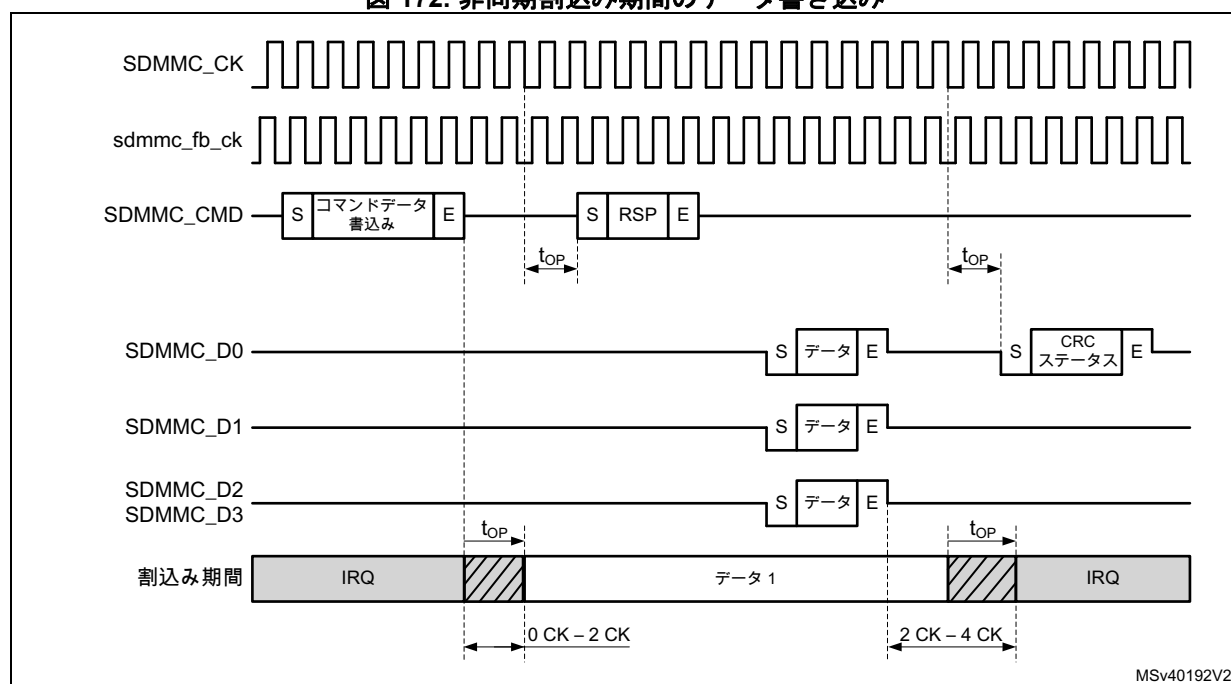


図 172. 非同期割込み期間のデータ書き込み



オープンエンド複数ブロックデータを転送し、DTMODE「STOP_TRANSMISSION コマンドで終わるブロックデータ転送」を使用する場合、SDMMC は最後のデータブロックから CMD12 STOP_TRANSMISSION コマンドの終了まで割込み期間をマスクします。

割込み期間は、メモリと I/O の両方の操作に適用されます。

4 ビットモードでは、表 231 に従って他の信号と区別できます。

表 231. 4 ビットモードのスタート、割込み、CRC ステータスの信号検出

SDMMC データライン	START	割込み	CRC ステータス
SDMMC_D0	0	1 または CRC ステータス	0
SDMMC_D1	0	0	X
SDMMC_D2	0	1 または 読出しウェイト	X
SDMMC_D3	0	1	X

SD I/O のサスペンドとレジューム

この機能は、SDIO バージョン 4.00 以降ではサポートされていません。

マルチファンクション SD I/O または I/O とメモリの両方の機能を持つカードでは、eMMC/SD バスへのアクセスを共有する複数のデバイス (I/O とメモリ) があります。複数デバイス間でホストへのアクセスを共有するため、SD I/O およびコンボカードはオプションでサスペンド/レジュームの概念を実装します。カードがサスペンド/レジュームをサポートするときには、ホストは 1 つの機能またはメモリへのデータ転送を一時的に中止 (サスペンド) して、別の機能またはメモリへの、より優先順位の高い転送用にバスを開放できます。この優先順位の高い転送が完了すると、本来の転送がアポートした所から再開 (レジューム) されます。

バス上でサスペンド/レジューム操作を行うには、ホストは以下の手順を実行します。

1. SDMMC_D[3:0] ラインを現在使用している機能を特定します。
2. 優先順位が低いトランザクションや時間のかかるトランザクションをサスペンドするように要求します。
3. トランザクションのサスペンドの完了を待ちます。
4. 優先順位の高いトランザクションを開始します。
5. 優先順位の高いトランザクションの完了を待ちます。
6. サスペンドされたトランザクションをリストアします。

サスペンドコマンドを受信したカードは、現在のバスステータスで応答します。バスがカードによってサスペンドされていた場合のみ、バスステータスがサスペンドを完了したことを示します。

次のような個別のサスペンドケースの条件があります。

- データ転送開始前に許可したサスペンドリクエスト。
- (データが同時に転送されたため) 許可されなかったサスペンドリクエスト。ホストは許可されるまでリクエストをチェックし続けます。(データ転送がサスペンドしました)。
- 書き込みビジー中のサスペンドリクエスト。
- 複数書き込み中のサスペンドリクエスト。
- 読出しウェイト中のサスペンドリクエスト。

ホストでバスが開放されたかどうか確認するためには、サスペンドが完了したというサスペンドリクエストのステータスを確認する必要があります。

サスペンドリクエストのレスポンスのバスステータスがサスペンド完了を示すと、カードはバスを開放します。このとき、サスペンドされた動作の状態を、その後で他の動作を開始できるように保存する必要があります。

サスペンドコマンドは CMDSUSPEND ビットをセットして送信する必要があります。これにより、バスがサスペンドされた（レスポンスビット BS = 0）ときに、サスペンドコマンドレスポンスの後に割込み期間を開始できます。

ハードウェアは、サスペンド操作を再開させるときに転送する残りのデータの数を保存しません。転送されたデータを判定して、正しい残りのデータバイト数で再開することはファームウェア次第となります。

カードからデータを受信している間、SDMMC は読出しデータブロック終了（Wait_R の DPSM）以後の読出し操作をサスペンドできます。カードからサスペンド確認応答レスポンスを受信した後、ファームウェアで次の手順を実行する必要があります。

1. 通常の受信プロセスは、DTHOLD ビットをセットすることで停止する必要があります。
 - a) FIFO の残りのデータバイト数を、受信 FIFO がエンプティになる（RXFIFOE フラグがセットされる）まで読み出し、IDMAEN = 0 の場合は、FIFO を FIFORST でリセットする必要があります。
2. すべてのデータが FIFO から読み出されたことと、サスペンドが完了したことの確認は、DHOLD フラグによって示されます。
 - a) 動作再開時に読み出されるべき残りのデータバイト数（データブロックの倍数）は、DATACOUNT によって示される残りのバイト数から判定する必要があります。

注： サスペンド手順中に DTIMEOUT フラグが発生した場合、これは無視する必要があります。

カードからデータの受信を再開するには、ファームウェアで次の手順を実行する必要があります。

1. 残りのデータバイト数（データブロックの倍数）を DATALENGTH にプログラムする必要があります。
2. DPSM を、DTHOLD ビットでデータを受信するように設定する必要があります。
3. CMDTRANS ビットと CMDSUSPEND ビットをセットして、レジュームコマンドを CPSM から送信する必要があります。これによって、データ転送が再開されたとき（レスポンスビット DF = 1）に割込み期間が終了し、DPSM が有効になり、その後、カードがデータ送信を再開します。

カードにデータを送信している間、SDMMC は書込みデータブロック CRC ステータス終了（ビジーの DPSM）以後の書込み操作をサスペンドできます。カードにサスペンドコマンドを送信する前に、ファームウェアで次の手順を実行する必要があります。

1. DHOLD フラグ（IDMAEN = 0 の場合は DBCKEND フラグも）を有効にします。
2. DTHOLD をセットすることで、DPSM が新しいデータブロックの送信を開始できないようにする必要があります。
3. IDMAEN = 0 の場合：DBCKEND フラグを受信すると、データ転送が停止します。ファームウェアは FIFO への入力を停止でき、その後 FIFO を FIFORST でリセットする必要があります。FIFOに残っているバイトは、操作再開時に再書込みする必要があります。
4. DHOLD フラグを受信すると、データ転送が停止します。再開時に書き込まれるべき残りのデータバイト数は、DATACOUNT によって示される残りのバイト数から判定する必要があります。
5. カードをサスペンドするために、CMDSUSPEND ビットをセットしてサスペンドコマンドを CPSM から送信する必要があります。これにより、バスがサスペンドされた（レスポンスビット BS = 0）ときに、サスペンドコマンドレスポンスの後に割込み期間を開始できます。

カードへのデータ送信を再開するには、ファームウェアで次の手順を実行する必要があります。

1. 残りのデータバイト数を DATALENGTH にプログラムする必要があります。
2. DTHOLD をセットして DPSM を送信用に設定し、CMDTRANS ビットと CMDSUSPEND ビットをセットして CPSM にレジュームコマンドを送信させることで DPSM を有効にする必要があります。これにより、割込み期間が終了し、データ転送が開始されます。DPSM は、SDMMC_D0

がビジーの信号を発していない場合 Wait_S 状態に移行し、ビジーの信号を発している場合はビジー状態に移行します。

3. IDMAEN = 1 の場合：残りのバイトを転送するために IDMA を再プログラムする必要があります。
4. IDMAEN = 0 の場合：ファームウェアが残りのデータを FIFO に入力することを開始する必要があります。

SD I/O 読出しウェイト

ブロックのギャップの間にデータ転送を一時停止する 2 つの方法があります。

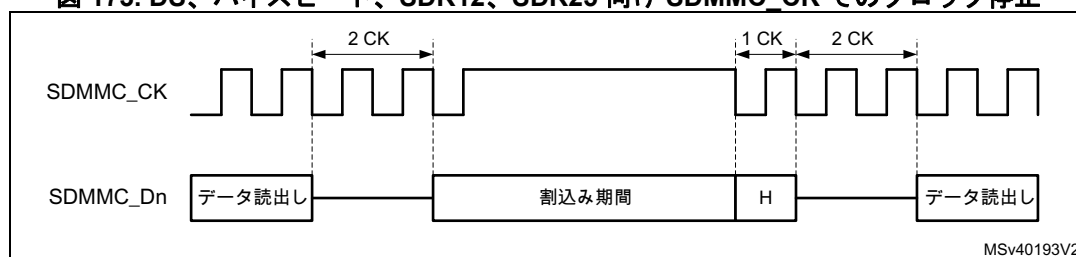
1. SDMMC_CLK の停止
2. SDMMC_D2 に読出しウェイト信号を使用

SDMMC は、表 230 に従ったレジスタ設定で読出しウェイトを実行できます。

SDMMC 動作モード (DS、ハイスピード、SDR12、SDR25) または (SDR50、SDR104、DDR) に応じて、各方法には異なる特性があります。

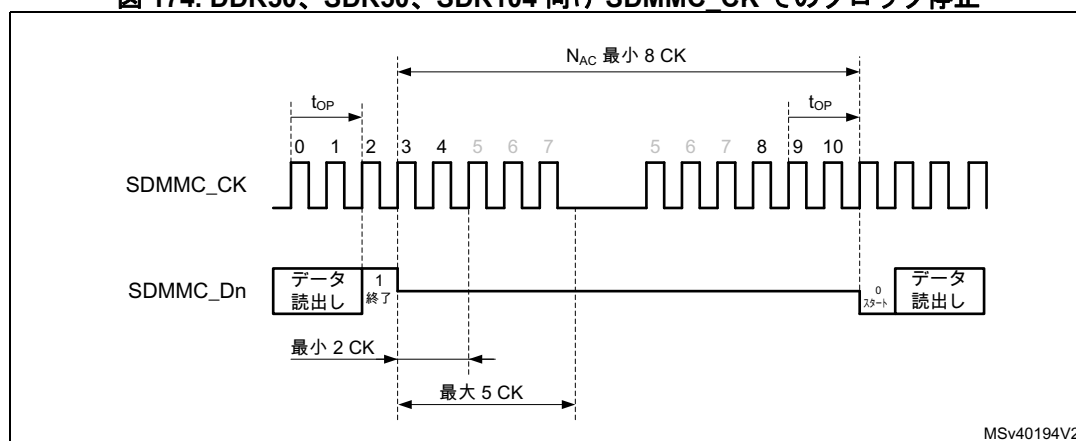
DS、ハイスピード、SDR12、SDR25 の場合 SDMMC_CLK を停止することで読出し動作を一時停止するタイミングは、終了ビットの 2 SDMMC_CLK サイクル後に、SDMMC_CLK を停止できます。準備が整うと、ホストはクロックを再開することで再開します。図 173 を参照してください。

図 173. DS、ハイスピード、SDR12、SDR25 向け SDMMC_CLK でのクロック停止



SDR50、SDR104、DDR50 の場合 SDMMC_CLK を停止することで読出し動作を一時停止するタイミングは、終了ビット後の最小 2 SDMMC_CLK サイクルおよび最大 5 SDMMC_CLK サイクルで、SDMMC_CLK を停止できます。準備が整うと、ホストはクロックを再開することで再開します。図 174 を参照してください。(DDR50 モードでは、クロックラインがローの場合、SDMMC_CLK は立下がりエッジの後でのみ停止する必要があります。)

図 174. DDR50、SDR50、SDR104 向け SDMMC_CLK でのクロック停止



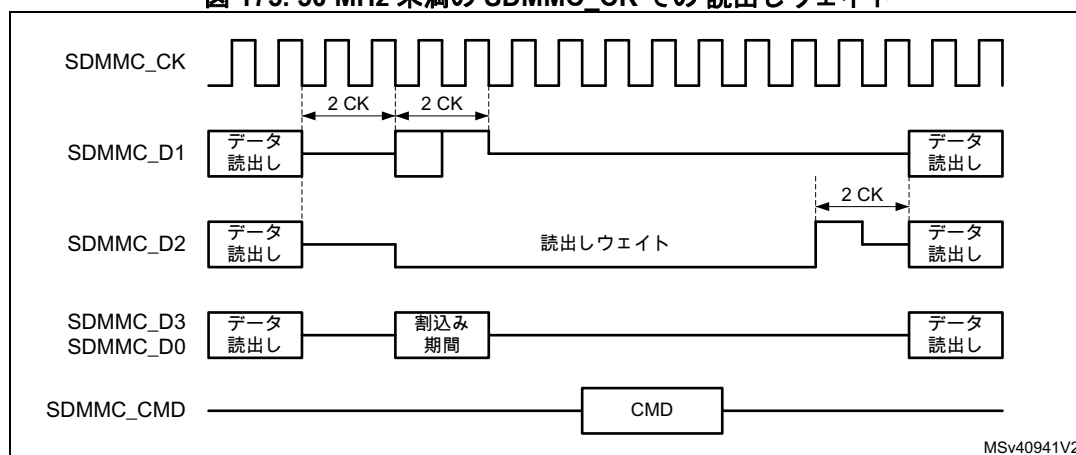
読出しウェイトの SDMMC_CK クロック停止では、RWSTART がセットされると、DSPM は現在受信したデータブロックの CRC の終了ビット後にクロックを停止します。RWSTOP ビットに 1 が書き込まれた後、クロックは再開し、その後、DSPM はカードからのスタートビットを待ちます。

SDMMC_CK が停止されると、コマンドをカードに発行できません。読出しウェイトインターバル中、SDMMC は SDMMC_D1 上の SDIO 割込みを検出できます。

オプションの SDMMC_D2 での読出しウェイト信号 (RW) 操作は、SD の 1 ビットおよび 4 ビットモードについてのみ定義されます。読出しウェイト操作によって、ホストは複数のレジスタ (IO_RW_EXTENDED、CMD53) を読み出しているカードにデータ転送を一時的にストールするよう信号を送り、その間にホストは SD I/O デバイス内の機能にコマンドを送信できます。カードが読出しウェイトプロトコルをサポートしているかどうかを判断するには、ホストは内部カードレジスタの機能ビットをテストする必要があります。

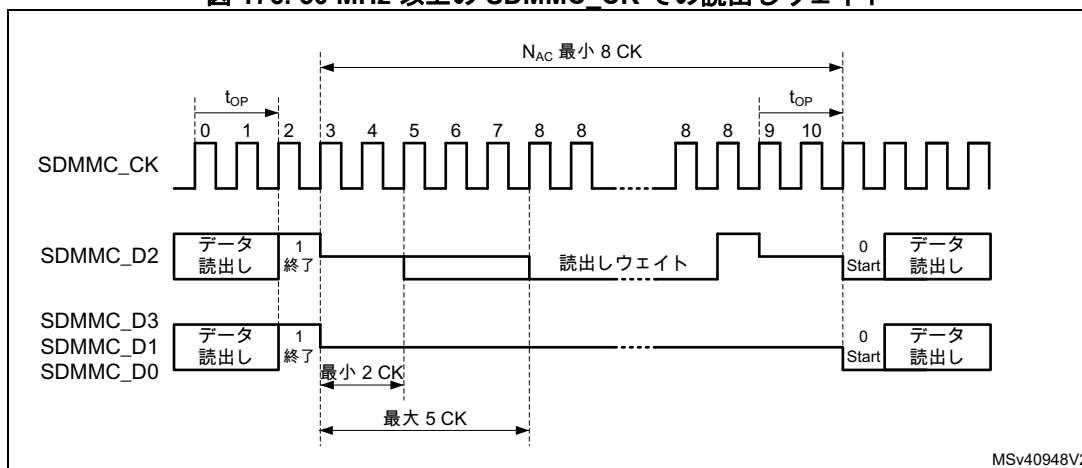
50 MHz 未満の SDMMC_CK での読出しウェイトのタイミング (DS、ハイスピード、SDR12、SDR25) は、カードによって SDMMC_D1 上に生成された割込み期間に基づきます。割込み期間中に SDMMC_D2 をローでアサートすることで、ホストはカードに読出しウェイトに移行するようリクエストします。読出しウェイトを終了するには、ホストは SDMMC_D2 をハイインピーダンスにする前の 1 SDMMC_CK サイクル中に SDMMC_D2 をハイにする必要があります。図 175 を参照してください。

図 175. 50 MHz 未満の SDMMC_CK での 読出しウェイト



50 MHz 以上の SDMMC_CK の SDR50、SDR104 および DDR50 の場合、カードは SDMMC_D2 での読出しウェイトリクエストを非同期イベントとして処理します。最小 2 SDMMC_CK サイクルおよび最大 5 SDMMC_CK サイクルの後に SDMMC_D2 をローでアサートすることで、ホストはカードに読出しウェイトに移行するようリクエストします。読出しウェイトを終了するには、ホストは SDMMC_D2 をハイインピーダンスにする前の 1 SDMMC_CK サイクル中に SDMMC_D2 をハイにする必要があります。ホストは SDMMC_CK クロックに基づいて SDMMC_D2 を立ち上げる必要があります (図 176 を参照してください)。

図 176. 50 MHz 以上の SDMMC_CK での読出しウェイト



読出しウェイトの SDMMC_D2 信号では、RWSTART がセットされると、DSPM は現在受信したデータブロックの CRC の終了ビット後に SDMMC_D2 を駆動します。SDMMC_D2 上の読出しウェイト信号は、RWSTOP ビットに 1 を書き込むと削除されます。DSPM は、さらに 2 SDMMC_CK クロックサイクルの間、R_W 状態にとどまり、1 クロックサイクルで SDMMC_D2 を 1 に（SDIO 仕様に従って）駆動し、その後、DSPM はカードからのスタートビットを待ちます。

SDMMC_D2 上で読出しウェイト信号がある間、コマンドをカードに発行できます。読出しウェイトインターバル中、SDMMC は SDMMC_D1 上の SDIO 割込みを検出できます。

24.6.2 CMD12 送信タイミング

CMD12 は、データ転送の停止／アボートに使用され、送信停止コマンドの終了ビット後 2 クロックサイクルでカードのデータ送信は終了します。

表 232. CMD12 の使用事例

データ操作	送信停止コマンド CMD12 の説明
SDMMC ストリーム書込み	送信停止コマンドを送信すると、データ転送が停止／アボートされます。
SDMMC オープンエンド複数ブロックの書込み	送信停止コマンドを送信すると、データ転送が停止／アボートされます。カードがエラーを検出した場合、ホストは送信停止コマンドを送信して操作をアボートする必要があります。
事前定義されたブロック数での SDMMC ブロック書込み	この種類の複数ブロック書込み終了時に、送信停止コマンドは必要ありません。（カードが最後のブロックを受信した後に送信停止コマンドを送信すると、不正なコマンドとみなされます。）カードがエラーを検出した場合、ホストは送信停止コマンドを送信して操作をアボートする必要があります。
SDMMC ストリーム読出し	送信停止コマンドを送信すると、データ転送が停止／アボートされます。
SDMMC オープンエンド複数ブロックの読出し	送信停止コマンドを送信すると、データ転送が停止／アボートされます。カードがエラーを検出した場合、ホストは送信停止コマンドを送信して操作をアボートする必要があります。
事前定義されたブロック数での SDMMC ブロック読出し	この種類の複数ブロック読出し終了時に、送信停止コマンドは必要ありません。（カードが最後のブロックを送信した後に送信停止コマンドを送信すると、不正なコマンドとみなされます。）トランザクションは、送信停止コマンドを送信することでアボートできます。カードがエラーを検出した場合、ホストは送信停止コマンドを送信して操作をアボートする必要があります。

すべてのデータ書込みと読み出しは、送信停止コマンド CMD12 でいつでもアボートできます。データ転送進行中は、次のデータアボート手順が適用されます。

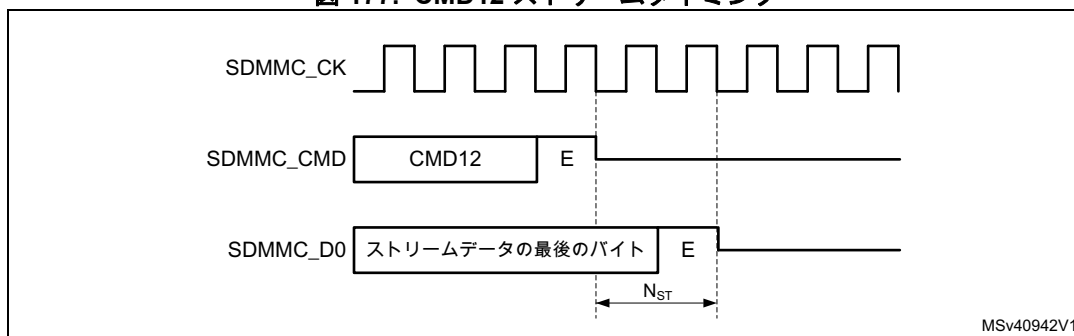
1. レジスタで CMD12 送信停止コマンドをロードして、CMDSTOP ビットをセットします。
 - a) これにより、コマンドが DPSM に送信されると、CPSM Abort 信号が生成されます。
2. 直ちにコマンドを送信するために CPSM を設定します (WAITPEND をクリアします)。
 - a) カードはデータを送信すると、送信停止コマンドの終了ビットの 2 サイクル後にデータ転送を停止します。
データが送信されていない場合は、カードは新しいデータの送信を開始しません。
 - b) ホストはデータを送信すると、送信停止コマンドの終了ビット後に、終了ビットが続く最後の 1 つのデータビットを送信します。
データが送信されていない場合は、ホストは新しいデータの送信を開始しません。
3. IDMAEN = 0 の場合、FIFO は FIFORST でリセットされる必要があります。
 - a) カードにデータを書き込む場合。CMDREND フラグで、ファームウェアは FIFO へのデータ書込みを停止する必要があります。その後、FIFO を FIFORST でリセットする必要があります。これにより FIFO が一掃されます。
 - b) カードからデータを読み出す場合。CMDREND フラグで、ファームウェアは FIFO から残りのデータを読み出す必要があります。その後、FIFO を FIFORST でリセットする必要があります。
4. IDMAEN = 1 の場合、ハードウェアが FIFO を管理します。
 - a) カードにデータを書き込む場合。CPSM Abort 信号で、ハードウェアは IDMA を停止し、その後、FIFO が一掃されます。
 - b) カードからデータを読み出す場合。CPSM Abort 信号で、ハードウェアは IDMA に、FIFO から RAM に残りのデータを転送するように命令します。
5. FIFO がエンプティになるか、リセットされると、DABORT フラグが生成されます。

ストリーム動作と CMD12

転送する最後のバイトの後にストリーム転送を停止するためには、CMD12 終了ビットのタイミングを最後のバイトのデータストリーム終了と合わせて送信する必要があります。次の書込みストリームデータ手順が適用されます。

1. DPSM で、DTMODE = MCC ストリームデータ転送の場合、ストリームデータを初期化します。
2. CMDTRANS = 1 で CPSM から WRITE_DATA_STREAM コマンドを送信します。
3. CMDSTOP ビットをセットして、コマンドレジスタで CMD12 をプリロードします。
4. CPSM を設定して、最後のデータ (DATALENGTH による) のペンディング待機 (WAITPEND = 1) 終了後にのみコマンドを送信します。
5. CPSM を有効にして、STOP_TRANSMISSION コマンドを送信すると、ストリームデータ終了ビットとコマンド終了ビットが揃えられます。
 - a) DATALENGTH > 5 バイトの場合、データ転送終了ビットと合わせるために CPSM でコマンド CMD12 が待たされます。
 - b) DATALENGTH < 5 バイトの場合、コマンド CMD12 が先に開始され、DPSM はデータ転送終了を CMD12 の終了ビットと合わせるために Wait_S 状態にとどまります。
6. 書込みストリームデータは、WAITPEND ビットをクリアすることでいつでもアボートできます。これにより、プリロードされた CMD12 が直ちに送信され、書込みデータストリームが停止されます。

図 177. CMD12 ストリームタイミング



最後のバイトの後に読出しストリームの転送を停止するために、CMD12 終了ビットのタイミングを最後のデータストリームバイトの後に発生させる必要があります。次の読出しストリームデータ手順が適用されます。

1. すべてのデータが DPSM によって受信され、FIFO から読み出されるのを待ちます (DATAEND フラグ)。
 - a) カードがより多くのデータを送信しても、DPSM は DATALENGTH で示された以上のデータを受信しません。
2. CPSM で CMD12 を送信します。
 - a) CMD12 によりカードのデータ送信が停止されます。

注： カードがデータを送信し続けても、DATACOUNT = 0 の場合、SDMMC はカードからそれ以上のデータを受信しません。

ブロック動作と CMD12

データの終了時にブロック転送を停止するためには、最後のブロック終了ビットの後に CMD12 終了ビットを送信する必要があります。

カードにデータを書き込んでいる場合、書込みデータブロックの CRC トークン終了ビットの後に、CMD12 終了ビットを送信する必要があります。これには、データブロック送信タイミングと紐づいた CMD12 送信が必要になります。オープンエンド複数ブロック書込みを停止するには、次の手順が適用されます。

1. データ転送開始前に、DTMODE を「STOP_TRANSMISSION コマンドで終わるブロックデータ転送」にセットします。
2. すべてのデータが DPSM によって送信され、CRC トークンが受信されるのを待ちます (DATAEND フラグ)。
 - a) DPSM は DATALENGTH で示された以上のデータを送信しません。
3. CPSM で CMD12 を送信します。
 - a) CMD12 によってカードはアイドルモードにセットされます。

カードからデータを読み出している場合、カード読出しデータブロックの最後のデータビットと同じタイミングで最も速く CMD12 終了ビットを送信する必要があります。これには、データブロック受信タイミングと紐づいた CMD12 送信が必要になります。以下のオープンエンド複数ブロックの読出しデータブロックの手順が適用されます。

1. データ転送開始前に、DTMODE を「STOP_TRANSMISSION コマンドで終わるブロックデータ転送」にセットします。
2. すべてのデータが DPSM によって受信され、FIFO から読み出されるのを待ちます (DATAEND フラグ)。
 - a) カードがより多くのデータを送信しても、DPSM は DATALENGTH で示された以上のデータを受信しません。
3. CPSM で CMDSTOP ビットをセットして CMD12 を送信します。
 - a) CMD12 によってカードの追加データ送信が停止され、カードがアイドルモードにセットされます。カードによって進行中のあらゆるブロック転送がアボートされます。

注： カードがデータを送信し続けても、**DATACOUNT = 0** の場合、**SDMMC** はカードからそれ以上のデータを受信しません。

24.6.3 SLEEP (CMD5)

eMMC カードは、CMD5 によって SLEEP 状態と STANDBY 状態を切り替える場合があります。SLEEP 状態では、カードの消費電力が最小限になり、Vcc 電源がオフに切り替わります。

CMD5 (SLEEP) は、STANDBY 状態から SLEEP 状態への遷移を開始するために使用します。遷移フェーズ中、カードはビジーを示し、SDMMC_D0 をプルダウンします。カードが SDMMC_DO ラインのプルダウンを停止すると、SLEEP 状態に到達します。

カードを SLEEP 状態にセットするには、次の手順が適用されます。

1. BUSYD0END の割込みを有効にします。
2. CMD5 (SLEEP) を送信します。
3. BUSYD0END 割込みで、カードが SLEEP 状態になります。
4. Vcc 電源をオフに切り替えられます。

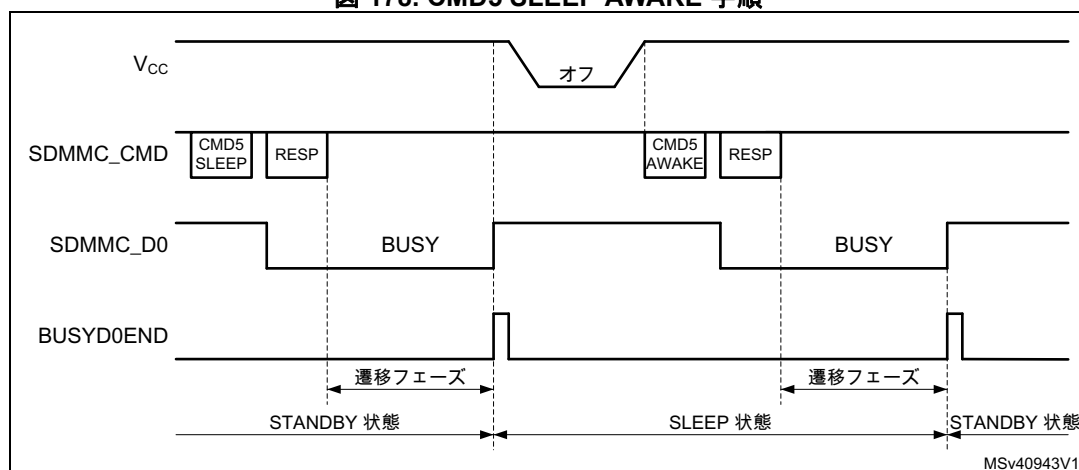
CMD5 (AWAKE) は、SLEEP 状態から STANDBY 状態への遷移を開始するために使用します。遷移フェーズ中、カードはビジーを示し、SDMMC_D0 をプルダウンします。カードが SDMMC_DO ラインのプルダウンを停止すると、STANDBY 状態に到達します。

カードを SLEEP 状態にセットするには、次の手順が適用されます。

1. Vcc 電源をオンに切り替え、最低動作レベルに到達するまで待ちます。
2. BUSYD0END の割込みを有効にします。
3. CMD5 (AWAKE) を送信します。
4. BUSYD0END 割込みで、カードが STANDBY 状態になります。

SLEEP 状態に到達した後でのみ、Vcc 電源をオフに切り替えられます。Vcc 電源は、CMD5 (AWAKE) が送信される前に再起動する必要があります。

図 178. CMD5 SLEEP AWAKE 手順



24.6.4 割込みモード (Wait-IRQ)

ホストとカードは、同時に割込みモード (Wait-IRQ) の移行と終了を行います。割込みモードでは、データ転送はありません。許可されるメッセージは、カードまたはホストからの割込みサービスリクエストレスポンスだけです。割込みモードが正しく動作するために、オープンドレインモードで実現可能な SDMMC_CMD データレート (これは静電容量負荷とプルアップ抵抗によって変わります) に従って SDMMC_CLK 周波数をセットする必要があります。CLKDIV は >1 にセットし、SETCLKRX で `sdmmc_io_in_ck` または `SDMMC_CLKIN` ソースを選択する必要があります。

ホストは、カードが STANDBY 状態であることを CMD40 (GO_IRQ_STATE) を発行する前に確認する必要があります。割込みレスポンスを待っている間、SDMMC_CLK クロック信号をアクティブに保つ必要があります。

割込みモード (IRQ 状態) のカード：

- 内部カード割込みイベントを待っています。イベントが発生すると、カードは割込みサービスリクエストレスポンスの送信を開始します。レスポンスはオープンドレインモードで送信されます。
- 内部カード割込みイベントを待っている間、カードは SDMMC_CMD ラインのスタートビットも監視します。スタートビットを検出すると、カードは割込みモードをアボートして、STANDBY 状態に切り替わります。

割込みモードのホスト (割込みを待つ CPSM ウェイトステート)：

- カード割込みサービスリクエストレスポンス (スタートビット) を待っています。
- カード割込みサービスリクエストレスポンスを待っている間、ホストは (WAITINT レジスタビットをクリアすることで) 割込みモードをアボートする場合があります、オープンドレインモードで `RCA = 0x0000` で割込みサービスリクエストレスポンス R5 を送信します。

割込みサービスリクエストレスポンスを送信すると、送信者ビット単位が SDMMC_CMD ビットストリームを監視します。割込みサービスリクエストレスポンスビットが SDMMC_CMD ラインのビットに対応しない送信者は、送信を停止します。送信側が複数の場合、完全な割込みサービスリクエストレスポンスの送信に成功するのは 1 つだけです。ホストが同時に送信した場合、ホストは送信ビット後に送信ができなくなります。

割込みモードを処理するには、次の手順が適用されます。

- オープンドレインモードで実現可能な SDMMC_CMD データレートに従って SDMMC_CLK 周波数をセットし、CLKDIV は >1 にセットし、SETCLKRX は `sdmmc_io_in_ck` を選択する必要があります。
- コマンドレジスタで CMD40 (GO_IRQ_STATE) をロードします。

3. WAITINT レジスタビットをセットして、割込みの待機を有効にします。
4. 直ちにコマンドを送信するために CPSM を設定します。
 - a) これにより、CMD40 が送信され、CPSM がウェイト状態で中止し、割込みサービスリクエストレスポンス待ちになります。
5. 割込みの待機状態（CPSM ウェイト状態）を終了する方法：
 - a) 割込みサービスリクエストレスポンスのスタートビットを検出すると、CPSM はレスポンスを受け取る受信状態に移行します。レスポンスの完全な受信は、CMDREND またはコマンド CRC エラーフラグによって示されます。
 - b) 割込みモードをアポートするためには、ホストは WAITINT レジスタビットをクリアします。これにより、ホストは独自に割込みサービスリクエストレスポンスを送信します。これで CPSM は受信状態に移行します。レスポンスの完全な受信は、CMDREND またはコマンド CRC エラーフラグによって示されます。

注： 同時送信した割込みサービスリクエストレスポンスのスタートビットが衝突した場合、ホストは送信ビット後にバスアクセスを失います。

24.6.5 ブート操作

ブート操作モードでは、ホストは 2 つのブート操作機能のうちいずれかでカードからブートデータを読み出せます。

1. 通常ブート（CMD ラインがローのまま）
2. 代替ブート（CMD0 を引数 0xFFFFF0 で送信）

ブートデータは、カードレジスタ設定に基づいて、次の設定オプションに応じて読み出すことができます。

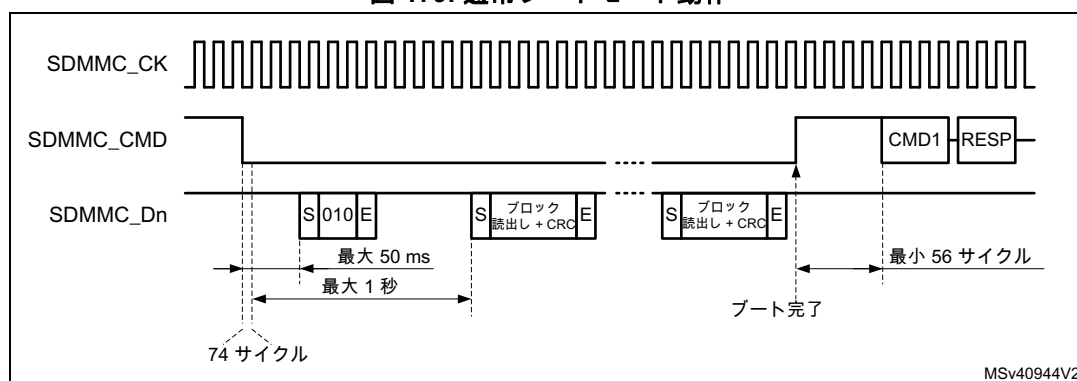
- ブートデータが読み出されるパーティション（EXT_CSD Byte[179]）
- ブートデータサイズ（EXT_CSD Byte[226]）
- ブート中のバス設定（EXT_CSD Byte[177]）
- カードからのブート確認応答の受信（EXT_CSD Byte[179]）

ブート確認応答が有効である場合、CMD ラインがローに移行、または CMD0 を引数 0xFFFFF0 でブートモードがリクエストされた後の 50 ms 以内に、カードが SDMMC_D0 でパターン 010 を送信します。ブート確認応答タイムアウト（ACKTIMEOUT）と確認応答ステータス（ACKFAIL）が提供されます。

通常ブート動作

カードのパワーアップまたはリセットから最初のコマンドが発行されるまで最低 74 クロックサイクルの間、SDMMC_CMD ラインがローで保たれる場合、カードはブートモードが開始されていると認識します。CMD ラインがローになった後 1 秒以内に、カードは SDMMC_Dn ラインで最初のブートコードデータの送信を開始します。ホストは、すべてのブートデータが読み出されるまで SDMMC_CMD ラインをローのままにしておく必要があります。ホストは、SDMMC_CMD ラインをハイにすることでブートモードを終了できます。

図 179. 通常ブートモード動作



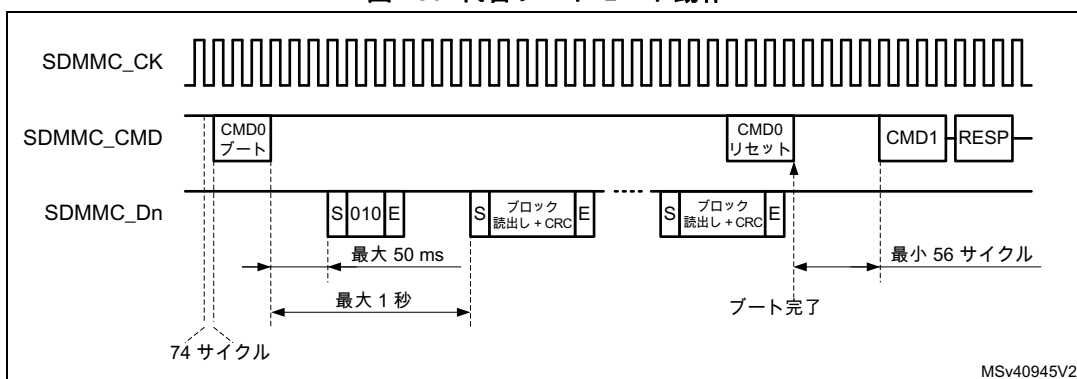
通常ブート手順を実行するには、次の手順が必要です。

1. カードをリセットします。
2. ブート確認応答をリクエストする場合、BOOTACKEN を有効にして、ACKTIME をセットし、ACKFAIL および ACKTIMEOUT 割込みを有効にします。
3. 受信モード (DTPDIR) で DPSM と DATALENGTH で受信するデータバイト数をセットしてデータ受信を有効にします。
4. ブートコマンドの終了確認に DTIMEOUT、DATAEND、および CMDSENT 割込みを有効にします。
5. BOOTMODE で通常ブート動作モードを選択し、BOOTEN でブートを有効にします。ブート手順は、CPSMEN で CPSM を有効にすることで開始されます。これにより次のことが発生します。
 - SDMMC_CMD がローに駆動されます。(BOOTMODE = 通常ブート)。
 - ACK タイムアウトが開始します。
 - DPSM が有効になります。
6. ブート確認応答の誤受信は、ACKFAIL フラグまたは ACKTIMEOUT フラグが有効時に検出できます。
 - 正しくないブート確認応答が受信された場合、ACKFAIL フラグが発生します。
 - ブート確認応答が時間内に受信されなかった場合、ACKTIMEOUT フラグが発生します。
7. すべてのブートデータが受信されると、DATAEND フラグが発生します。
 - データ CRC が失敗した場合、DCRCFAIL フラグも生成されます。
 - データタイムアウトが発生した場合、DTIMEOUT フラグも生成されます。
8. 最後のデータが受信されると、FIFO がエンptyになるまで FIFO からデータを読み出し、その後、データの終了 DATAEND フラグが生成されます。
 - SDMMC がすべてのデータを完全に受信して、DPSM が無効になります。
9. ブート手順は、ファームウェアが BOOTEN をクリアすることで終了し、これにより SDMMC_CMD ラインがハイになります。CMDSENT フラグが 56 サイクル後に生成され、新しいコマンドを送信できることを示します。
 - a) すべてのデータが受信される前にブート手順がファームウェアによってアボートされた場合、CPSM Abort 信号によってデータ受信が停止され、DPSM が無効にされ、これにより、有効になっていれば DABORT フラグがトリガされます。
10. CMDSENT フラグがブート手順の終了を伝え、カードが新しいコマンドを受信できるようになります。

代替ブート動作

カードのパワーアップまたはリセットの後、ホストが CMD0 が発行されるまでの 74 クロックサイクル後に引数 0xFFFFFFFFFA で CMD0 を送信する場合、カードはブートモードが開始されていると認識します。引数 0xFFFFFFFFFA で CMD0 が送信された後 1 秒以内に、カードは SDMMC_Dn ラインで最初のブートコードデータの送信を開始します。マスタが CMD0 (リセット) を送信してブート動作を終了します。

図 180. 代替ブートモード動作



MSv40945V2

代替ブート手順を実行するには、次の手順が必要です。

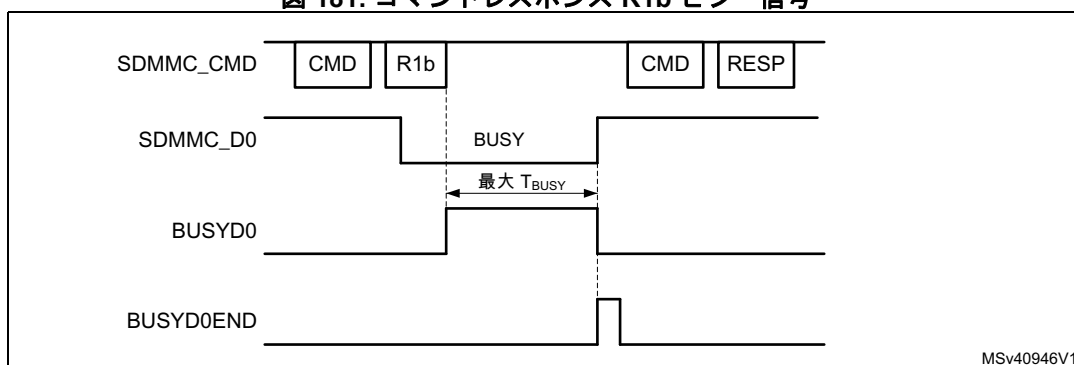
1. SDMMC をパワーオフ状態に移行させ、カードをリセットします。
2. SDMMC をパワーオン状態に移行させます。これにより、どのコマンドの前にも 74 SDMMC_CK サイクルがクロック供給されることが保証されます。
3. ブート確認応答をリクエストする場合、BOOTACKEN を有効にして、ACKTIME をセットし、ACKTIMEOUT フラグを有効にします。
4. 受信モード (DTCIR) で DPSM と DATALENGTH で受信するデータ数をセットしてデータ受信を有効にします。DTIMEOUT と DATAEND フラグを有効にします。
5. BOOTMODE で代替ブート動作モードを選択し、コマンドレジスタで 0xFFFFFFFFFA の引数で CMD0 をロードします。ブートの終了コマンド確認に CMDSENT フラグを有効にして、BOOTEN でブートを有効にします。ブート手順は、CPSMEN で CPSM を有効にすることで開始されます。これによって以下のことが起きます。
 - ロードされたコマンドと引数が送信されます。(BOOTMODE = 代替ブート)。
 - ACK タイムアウトが開始します。
 - DPSM が有効になります。
6. コマンドが送信されると、CMDSENT フラグが生成され、その時に BOOTEN ビットをクリアする必要があります。
7. ブート確認応答の受信は、ACKFAIL フラグが有効時に検出できます。
 - ブート確認応答が時間内に受信されなかった場合、ACKTIMEOUT フラグが発生します。
8. すべてのブートデータが受信されると、DATAEND フラグが発生します。
 - データ CRC が失敗した場合、DCRCFAIL フラグも生成されます。
 - データタイムアウトが発生した場合、DTIMEOUT フラグも生成されます。
9. 最後のデータが受信されると、FIFO がエンプティになるまで FIFO からデータを読み出し、その後、データの終了 DATAEND フラグが生成されます。
 - SDMMC がすべてのデータを完全に受信して、DPSM が無効になります。

10. BOOTMODE = 代替モードで CMD0(リセット)を送信してブート手順を終了する前に BOOTEN ビットをクリアする必要があります。これにより、CMDSENT フラグがコマンドの後 56 サイクルで発生します。
 - すべてのデータが受信される前にブート手順がファームウェアによってアボートされた場合、CPSM Abort 信号によってデータ受信が停止され、DPSM が無効にされ、これにより、有効になっていれば DABORT フラグがトリガされます。
11. CMDSENT フラグがブート手順の終了を伝え、カードが新しいコマンドを受信できるようになります。RESET コマンドが正常に送信されると、ブート動作を終了するために BOOTMODE 制御ビットをクリアしなければなりません。

24.6.6 レスポンス R1b の処理

R1b レスポンスのあるコマンドを送信する場合、ビジー信号は BUSYD0 レジスタビットに反映され、BUSYD0END フラグでビジーの開放を反映します。SDMMC_D0 ラインは、R1b レスポンスの終了時にサンプリングされ、BUSYD0 レジスタビットで信号送信されます。BUSYD0 レジスタビットは、BUSYD0END フラグが生成されると同時に、SDMMC_D0 ラインがビジーを開放したときにビジーでない状態にリセットされます。

図 181. コマンドレスポンス R1b ビジー信号



期待される最大ビジー時間を、コマンド送信前に DATATIME レジスタでセットする必要があります。R1b レスポンスビジーがプログラムされた時間より長くアクティブになっていると、DTIMEOUT が有効になっている場合、そのフラグがセットされます。

R1b レスポンスでコマンドを送信したときに SDMMC_D0 ビジー信号を検出するには、次の手順が適用されます。

- CMDREND フラグを有効にします。
- CPSM でコマンドを送信します。
- CMDREND フラグで、BUSYD0 レジスタビットをチェックします。
 - BUSYD0 がビジーではない場合、ビジー開放の信号をファームウェアに送信します。
 - BUSYD0 がビジーである場合、BUSYD0END フラグを待ちます。
- BUSYD0END フラグで、ビジー開放の信号をファームウェアに送信します。
- DTIMEOUT フラグで、ビジーがプログラムされた時間以上にアクティブになります。

24.6.7 リセットとカードサイクルパワー

リセット

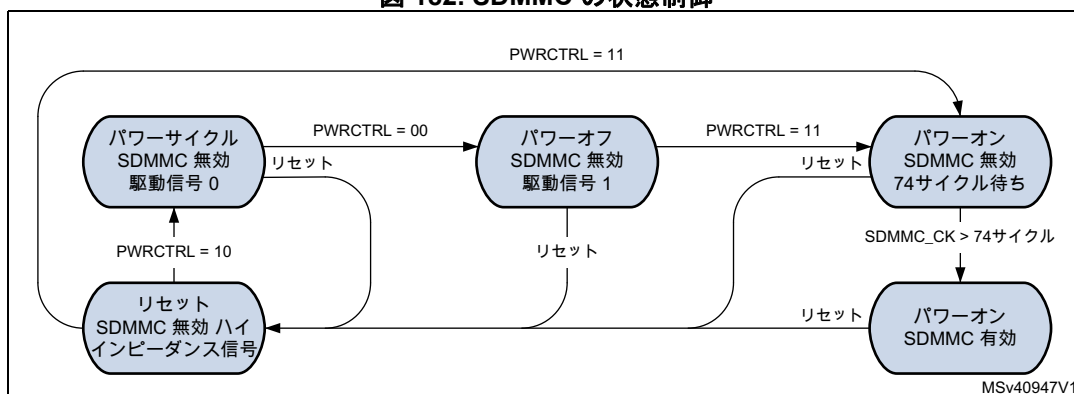
リセット後、SDMMC はリセット状態になります。この状態では、SDMMC が無効になり、コマンドやデータが転送できません。SDMMC_D[7:0] および SDMMC_CMD はハイインピーダンスになり、SDMMC_CK はローに駆動されます。

パワーオン状態に移行する前に、SDMMC を設定する必要があります。

パワーオン状態では、SDMMC_CK クロックが実行しています。最初に 74 SDMMC_CK サイクルがクロック供給され、その後に SDMMC が有効になり、コマンドとデータを転送できるようになります。

SDMMC 状態は、図 182 に従った PWRCTRL レジスタビットでファームウェアによって制御されます。

図 182. SDMMC の状態制御

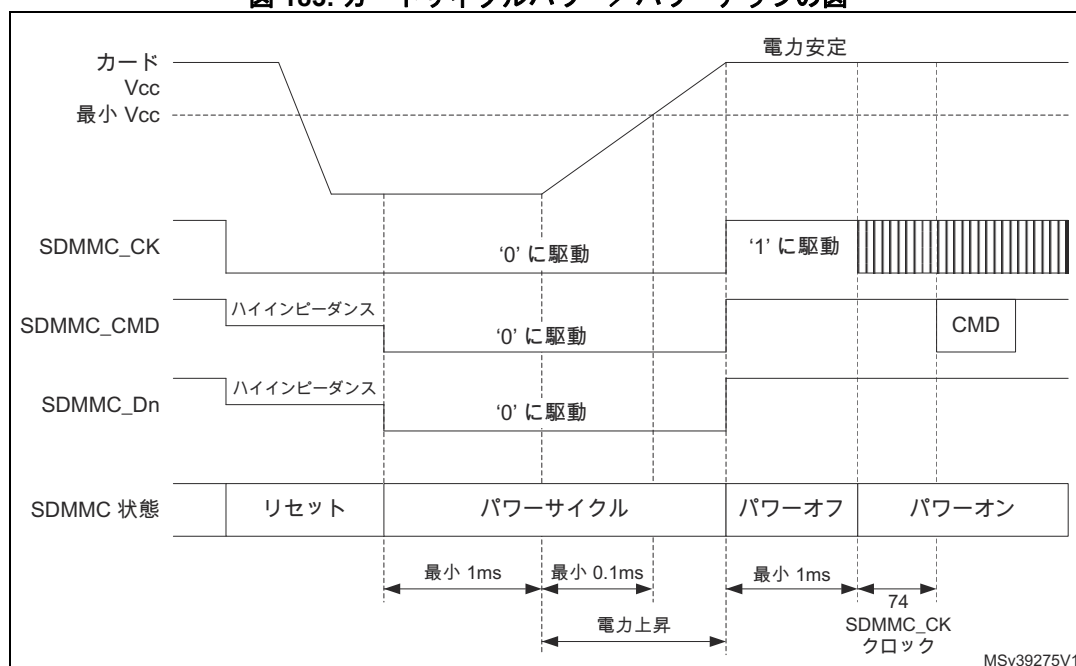


カードサイクルパワー

カードサイクルパワーを実行するには、次の手順が適用されます。

1. RCC.SDMMCxRST レジスタビットで SDMMC をリセットします。これにより、SDMMC がリセット状態にリセットされ、CPSM と DPSM はアイドル状態へ移行します。
2. カードへの Vcc 電源を無効にします。
3. パワーサイクル状態で SDMMC をセットします。これにより、SDMMC_D[7:0]、SDMMC_CMD および SDMMC_CK がローに駆動され、カードが信号ラインから電源供給されることを防ぎます。
4. 最低 1 ms 後に、カードへの Vcc 電源を有効にします。
5. 電力上昇時間の後、最低 1 ms の間 SDMMC をパワーオフ状態にセットします。SDMMC_D[7:0]、SDMMC_CMD および SDMMC_CK がセットされ“1”を駆動します。
6. 1 ms の遅延後、SDMMC をパワーオン状態にセットし、SDMMC_CK クロックを有効にします。
7. 74 SDMMC_CK サイクル後、最初のコマンドをカードに送信できるようになります。

図 183. カードサイクルパワー／パワーアップの図



24.7 ハードウェアフロー制御

データ転送中のハードウェアフロー制御機能は、FIFO アンダーラン (TX モード) およびオーバーラン (RX モード) エラーを回避するために使用します。

この機能は、データ転送中に SDMMC_CK を停止し、SDMMC ステートマシンをフリーズします。FIFO を使用してデータを送受信できない場合、データ転送がストールされます。送信 FIFO がハーフフルになるか、DATALENGTH に応じてすべてのデータが格納されるか、受信 FIFO がハーフエンptyになるまで、データ転送がストールされたままとなります。SDMMC_CK によってクロック供給されるステートマシンだけが停止し、AHB インタフェースは動作を続けます。したがって、フロー制御がアクティブであっても、FIFO がいっぱいになったり、空になることがあります。

IDMA のリンクされたリストの転送エラーが発生すると、ハードウェアフロー制御が無効になります。その結果、IDMA のリンクされたリストの転送エラーがいつ発生するかによって、アンダーランエラーまたはオーバーランエラーも発生することがあります ([セクション : IDMA 転送エラー管理](#)を参照)。

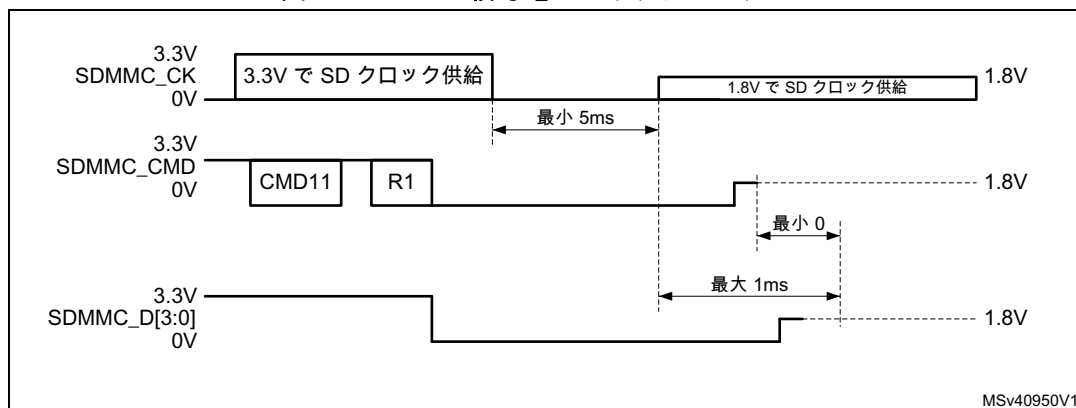
データ転送中にハードウェアフロー制御を有効にするには、HWFC_EN レジスタビットを 1 にセットする必要があります。リセット後、ハードウェアフロー制御は無効になります。

ハードウェアフロー制御は、SDMMC_Dn データが SDMMC_CK とサイクル調整されている場合のみに使用する必要があります。t_{OP} および Dt_{OP} の遅延が 1 サイクルを超える SDR104 モードの場合など DLYB 遅延ブロックからの sdmmc_fb_ck が使用されている場合は、常にハードウェアフロー制御は使用できません。

24.8 超ハイスピードフェーズ I (UHS-I) の電圧スイッチ

UHS-I モード (SDR12、SDR25、SDR50、SDR104、および DDR50) は、1.8 V 信号のサポートが必須です。パワーアップ後、カードは 3.3 V モードで開始します。CMD11 が電圧スイッチシーケンスを呼び出し、1.8 V モードにします。電圧シーケンスが正常に完了すると、カードはデフォルトの SDR12 で UHS-I モードに移行し、カードの入力および出力タイミングが変わります。

図 184. CMD11 信号電圧スイッチシーケンス



信号電圧スイッチシーケンスを実行するには、次の手順が必要です。

1. 電圧スイッチ手順を開始する前に、SDMMC_CK 周波数は、100~400 kHz の範囲にセットする必要があります。
2. ホストは、CMD11 を送信する前に VSWITCHEN ビットをセットして電圧スイッチを開始します。
3. カードが R1 レスポンスを返します。
 - レスポンス CRC がパスして、電圧スイッチ手順を続行する場合、ホストは電圧スイッチシーケンスが完了するまで CMD および SDMMC_D[3:0] 信号を駆動しません。レスポンス後の数サイクルで、SDMMC_CK は停止し、CKSTOP フラグがセットされます。
 - レスポンス CRC が失敗した場合 (CCRCFAIL フラグ)、またはタイムアウトまでにレスポンスを受信しなかった場合 (CTIMEOUT フラグ)、電圧スイッチ手順は停止します。
4. カードが、R1 レスポンス後の次のクロックで CMD および SDMMC_D[3:0] をローに駆動します。
5. ホストは、R1 レスポンスを受信した後、BUSYD0 レジスタビットで SDMMC_D0 ラインを監視します。SDMMC_D0 ラインは、レスポンス後 2 SDMMC_CK クロックサイクルでサンプリングされます。ファームウェアは、CKSTOP フラグの後に BUSYD0 レジスタビットを読み出す場合があります。
 - BUSYD0 がローで検出されると、ホストファームウェアは電圧レギュレータを 1.8 V に切り替え、レジスタビット VSWITCH をセットして SDMMC に電圧スイッチシーケンスのタイミングが重要なセクションを開始するよう命令します。ハードウェアは、最低 5 ms の間 SDMMC_CK をローに保って停止させ続けます。
 - BUSYD0 がハイで検出されると、ホストは電圧スイッチシーケンスをアボートして、カードをサイクルパワーします。
6. SDMMC_CK がローで検出された後、カードは信号電圧を 1.8 V に切り替え始めます。
7. 最低 5 ms 後に、ホスト SDMMC ハードウェアは SDMMC_CK を再開します。
8. SDMMC_CK の遷移を検出してから 1 ms 以内に、カードは最低 1 SDMMC_CK サイクルの間 CMD と DAT[3:0] をハイに駆動してから、CMD と DAT[3:0] の駆動を停止させます。
9. ホスト SDMMC ハードウェアでは、SDMMC_CK が再開してから 1ms で SDMMC_D0 が BUSYD0 にサンプリングされ、VSWEND フラグがセットされます。

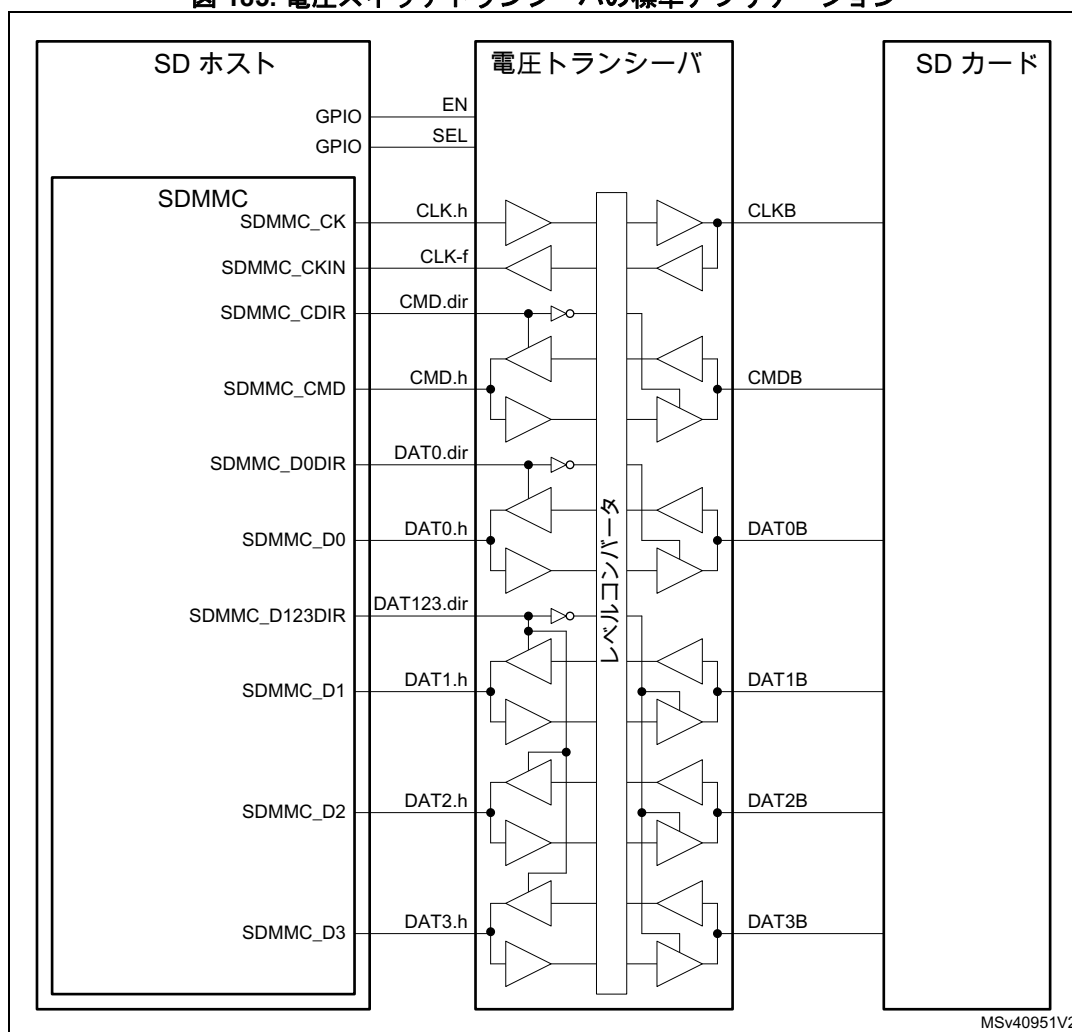
10. VSWEND フラグで、ホストは BUSYD0 レジスタビットを使って SDMMC_D0 ラインをチェックし、電圧スイッチシーケンスの完了を確認します。
- BUSYD0 がハイで検出される場合、電圧スイッチが正常に完了しています。
 - BUSYD0 がローで検出される場合、電圧スイッチは失敗であり、ホストはカード電源をサイクルパワーします。

SDMMC_CK を停止するための最低 5 ms の時間は、クロック分周回路 CLKDIV で設定されている最大周波数 25 MHz (SD モード) を備えた内部のゲートされていない SDMMC_CK クロックから導出されます。5ms を超える時間は、 2^{12} サイクル (10.24ms @ 400 kHz) でカウントされます。より低い SDMMC_CK 周波数がクロック分周回路 CLKDIV で選択されている場合、SDMMC_CK クロックが停止する時間は長くなります。

カードが SDMMC_Dn と SDMMC_CMD をハイに駆動する最大 1 ms の時間は、クロック分周回路 CLKDIV で設定されている最大周波数 25 MHz (SD モード) を備えた内部のゲートされていない SDMMC_CK から導出されます。SDMMC は、 2^9 サイクル (1.28ms @ 25 MHz) でカウントされる 1ms を超える時間以降、ラインをチェックします。より低い SDMMC_CK 周波数がクロック分周回路 CLKDIV で選択されている場合、ラインをチェックする時間は長くなります。

信号電圧レベルは、ST マイクロエレクトロニクスの ST6G3244ME のような外部電圧変換トランシーバを通じてサポートされます。

図 185. 電圧スイッチトランシーバの標準アプリケーション



MSv40951V2

外部ドライバ（電圧スイッチトランシーバ）とインタフェース接続するために、標準信号とは別に、SDMMC は次の信号を使用します。

SDMMC_CKIN フィードバック入力クロック

SDMMC_CDIR CMD 信号の I/O 方向制御

SDMMC_D0DIR SDMMC_D0 信号の I/O 方向制御

SDMMC_D123DIR SDMMC_D1、SDMMC_D2、および SDMMC_D3 信号の I/O 方向制御

電圧トランシーバ信号 **EN** および **SEL** は、汎用 IO を通じて処理されます。

SDMMC_CDIR、SDMMC_D0DIR、および SDMMC_D123DIR 信号の極性は、SDMMC_POWER.DIRPOL 制御ビットで選択できます。

24.9 SDMMC 割込み

表 233. SDMMC 割込み

項目 (割込みの 略称)	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法	SLEEP モードの 終了
SDMMC	コマンドレスポンス CRC 失敗	CCRCFAIL	CCRCFAILIE	CCRCFAILC	可能
SDMMC	データブロック CRC 失敗	DCRCFAIL	DCRCFAILIE	DCRCFAILC	可能
SDMMC	コマンドレスポンス タイムアウトです。	CTIMEOUT	CTIMEOUTIE	CTIMEOUTC	可能
SDMMC	データタイムアウト です。	DTIMEOUT	DTIMEOUTIE	DTIMEOUTC	可能
SDMMC	送信 FIFO アンダーラン	TXUNDERR	TXUNDERRIE	TXUNDERRC	可能
SDMMC	受信 FIFO オーバーラン	RXOVERR	RXOVERRIE	RXOVERRC	可能
SDMMC	コマンドレスポンス 受信	CMDREND	CMDRENDIE	CMDREND C	可能
SDMMC	コマンド送信	CMDSENT	CMDSENTIE	CMDSENTC	可能
SDMMC	データ転送終了	DATAEND	DATAENDIE	DATAENDC	可能
SDMMC	データ転送保留	DHOLD	DHOLDIE	DHOLD C	可能
SDMMC	データブロック 送受信	DBCKEND	DBCKENDIE	DBCKEND C	可能
SDMMC	データ転送アボート	DABORT	DABORTIE	DABORTC	可能
SDMMC	送信 FIFO ハーフエンプティ	TXFIFOHE	TXFIFOHEIE	該当なし	可能
SDMMC	受信 FIFO ハーフフル	RXFIFOHF	RXFIFOHFIE	該当なし	可能
SDMMC	送信 FIFO フル	TXFIFO F	該当なし	該当なし	可能
SDMMC	受信 FIFO フル	RXFIFO F	RXFIFO FIE	該当なし	可能
SDMMC	送信 FIFO エンプティ	TXFIFOE	TXFIFOEIE	該当なし	可能
SDMMC	受信 FIFO エンプティ	RXFIFOE	該当なし	該当なし	可能
SDMMC	コマンドレスポンスの ビジー終了	BUSYD0END	BUSYD0ENDIE	BUSYD0END C	可能
SDMMC	SDIO 割込み	SDIOIT	SDIOITIE	SDIOITC	可能
SDMMC	ブート確認応答失敗	ACKFAIL	ACKFAILIE	ACKFAILC	可能
SDMMC	ブート確認応答 タイムアウト	ACKTIMEOUT	ACKTIMEOUTIE	ACKTIMEOUTC	可能
SDMMC	電圧スイッチ タイミング	VSWEND	VSWENDIE	VSWEND C	可能

表 233. SDMMC 割込み (続き)

項目 (割込みの 略称)	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法	SLEEP モードの 終了
SDMMC	SDMM_CK が電圧 スイッチで停止	CKSTOP	CKSTOPIE	CKSTOPC	可能
SDMMC	IDMA 転送エラー	IDMATE	IDMATEIE	IDMATEC	可能
SDMMC	IDMA バッファ転送 完了	IDMABTC	IDMABTCIE	IDMABTCC	可能

24.10 SDMMC レジスタ

デバイスは、AHB スレーブインタフェース経由でアクセス可能な 32 ビットの制御レジスタによってシステムと通信します。

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。バイト（8-bit）およびハーフワード（16-bit）のアクセスは、AHB バスエラーをトリガします。

24.10.1 SDMMC 電源制御レジスタ (SDMMC_POWER)

アドレスオフセット：0x000

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIR POL	VSWI TCHEN	VSWI TCH	PWRCTRL[1:0]	
											rw	rw	rw	rw	rw

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **DIRPOL**：データおよびコマンドの方向信号の極性選択

このビットは、SDMMC がパワーオフ状態（PWRCTRL = 00）のときのみ書き込むことができます。

0：方向信号がローのときに電圧トランシーバの IO が出力として駆動されます。

1：方向信号がハイのときに電圧トランシーバの IO が出力として駆動されます。

ビット 3 **VSWITCHEN**：電圧スイッチ手順有効

このビットは、CPSM が無効（CPSMEN = 0）のときのみファームウェアで書き込むことができます。

このビットは、電圧スイッチコマンドレスポンスの後 SDMMC_CLK を停止するために使用されます。

0：コマンドレスポンスを正常に受信した後、SDMMC_CLK クロックに変化はありません。

1：コマンドレスポンスを正常に受信した後、SDMMC_CLK クロックが停止します。

ビット 2 **VSWITCH**：電圧スイッチシーケンス開始

このビットは、電圧スイッチシーケンスのタイミングの重要なセクションの開始に使用されます。

0：電圧スイッチシーケンスは開始されずアクティブではありません。

1：電圧スイッチシーケンスが開始されているかアクティブです。

ビット 1:0 **PWRCTRL[1:0]**：SDMMC の状態制御ビット

これらのビットは、SDMMC がパワーオン状態ではないとき（PWRCTRL ≠ 11）のみ書き込むことができます。

これらのビットは、SDMMC 信号の機能状態を定義するために使用されます。

00：リセット後、リセット：SDMMC は無効化され、カードへのクロックは停止され、SDMMC_D[7:0] および SDMMC_CMD はハイインピーダンスになり、SDMMC_CLK はローに駆動されます。

00 が書き込まれた場合、パワーオフ：SDMMC は無効化され、カードへのクロックは停止され、SDMMC_D[7:0]、SDMMC_CMD および SDMMC_CLK はハイに駆動されます。

01：予約済み。（01 が書き込まれた場合、PWRCTRL 値は変化しません）

10：パワーサイクル、SDMMC は無効化され、カードへのクロックは停止され、SDMMC_D[7:0]、SDMMC_CMD および SDMMC_CLK はローに駆動されます。

11：パワーオン：カードはクロック供給されます。最初の 74 SDMMC_CLK サイクルは SDMMC が無効化されたままになります。74 サイクル後、SDMMC は有効になり、SDMMC_D[7:0]、SDMMC_CMD および SDMMC_CLK は SDMMC の動作に従って制御されます。

追加の書き込みは無視され、PWRCTRL 値は 11 のままとなります。

24.10.2 SDMMC クロック制御レジスタ (SDMMC_CLKCR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

SDMMC_CLKCR レジスタは、SDMMC_CK の出力クロック、sdmmc_rx_ck 受信クロック、バス幅を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUS SPEED	DDR	HWFC _EN	NEG EDGE
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WID BUS[1:0]		Res.	PWR SAV	Res.	Res.	CLKDIV[9:0]									
rw	rw		rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **SELCLKRX[1:0]** : 受信クロックの選択

これらのビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

00 : sdmmc_io_in_ck が受信クロックとして選択されます。

01 : SDMMC_CKIN フィードバッククロックが受信クロックとして選択されます。

10 : sdmmc_fb_ck 調整フィードバッククロックが受信クロックとして選択されます。

11 : 予約済み (sdmmc_io_in_ck を選択)

ビット 19 **BUSPEED** : SDMMC の動作モードを選択するためのバス速度

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

0 : DS、ハイスピード、SDR12、SDR25、レガシー互換、高速 SDR、高速 DDR のバス速度モードが選択されます。

1 : SDR50、DDR50、SDR104、HS200 のバス速度モードが選択されます。

ビット 18 **DDR** : データレート信号の選択

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

DDR レートは、4 ビットまたは 8 ビット幅のバスモードでのみ選択する必要があります。(WIDBUS > 00)。WIDBUS = 00 (1 ビット幅バス) の場合、DDR = 1 は効果がありません。

DDR レートは、クロック分周が 1 より大きい場合のみ選択する必要があります。(CLKDIV > 0)。

0 : SDR シングルデータレートシグナリング

1 : DDR ダブルデータレートシグナリング

ビット 17 **HWFC_EN** : ハードウェアフロー制御イネーブル

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

0 : ハードウェアフロー制御は無効です。

1 : ハードウェアフロー制御は有効です。

ハードウェアフロー制御が有効なときの TXFIFOE および RXFIFOE フラグ変更の意味については、[セクション 24.10.11](#) の SDMMC ステータスレジスタの定義を参照してください。

ビット 16 **NEGEDGE** : データとコマンドの SDMMC_CLK 位相反転選択ビット

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

クロック分周 = 1 (CLKDIV = 0) の場合、このビットは意味を持ちません。コマンドとデータは SDMMC_CLK 立下がりエッジで変更されます。

0 : クロック分周 > 1 (CLKDIV > 0) および DDR = 0 の場合 :

- コマンドとデータは SDMMC_CLK 立上がりエッジに続く sdmmc_ker_ck 立下がりエッジで変更されます。
- SDMMC_CLK エッジは sdmmc_ker_ck 立上がりエッジで発生します。

クロック分周 > 1 (CLKDIV > 0) および DDR = 1 の場合 :

- コマンドは SDMMC_CLK 立上がりエッジに続く sdmmc_ker_ck 立下がりエッジで変更されます。
- データは SDMMC_CLK エッジに続く sdmmc_ker_ck 立下がりエッジで変更されます。
- SDMMC_CLK エッジは sdmmc_ker_ck 立上がりエッジで発生します。

1 : クロック分周 > 1 (CLKDIV > 0) および DDR = 0 の場合 :

- コマンドとデータは SDMMC_CLK 立下がりエッジを生成する同じ sdmmc_ker_ck 立上がりエッジで変更されます。

クロック分周 > 1 (CLKDIV > 0) および DDR = 1 の場合 :

- コマンドは SDMMC_CLK 立下がりエッジを生成する同じ sdmmc_ker_ck 立上がりエッジで変更されます。
- データは SDMMC_CLK エッジに続く SDMMC_CLK 立下がりエッジで変更されます。
- SDMMC_CLK エッジは sdmmc_ker_ck 立上がりエッジで発生します。

ビット 15:14 **WIDBUS[1:0]** : ワイドバスモード有効ビット

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

00 : デフォルトの 1 ビット幅のバスモード : SDMMC_D0 使用 (DDR をサポートしません)

01 : 4 ビット幅のバスモード : SDMMC_D[3:0] 使用

10 : 8 ビット幅のバスモード : SDMMC_D[7:0] 使用

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **PWRSAPV** : 節電構成ビット

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

節電のため、バスがアイドル状態のときには、PWRSAPV をセットして SDMMC_CLK クロック出力を無効にできます。

0 : SDMMC_CLK クロックは常に有効です。

1 : SDMMC_CLK はバスがアクティブなときに限り有効です。

ビット 11:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **CLKDIV[9:0]** : クロック分周ファクタ

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

このフィールドは、入力クロック (sdmmc_ker_ck) と出力クロック (SDMMC_CLK) の間の分周ファクタを定義します。SDMMC_CLK 周波数 = sdmmc_ker_ck / [2 * CLKDIV]。

0x000 : SDMMC_CLK 周波数 = sdmmc_ker_ck / 1 (DDR をサポートしません)

0x001 : SDMMC_CLK 周波数 = sdmmc_ker_ck / 2

0x002 : SDMMC_CLK 周波数 = sdmmc_ker_ck / 4

0x0XX : 同様に続く

0x080 : SDMMC_CLK 周波数 = sdmmc_ker_ck / 256

0xXXX : 同様に続く

0x3FF : SDMMC_CLK 周波数 = sdmmc_ker_ck / 2046

- 注：
- 1 SD/SDIO カードまたは eMMC が識別モードのときには、SDMMC_CLK 周波数は 400 kHz 未満である必要があります。
 - 2 クロック周波数は、相対カードアドレスがすべてのカードに割り当てられているときには、最大カードバス周波数に変更できます。
 - 3 このレジスタへの各書き込みアクセス間には、7 sdmmc_hclk クロック周期以上が必要です。SDMMC_CLK は、SD I/O カードの読出しウェイトインターバルの間も、停止することができます。この場合、SDMMC_CLKCR レジスタは SDMMC_CLK を制御しません。

24.10.3 SDMMC 引数レジスタ (SDMMC_ARGR)

アドレスオフセット：0x008

リセット値：0x0000 0000

SDMMC_ARGR レジスタは、コマンドメッセージの一部としてカードに送信される 32 ビットのコマンド引数を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMDARG[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **CMDARG[31:0]**：コマンド引数

これらのビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。コマンドメッセージの一部としてカードに送られるコマンド引数です。コマンドが引数を含む場合、コマンドをコマンドレジスタに書き込む前に、このレジスタにロードする必要があります。

24.10.4 SDMMC コマンドレジスタ (SDMMC_CMDR)

アドレスオフセット：0x00C

リセット値：0x0000 0000

SDMMC_CMDR レジスタは、コマンドインデックスおよびコマンドタイプビットを含みます。コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。コマンドタイプビットは、コマンドパスステートマシン (CPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMD SUS PEND
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ブート EN	ブート MODE	DT HOLD	CPSM EN	WAITP END	ウェイト INT	WAITRESP[1:0]		CMD STOP	CMD TRANS	CMDINDEX[5:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

- ビット 16 **CMDSPEND** : CPSM がコマンドをサスペンドまたはレジャーコマンドとして処理し、割り込み期間の開始/終了の信号を伝えます。
- このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
CMDSPEND = 1 および CMDTRANS = 0 のサスペンドコマンドで、レスポンスビット BS=0 の場合、割り込み期間を開始します。
CMDSPEND = 1 および CMDTRANS = 1 のレジャーコマンドとデータで、レスポンスビット DF=1 の場合、割り込み期間を終了します。
- ビット 15 **BOOTEN** : ブートモード手順の有効
- 0 : ブートモード手順は無効です。
1 : ブートモード手順は有効です。
- ビット 14 **BOOTMODE** : 使用するブートモード手順の選択
- このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
0 : 通常ブートモード手順を選択します。
1 : 代替ブートモード手順を選択します。
- ビット 13 **DTHOLD** : DPSM での新しいデータブロックの送受信の保留
- このビットがセットされると、DPSM は Wait_S 状態から送信状態へ、または Wait_R 状態から受信状態へ移行しません。
- ビット 12 **CPSMEN** : コマンドパスステートマシン (CPSM) 有効ビット
- このビットは、CPSM がアイドル状態に移行すると、ファームウェアによって 1 が書き込まれ、ハードウェアによってクリアされます。
このビットがセットされると、CPSM が有効になります。
DTEN = 1 の場合、コマンドは転送されず、ブート手順も開始されません。CPSMEN は 0 にクリアされます。
SDMMC_CK を停止した読み出しウェイト状態の間、コマンドは送信されず、CPSMEN は 0 のままとなります。
- ビット 11 **WAITPEND** : CPSM の DPSM からのデータ転送終了待ち (CmdPend 内部信号)
- このビットがセットされると、CPSM は、データ転送の終了を待ってから、コマンドの送信を開始します。
DTMODE = eMMC ストリームデータ転送、WIDBUS = 1 ビット幅バスモード、DPSMACT = 1 かつ DTDIR = ホストからカードの場合のみ、WAITPEND は考慮されます。
- ビット 10 **WAITINT** : CPSM の割り込みリクエスト待ち
- このビットがセットされると、CPSM は、コマンドタイムアウトを無効にして、割り込みリクエスト (レスポンス) を待ちます。
CPSM ウェイトステートでこのビットがクリアされると、割り込みモードのアポートが発生します。
- ビット 9:8 **WAITRESP[1:0]** : レスポンスビット待ち
- このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
これらは、CPSM がレスポンスを待つかどうか、および、待つ場合にはレスポンスの種類を設定するために使用されます。
00 : レスポンスなし。CMDSENT フラグを待ちます。
01 : ショートレスポンス。CMDREND または CCRCFAIL フラグを待ちます。
10 : ショートレスポンス。CMDREND フラグ (CRC なし) を待ちます。
11 : ロングレスポンス。CMDREND または CCRCFAIL フラグを待ちます。
- ビット 7 **CMDSTOP** : CPSM がコマンドを送信停止コマンドとして処理し、DPSM に Abort 信号を伝えます
- このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
このビットがセットされている場合、CPSM はコマンド送信時に Abort 信号を DPSM に発行します。
- ビット 6 **CMDTRANS** : CPSM がコマンドをデータ転送コマンドとして処理し、割り込み期間を停止し、DPSM に DataEnable 信号を伝えます。
- このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
このビットがセットされている場合、CPSM はコマンド送信時に割り込み期間の終了と DataEnable 信号を DPSM に発行します。

ビット 5:0 **CMDINDEX[5:0]** : コマンドインデックス

このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。
コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。

- 注 :
- 1 このレジスタへの各書き込みアクセス間には、7 sdmmc_hclk クロック周期以上が必要です。
 - 2 MultiMediaCard はショートレスポンス (48 ビット) またはロングレスポンス (136 ビット) の 2 種類のレスポンスを送信できます。SD カードと SD I/O カードはショートレスポンスのみ送信でき、引数はレスポンスのタイプによって異なることがあります。ソフトウェアは、送信コマンドに応じてレスポンスのタイプを区別します。

24.10.5 SDMMC コマンドレスポンスレジスタ (SDMMC_RESPCMDR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

SDMMC_RESPCMDR レジスタは、最後に受信したコマンドレスポンスのコマンドインデックスフィールドを含みます。コマンドレスポンス送信にコマンドインデックスフィールドが含まれていない場合 (ロングまたは OCR レスポンス)、RESPCMD フィールドは未知ですが、11111b (レスポンスの予約フィールドの値) が含まれている必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
										r	r	r	r	r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **RESPCMD[5:0]** : レスポンスコマンドインデックス

読み専用ビットフィールドです。最後に受信したコマンドレスポンスのコマンドインデックスを含みます。

24.10.6 SDMMC レスポンス x レジスタ (SDMMC_RESPxR)

アドレスオフセット : $0x010 + 0x004 * x$, ($x = 1$ から 4)

リセット値 : 0x0000 0000

SDMMC_RESP1/2/3/4R レジスタは、受信したレスポンスの一部であるカードのステータスを含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CARDSTATUS[31:0]** : 下表に従ったカードステータス

[表 234](#)を参照してください。

カードステータスサイズは、レスポンスのタイプによって 32 または 128 ビットです。

表 234. レスポンスタイプと SDMMC_RESPxR レジスタ

レジスタ ⁽¹⁾	ショートレスポンス	ロングレスポンス
SDMMC_RESP1R	カードステータス [31:0]	カードステータス [127:96]
SDMMC_RESP2R	すべて 0	カードステータス [95:64]
SDMMC_RESP3R	すべて 0	カードステータス [63:32]
SDMMC_RESP4R	すべて 0	カードステータス [31:0] ⁽²⁾

1. カードステータスの最上位ビットから先に受信します。
2. SDMMC_RESP4R レジスタ LSB は常に 0 です。

24.10.7 SDMMC データタイマレジスタ (SDMMC_DTIMER)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

SDMMC_DTIMER レジスタは、データタイムアウト期間(カードバスクロック周期単位)を含みます。

カウンタは SDMMC_DTIMER レジスタから値をロードして、データパスステートマシン (DPSM) が Wait_R またはビジー状態に入ったときに、デクリメントを開始します。DPSM がこのどちらかの状態のときにタイマが 0 に達した場合、タイムアウトステータスフラグがセットされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATETIME[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATETIME[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **DATETIME[31:0]** : データおよび R1b ビジータイムアウト期間

このビットに書き込めるのは、CPSM および DPSM がアクティブではないときだけです (CPSMACT = 0 および DPSMACT = 0)。

カードバスクロック周期で表されたデータおよび R1b ビジータイムアウト期間です。

注 : データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。

24.10.8 SDMMC データ長レジスタ (SDMMC_DLENR)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

SDMMC_DLENR レジスタは、転送されるデータバイト数を含みます。値は、データ転送が開始されたときに、データカウンタにロードされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:16]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATALENGTH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATALENGTH[24:0]** : データ長の値

このレジスタは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

転送されるデータバイト数です。

DDR = 1 の場合、DATALENGTH は 2 の倍数に切り詰められます。(最後の奇数バイトは転送されません)

DATALENGTH = 0 の場合、データは転送されず、CPSMEN および CMDTRANS = 1 でリクエストされたときは、コマンドも転送されません。DTEN および CPSMEN は 0 にクリアされます。

注 : ブロックデータ転送では、データ長レジスタの値はブロックサイズの倍数である必要があります (SDMMC_DCTRL を参照してください)。データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。

SDMMC マルチバイト転送の場合、データ長レジスタ内の値は 1 から 512 ままでなければなりません。

24.10.9 SDMMC データ制御レジスタ (SDMMC_DCTRL)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

SDMMC_DCTRL レジスタは、データパスステートマシン (DPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FIFO RST	ブート ACK EN	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE[3:0]				DTMODE[1:0]		DTDIR	DTEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **FIFORST** : FIFO リセット、残りのデータを一扫します。

このビットは、IDMAEN=0 および DPSM がアクティブ (DPSMACT = 1) のときのみファームウェアで書き込むことができます。このビットは、転送エラーまたは転送保留が発生したときのみ有効になります。

0 : FIFO に影響がありません。

1 : 残りのデータを一扫して FIFO ポインタをリセットします。このビットは、DPSM がインアクティブ (DPSMACT = 0) になったとき、ハードウェアによって自動的に 0 にクリアされます。

ビット 12 **BOOTACKEN** : ブート確認応答の受信を有効にします。

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : ブート確認応答は無効になり、受信の予定はありません。

1 : ブート確認応答は有効になり、受信の予定があります。

ビット 11 **SDIOEN** : SD I/O 割込み有効機能

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

このビットがセットされると、DPSM は SD I/O カード固有の割込み操作を可能にします。

ビット 10 **RWMOD** : 読出しウェイトモード

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : SDMMC_D2 を使用する読出しウェイト制御です。

1 : SDMMC_CLK を停止する読出しウェイト制御です。

ビット 9 **RWSTOP** : 読出しウェイト停止

このビットは、ファームウェアによって書き込まれ、DPSM が R_W 状態から Wait_R またはアイドル状態に移行すると、ハードウェアによって自動クリアされます。

0 : 読出しウェイト停止はありません。

1 : DPSM が R_W 状態の場合、読出しウェイト停止が有効になります。

ビット 8 **RWSTART** : 読出しウェイト開始

このビットがセットされると、読出しウェイト動作が開始されます。

ビット 7:4 **DBLOCKSIZE[3:0]** : データブロックサイズ

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

ブロックデータ転送モードが選択されているときのデータブロック長を定義します。

0000 : ブロック長 = 2^0 = 1 バイト

0001 : ブロック長 = 2^1 = 2 バイト

0010 : ブロック長 = 2^2 = 4 バイト

0011 : ブロック長 = 2^3 = 8 バイト

0100 : ブロック長 = 2^4 = 16 バイト

0101 : ブロック長 = 2^5 = 32 バイト

0110 : ブロック長 = 2^6 = 64 バイト

0111 : ブロック長 = 2^7 = 128 バイト

1000 : ブロック長 = 2^8 = 256 バイト

1001 : ブロック長 = 2^9 = 512 バイト

1010 : ブロック長 = 2^{10} = 1024 バイト

1011 : ブロック長 = 2^{11} = 2048 バイト

1100 : ブロック長 = 2^{12} = 4096 バイト

1101 : ブロック長 = 2^{13} = 8192 バイト

1110 : ブロック長 = 2^{14} = 16384 バイト

1111 : 予約済み

DATALENGTH が DBLOCKSIZE の整数倍ではない場合、転送されたデータは DBLOCKSIZE の倍数に切捨てられます。(残りのデータは転送されません。)

DDR = 1 の場合、DBLOCKSIZE = 0000 は使用してはなりません。(データは転送されません。)

ビット 3:2 **DTMODE[1:0]** : データ転送モード選択

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

00 : ブロック数で終了するブロックデータ転送

01 : SDIO マルチバイトデータ転送

10 : eMMC ストリームデータ転送 (WIDBUS では 1 ビット幅のバスモードを選択する必要があります)

11 : STOP_TRANSMISSION コマンドで終了するブロックデータ転送 (DTEN で開始するデータ転送では使用しません)

ビット 1 **DTDIR** : データ転送方向選択

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : ホストからカードへ

1 : カードからホストへ

ビット 0 **DTEN** : データ転送有効ビット

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。このビットは、データ転送が完了したとき、ハードウェアによってクリアされます。

このビットは、関連するデータ転送コマンドが使用されないときのみ、データを転送するために使用する必要があります。つまり、SD カードや eMMC カードでは使用しないでください。

0 : CPSM データ転送コマンドなしでデータ転送を開始しません。

1 : CPSM データ転送コマンドなしでデータ転送を開始します。

24.10.10 SDMMC データカウンタレジスタ (SDMMC_DCNT)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

SDMMC_DCNT レジスタは、DPSM がアイドル状態から Wait_R または Wait_S 状態へ移行すると、データ長レジスタから値をロードします (SDMMC_DLENR を参照してください)。データが転送されると、カウンタは 0 に達するまで値をデクリメントします。その後、DPSM はアイドル状態へ移行し、エラーがなく、送信データ転送保留がない場合、データステータス終了フラグ (DATAEND) がセットされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:16]								
							r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATACOUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATACOUNT[24:0]** : データカウント値

読み出すと、残りの転送バイト数が返されます。書き込み操作は何の影響も与えません。

注 : このレジスタは、データ転送を完了または保留したときだけ読み出す必要があります。エラーイベントの後に読み出す場合、読出しデータカウンタ値は実際に転送されたデータバイト数と異なる場合があります。

24.10.11 SDMMC ステータスレジスタ (SDMMC_STAR)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

SDMMC_STAR レジスタは、読出し専用レジスタです。2 種類のフラグを含みます。

- スタティックフラグ (ビット [28, 21, 11:0]) : これらのビットは、SDMMC 割込みクリアレジスタ (SDMMC_ICR を参照) に書き込んでクリアされるまで、アサートされたままです。
- ダイナミックフラグ (ビット [20:12]) : これらのビットは、基礎回路の状態に応じて状態が変わります (たとえば、FIFO フルおよびエンプティフラグは、データが FIFO に書き込まれるとアサートおよびネゲートされます)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTC	IDMA TE	CK STOP	VSW END	ACK TIME OUT	ACK FAIL	SDIOIT	BUSY D0END	BUSY D0	RX FIFOE	TX FIFOE	RX FIFO	TX FIFO
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HF	TX FIFO HE	CPSM ACT	DPSM ACT	DA BORT	DBCK END	DHOLD	DATA END	CMD SENT	CMDR END	RX OVERR	TX UNDER R	D TIME OUT	C TIME OUT	DCRC FAIL	CCRC FAIL
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

- ビット 28 **IDMABTC** : IDMA バッファ転送完了
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 27 **IDMATE** : IDMA 転送エラー
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 26 **CKSTOP** : SDMMC_CK が電圧スイッチ手順で停止されます。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 25 **VSWEND** : 電圧スイッチの重要なタイミングセクションの完了
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 24 **ACKTIMEOUT** : ブート確認応答タイムアウト
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 23 **ACKFAIL** : ブート確認応答受信 (ブート確認応答チェック失敗)
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 22 **SDIOIT** : SDIO 割込みが受信されました。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 21 **BUSYD0END** : CMD レスポンス検出後の SDMMC_D0 ビジーの終了
これは、CMD レスポンス後のビジーの終了のみを示します。このビットは、データ転送によってビジー信号を送信しません。割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
0 : カード SDMMC_D0 信号は、ビジーからビジーでない状態への変更を伝えません。
1 : カード SDMMC_D0 信号は、ビジーからビジーでない状態に変更されました。
- ビット 20 **BUSYD0** : CMD レスポンスの終了時と、CMD レスポンス後の 2 SDMMC_CK サイクルでサンプリングされた SDMMC_D0 ラインの反転された値 (ビジー)
このビットは、SDMMC_D0 ラインがビジーからビジーでない状態に変わる際に、ビジーでない状態にリセットされます。このビットは、データ転送によってビジー信号を送信しません。これはハードウェアステータスフラグのみで、割込みは生成されません。
0 : カードが SDMMC_D0 でビジーでない状態を伝えます。
1 : カードが SDMMC_D0 でビジー状態であることを伝えます。
- ビット 19 **RXFIFOE** : 受信 FIFO エンプティ
これはハードウェアステータスフラグのみで、割込みは生成されません。このビットは、1 つの FIFO の位置がフルになったときにクリアされます。
- ビット 18 **TXFIFOE** : 送信 FIFO エンプティ
このビットは、1 つの FIFO の位置がフルになったときにクリアされます。
- ビット 17 **RXFIFOF** : 受信 FIFO フル
このビットは、1 つの FIFO の位置がエンプティになったときにクリアされます。
- ビット 16 **TXFIFOF** : 送信 FIFO フル
これはハードウェアステータスフラグのみで、割込みは生成されません。このビットは、1 つの FIFO の位置がエンプティになったときにクリアされます。
- ビット 15 **RXFIFOHF** : 受信 FIFO ハーフフル
FIFO には少なくともワード数の半分入っています。このビットは、FIFO がハーフ +1 空になったときにクリアされます。
- ビット 14 **TXFIFOHE** : 送信 FIFO ハーフエンプティ
FIFO に少なくともワード数の半分を書込み可能です。このビットは、FIFO がハーフ +1 フルになったときにクリアされます。
- ビット 13 **CPSMACT** : コマンドパスステートマシンアクティブ (アイドル状態ではない)
これはハードウェアステータスフラグのみで、割込みは生成されません。

- ビット 12 **DPSMACT** : データバスステートマシンアクティブ (アイドル状態ではない)
これはハードウェアステータスフラグのみで、割込みは生成されません。
- ビット 11 **DABORT** : CMD12 によるデータ転送のアボート
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 10 **DBCKEND** : データブロック送受信
次のような場合に DBCKEND が生成されます。
- CRC 確認に成功し、DPSM が R_W 状態へ移行
または
- IDMAEN = 0 かつ送信データ転送が保留され、DATACOUNT > 0 であり、DPSM が Wait_S へ移行
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 9 **DHOLD** : データ転送保留
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 8 **DATAEND** : データ転送の正常終了
データカウンタの DATACOUNT が 0 で、発生したエラーがなく、送信データ転送保留がない場合に DATAEND がセットされます。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 7 **CMDSENT** : コマンドが送信されました (レスポンス不要)。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 6 **CMDREND** : コマンドレスポンス受信 (CRC 確認成功、または CRC なし)
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 5 **RXOVERR** : 受信 FIFO オーバーランエラー (IDMA が有効な場合はハードウェアでマスクされます)
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 4 **TXUNDERR** : 送信 FIFO アンダーランエラー (IDMA が有効な場合はハードウェアでマスクされます)
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 3 **DTIMEOUT** : データタイムアウトです。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 2 **CTIMEOUT** : コマンドレスポンスタイムアウトです。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
コマンドタイムアウト期間は、固定値の 64 SDMMC_CLK クロック周期です。
- ビット 1 **DCRCFAIL** : データブロック送受信 (CRC 確認失敗) しました。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。
- ビット 0 **CCRCFAIL** : コマンドレスポンス受信 (CRC 確認失敗) しました。
割込みフラグは、SDMMC_ICR で対応する割込みクリアビットに書き込むことでクリアされます。

注 : **FIFO 割込みフラグは、IDMA モード使用時 SDMMC_MASKR でマスクする必要があります。**

24.10.12 SDMMC 割込みクリアレジスタ (SDMMC_ICR)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

SDMMC_ICR レジスタは、書き込み専用レジスタです。ビットに 1 を書き込むと、SDMMC_STAR ステータスレジスタの対応するビットがクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTCC	IDMA TEC	CK STOPC	VSW ENDC	ACK TIME OUTC	ACK FAILC	SDIO ITC	BUSY D0 ENDC	Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw	rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	D ABORT C	DBCK ENDC	DHOLD C	DATA ENDC	CMD SENTC	CMDR ENDC	RX OVERR C	TX UNDER RC	D TIME OUTC	C TIME OUTC	DCRC FAILC	CCRC FAILC
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **IDMABTCC** : IDMA バッファ転送完了クリアビット

IDMABTC フラグをクリアするために、ソフトウェアによってセットされます。

0 : IDMABTC はクリアされません。

1 : IDMABTC はクリアされます。

ビット 27 **IDMATEC** : IDMA 転送エラークリアビット

IDMATE フラグをクリアするために、ソフトウェアによってセットされます。

0 : IDMATE はクリアされません。

1 : IDMATE はクリアされます。

ビット 26 **CKSTOPC** : CKSTOP フラグクリアビット

CKSTOP フラグをクリアするために、ソフトウェアによってセットされます。

0 : CKSTOP はクリアされません。

1 : CKSTOP はクリアされます。

ビット 25 **VSWENDC** : VSWEND フラグクリアビット

VSWEND フラグをクリアするために、ソフトウェアによってセットされます。

0 : VSWEND はクリアされません。

1 : VSWEND はクリアされます。

ビット 24 **ACKTIMEOUTC** : ACKTIMEOUT フラグクリアビット

ACKTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。

0 : ACKTIMEOUT はクリアされません。

1 : ACKTIMEOUT はクリアされます。

ビット 23 **ACKFAILC** : ACKFAIL フラグクリアビット

ACKFAIL フラグをクリアするために、ソフトウェアによってセットされます。

0 : ACKFAIL はクリアされません。

1 : ACKFAIL はクリアされます。

ビット 22 **SDIOITC** : SDIOIT フラグクリアビット

SDIOIT フラグをクリアするために、ソフトウェアによってセットされます。

0 : SDIOIT はクリアされません。

1 : SDIOIT はクリアされます。

ビット 21 **BUSYD0ENDC** : BUSYD0END フラグクリアビット

BUSYD0END フラグをクリアするために、ソフトウェアによってセットされます。

0 : BUSYD0END はクリアされません。

1 : BUSYD0END がクリアされます。

ビット 20:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **DABORTC** : DABORT フラグクリアビット

DABORT フラグをクリアするために、ソフトウェアによってセットされます。

0 : DABORT はクリアされません。

1 : DABORT はクリアされます。

ビット 10 **DBCKENDC** : DBCKEND フラグクリアビット

DBCKEND フラグをクリアするために、ソフトウェアによってセットされます。

0 : DBCKEND はクリアされません。

1 : DBCKEND はクリアされます。

ビット 9 **DHOLD C** : DHOLD フラグクリアビット

DHOLD フラグをクリアするために、ソフトウェアによってセットされます。

0 : DHOLD はクリアされません。

1 : DHOLD はクリアされます。

ビット 8 **DATAENDC** : DATAEND フラグクリアビット

DATAEND フラグをクリアするために、ソフトウェアによってセットされます。

0 : DATAEND はクリアされません。

1 : DATAEND はクリアされます。

ビット 7 **CMDSENTC** : CMDSENT フラグクリアビット

CMDSENT フラグをクリアするために、ソフトウェアによってセットされます。

0 : CMDSENT はクリアされません。

1 : CMDSENT はクリアされます。

ビット 6 **CMDREND C** : CMDREND フラグクリアビット

CMDREND フラグをクリアするために、ソフトウェアによってセットされます。

0 : CMDREND はクリアされません。

1 : CMDREND はクリアされます。

ビット 5 **RXOVERRC** : RXOVERR フラグクリアビット

RXOVERR フラグをクリアするために、ソフトウェアによってセットされます。

0 : RXOVERR はクリアされません。

1 : RXOVERR はクリアされます。

ビット 4 **TXUNDERRC** : TXUNDERR フラグクリアビット

TXUNDERR フラグをクリアするために、ソフトウェアによってセットされます。

0 : TXUNDERR はクリアされません。

1 : TXUNDERR はクリアされます。

ビット 3 **DTIMEOUTC** : DTIMEOUT フラグクリアビット

DTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。

0 : DTIMEOUT はクリアされません。

1 : DTIMEOUT はクリアされます。

ビット 2 **CTIMEOUTC** : CTIMEOUT フラグクリアビット

CTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。

0 : CTIMEOUT はクリアされません。

1 : CTIMEOUT はクリアされます。

ビット 1 **DCRCFAILC** : DCRCFAIL フラグクリアビット

DCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。

0 : DCRCFAIL はクリアされません。

1 : DCRCFAIL はクリアされます。

ビット 0 **CCRCFAILC** : CCRCFAIL フラグクリアビット

CCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。

0 : CCRCFAIL はクリアされません。

1 : CCRCFAIL はクリアされます。

24.10.13 SDMMC マスクレジスタ (SDMMC_MASKR)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

割込みマスクレジスタは、対応するビットを 1 にセットすることによって、割込みリクエストを生成するステータスフラグを決めます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTCIE	Res.	CK STOP IE	VSW ENDIE	ACK TIME OUTIE	ACK FAILIE	SDIO ITIE	BUSY D0 ENDIE	Res.	Res.	TX FIFO EIE	RX FIFO FIE	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HFIE	TX FIFO HEIE	Res.	Res.	DA BORT IE	DBCK ENDIE	DHOLD IE	DATA ENDIE	CMD SENT IE	CMDR ENDIE	RX OVER RIE	TX UNDER RIE	D TIME OUTIE	C TIME OUTIE	DCRC FAILIE	CCRC FAILIE
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **IDMABTCIE** : IDMA バッファ転送完了割込み有効

IDMA がメモリバッファに属するすべてのデータを転送したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : IDMA バッファ転送完了割込みは無効です。

1 : IDMA バッファ転送完了割込みは有効です。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **CKSTOPIE** : 電圧スイッチクロック停止割込み有効

電圧スイッチクロック停止による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 電圧スイッチクロック停止割込みは無効です。

1 : 電圧スイッチクロック停止割込みは有効です。

ビット 25 **VSWENDIE** : 電圧スイッチの重要なタイミングセクション完了割込み有効

電圧スイッチの重要なタイミングセクションが完了したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 電圧スイッチの重要なタイミングセクション完了割込みは無効です。

1 : 電圧スイッチの重要なタイミングセクション完了割込みは有効です。

ビット 24 **ACKTIMEOUTIE** : 確認応答タイムアウト割込み有効

確認応答タイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 確認応答タイムアウト割込みは無効です。

1 : 確認応答タイムアウト割込みは有効です。

- ビット 23 **ACKFAILIE** : 確認応答失敗割込み有効
確認応答失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : 確認応答失敗割込みは無効です。
1 : 確認応答失敗割込みは有効です。
- ビット 22 **SDIOITIE** : SDIO モード割込み受信割込み有効
SDIO モード割込みを受信したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : SDIO モード割込み受信割込みは無効です。
1 : SDIO モード割込み受信割込みは有効です。
- ビット 21 **BUSYD0ENDIE** : BUSYD0END 割込み有効
SDMMC_D0 信号が CMD レスポンスの後にビジーからビジーではない状態に変更したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : BUSYD0END 割込みは無効です。
1 : BUSYD0END 割込みは有効です。
- ビット 20:19 予約済みであり、リセット値に保持する必要があります。
- ビット 18 **TXFIFOEIE** : Tx FIFO エンプティ割込み有効
Tx FIFO エンプティによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Tx FIFO エンプティ割込みは無効です。
1 : Tx FIFO エンプティ割込みは有効です。
- ビット 17 **RXFIFOIE** : Rx FIFO フル割込み有効
Rx FIFO がフルになったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Rx FIFO フル割込みは無効です。
1 : Rx FIFO フル割込みは有効です。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **RXFIFOHFIE** : Rx FIFO ハーフフル割込み有効
Rx FIFO がハーフフル状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Rx FIFO ハーフフル割込みは無効です。
1 : Rx FIFO ハーフフル割込みは有効です。
- ビット 14 **TXFIFOHEIE** : Tx FIFO ハーフエンプティ割込み有効
Tx FIFO がハーフエンプティ状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Tx FIFO ハーフエンプティ割込みは無効です。
1 : Tx FIFO ハーフエンプティ割込みは有効です。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **DABORTIE** : データ転送アボート割込み有効
アボートされているデータ転送による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : データ転送アボート割込みは無効です。
1 : データ転送アボート割込みは有効です。
- ビット 10 **DBCKENDIE** : データブロックエンド割込み有効
データブロックエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : データブロックエンド割込みは無効です。
1 : データブロックエンド割込みは有効です。

ビット 9 DHOLDIE : データホールド割込み有効

DPSM の Wait_S 状態で新しいデータの送信が保留されている場合に生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データホールド割込みは無効です。

1 : データホールド割込みは有効です。

ビット 8 DATAENDIE : データエンド割込み有効

データエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データエンド割込みは無効です。

1 : データエンド割込みは有効です。

ビット 7 CMDSENTIE : コマンド送信割込み有効

コマンド送信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド送信割込みは無効です。

1 : コマンド送信割込みは有効です。

ビット 6 CMDRENDIE : コマンドレスポンス受信割込み有効

コマンドレスポンスの受信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドレスポンス受信割込みは無効です。

1 : コマンドレスポンス受信割込みは有効です。

ビット 5 RXOVERRIE : Rx FIFO オーバーランエラー割込み有効

Rx FIFO オーバーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO オーバーランエラー割込みは無効です。

1 : Rx FIFO オーバーランエラー割込みは有効です。

ビット 4 TXUNDERRIE : Tx FIFO アンダーランエラー割込み有効

Tx FIFO アンダーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO アンダーランエラー割込みは無効です。

1 : Tx FIFO アンダーランエラー割込みは有効です。

ビット 3 DTIMEOUTIE : データタイムアウト割込み有効

データタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データタイムアウト割込みは無効です。

1 : データタイムアウト割込みは有効です。

ビット 2 CTIMEOUTIE : コマンドタイムアウト割込み有効

コマンドタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドタイムアウト割込みは無効です。

1 : コマンドタイムアウト割込みは有効です。

ビット 1 DCRCFAILIE : データ CRC 失敗割込み有効

データ CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ CRC 失敗割込みは無効です。

1 : データ CRC 失敗割込みは有効です。

ビット 0 CCRCFAILIE : コマンド CRC 失敗割込み有効

コマンド CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド CRC 失敗割込みは無効です。

1 : コマンド CRC 失敗割込みは有効です。

24.10.14 SDMMC 確認応答タイマレジスタ (SDMMC_ACKTIMER)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

SDMMC_ACKTIMER レジスタは、確認応答タイムアウト期間 (SDMMC_CK バスクロック周期単位) を含みます。

カウンタは SDMMC_ACKTIMER レジスタから値をロードして、データパスステートマシン (DPSM) が Wait_Ack 状態に入ったときに、デクリメントを開始します。DPSM がこの状態のときにタイマが 0 に達した場合、確認応答タイムアウトステータスフラグがセットされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:16]								
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACKTIME[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **ACKTIME[24:0]** : ブート確認応答タイムアウト期間

このビットは、CPSM が無効 (CPSMEN = 0) のときのみファームウェアで書き込むことができます。カードバスクロック周期で表されたブート確認応答タイムアウト期間です。

注 : データ転送は、データ制御レジスタに書き込まれる前に、確認応答タイマレジスタに書き込まれなければならない。

24.10.15 SDMMC データ FIFO レジスタ x (SDMMC_FIFORx)

アドレスオフセット : $0x080 + 0x004 * x$ ($x = 0$ から 15)

リセット値 : 0x0000 0000

受信および送信 FIFO は、ワード (32 ビット) 幅のレジスタとしてのみ読み書きできます。FIFO は、連続したアドレスに 16 個のエントリを含みます。このため、CPU はそのロードとストアで複数のオペランドを使用して、FIFO から読み書きできるようになります。FIFO レジスタインタフェースは、FIFO 内でデータが正しく配置されるよう処理します。CPU が使用する FIFO レジスタアドレスは重要です。

ハーフワードまたはバイトアクセスで SDMMC_FIFOR にアクセスすると、AHB バスフォールトが生成されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIFODATA[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODATA[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **FIFODATA[31:0]** : 受信および送信 FIFO データ

このレジスタは、DPSM がアクティブ (DPSMACT = 1) のときのみファームウェアで読み書きできます。FIFO データは、32 ビットワードの 16 個のエントリを占有します。

24.10.16 SDMMC DMA 制御レジスタ (SDMMC_IDMACTRLR)

アドレスオフセット : 0x050

リセット値 : 0x0000 0000

受信および送信 FIFO は、32 ビット幅のレジスタとして読み書きできます。FIFO は、32 の連続したアドレスに 32 個のエントリを含みます。このため、CPU はそのロードとストアで複数のオペランドを使用して、FIFO から読み書きできるようになります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMAB MODE	IDMA EN
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **IDMABMODE** : バッファモード選択

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : シングルバッファモード

1 : リンクされたリストモード

ビット 0 **IDMAEN** : IDMA 有効化

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : IDMA は無効です。

1 : IDMA は有効です。

24.10.17 SDMMC IDMA バッファサイズレジスタ (SDMMC_IDMABSIZER)

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

リンクされたリストの設定では、SDMMC_IDMABSIZER レジスタにバッファサイズが格納されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMA BNDT [11]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABNDT[10:0]											Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw					

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:5 **IDMABNDT[11:0]** : バッファごとのバイト数

この 12 ビット値は、32 ビットワードでのバッファサイズを得るには、8 倍する必要があります、バイト単位でのバッファのサイズを得るには 32 倍する必要があります。

例 : IDMABNDT = 0x001 : バッファサイズ = 8 ワード = 32 バイト。

例 : IDMABNDT = 0x800 : バッファサイズ = 16384 ワード = 64 KB。

これらのビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

24.10.18 SDMMC IDMA バッファベースアドレスレジスタ (SDMMC_IDMABASER)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

SDMMC_IDMABASER レジスタには、シングルバッファ設定およびリンクされたリスト設定でメモリバッファベースアドレスが格納されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABASE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

ビット 31:0 **IDMABASE[31:0]** : バッファメモリベースアドレスビット [31:2] は、ワード境界に合わせる必要があります (ビット [1:0] は常に 0 で読出し専用です)。

このレジスタは、DPSM がインアクティブ (DPSMACT = 0) のときファームウェアによって書き込むことができ、DPSM がアクティブ (DPSMACT = 1) のときはファームウェアによって動的に書き込むことができます。

24.10.19 SDMMC IDMA リンクされたリストのアドレスレジスタ (SDMMC_IDMALAR)

アドレスオフセット : 0x064

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ULA	ULS	ABR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMALA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

ビット 31 **ULA** : リンクされたリストモード (SDMMC_IDMACTRLR.IDMABMODE でリンクされたリストモードを選択) の場合、リンクされたリストから SDMMC_IDMALAR を更新します。

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : SDMMC_IDMALAR は更新されず、最後のリンクされたリスト項目です。

1 : SDMMC_IDMALAR は、リンクされたリストのテーブルから更新されます。

ビット 30 **ULS** : リンクされたリストモードの場合、次のリンクされたリストから SDMMC_IDMABSIZE を更新します (SDMMC_IDMACTRLR.IDMABMODE でリンクされたリストモードを選択し、ULA = 1 の場合)。

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

0 : SDMMC_IDMABSIZE は、次のリンクされたリストのテーブルからは更新されません。

1 : SDMMC_IDMABSIZE は、次のリンクされたリストのテーブルから更新されます。

ビット 29 **ABR** : リンクされたリストバッファの準備完了確認応答

このビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

ソフトウェアでプログラムされたレジスタ情報から最初のリンクされたリストバッファを開始する場合、このビットは考慮されません。ABR は、後続してロードされるリンクされたリストの項目においてのみ考慮されます。

0 : ロードされたリンクされたリストバッファの準備ができていません (これにより、リンクされたリストの IDMA 転送エラーが生成されます)。

1 : ロードされたリンクされたリストバッファの準備完了確認応答。リンクされたリストバッファのデータは IDMA で転送されます。

ビット 28:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:2 **IDMALA[13:0]** : ワード境界調整された、リンクされたリスト項目のアドレスオフセット

次のリンクされたリスト項目構造のベースへの、リンクされたリスト項目のオフセットポインタ。

リンクされたリスト項目のベースアドレスは IDMAA + IDMALA となります。

これらのビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

24.10.20 SDMMC IDMA リンクされたリストメモリのベースレジスタ (SDMMC_IDMABAR)

アドレスオフセット : 0x068
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABA[29:14]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

ビット 31:2 **IDMABA[29:0]** : ワード境界調整された、リンクされたリストメモリのベースアドレス
リンクされたリストメモリのベースポインタ。
これらのビットは、DPSM がインアクティブ (DPSMACT = 0) のときのみファームウェアで書き込むことができます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

24.10.21 SDMMC レジスタマップ

表 235. SDMMC レジスタマップ

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	SDMMC_ POWER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRPOL	VSWITCHEN	VSWITCH	PWRCtrl[1:0]	
	リセット値																													0	0	0	0	0
0x004	SDMMC_ CLKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUSPEED	DDR	HWFC_EN	NEGEDGE	WIDBUS[1:0]		Res.	PWRSV	Res.	Res.	CLKDIV[9:0]										
	リセット値											0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0
0x008	SDMMC_ ARGR	CMDARG[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C	SDMMC_ CMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDSPEND	BOOTEN	BOOTMODE	DTHOLD	CPSMEN	WAITPEND	WAITINT	WAITRESPI[1:0]		CMDSTOP	CMDTRANS	CMDINDEX[5:0]						
	リセット値																																	
0x010	SDMMC_ RESPCMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		RESPCMD[5:0]					
	リセット値																											0	0	0	0	0	0	
0x014	SDMMC_ RESP1R	CARDSTATUS[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x018	SDMMC_ RESP2R	CARDSTATUS[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x01C	SDMMC_ RESP3R	CARDSTATUS[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x020	SDMMC_ RESP4R	CARDSTATUS[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x024	SDMMC_ DTIMER	DATETIME[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x028	SDMMC_ DLENR	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:0]																										
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x02C	SDMMC_ DCTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFIRST	BOOTACKEN	SIOEN	RWMOD	RWSTOP	RWSTART	DBLOCK SIZE[3:0]			DTMODE[1:0]		DTDIR	DTEN	
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0	

表 235. SDMMC レジスタマップ (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x030	SDMMC_ DCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:0]																												
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x034	SDMMC_ STAR	Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOT	BUSYDOEND	BUSYD0	RXFIOE	TXFIOE	RXFIOF	TXFIOF	RXFIOFH	TXFIOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL				
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x038	SDMMC_ ICR	Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWENDC	ACKTIMEOUTC	ACKFAILC	SDIOTC	BUSYDOENDC		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DABORTC	DBCKENDC	DHOLD	DATAENDC	CMDSENTC	CMDREND	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC				
	リセット値				0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0				
0x03C	SDMMC_ MASKR	Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILE	SDIOTIE	BUSYDOENDIE		Res.	Res.	TXFIOEIE	RXFIOFIE	Res.	TXFIOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILE	CCRCFAILE				
	リセット値				0		0	0	0	0	0	0				0	0		0	0		0	0	0	0	0	0	0	0	0	0	0	0				
0x040	SDMMC_ ACKTIMER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:0]																												
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x044 - 0x04C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
0x050	SDMMC_ IDMACTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABMODE	IDMAEN				
	リセット値																														0	0					
0x054	SDMMC_ IDMABSIZE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABNDT[11:0]													Res.	Res.	Res.	Res.				
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0									
0x058	SDMMC_ IDMABASER	IDMABASE[31:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x05C - 0x060	予約済み	Res.																																			
0x064	SDMMC_ IDMALAR	ULA	ULS	ABR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMALA[13:0]													Res.	Res.						
	リセット値	0	0	0													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x068	SDMMC_ IDMABAR	IDMABA[29:0]																													Res.	Res.					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			Res.	Res.			
0x06C - 0x07C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			

表 235. SDMMC レジスタマップ (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x080 + 0x04 * x (x = 0 ~ 15)	SDMMC_ FIFOR	FIFODATA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

25 遅延ブロック (DLYB)

25.1 概要

遅延ブロック (DLYB) は、入力クロックからディフェーズ (位相をずら) した出力クロックを生成するために使用します。出力クロックのフェーズは、ユーザアプリケーションでプログラムする必要があります。次に、出力クロックを使用して、SDMMC または Octo-SPI インタフェースなどの別のペリフェラルで受信するデータにクロック供給します。

遅延は電圧や温度に依存するため、場合によってはアプリケーションで出力クロックの位相を再設定し、受信データと中央揃えを再度行う必要があるかもしれません。

25.2 DLYB の主な機能

遅延ブロックは、次のような機能を備えています。

- 入力クロック周波数は 25 MHz から通信インタフェースでサポートされる最大周波数までの範囲 (データシートを参照)
- 最大 12 のオーバーサンプリングフェーズ

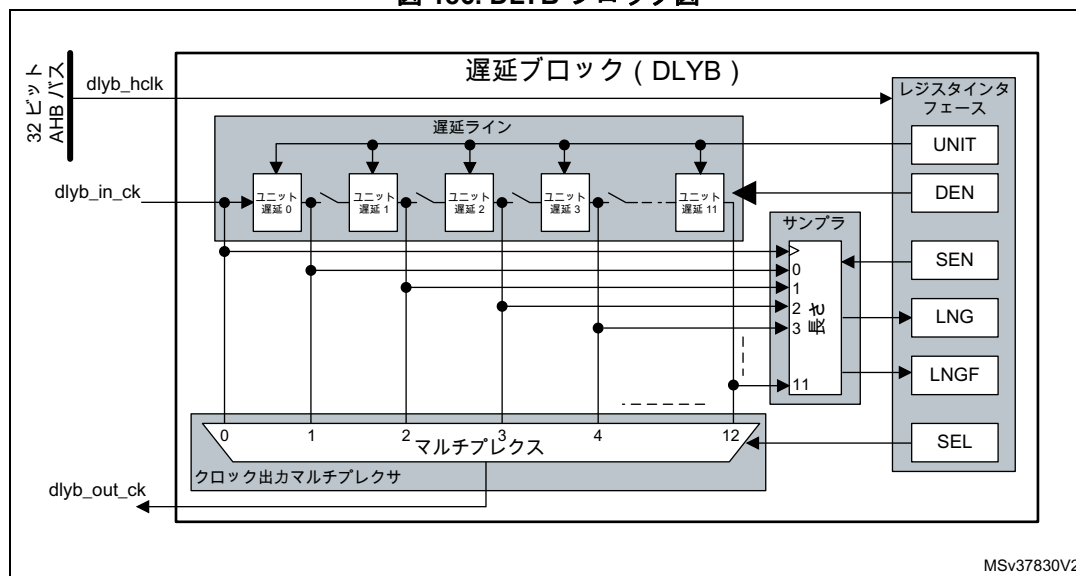
25.3 DLYB の機能説明

25.3.1 DLYB 図

遅延ブロックは、次のようなサブブロックから成ります (次の図に示します)。

- DLYB レジスタへの AHB アクセスを可能にするレジスタインタフェースブロック
- ユニット遅延をサポートする遅延ライン
- 遅延ライン長さサンプリング
- 出力クロック選択マルチプレクサ

図 186. DLYB ブロック図



25.3.2 DLYB ピンおよび内部信号

表 236 に DLYB 内部信号を示します。

表 236. DLYB 内部入力／出力信号

信号名	信号タイプ	説明
dlyb_hclk	デジタル入力	遅延ブロックレジスタのインタフェースクロック
dlyb_in_ck	デジタル入力	遅延ブロック入力クロック
dlyb_out_ck	デジタル出力	遅延ブロック出力クロック

25.3.3 概要

遅延ブロックは、DLYB 制御レジスタ (DLYB_CR) の DEN ビットをセットすることによって有効にします。長さサンプリングは、DLYB_CR レジスタの SEN ビットによって有効にします。

遅延ブロックが有効なとき、ユニット遅延によって追加される遅延は DLYB 設定レジスタ (DLYB_CFGR) の UNIT[6:0] フィールドで定義します。

注： UNIT[6:0] は、出力クロックが無効 (SEN = 1) のときのみプログラム可能です。

遅延ブロックが有効なとき、出力クロックの位相は DLYB_CFGR レジスタの SEL[3:0] フィールドによって選択します。

注： SEL は出力クロックが無効 (SEN = 1) のときのみプログラム可能です。

遅延ラインの長さを 1 周期にまたがるように設定することにより、出力クロックは入力クロックの 1 周期にわたって位相をずらすことができます。遅延ラインの長さは SEN ビットによって長さサンプリングを有効にすることで設定でき、それにより DLYB_CFGR の遅延ライン長さ (LNG[11:0] および長さ有効フラグ (LNGF) へのアクセスが可能になります。

入力クロックの 1 周期より短い出力クロック遅延が必要な場合は、遅延線の長さを短くすることができます。これにより、より小さなユニット遅延が可能になり、より高い分解能が得られます。

遅延ライン長さが設定されると、出力クロックマルチプレクサによって、位相がずらされた出力クロックを選択することができます。この選択は SEL[3:0] で行います。出力クロックは SEN が 0 にセットされている場合にのみ、選択された位相で使用できます。

下の表に、遅延ブロック制御の概要を示します。

表 237. 遅延ブロック制御

DEN	SEN	UNIT	SEL	LNG	LNGF	出力クロック
0	0	無視	無視	無視	無視	有効 (= 入力クロック)
x	1	ユニット遅延	出力クロックフェーズ	長さ	長さフラグ	無効
1	0	ユニット遅延 ⁽¹⁾	出力クロックフェーズ ⁽²⁾	無視	無視	有効 (= 選択されたフェーズ)

1. ユニット遅延は SEN = 1 の場合にのみ変更できます。

2. 出力クロックの位相は SEN = 1 の場合にのみ変更できます。

25.3.4 遅延ライン長さ設定手順

LNG[11:0] を使用して、入力クロック周期に対する遅延ライン長さを決定します。遅延ライン長さは、入力クロックの 1 周期全体をカバーできるような長さに設定しなければなりません。

遅延線には 12 個のユニット遅延要素がありますが、次の手順の説明では 0~10 の間の長さが返されることに注意してください。1 入力クロックサイクル全体にわたって遅延が調整されることを保証するために、最上位の遅延出力値が使用されるからです。クロック周波数および UNIT 値によっては、ユニット遅延要素 10 もクロックサイクル長から切捨てられることがあります。

チューニング手順全体を通じて、クロック入力（フリーランニングクロック）が必要です。

入力クロック 1 周期に対する遅延ライン長さを設定するには、以下のシーケンスに従います。

1. DEN ビットを 1 に設定して遅延ブロックを有効にします。
2. SEN ビットを 1 に設定して長さサンプリングを有効にします。
3. SEL[3:0] を 12 に設定してすべての遅延セルを有効にします。
4. UNIT[6:0] = 0~127 に対して（遅延ライン長さが設定されるまで、このステップを繰り返すこと）：
 - a) UNIT[6:0] の値を更新し、長さフラグ LNGF が 1 にセットされるのを待ちます。
 - b) LNG[11:0] を読み出します。LNG[10:0] > 0、かつ LNG[11] または LNG[10] = 0 の場合、遅延ライン長さは入力クロック 1 周期に対して設定されます。
5. 入力クロック 1 周期にわたるユニット遅延の数 (N) を決定します。N = 0~10 について、LNG[N] = 1 の場合、入力クロック周期にわたるユニット遅延の数 = N となります。
6. SEN ビットを 0 クリアして長さサンプリングを無効にします。

入力クロックの 1 周期より短い出力クロック遅延が必要な場合は、遅延線の長さを入力クロックの 1 周期より短くすることができます。これにより、より小さなユニット遅延が可能になり、短い時間間隔にわたってより高い分解能が得られます。

25.3.5 出力クロック位相の設定手順

遅延ライン長さを入力クロック 1 周期に対して設定する場合、入力クロック 1 周期の期間中に発生するユニット遅延の間で出力クロックフェーズを選択することができます。

出力クロックフェーズを選択するには以下の手順に従います。

1. SEN ビットを 1 にセットすることで、出力クロックを無効にし、位相選択 SEL[3:0] ビットへのアクセスを有効にします。
2. 希望する出力クロックの位相値で SEL[3:0] をプログラムします。
3. SEN を 0 クリアすることにより、選択された位相で出力クロックを有効にします。

Octo-SPI の使用例

遅延ブロックは Octo-SPI インタフェースと組み合わせて使用され、入力データサンプリング信号をシフトできるようになります。このサンプリング信号は、フィードバッククロックまたはデータストロープ (DQS) 信号であり、特定のタイプのデバイスによって供給されます。DQS を使用する場合、DQS は不連続信号であるため、事前にフリーランニングクロックを使用して較正手順を実行する必要があります。

SDR（シングルデータレート）モードの場合、ユーザは通常、サンプリングエッジが有効なデータ 位相の中央に位置するように、サンプリング信号を半周期シフトする必要があります。

DDR（デュアルデータレート）モードの場合、周期の開始時と中間時にデータが遷移するため、較正が完了したとき、標準値は N/4 に近くなる必要があります。

高周波数でタイミング制約が厳しい場合、遅延設定の粒度（10）が粗すぎる場合があります。ほとんどの場合、サンプリングクロック周期全体をカバーする可能遅延値は必要ないため、サンプリングエッジ位置の精度を向上させるために、アプリケーションによって「ユニット」値を上書きすることができます（例：2 倍小さい「ユニット」値を提供すると、タイミング精度は 2 倍向上します。これに相当することは、最大可能遅延値を 2 で割ることです）。

SDMMC の使用例

遅延ブロックは、SDMMC インタフェースの可変遅延と組み合わせて使用されます。サンプリングポイントを正しく調整するには、遅延値が SDMMC_CLK クロック周期全体をカバーする必要があります。遅延線の長さを調整した後、個々の遅延をサンプリングポイントの調整に使用して、最適なサンプリングポイントを見つけます。

25.4 DLYB レジスタ

すべてのレジスタはワード、ハーフワード、バイトでアクセスすることができます。

25.4.1 DLYB 制御レジスタ (DLYB_CR)

アドレスオフセット：0x000

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEN	DEN
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SEN** : サンプラ長さ有効ビット

0 : サンプラ長さと、UNIT[6:0] および SEL[3:0] へのレジスタのアクセスは無効、出力クロックは有効です。

1 : サンプラ長さと、UNIT[6:0] および SEL[3:0] へのレジスタのアクセスは有効、出力クロックは無効です。

ビット 0 **DEN** : 遅延ブロック有効ビット

0 : DLYB は無効です。

1 : DLYB は有効です。

25.4.2 DLYB 設定レジスタ (DLYB_CFGR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LNGF	Res.	Res.	Res.	LNG[11:0]											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UNIT[6:0]							Res.	Res.	Res.	Res.	SEL[3:0]			
	rW	rW	rW	rW	rW	rW	rW					rW	rW	rW	rW

ビット 31 **LNGF** : 長さ有効フラグ

このフラグは UNIT[6:0] ビットの変更後、LNG[11:0] に含まれる遅延ライン長さの値が有効になった時を知らせます。

0 : LNG の長さの値は無効です。

1 : LNG の長さの値は有効です。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **LNG[11:0]** : 遅延ライン長さの値

これらのビットには、入カクロックの立上がりエッジでサンプリングされた 12 個のユニット遅延の値が反映されます。

値は LNGF = 1 のときにのみ有効です。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:8 **UNIT[6:0]** : ユニット遅延セルの遅延

これらのビットは、SEN = 1 のときにのみ書き込むことができます。

ユニット遅延 = 初期遅延 + UNIT[6:0] x 遅延ステップ (刻み)

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **SEL[3:0]** : 出カクロックの位相

これらのビットは、SEN = 1 のときにのみ書き込むことができます。

出カクロックの位相 = 入カクロック + SEL[3:0] x ユニット遅延

25.4.3 DLYB レジスタマップ

表 238. DLYB レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DLYB_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEN	DEN	
	リセット値																														0	0	
0x004	DLYB_CFGR	LNGF	Res.	Res.	Res.	LNG										Res.	UNIT							Res.	Res.	Res.	Res.	SEL					
	リセット値	0				0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0					0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

26 アナログデジタルコンバータ (ADC1/2)

26.1 概要

このセクションでは、最大 2 つ実装されている ADC について説明します。

- ADC1 および ADC2 は密接に結合されており、デュアルモードで動作できます (ADC1 はマスターです)。

各 ADC は、12 ビットの逐次比較型アナログデジタルコンバータで構成されています。

各 ADC には、最大 20 の多重化チャネルがあります。さまざまなチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めで 16 ビットのデータレジスタに格納されます。

ADC は AHB バスに配置され、データを高速で処理できます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

組み込みのオーバーサンプリング回路により、CPU の計算負荷を軽減しながらアナログ性能を高めることができます。

低周波数で非常に低い消費電力を可能にするために、効率的な低電力モードが実装されています。

26.2 ADC の主な機能

- ハイパフォーマンス機能
 - 最大 2 つの ADC、デュアルモードで動作可能
 - ADC1 は、18 個の外部チャネルと 2 つの内部チャネルに接続
 - ADC2 は、18 個の外部チャネルと 2 つの内部チャネルに接続
 - 12、10、8、または 6 ビットに設定可能な分解能
 - ADC 変換時間は AHB バスクロック周波数から独立
 - 分解能を下げることで変換時間を短縮可
 - シングルエンドまたは差動入力を管理
 - 高速データ処理を可能にする AHB スレーブバスインタフェース
 - 自己較正
 - チャネル単位でプログラム可能なサンプリング時間
 - 柔軟なサンプリング時間制御
 - 最大 4 つのインジェクトチャネル (アナログ入力をレギュラまたはインジェクトチャネルへ完全に割り当て設定可能)
 - 高速コンテキスト切り替えを可能にするためにインジェクトチャネルの内容を準備するハードウェアによる支援
 - 組み込みのデータコヒーレンシによるデータ配置
 - データは、レギュラチャネル変換において、DMA で管理可能
 - 4 つのインジェクトチャネル専用のデータレジスタ
- 低電力機能
 - 低周波数での動作時に ADC 消費電流を低減するための速度適応型低電力モード
 - 最適な ADC 性能を維持しながら、低いバス周波数を適用可能

- 低 AHB バスクロック周波数アプリケーション（自動遅延モード）で ADC オーバーランを避けるための自動制御を提供
- オーバーサンプリング回路
 - 16 ビットデータレジスタ
 - 2 ~ 256 倍までオーバーサンプリング比を調整可能
 - 最大 8 ビットまでプログラム可能なデータシフト
- データプレコンディショニング
 - オフセット補正
- アナログ入力チャネル
 - 外部アナログ入力（ADC から）：
 - GPIO パッドから最大 6 個の高速チャネル
 - GPIO パッドから最大 12 個の低速チャネル
 - 内部温度センサ用の 1 つのチャネル (V_{SENSE})
 - 内部基準電圧用の 1 つのチャネル (V_{REFINT})
 - 外部 VBAT 電源ピンを監視するための 1 つのチャネル
 - 内部 V_{DDCORE} 電源を監視するための 1 つのチャネル
- 変換開始は、次のように開始可能
 - レギュラ変換とインジェクト変換の両方について、ソフトウェアによって開始
 - レギュラ変換とインジェクト変換の両方について、極性が設定可能なハードウェアトリガによって開始（内部タイマイイベントまたは GPIO 入力イベント）
- 変換モード
 - 各 ADC では単一チャネルの変換、または一連のチャネルをスキャン可能
 - シングルモードは、選択された入力をトリガごとに 1 回変換します。
 - 連続モードは、選択された入力を連続的に変換します。
 - 不連続モード
- ADC レディ時、サンプリング終了時、（レギュラまたはインジェクト）変換終了時、（レギュラまたはインジェクト）シーケンス変換終了時、アナログウォッチドッグ 1、2、3、またはオーバーランイベント時に割込みを生成
- ADC ごとの 3 つのアナログウォッチドッグ
 - ウォッチドッグはフィルタリングを行って、範囲外のデータを無視できます。
- ADC 入力電圧範囲： $V_{SSA} \leq V_{IN} \leq V_{REF+}$

図 187 に、1 つの ADC のブロック図を示します。

26.3 ADC の実装

表 239. ADC の機能

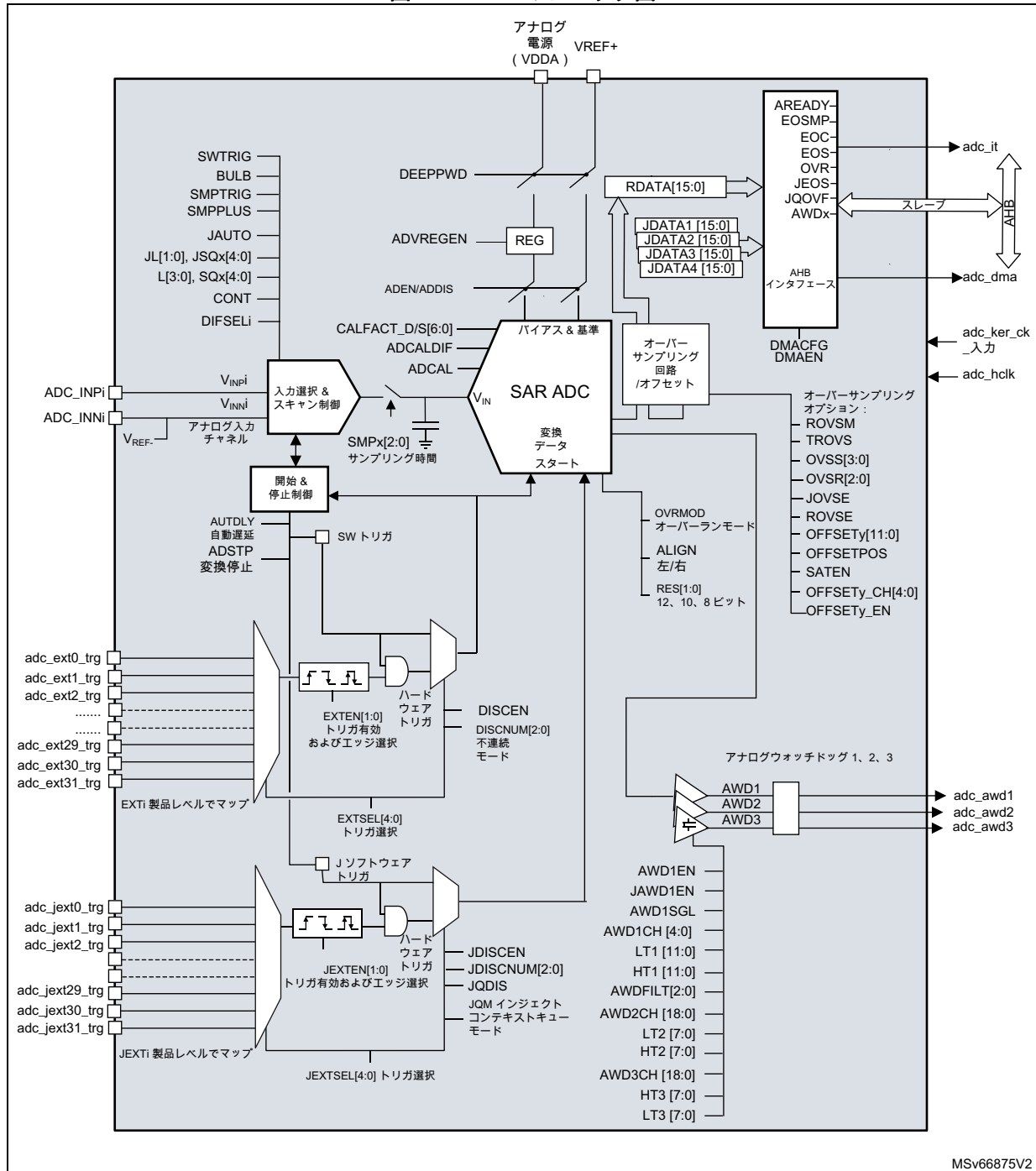
A/D コンバータのモード／機能	ADC1	ADC2
分解能	12 ビット	
最大サンプリング速度	5 Msps (分解能 : 12 ビット)	
デュアルモード動作	X	
ハードウェアオフセット較正	X	
ハードウェア直線性較正	-	
シングルエンド入力	X	
差動入力	X	
インジェクトチャネル変換	X	
オーバーサンプリング	最大 x256	
データレジスタ	16 ビット	
データレジスタ FIFO 深さ	3 ステージ	
DMA サポート	X	
ADF へのパラレルデータ出力	-	
オフセット補正	X	
ゲイン補正	-	
アナログウォッチドッグの数	3	
オプションレジスタ	X	

26.4 ADC の機能説明

26.4.1 ADC のブロック図

図 187 に ADC のブロック図を、表 240 に ADC ピンの説明を示します。

図 187. ADC のブロック図



26.4.2 ADC ピンおよび内部信号

表 240. ADC の入出力ピン

ピン名	信号タイプ	説明
VDDA	入力、アナログ電源供給	ADC のアナログ電源と正基準電圧
VSSA	入力、アナログ供給 グラウンド	V _{SS} に等しいアナログ電源供給のグラウンド。
VREF+	入力、アナログ 基準電圧正	A/D コンバータのハイレベル／正基準電圧。
VREF-	入力、アナログ 基準電圧負	ADC のローレベル／負基準電圧。V _{REF-} は内部的に V _{SSA} に接続されています。
ADC1/2_INNi/INPi	正負の外部アナログ入力 信号	20 個の正負の外部アナログ入力チャネル（詳細については、 セクション 26.4.4 : ADC 接続性 を参照）

表 241. ADC 内部入力／出力信号

内部信号名	信号タイプ	説明
V _{INPi}	正のアナログ入力 チャネル	ADC1/2_INPi 外部チャネルまたは内部チャネルのいずれかに接続された正の内部アナログ入力チャネル。
V _{INNi}	負のアナログ入力 チャネル	ADC1/2_INNi 外部チャネルまたは内部チャネルのいずれかに接続された負の内部アナログ入力チャネル。
adc_ext_trgi	入力	レギュラ変換の ADC 外部トリガ入力。これらの入力は、ADC マスタと ADC スレーブの間で共有されています。
adc_jext_trgi	入力	インジェクト変換の ADC 外部トリガ入力。これらの入力は、ADC マスタと ADC スレーブの間で共有されています。
adc_awdx	出力	オンチップタイマに接続された内部アナログウォッチドッグの出力信号 (x = アナログウォッチドッグの数 1、2、3)
adc_ker_ck_input	出力	ADC カーネルクロック
adc_hclk	入力	ADC ペリフェラルクロック
adc_it	出力	ADC 割込み
adc_dma	出力	ADC DMA リクエスト

表 242. ADC 相互接続

信号名	転送元／転送先
ADC1 V _{INP} [16]	VSENSE（内部温度センサ出力電圧）。
ADC1 V _{INP} [17]	V _{REFINT} （内部基準電圧からの出力電圧）。
ADC2 V _{INP} [16]	V _{BAT} /4（4で分圧された VBAT ピン入力電圧）。
ADC2 V _{INP} [17]	V _{DDCORE} （内部デジタルコア電圧）。
adc_ext_trg0	tim1_oc1
adc_ext_trg1	tim1_oc2
adc_ext_trg2	tim1_oc3
adc_ext_trg3	tim2_oc2
adc_ext_trg4	tim3_trgo
adc_ext_trg5	tim4_oc4

表 242. ADC 相互接続 (続き)

信号名	転送元/転送先
adc_ext_trg6	exti11
adc_ext_trg7	tim8_trgo
adc_ext_trg8	tim8_trgo2
adc_ext_trg9	tim1_trgo
adc_ext_trg10	tim1_trgo2
adc_ext_trg11	tim2_trgo
adc_ext_trg12	tim4_trgo
adc_ext_trg13	tim6_trgo
adc_ext_trg14	tim15_trgo
adc_ext_trg15	tim3_oc4
adc_ext_trg16	exti15
adc_ext_trg17	予約済み
adc_ext_trg18	lptim1_ch1
adc_ext_trg19	lptim2_ch1
adc_ext_trg20	予約済み
adc_ext_trg21	予約済み
adc_ext_trg22	予約済み
adc_ext_trg23	予約済み
adc_ext_trg24	予約済み
adc_ext_trg25	予約済み
adc_ext_trg26	予約済み
adc_ext_trg27	予約済み
adc_ext_trg28	予約済み
adc_ext_trg29	予約済み
adc_ext_trg30	予約済み
adc_ext_trg31	予約済み
adc_jext_trg0	tim1_trgo
adc_jext_trg1	tim1_oc4
adc_jext_trg2	tim2_trgo
adc_jext_trg3	tim2_oc1
adc_jext_trg4	tim3_oc4
adc_jext_trg5	tim4_trgo
adc_jext_trg6	exti15
adc_jext_trg7	tim8_oc4
adc_jext_trg8	tim1_trgo2
adc_jext_trg9	tim8_trgo
adc_jext_trg10	tim8_trgo2

表 242. ADC 相互接続 (続き)

信号名	転送元／転送先
adc_jext_trg11	tim3_oc3
adc_jext_trg12	tim3_trgo
adc_jext_trg13	tim3_oc1
adc_jext_trg14	tim6_trgo
adc_jext_trg15	tim15_trgo
adc_jext_trg16	予約済み
adc_jext_trg17	予約済み
adc_jext_trg18	lptim1_ch1
adc_jext_trg19	lptim2_ch1
adc_jext_trg20	予約済み
adc_jext_trg21	予約済み
adc_jext_trg22	予約済み
adc_jext_trg23	予約済み
adc_jext_trg24	予約済み
adc_jext_trg25	予約済み
adc_jext_trg26	予約済み
adc_jext_trg27	予約済み
adc_jext_trg28	予約済み
adc_jext_trg29	予約済み
adc_jext_trg30	予約済み
adc_jext_trg31	予約済み

26.4.3 ADC クロック

デュアルクロックドメインアーキテクチャ

デュアルクロックドメインアーキテクチャは、ADC クロックが AHB バスクロックから独立していることを意味します。

ADC 入力クロックは、2 つの異なるクロックソースから選択できます (図 188: ADC クロック構成)。

1. ADC クロックは、AHB クロックから独立した、非同期の特定のクロックソース (adc_ker_ck_input) にすることができます。

ADC 専用クロックの生成方法の詳細については、リセットおよびクロック制御 (RCC) のセクションを参照してください。この構成を選択するには、ADC_CCR レジスタの CKMODE[1:0] ビットが「00」である必要があります。

2. ADC クロックは、AHB クロックインタフェースから生成し、1、2、または 4 のプログラム可能な係数で分周することができます。この構成を選択するには、ADC_CCR の CKMODE[1:0] ビットが「00」以外の値である必要があります。プログラム可能な分周比は、ADC_CCR の CKMODE[1:0] ビットで設定することができます。

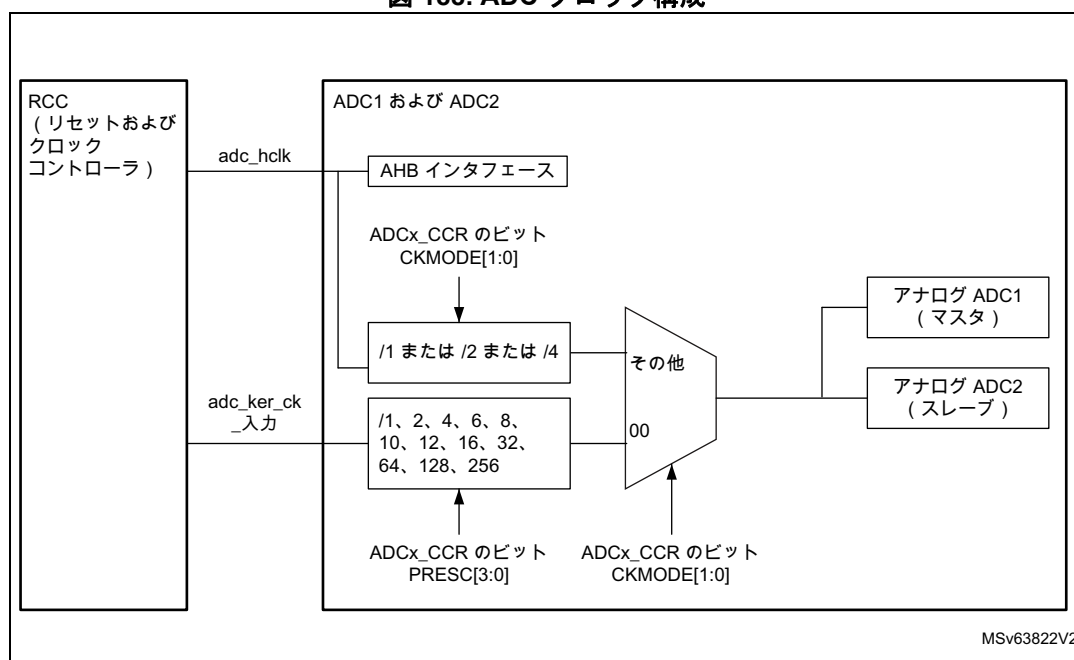
AHB プリスケアラが 1 (RCC_CFGR レジスタで HPRE[3:0] = 0xxx) の場合のみ、プリスケアラ係数 1 (CKMODE[1:0] = 01) を使用できます。

オプション 1 には、選択された AHB クロック構成にかかわらず、最大の ADC クロック周波数に到達できるという利点があります。ADC クロックは、最終的には 1、2、4、6、8、12、16、32、64、128、256 の比率で分周できます。このとき、ADC_CCR レジスタの PRESC[3:0] ビットで設定されたプリスケアラを使用します。

オプション 2 には、クロックドメインの再同期を迂回するという利点があります。これは、ADC がタイマによってトリガされるときと、アプリケーションが ADC の確実で精密なトリガを必要とする場合に便利です（そうしないと、トリガインスタンスの不確実性は、2 つのクロックドメイン間の再同期によって高まります）。

CKMODE[1:0] ビットで設定されたクロックは、デバイスデータシートに指定された動作周波数と一致している必要があります。

図 188. ADC クロック構成



ADC クロックと AHB クロックの間におけるクロック比の制約

通常、ADC クロックと AHB クロックの間で注意すべき制約はありませんが、インジェクトチャンネルがプログラムされている場合は例外です。この場合、次の比率に注意する必要があります。

- すべてのチャンネルの分解能が 12 ビットまたは 10 ビットである場合、 $F_{\text{adc_hclk}} \geq F_{\text{ADC}} / 4$
- いくつかのチャンネルの分解能が 8 ビットである場合（それを下回る分解能のチャンネルがない場合）、 $F_{\text{adc_hclk}} \geq F_{\text{ADC}} / 3$
- いくつかのチャンネルの分解能が 6 ビットである場合、 $F_{\text{adc_hclk}} \geq F_{\text{ADC}} / 2$

ADC クロック間の制約

いくつかの ADC インタフェースを同時に使用している場合は、RCC ブロックから同じクロックソースを使用する必要があります。すべての ADC インタフェースに対するプリスケアラ比はありません。

26.4.4 ADC 接続性

ADC 入力は、以下に示すように内部ソースに加えて外部チャンネルにも接続されています。

図 189. ADC1 接続性

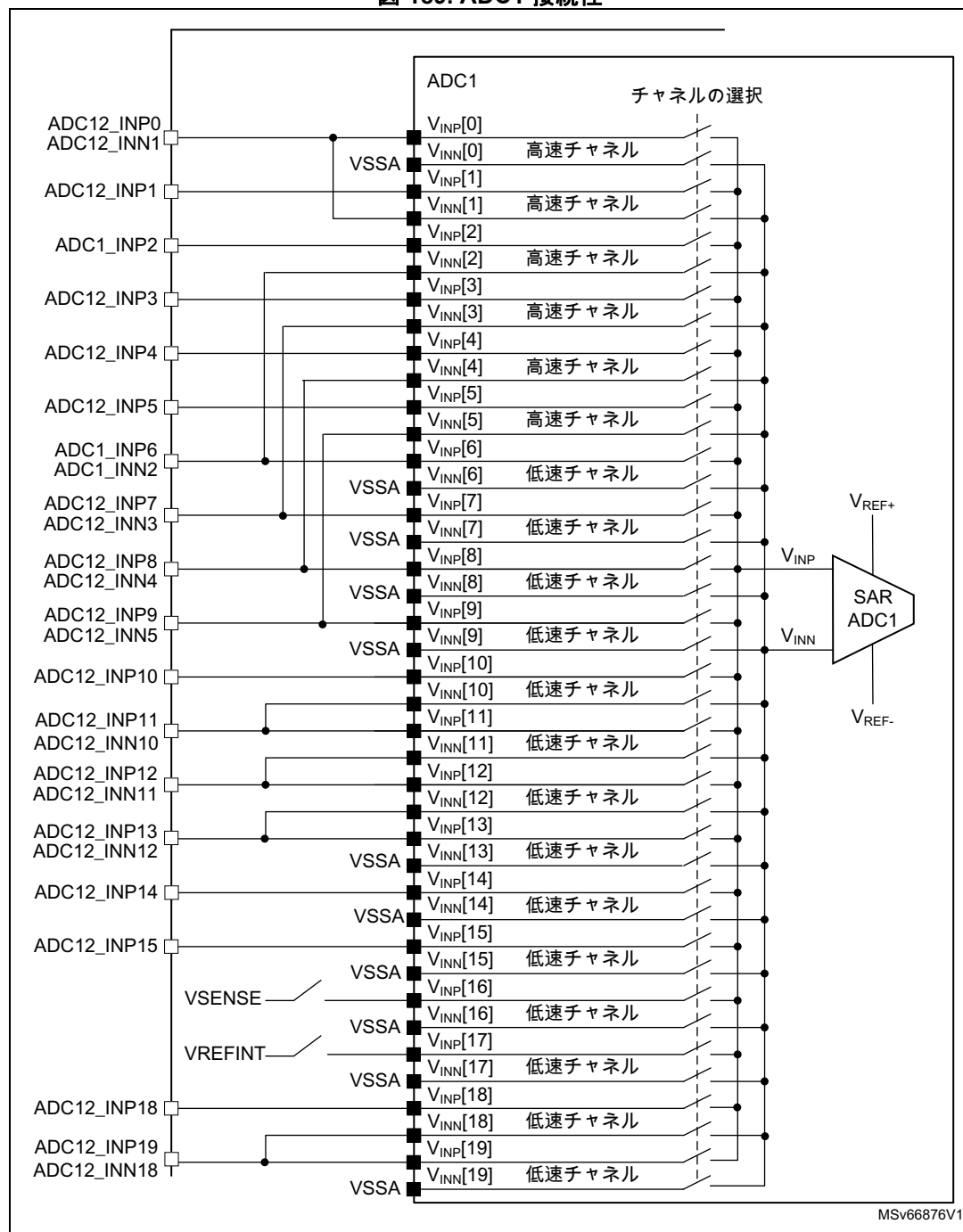
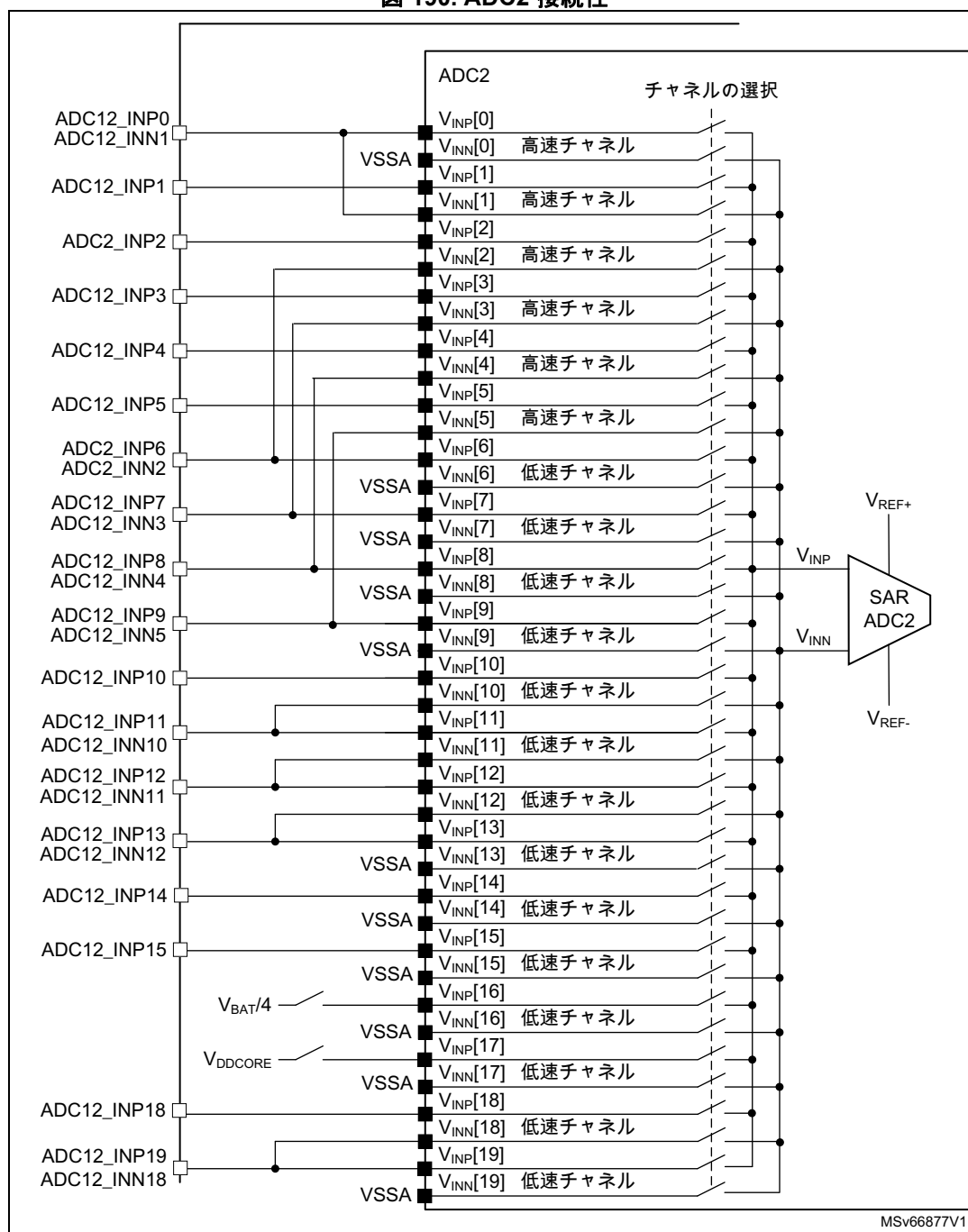


図 190. ADC2 接続性



26.4.5 スレーブ AHB インタフェース

ADC は、制御／ステータスレジスタおよびデータアクセスのために AHB スレーブポートを実装しています。AHB インタフェースの機能を次に示します。

- ワード (32 ビット) アクセス
- シングルサイクルレスポンス
- レジスタへのすべての読み出し／書き込みアクセスに対してゼロウェイトステートの応答。

AHB スレーブインタフェースでは分割／再試行リクエストをサポートしておらず、また AHB エラーも発生しません。

26.4.6 ADC ディープパワーダウンモード (DEEPPWD) および ADC 電圧レギュレータ (ADVREGEN)

デフォルトでは、リーク電流を低減するために電源が内部的にオフになっている (ADC_CR レジスタのビット DEEPPWD のリセット状態が 1 である) 場合、ADC はディープパワーダウンモードになります。

ADC の動作を開始するには、まずビット DEEPPWD = 0 をセットしてディープパワーダウンモードを終了する必要があります。

そして、ADC_CR レジスタでビット ADVREGEN = 1 にセットすることによって、ADC 内部電圧レギュレータを有効にする必要があります。ソフトウェアは、較正を起動するか ADC を有効化する前に、ADC 電圧レギュレータの起動時間の間待つ必要があります。(T_{ADCVREG_STUP}) この遅延は、ソフトウェアによって実装する必要があります。

ADC 電圧レギュレータの起動時間については、デバイスデータシートで T_{ADCVREG_STUP} パラメータを参照してください。

ADC の動作が完了すると、ADC を無効化できます (ADEN = 0)。ADC 電圧レギュレータを無効にすることで節電できます。これは、ビット ADVREGEN に 0 を書き込んで実行します。

そして、リーク電流を削減してさらに節電するには、ADC_CR レジスタでビット DEEPPWD = 1 をセットして ADC ディープパワーダウンモードに再移行することもできます。これは特に STOP モードに移行する前に有益です。

注： DEEPPWD = 1 を書き込むと、ADC 電圧レギュレータは自動的に無効になり、ビット ADVREGEN が自動的にクリアされます。

内部電圧レギュレータが無効なとき (ADVREGEN = 0) には、内部アナログ較正が保持されます。

ADC ディープパワーダウンモード (DEEPPWD = 1) では、内部アナログ較正が失われ、較正を再起動するか、以前に保存した較正係数を再調整する必要があります ([セクション 26.4.8: 較正 \(ADCAL、ADCALDIF、ADC_CALFACT\)](#) を参照してください)。

26.4.7 シングルエンドおよび差動入力チャンネル

ADC_DIFSEL レジスタのビット DIFSEL[i] にプログラムすることで、チャンネルを、シングルエンド入力または差動入力のいずれかに設定できます。この設定は、ADC が無効 (ADEN = 0) のときに書き込む必要があります。シングルエンドチャンネルに対応する DIFSEL[i] ビットは、常に 0 でプログラムされていることに注意してください。

シングルエンド入力モードで、チャンネル「i」に変換されるアナログ電圧は、外部電圧 V_{INP[i]} (正の入力) と V_{REF-} (負の入力) の差です。

差動入力モードで、チャンネル「i」に変換されるアナログ電圧は、外部電圧 V_{INP[i]} (正の入力) と V_{INN[i]} (負の入力) の差です。

差動モードの出力データは、符号なしデータです。 $V_{\text{INP}[i]}$ が $V_{\text{REF-}}$ のとき $V_{\text{INN}[i]}$ は $V_{\text{REF+}}$ で、出力データは 0x000 (12 ビットの分解能モード) です。 $V_{\text{INP}[i]}$ が $V_{\text{REF+}}$ のとき $V_{\text{INN}[i]}$ は $V_{\text{REF-}}$ で、出力データは 0xFFFF です。

$$\text{Converted value} = \frac{\text{ADC_Full_Scale}}{2} \times \left[1 + \frac{V_{\text{INP}} - V_{\text{INN}}}{V_{\text{REF+}}} \right]$$

ADC が差動モードに設定されている場合、両方の入力を $(V_{\text{REF+}}) / 2$ の電圧でバイアスする必要があります。

入力信号は差分とみなされます (共通モードの電圧は固定する必要があります)。

内部チャネル (V_{REFINT} および V_{SENSE} など) はシングルエンドモードのみで使用されます。

各 ADC への入力チャネルの接続方法の詳細については、[セクション 26.4.4: ADC 接続性](#)を参照してください。

注意 : 差動入力モードでチャネル「i」を設定する場合、負の入力電圧 $V_{\text{INN}[i]}$ は別のチャネルに接続されます。その結果、このチャネルはシングルエンドモードや差動モードでは使用できなくなり、この先変換するために設定してはいけません。ADC1/ADC2 では一部のチャネルを共有しています。これにより、ほかの ADC のチャネルが使用不可になる場合があります。唯一の例外は、ADC マスタとスレーブのインタリーブモードです。

26.4.8 較正 (ADCAL、ADCALDIF、ADC_CALFACT)

各 ADC は、ADC のパワーオン/オフシーケンスを含むすべての較正シーケンスを駆動する、自動較正手順を提供しています。この処理時、ADC は、次の ADC のパワーオフまで ADC に内部で適用される、7 ビット幅の較正係数を計算します。較正処理時、アプリケーションでは ADC を使用してはならず、較正が完了するまで待つ必要があります。

較正は、すべての ADC 動作に先立って行われます。較正は、プロセスまたはバンドギャップのばらつきによりチップごとに異なるオフセットエラーを削除します。

シングルエンド入力変換に適用される較正係数は、差動入力変換に適用される係数とは異なります。

- シングルエンド入力変換に適用される較正を起動する前に、ADCALDIF = 0 を書き込みます。
- 差動入力変換に適用される較正を起動する前に、ADCALDIF = 1 を書き込みます。

較正は、ビット ADCAL = 1 をセットすることによって、ソフトウェアによって開始されます。較正は、ADC が無効のとき (ADEN = 0 のとき) だけ開始できます。すべての較正シーケンス時、ADCAL ビットは 1 のままです。較正が完了すると、ハードウェアによってクリアされます。このとき、関連する較正係数は、アナログ ADC と、ADC_CALFACT レジスタのビット CALFACT_S[6:0] または CALFACT_D[6:0] に内部的に格納されます (シングルエンドであるか差動入力較正であるかによって異なる)。

ADC が無効の場合 (ADEN = 0)、内部アナログ較正が保持されます。ただし、ADC が長時間無効である場合は、ADC を再度有効にする前に、新しい較正サイクルを実行することを推奨します。

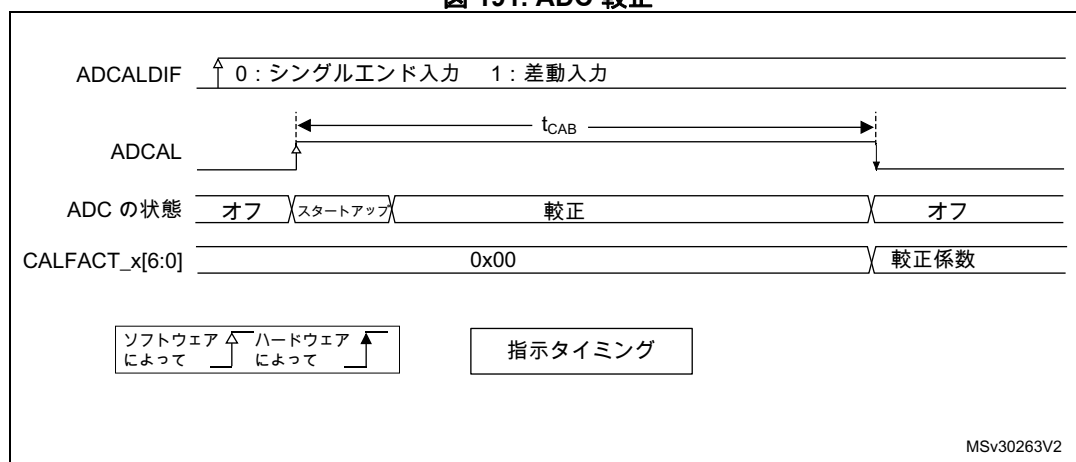
内部アナログ較正は ADC の電源がオフになるたびに失われます (たとえば、製品が STANDBY または VBAT モードになった場合)。この場合、ADC の再較正に時間をかけるのを避けるためには、ソフトウェアが前回の較正時に較正係数を保存していた場合にかぎり、再較正せずに ADC_CALFACT レジスタに較正係数を再度書き込むことができます。

ADC が有効であり、変換中でない場合 (ADEN = 1 かつ ADSTART = 0 かつ JADSTART = 0)、較正係数を書き込むことができます。その場合、次の変換開始時に、較正係数がアナログ ADC に自動的にインジェクトされます。このローディングは透過的であり、変換開始のサイクル遅延は増加しません。 $V_{\text{REF+}}$ 電圧が 10% を超えて変化した場合、再較正することを推奨します。

ソフトウェアでの ADC の較正手順

1. DEEPPWD = 0、ADVREGEN = 1 であり、ADC 電圧レギュレータの起動時間が経過していることを確認します。
2. ADEN = 0 であることを確認します。
3. ADCALDIF = 0 (シングルエンド入力) または ADCALDIF = 1 (差動入力) にセットして、この較正の入力モードを選択します。
4. ADCAL = 1 にセットします。
5. ADCAL = 0 になるまで待ちます。
6. 較正係数を ADC_CALFACT レジスタから読み出すことができます。

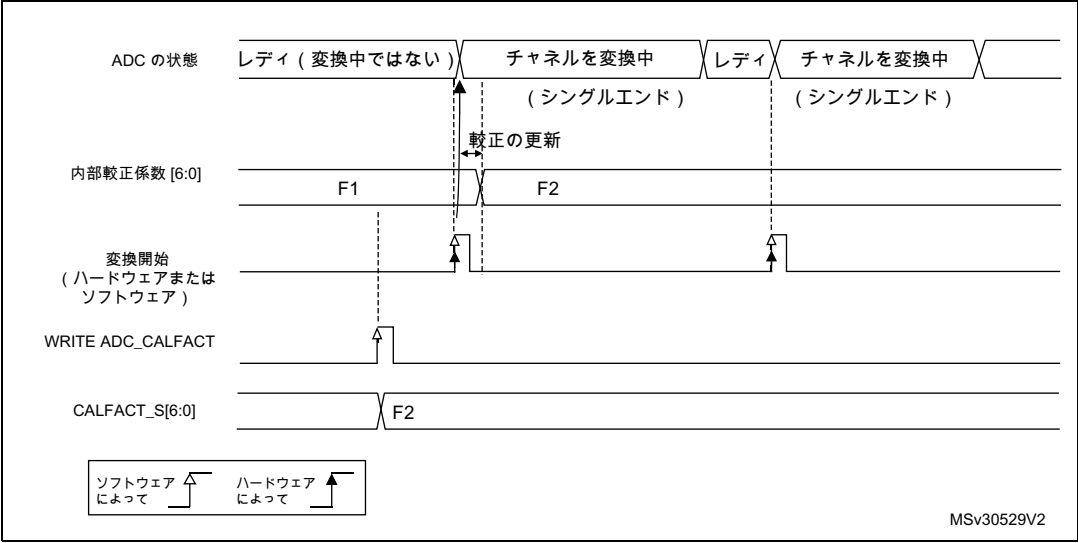
図 191. ADC 較正



ソフトウェアでの ADC への較正係数の再インジェクト手順

1. ADEN = 1 かつ ADSTART = 0 かつ JADSTART = 0 である (ADC が有効で変換中でない) ことを確認します。
2. 新しい較正係数で CALFACT_S および CALFACT_D を書き込みます。
3. 変換を起動すると、較正係数がアナログ ADC にインジェクトされます。これは、内部アナログ較正係数が、シングルエンド入力チャネルの場合はビット CALFACT_S、差動入力チャネルの場合はビット CALFACT_D に格納された較正係数と異なる場合にのみ有効です。

図 192. ADC 較正係数の更新

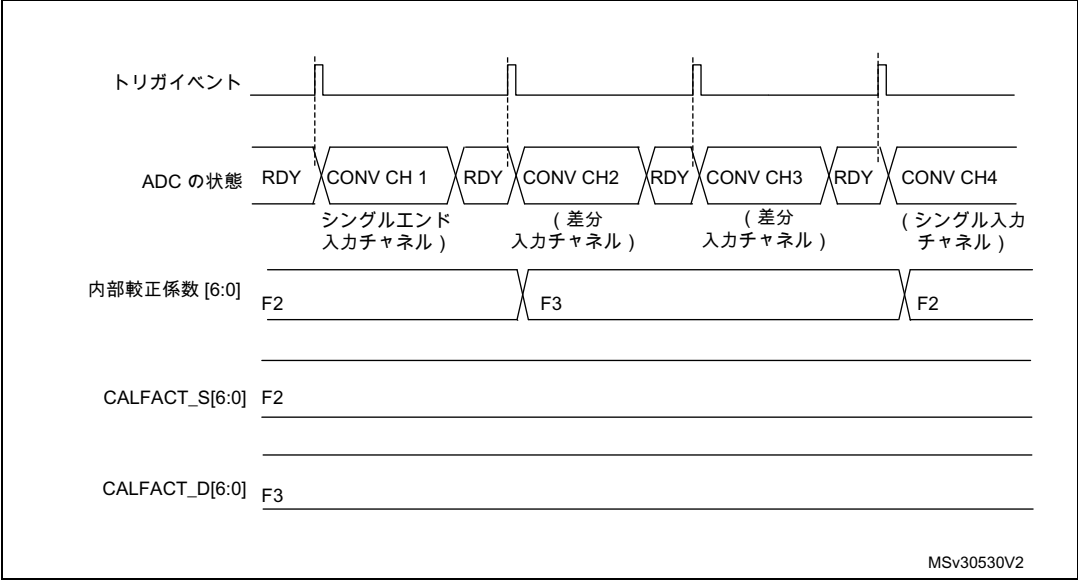


シングル ADC によるシングルエンドおよび差動アナログ入力の変換

ADC で、差動入力とシングルエンド入力の両方を変換することを想定している場合、ADCALDIF = 0 と ADCALDIF = 1 の 2 つの較正を実行する必要があります。手順は、次のとおりです。

1. ADC を無効にします。
2. シングルエンド入力モードで ADC を較正します (ADCALDIF = 0)。これにより、レジスタ CALFACT_S[6:0] が更新されます。
3. 差動入力モードで ADC を較正します (ADCALDIF = 1)。これにより、レジスタ CALFACT_D[6:0] が更新されます。
4. ADC を有効化し、チャンネルを設定して変換を起動します。シングルエンド入力チャンネルから差動入力チャンネル (またはその逆) に切り替えられるたびに、アナログ ADC では自動的に較正が行われます。

図 193. シングルエンドチャンネルと差動チャンネルの混合



26.4.9 ADC オン / オフ制御 (ADEN、ADDIS、ADRDY)

最初に、[セクション 26.4.6 : ADC ディープパワーダウンモード \(DEEPPWD\) および ADC 電圧レギュレータ \(ADVREGEN\)](#) で説明している手順に従ってください。

一度 DEEPPWD = 0 および ADVREGEN = 1 になると ADC を有効化することができ、[図 194](#) にあるように、ADC は正確な変換を開始する前に、安定時間 t_{STAB} を必要とします。ADC は、次の 2 つの制御ビットによって有効化または無効化されます。

- ADC は、ADEN = 1 によって有効化されます。ADC の動作準備ができると、ADRDY フラグがセットされます。
- ADC は、ADDIS = 1 によって無効化されます。アナログ ADC が実質的に無効になると、ADEN および ADDIS はハードウェアによって自動的にクリアされます。

その後、ADSTART = 1 をセットすることによって ([セクション 26.4.18 : 外部トリガおよびトリガ極性での変換 \(EXTSEL、EXTEN、JEXTSEL、JEXTEN\)](#) を参照)、またはトリガが有効な場合に外部トリガイベントが発生したときに、レギュラ変換を開始できます。

JADSTART = 1 をセットすることによって、またはインジェクトトリガが有効な場合に外部インジェクトトリガイベントが発生したときに、インジェクト変換を開始できます。

ソフトウェアでの ADC の有効化手順

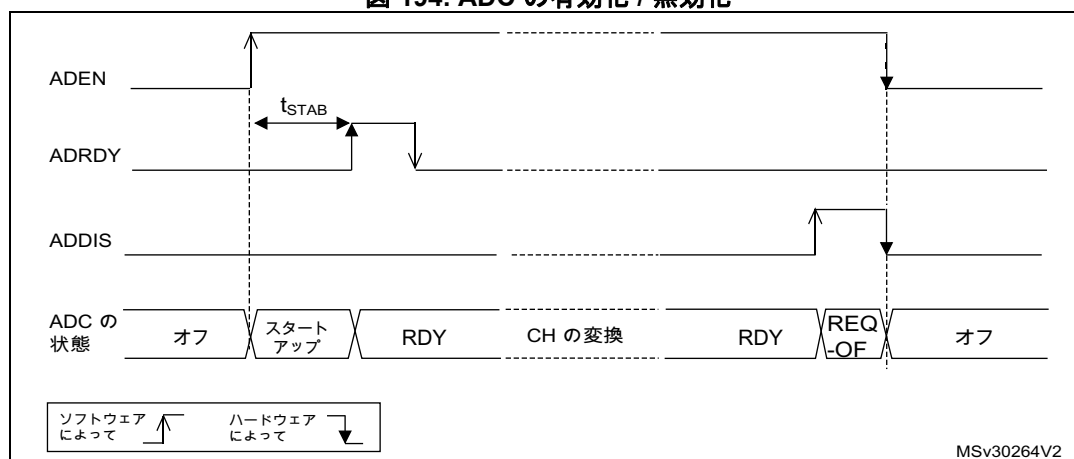
1. ADC_ISR レジスタの ADRDY ビットに「1」を書き込んでクリアします。
2. ADEN = 1 にセットします。
3. ADRDY = 1 になるまで待ちます (ADRDY は ADC 起動時間後にセットされます)。これは、対応する割込みを使用することで実行できます (ADRDYIE = 1 をセットします)。
4. ADC_ISR レジスタの ADRDY ビットに「1」を書き込んでクリアします (オプション)。

注意 : ADCAL がセットされているとき、および、ADCAL ビットがハードウェアによってクリアされた (較正の終了) 後の 4 ADC クロックサイクル中は、ADEN ビットをセットできません。

ソフトウェアでの ADC の無効化手順

1. ADSTART = 0 かつ JADSTART = 0 であることを確認し、変換が実行中でないことを確認します。必要な場合は、ADSTP = 1 および JADSTP = 1 にセットして、実行中のレギュラ変換とインジェクト変換を停止し、ADSTP = 0 および JADSTP = 0 になるまで待ちます。
2. ADDIS = 1 にセットします。
3. アプリケーションによって必要とされる場合、ADEN = 0 になり、アナログ ADC が実質的に無効になるまで待ちます (ADEN = 0 になると、ADDIS は自動的にリセットされます)。

図 194. ADC の有効化 / 無効化



26.4.10 ADC 制御ビット書込み時の制約

ソフトウェアでは、ADC が無効である場合のみ（ADEN は 0 である必要があります）、ADC クロック（RCC セクションを参照）、ADC_DIFSEL レジスタの制御ビット DIFSEL[i]、および ADC_CR レジスタの制御ビット ADCAL と ADEN を設定して有効化するために、RCC 制御ビットを書き込むことができます。

ADC が有効であり、ADC を無効にするための保留中のリクエストがない場合のみ（ADEN=1 かつ ADDIS=0 である必要があります）、ソフトウェアは ADC_CR レジスタの制御ビット ADSTART、JADSTART、および ADDIS を書き込むことができます。

ADC_CFGR、ADC_SMPRx、ADC_TRy、ADC_SQRy、ADC_JDRy、ADC_OFRy、ADC_OFCHRy、および ADC_IER レジスタのその他すべての制御ビットについては、次のとおりです。

- レギュラ変換の設定に関連する制御ビットについては、ADC が有効（ADEN = 1）であり、実行中のレギュラ変換がない（ADSTART = 0）場合のみ、ソフトウェアでこれらを書き込むことができます。
- インジェクト変換の設定に関連する制御ビットについては、ADC が有効（ADEN = 1）であり、実行中のインジェクト変換がない（JADSTART = 0）場合のみ、ソフトウェアでこれらを書き込むことができます。
- アナログデジタル変換の実行中に ADC_TRy レジスタの内容を変更できます（詳細については、[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx\)](#) を参照)。

ADC が有効で、変換可能な場合、および ADC を無効にするための保留中のリクエストがない場合のみ（ADSTART = 1 または JADSTART = 1、かつ ADDIS = 0）、ソフトウェアは ADC_CR レジスタの制御ビット ADSTP または JADSTP を書き込むことができます。

ソフトウェアは、ADC が有効である場合（ADEN = 1）、いつでもレジスタ ADC_JSQR を書き込むことができます。詳細については、[セクション 26.6.16 : ADC インジェクトシーケンスレジスタ \(ADC_JSQR\)](#) を参照してください。

注： これらの禁止された書込みアクセスを防ぐハードウェア保護はありません。ADCの挙動は不明な状態になる場合があります。この状況を回復するには、ADC を無効にする必要があります（ADEN = 0 と ADC_CR レジスタのすべてのビットをクリアします）。

26.4.11 チャネルの選択 (SQRx、JSQRx)

ADC には、ADC ごとに最大 20 の多重化チャネルがあります。そのうち：

- 製品に応じて GPIO パッド (ADC_INP/INN[i]) から最大 18 個のアナログ入力がありますが、すべてを GPIO パッドで使用できるわけではありません。
- ADC は、4 個の内部アナログ入力に接続されています。
 - 内部温度センサ (V_{SENSE})
 - 内部基準電圧 (V_{REFINT})
 - V_{BAT} 監視チャネル (V_{BAT}/4)
 - 内部デジタルコア電圧 (V_{DDCORE})

内部アナログチャネルのうちの 1 つを変換するには、ADC_CCR レジスタのビット VREFEN、VBATEN、または TSEN をプログラムして、対応するアナログソースを最初に有効化する必要があります。

上記の内部アナログ入力の、外部 ADC ピンまたは内部信号への接続については、[セクション 26.4.2: ADC ピンおよび内部信号の ADC 相互接続の表](#)を参照してください。

変換は、レギュラとインジェクトの 2 つのグループに構成することができます。各グループは、任意のチャネルに対して任意の順序で行うことができる一連の変換で構成されます。たとえば、次のような順で変換順を設定することができます: ADC1/2_INP/INN3、ADC1/2_INP/INN8、ADC1/2_INP/INN2、ADC1/2_INN/INP2、ADC1/2_INP/INN0、ADC1/2_INP/INN2、ADC1/2_INP/INN2、ADC1/2_INP/INN15。

- **レギュラグループ**は、最大 16 の変換で構成されます。レギュラチャネルと変換シーケンス内での順序は、ADC_SQRy レジスタで選択する必要があります。レギュラグループの合計変換数は、ADC_SQR1 レジスタの L[3:0] ビットに書き込む必要があります。
- **インジェクトグループ**は、最大 4 つの変換で構成されます。インジェクトチャネルと変換シーケンス内での順序は、ADC_JSQR レジスタで選択する必要があります。インジェクトグループの合計変換数は、ADC_JSQR レジスタの L[1:0] ビットに書き込む必要があります。

レギュラ変換が実行される可能性がある場合は、ADC_SQRy レジスタを変更しないでください。この場合、最初に ADSTP = 1 を書き込んで、ADC レギュラ変換を停止する必要があります ([セクション 26.4.17: 実行中の変換の停止 \(ADSTP、JADSTP\)](#) を参照)。

JADSTART が 1 にセットされているとき (インジェクト変換実行中) にソフトウェアが ADC_JSQR レジスタを動作中に変更できるのは、コンテキストキューが有効になっている (ADC_CFGR レジスタの JQDIS = 0) 場合だけです。 [セクション 26.4.21: インジェクト変換のコンテキストのキュー](#)を参照してください。

26.4.12 チャネル単位でプログラム可能なサンプリング時間 (SMPR1、SMPR2)

変換を開始する前に、ADC は測定する電圧ソースと ADC の内蔵サンプリングコンデンサの間の直接接続を確立する必要があります。このサンプリング時間は、入力電圧ソースが内蔵コンデンサを入力電圧レベルまでチャージできるだけの十分な長さが必要です。

各チャネルは、ADC_SMPR1 および ADC レジスタの SMP[2:0] ビットを使用してプログラムすることで異なるサンプリング時間でサンプリングできます。したがって、次のサンプリング時間の値から選択することができます。

- SMP = 000 : 2.5 ADC クロックサイクル
- SMP = 001 : 6.5 ADC クロックサイクル
- SMP = 010 : 12.5 ADC クロックサイクル
- SMP = 011 : 24.5 ADC クロックサイクル
- SMP = 100 : 47.5 ADC クロックサイクル
- SMP = 101 : 92.5 ADC クロックサイクル
- SMP = 110 : 247.5 ADC クロックサイクル
- SMP = 111 : 640.5 ADC クロックサイクル

合計変換時間は、次のように計算されます。

$$T_{\text{CONV}} = \text{サンプリング時間} + 12.5 \text{ ADC クロックサイクル}$$

例 :

$$F_{\text{adc_ker_ck}} = 30 \text{ MHz、サンプリング時間} = 2.5 \text{ ADC クロックサイクル} :$$

$$T_{\text{CONV}} = (2.5 + 12.5) \text{ ADC クロックサイクル} = 15 \text{ ADC クロックサイクル} = 500 \text{ ns}$$

ADC は、ステータスビット EOSMP をセットすることによって、サンプリングフェーズの終了を示します (レギュラ変換のみ)。

注 : ADC の変換モードに応じて、実際のサンプリング時間は、先にプログラムした SMP 値とは異なる場合がありますが、同等の合計変換時間 (T_{CONV}) は変わりません。

- スキャンまたは連続モードの最初の変換、ならびに不連続モードでのすべての変換では、実際のサンプリング時間が先に設定した値と比べて 0.5 クロックサイクル分短くなります。
- スキャンまたは連続モードの 2 番目とそれに続く変換では、設定されたサンプリング時間に 0.5 サイクル追加されます。この追加の 0.5 クロックサイクルが、前の変換サイクルと重なります。

サンプリング時間の制約

各チャネルで、SMP[2:0] ビットはデータシートの ADC 特性セクションで指定された最小サンプリング時間を考慮してプログラムする必要があります。

バルブサンプリングモード

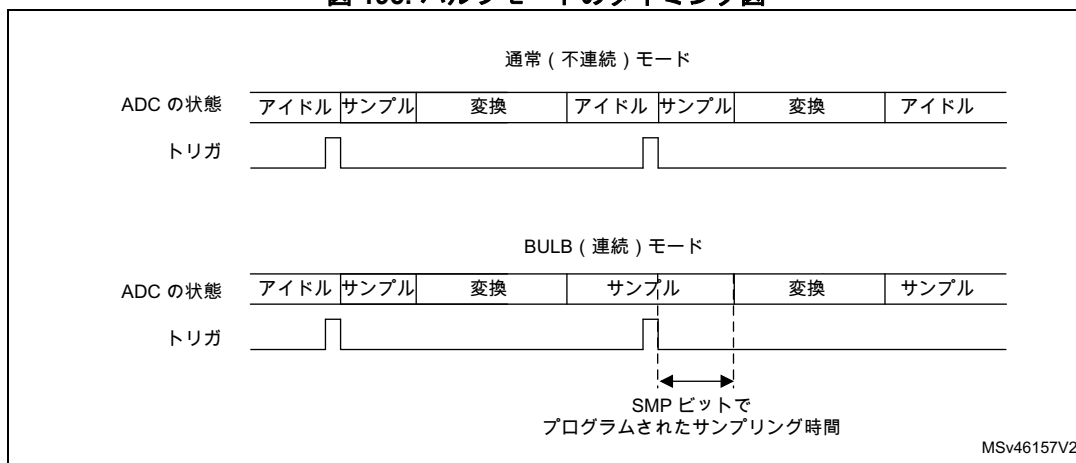
ADC レジスタで BULB ビットがセットされているとき、サンプリング期間は最後の ADC 変換の直後に開始します。ハードウェアまたはソフトウェアトリガは、サンプリング時間が ADC_SMPR1 レジスタでプログラムされた後、変換を開始します。最初の ADC 変換は、ADC が有効化された後、SMP ビットでプログラムされたサンプリング時間で実行されます。バルブモードは、2 番目の変換から開始するのに有効です。

最大サンプリング時間は制限されます (データシートの ADC 特性のセクションを参照)。

バルブモードは、連続変換モードとも、インジェクトチャネル変換とも互換性がありません。

BULB ビットがセットされているとき、ADC_CFGR2 の SMPTRIG ビットをセットすることはできません。

図 195. バルブモードのタイミング図



サンプリング時間制御トリガモード

SMPTRIG ビットがセットされているとき、SMPx ビットによってプログラムされたサンプリング時間は適用されません。サンプリング時間は、トリガ信号エッジによって制御されます。

ハードウェアトリガが選択されたとき、トリガ信号の各立ち上がりエッジによってサンプリング期間は開始します。立ち下がりエッジによってサンプリング期間は終了し、変換が開始されます。

ソフトウェアトリガが選択されたとき、ソフトウェアトリガは ADC_CR の ADSTART ビットではなく、SWTRIG ビットです。サンプリング期間を開始するには、SWTRIG ビットをセットする必要があります。サンプリング期間を終了して変換を開始するには、SWTRIG ビットをクリアする必要があります。

最大サンプリング時間は制限されます（データシートの ADC 特性のセクションを参照）。

このモードは、連続変換モードとも、インジェクトチャネル変換とも互換性がありません。

SMPTRIG ビットがセットされているとき、BULB ビットはセットできません。

I/O アナログスイッチ電圧ブースタ

V_{DDA} 電圧が低すぎる場合に、I/O アナログスイッチの抵抗が増加します。サンプリング時間は、状況に応じて調整する必要があります（対応する電気的特性については、デバイスデータシートを参照）。この抵抗は、BOOSTE ビットで内部電圧ブースタを有効にするか、PWR_PMCR レジスタの ADV_READY ビットによって V_{DD} ブースタ電圧 ($V_{DD} > 2.7\text{ V}$ の場合) を選択することで、低い V_{DDA} で最小化できます。

SMPPLUS 制御ビット

SMPPLUS ビットは、サンプリング時間を 2.5 ADC クロックサイクルから 3.5 ADC クロックサイクルに変更するのに使用できます。

26.4.13 シングル変換モード (CONT = 0)

シングル変換モードでは、ADC はすべてのチャネルを一度変換します。CONT ビットが 0 の時、次のいずれかによってこのモードが開始します。

- ADC_CR レジスタの ADSTART ビットをセット（レギュラチャネル）
- ADC_CR レジスタの JADSTART ビットをセット（インジェクトチャネル）
- 外部ハードウェアトリガイベント（レギュラまたはインジェクトチャネル）

レギュラシーケンス内で、各変換の完了後、

- 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
- EOC (レギュラ変換の完了) フラグがセットされます。
- EOCIE ビットがセットされている場合、割込みが生成されます。

インジェクトシーケンス内で、各変換の完了後、

- 変換データは 4 つの 16 ビット ADC_JDRy レジスタのうちの 1 つに格納されます。
- JEOP (インジェクト変換の完了) フラグがセットされます。
- JEOPIE ビットがセットされている場合、割込みが生成されます。

レギュラシーケンスの完了後、

- EOS (レギュラシーケンスの完了) フラグがセットされます。
- EOSIE ビットがセットされている場合、割込みが生成されます。

インジェクトシーケンスの完了後、

- JEOP (インジェクトシーケンスの完了) フラグがセットされます。
- JEOPIE ビットがセットされている場合、割込みが生成されます。

次に、ADC は、新しい外部レギュラトリガまたは外部インジェクトトリガが発生するか、ADSTART または JADSTART ビットが再びセットされるまで停止します。

注： 単一チャンネルを変換するには、長さが 1 のシーケンスをプログラムします。

26.4.14 連続変換モード (CONT = 1)

このモードはレギュラチャンネルにのみ適用されます。

連続変換モードでは、ソフトウェアまたはハードウェアのレギュラトリガイイベントが発生すると、ADC はチャンネルのレギュラ変換を一度すべて実行し、自動的に再起動して、シーケンスの各変換を連続的に実行します。CONT ビットが 1 の時、外部トリガまたは ADC_CR レジスタの ADSTART ビットをセットすることによって、このモードが開始されます。

レギュラシーケンス内で、各変換の完了後、

- 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
- EOC (end of conversion) フラグがセットされます。
- EOCIE ビットがセットされている場合、割込みが生成されます。

変換シーケンスの完了後、

- EOS (シーケンス完了) フラグがセットされます。
- EOSIE ビットがセットされている場合、割込みが生成されます。

次に、新しいシーケンスをすぐに再開して、ADC は変換シーケンスを連続的に繰り返します。

注： 単一チャンネルを変換するには、長さが 1 のシーケンスをプログラムします。

不連続モードと連続モードの両方を有効にすることはできません。DISCEN = 1 と CONT = 1 の両方をセットすることは禁じられています。

インジェクトチャンネルを連続して変換することはできません。唯一の例外は、連続モードに設定されたレギュラチャンネルの後にインジェクトチャンネルを自動的に変換するよう設定されている (JAUTO ビットを使用) 場合です。自動インジェクションモードのセクションを参照してください。

26.4.15 変換の開始 (ADSTART、JADSTART)

ソフトウェアは、ADSTART = 1 をセットすることによって ADC レギュラ変換を開始します。

ADSTART がセットされると、変換は、

- EXTEN = 0x0 (ソフトウェアトリガ) の場合、すぐに開始します。
- EXTEN が 0x0 ではない場合、選択されたレギュラハードウェアトリガの次のアクティブエッジで開始します。

ソフトウェアは、JADSTART = 1 をセットすることによって ADC インジェクト変換を開始します。

JADSTART がセットされると、変換は、

- JEXTEN = 0x0 (ソフトウェアトリガ) の場合、すぐに開始します。
- JEXTEN が 0x0 ではない場合、選択されたインジェクトハードウェアトリガの次のアクティブエッジで開始します。

注： 自動インジェクションモード (JAUTO = 1) では、ADSTART ビットを使用してレギュラ変換を開始し、続いて自動インジェクト変換を行います (JADSTART はクリアされたままでなければなりません)。

ADSTART および JADSTART では、ADC の動作が実行中かどうかを示す情報も提供しています。ADSTART = 0 および JADSTART = 0 が両方とも真であり、ADC がアイドルであることを示しているときには、ADC を再設定できます。

次のときに、ADSTART はハードウェアによってクリアされます。

- シングルモード時のソフトウェアのレギュラトリガ (CONT = 0、EXTSEL = 0x0)
 - レギュラ変換シーケンスの終了時 (EOS のアサート)、または DISCEN = 1 の場合のサブグループ処理の終了時
- すべての場合に (CONT = x、EXTSEL = x)
 - ソフトウェアによってアサートされた ADSTP 手順の実行後。

注： 連続モード (CONT = 1) では、シーケンスは自動的に再起動されるので、EOS がアサートされても、ADSTART はハードウェアによってクリアされません。

シングルモードでハードウェアトリガが選択されたとき (CONT = 0 かつ EXTSEL ≠ 0x00)、次のハードウェアトリガイベントで ADSTART の再リセットを必要としないソフトウェアを支援するために、EOS がアサートされても、ADSTART はハードウェアによってクリアされません。これにより、さらなるハードウェアトリガの見落としを防ぐことができます。

次のときに、JADSTART はハードウェアによってクリアされます。

- シングルモード時のソフトウェアのインジェクトトリガ (JEXTSEL = 0x0)
 - インジェクト変換シーケンスの終了時 (JEOS のアサート)、または JDISCEN = 1 の場合のサブグループ処理の終了時
- すべての場合に (JEXTSEL = x)
 - ソフトウェアによってアサートされた JADSTP 手順の実行後。

注： ソフトウェアトリガが選択されているとき、EOC フラグがまだハイの場合に ADSTART ビットをセットしてはいけません。

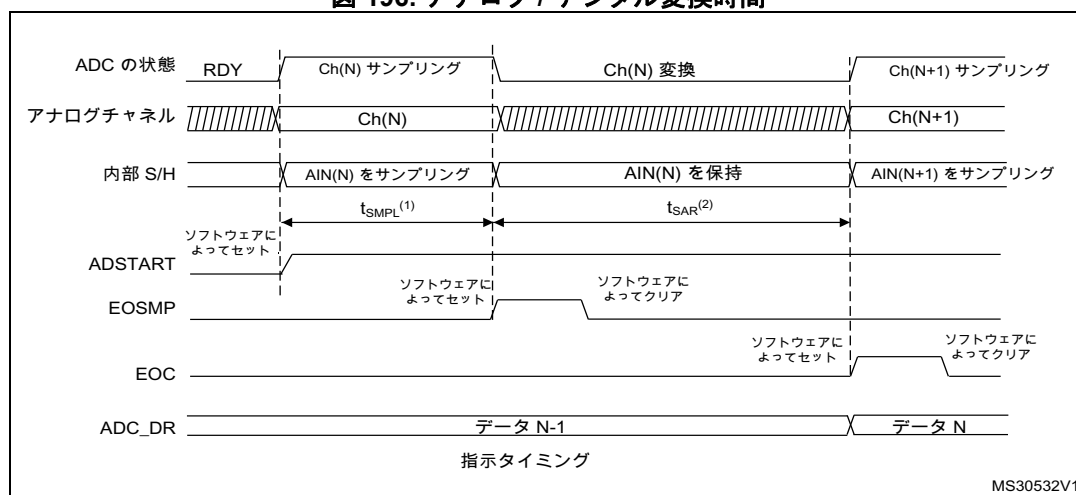
26.4.16 ADC のタイミング

変換の開始から変換の終了までの経過時間は、設定されたサンプリング時間に逐次比較時間（データ分解能に依存）を加えた合計です。

$$T_{\text{CONV}} = T_{\text{SMPL}} + T_{\text{SAR}} = [2.5 \text{ }_{\text{min}} + 12.5 \text{ }_{\text{12bit}}] \times T_{\text{ADC_CLK}}$$

$$T_{\text{CONV}} = T_{\text{SMPL}} + T_{\text{SAR}} = 83.33 \text{ ns }_{\text{min}} + 416.67 \text{ ns }_{\text{12bit}} = 500.0 \text{ ns (} F_{\text{ADC_CLK}} = 30 \text{ MHz の場合)}$$

図 196. アナログ / デジタル変換時間



1. T_{SMPL} は SMP[2:0] に依存します。

2. T_{SAR} は RES[2:0] に依存します。

26.4.17 実行中の変換の停止 (ADSTP、JADSTP)

ソフトウェアは、ADSTP = 1 をセットすることによって実行中のレギュラ変換を、また JADSTP = 1 をセットすることによって実行中のインジェクト変換を停止することができます。

変換を停止すると、実行中の ADC 動作がリセットされます。これによって ADC 動作が再設定され（チャンネル選択やトリガの変更など）、新しい動作の準備ができます。

レギュラ変換の実行中に、インジェクト変換を停止することもできる点に注意してください（その逆も可能）。これにより、たとえば、レギュラ変換の実行中にインジェクト変換シーケンスやトリガを再設定することができます（その逆も可能）。

ADSTP ビットがソフトウェアによってセットされると、実行中のレギュラ変換はアボートされ、部分的な結果は破棄されます（ADC_DR レジスタは現在の変換では更新されません）。

JADSTP ビットがソフトウェアによってセットされると、実行中のインジェクト変換はアボートされ、部分的な結果は破棄されます（ADC_JDRy レジスタは現在の変換では更新されません）。シーケンスも中止され、リセットされます（ADC を再起動すると、新しいシーケンスが再開されることを意味します）。

この手順が完了すると、ADSTP/ADSTART ビット（レギュラ変換の場合）、または JADSTP/JADSTART ビット（インジェクト変換の場合）がハードウェアによってクリアされ、ソフトウェアは ADC が完全に停止する前にビットがリセットされるまで ADSTART（または JADSTART）をポーリングする必要があります。

注： 自動インジェクションモード (JAUTO = 1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアボートされます (JADSTP を使用してはいけません)。

図 197. 実行中のレギュラ変換の停止

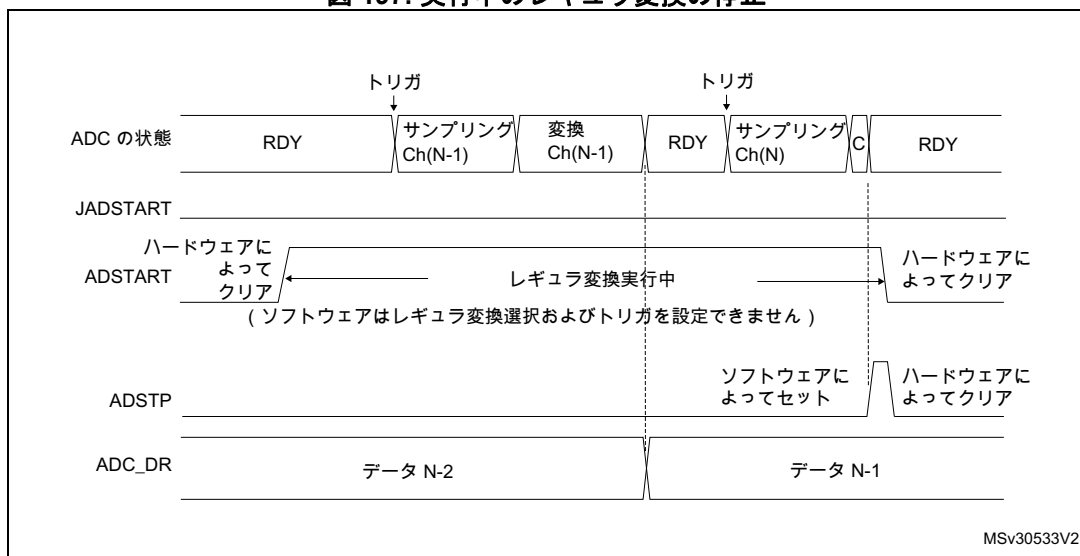
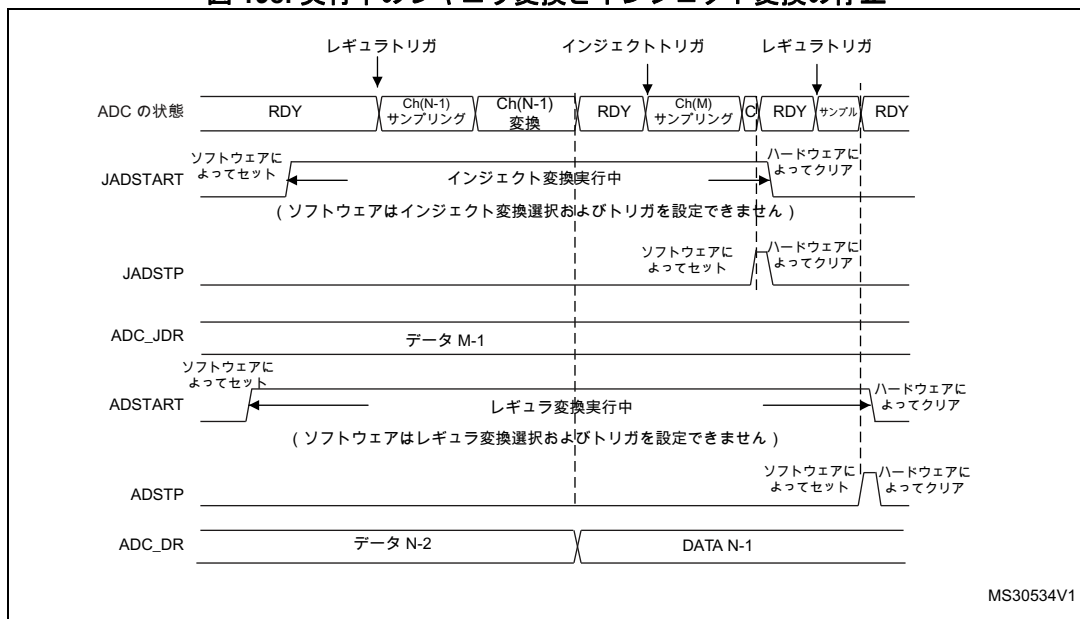


図 198. 実行中のレギュラ変換とインジェクト変換の停止



26.4.18 外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN、JEXTSEL、JEXTEN)

変換または変換シーケンスは、ソフトウェアによって、または外部イベント（タイマキャプチャや入力ピンなど）によってトリガできます。EXTEN[1:0] 制御ビット（レギュラ変換の場合）または JEXTEN[1:0] ビット（インジェクト変換の場合）が 0b00 以外の場合、外部イベントは選択した極性で変換をトリガすることができます。

インジェクトキューが有効である場合（ビット JQDIS = 0）、インジェクトされたソフトウェアのトリガはできません。

レギュラトリガ選択は、一度ソフトウェアでビット ADSTART = 1 をセットすると有効になります。インジェクトトリガ選択は、一度ソフトウェアでビット JADSTART = 1 をセットすると有効になります。

変換中に発生したハードウェアトリガは無視されます。

- ビット ADSTART = 0 の場合、発生したレギュラハードウェアトリガは無視されます。
- ビット JADSTART = 0 の場合、発生したインジェクトハードウェアトリガは無視されます。

表 243 に EXTEN[1:0] と JEXTEN[1:0] の値とトリガ極性の対応を示します。

表 243. レギュラ外部トリガのトリガ極性の設定

EXTEN[1:0]	転送元
00	ハードウェアトリガ検出が無効になり、ソフトウェアトリガ検出が有効になります。
01	立ち上がりエッジで検出するハードウェアトリガ
10	立ち下がりエッジで検出するハードウェアトリガ
11	立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

注：レギュラトリガの極性は動作中に変更することはできません。

表 244. インジェクト外部トリガのトリガ極性の設定

JEXTEN[1:0]	転送元
00	<ul style="list-style-type: none"> – JQDIS = 1（キュー無効）の場合：ハードウェアトリガ検出が無効になり、ソフトウェアトリガ検出が有効になります。 – JQDIS = 0（キュー有効）の場合、ハードウェアおよびソフトウェアトリガ検出は無効になります。
01	立ち上がりエッジで検出するハードウェアトリガ
10	立ち下がりエッジで検出するハードウェアトリガ
11	立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

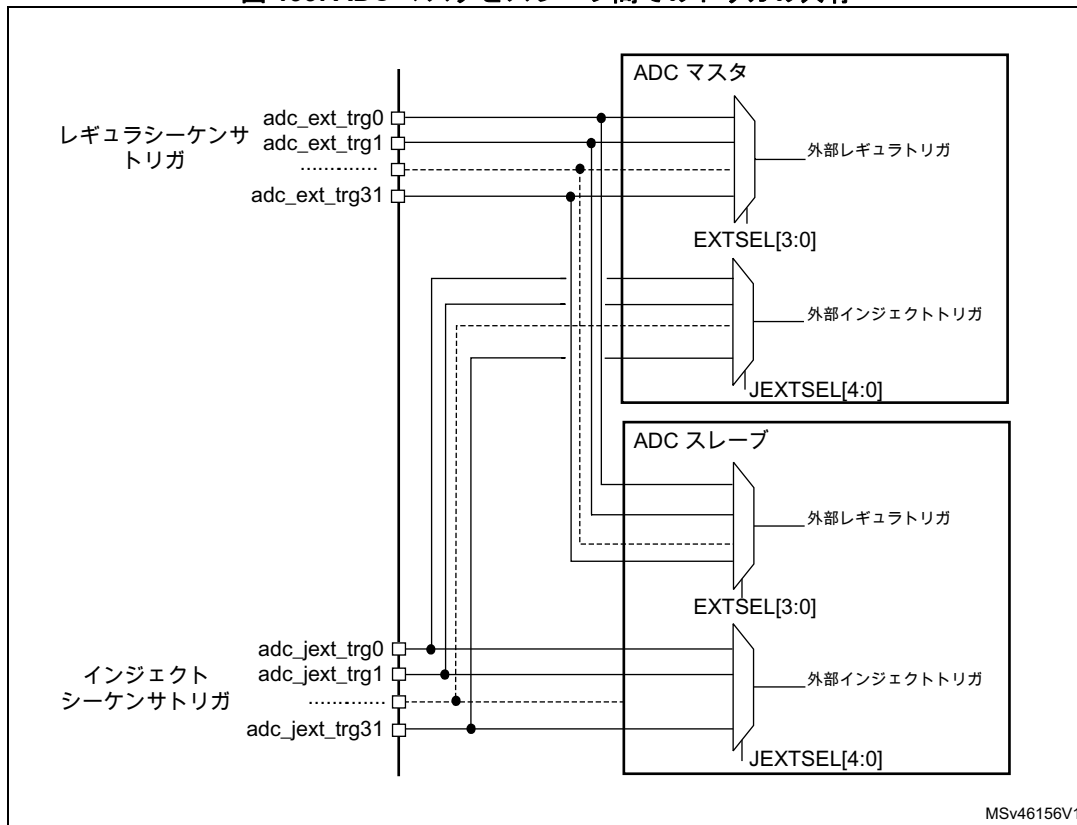
注：キューが有効（JQDIS = 0）になっている場合、インジェクトトリガの極性は動作中に予想して変更できます。[セクション 26.4.21：インジェクト変換のコンテキストのキュー](#)を参照してください。

EXTSEL および JEXTSEL 制御ビットでは、レギュラおよびインジェクトグループの変換をトリガできるイベントを 32 のイベントから選択します。

レギュラグループ変換は、インジェクトトリガによって中断することが可能です。

注： レギュラトリガの選択は動作中に変更することはできません。インジェクトトリガの選択は動作中に予想して変更できます。[セクション 26.4.21 : 1032 ページのインジェクト変換のコンテキストのキュー](#)を参照してください。

図 199. ADC マスタとスレーブ間でのトリガの共有



レギュラ変換に使用可能なすべての外部トリガのリストについては、[セクション 26.4.2 : ADC ピンおよび内部信号](#)の ADC 相互接続の表を参照してください。

26.4.19 インジェクトチャネルの管理

トリガインジェクションモード

トリガインジェクションを使用するには、ADC_CFGR レジスタの JAUTO ビットがクリアされていなければなりません。

1. 外部トリガによって、または、ADC_CR レジスタの ADSTART ビットをセットすることによって、レギュラチャネルグループの変換を開始します。
2. レギュラチャネルグループの変換中に外部インジェクトトリガが発生した場合、または ADC_CR レジスタの JADSTART ビットがセットされた場合、現在の変換はリセットされ、インジェクトチャネルシーケンススイッチが起動されます(すべてのインジェクトチャネルが一度変換されます)。
3. その後、レギュラチャネルグループのレギュラ変換が、最後に中断されたレギュラ変換から再開されます。
4. インジェクト変換中にレギュライベントが発生した場合、インジェクト変換は中断されず、インジェクトシーケンスの終了時にレギュラシーケンスが実行されます。[図 200](#) に対応するタイミング図を示します。

注： トリガインジェクションを使用する場合は、トリガイベントの間隔がインジェクションシーケンスより長くなるようにしなければなりません。たとえば、シーケンスの長さが 30 ADC クロックサイクルの場合（2.5 クロック周期のサンプリング時間を持つ 2 つの変換）、トリガの最小間隔は 31 ADC クロックサイクルでなければなりません。

自動インジェクションモード

ADC_CFGR レジスタの JAUTO ビットがセットされている場合、レギュラグループチャネルの変換後、インジェクトグループチャネルが自動的に変換されます。これを使用して、ADC_SQRy レジスタおよび ADC_JSQR レジスタにプログラミングされた最大 20 回の変換シーケンスを実行することができます。

このモードでは、レギュラ変換を開始し、続いてインジェクト変換を開始するために ADC_CR レジスタの ADSTART ビットをセットする必要があります（JADSTART はクリアされたままでなければなりません）。ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアボートされます（JADSTP ビットを使用することはできません）。

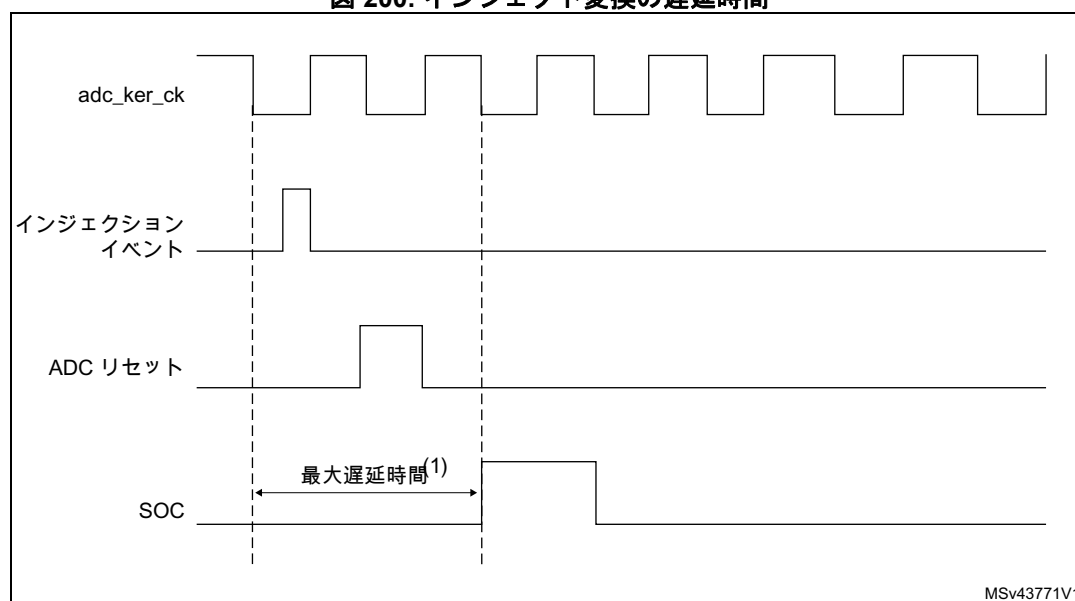
このモードでは、インジェクトチャネルの外部トリガは無効でなければなりません。

JAUTO ビットに加えて CONT ビットもセットされている場合、レギュラチャネルに続いてインジェクトチャネルが連続して変換されます。

注： 自動インジェクトモードと不連続モードを同時に使用することはできません。

JAUTO モードで、レギュラシーケンスのデータをエクスポートする際に DMA を使用する場合は、サーキュラモードでプログラムする必要があります（DMA_CCRx レジスタの CIRC ビットをセット）。CIRC ビットがリセットされると（シングルショットモード）、DMA 転送完了イベントによって JAUTO シーケンスが停止されます。

図 200. インジェクト変換の遅延時間



1. 最大遅延時間の値は、デバイスのデータシートの電気特性に記載されています。

26.4.20 不連続モード (DISCEN、DISCNUM、JDISCEN)

レギュラグループモード

このモードは、ADC_CFGR レジスタの DISCEN ビットをセットすることによって有効になります。

これを使用して、ADC_SQRy レジスタで選択された変換シーケンスの一部である短い n 回 ($n \leq 8$) の変換シーケンス (サブグループ) を変換できます。 n の値は、ADC_CFGR レジスタの DISCNUM[2:0] ビットに書き込むことによって指定されます。

外部トリガが発生すると、ADC_SQRy レジスタで選択された次の n 回の変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_SQR1 レジスタの L[3:0] ビットによって定義されます。

例：

- DISCEN = 1、 $n = 3$ 、変換されるチャネル = 1、2、3、6、7、8、9、10、11
 - 1 番目のトリガ：変換されるチャネルは 1、2、3 です (各変換で EOC イベントが生成されます)。
 - 2 番目のトリガ：変換されるチャネルは 6、7、8 です (各変換で EOC イベントが生成されます)。
 - 3 番目のトリガ：変換されるチャネルは 9、10、11 です (各変換で EOC イベントが生成されます)。EOS イベントはチャネル 11 の変換後に生成されます。
 - 4 番目のトリガ：変換されるチャネルは 1、2、3 です (各変換で EOC イベントが生成されます)。
 -
- DISCEN = 0、変換されるチャネル = 1、2、3、6、7、8、9、10、11
 - 1 番目のトリガ：シーケンス全体、すなわち、チャネル 1、次いで 2、3、6、7、8、9、10、および 11 が変換されます。各変換後に EOC イベントが生成され、最後の変換後には EOS イベントも生成されます。
 - 後続のすべてのトリガイイベントによって、シーケンス全体が再起動されます。

注： 上記の例に示したチャネル番号は、すべてのマイクロコントローラ上で使用できるわけではありません。レギュラグループが不連続モードで変換されるときには、ロールオーバーは発生しません (シーケンスの最後のサブグループの変換は n 回以下です)。

すべてのサブグループが変換されると、次のトリガでは、最初のサブグループの変換が開始されます。上の例では、4 番目のトリガで最初のサブグループのチャネル 1、2、および 3 が再変換されます。

不連続モードと連続モードの両方を有効にすることはできません。この場合 (DISCEN = 1、CONT = 1 の場合)、ADC は連続モードが無効化されたかのように挙動します。

インジェクトグループモード

このモードは、ADC_CFGR レジスタの JDISCEN ビットをセットすることによって有効になります。ADC_JSQR レジスタで選択されたシーケンスを、外部インジェクトトリガイイベント後に 1 チャンルずつ変換します。これは、「n」が 1 に固定されたレギュラチャネルの不連続モードに相当します。

外部トリガが発生すると、ADC_JSQR レジスタで選択された次のチャネル変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_JSQR レジスタの JL[1:0] ビットによって定義されます。

例：

- JDISCEN = 1、変換されるチャネル = 1、2、3
 - 1 番目のトリガ：チャネル 1 が変換されます (JEOC イベントが生成されます)。
 - 2 番目のトリガ：チャネル 2 が変換されます (JEOC イベントが生成されます)。
 - 3 番目のトリガ：チャネル 3 が変換され、JEOC イベントと JEOS イベントが生成されます。
 -

注： 上記の例に示したチャネル番号は、すべてのマイクロコントローラ上で使用できるわけではありません。すべてのインジェクトチャネルが変換されると、次のトリガでは、最初のインジェクトチャネルの変換が開始されます。上の例では、4 番目のトリガで最初のインジェクトチャネル 1 が再変換されます。自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO のセット時には、ビット DISCEN と JDISCEN はクリアされたままではありません。

26.4.21 インジェクト変換のコンテキストのキュー

コンテキストのキューは、次のインジェクト変換シーケンスに最大 2 つのコンテキストを予測して実装されます。この機能を有効にするには、ADC_CFGR レジスタの JQDIS ビットをリセットする必要があります。コンテキストキューが有効になっている場合、ハードウェアトリガによる変換のみ可能です。

このコンテキストの構成は次のとおりです。

- インジェクトトリガの設定 (ADC_JSQR レジスタの JEXTEN[1:0] ビット および JEXTSEL ビット)
- インジェクトシーケンスの定義 (ADC_JSQR レジスタの JSQx[4:0] ビット および JL[1:0] ビット)

コンテキストのすべてのパラメータは 1 つの ADC_JSQR レジスタ 内に定義され、このレジスタには最大 2 セットのパラメータのバッファ化を可能にする 2 つのバッファのキューが実装されます。

- JSQR レジスタは、インジェクト変換の実行中など、いつでも書き込むことができます。
- JSQR レジスタに書き込まれる各データは、コンテキストのキューに格納されます。
- 最初に、キューは空であり、JSQR レジスタへの最初の書込みアクセスによってコンテキストがすぐに変更され、ADC ではインジェクトトリガを受信する準備ができます。
- 一度インジェクトシーケンスが完了すると、キューは消費され、キューに格納された次の JSQR パラメータに従ってコンテキストが変化します。この新しいコンテキストは、次のインジェクト変換シーケンスに適用されます。

- キューのオーバーフローは、キューがフル状態のときにJSQR レジスタに書き込むと発生します。このオーバーフローは、フラグ JQOVF のアサーションによって示されます。オーバーフローが発生すると、オーバーフローを作成した JSQR レジスタの書き込みアクセスは無視され、コンテキストのキューは変更されません。JQOVFIE ビットがセットされている場合は、割込みを生成することができます。
- キューが空になると、ADC_CFGR レジスタの制御ビット JQM の値に応じて、2 つの挙動が考えられます。
 - JQM = 0 の場合、キューは ADC を有効化した直後に空になりますが、実行動作中に空になることはありません。キューは常に最後のアクティブコンテキストを維持し、さらに有効なインジェクトシーケンスの開始を最後のアクティブコンテキストに応じて処理します。
 - JQM = 1 の場合、キューはインジェクトシーケンス終了後、またはキューは破棄され空になります。このとき、キュー内にコンテキストはなく、ハードウェアトリガは無効になります。そのため、ソフトウェアが新しいインジェクトコンテキストを JSQR レジスタに再度書き込むまで、さらなるハードウェアインジェクトトリガは無視されます。
- JSQR レジスタを読み出すと、その時点でアクティブな現在の JSQR コンテキストが返されます。JSQR コンテキストが空である場合、JSQR は 0x0000 として読み出されます。
- キューは、JADSTP = 1 をセットしてインジェクト変換を停止した場合、または ADDIS = 1 をセットして ADC を無効化した場合に破棄されます。
 - JQM = 0 の場合、キューは最後のアクティブコンテキストによって維持されます。
 - JQM = 1 の場合、キューは空になり、トリガは無視されます。

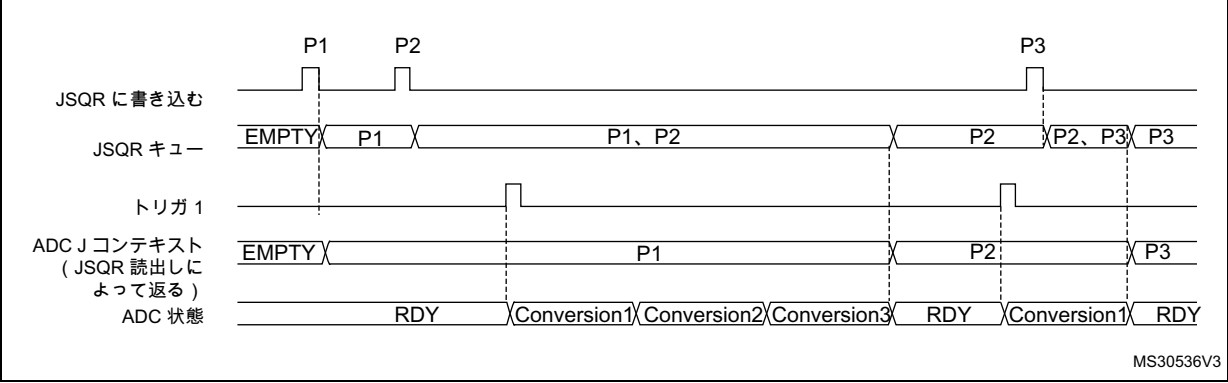
注： 不連続モードで設定された場合（ビット JDISCEN = 1）、コンテキストを変更し、キューを消費するのはインジェクトシーケンスの最後のトリガのみです。1 番目のトリガではキューを消費するだけで、次の不連続モードの例に示すように、その他のトリガは引き続き有効なトリガとなります（両方のコンテキストにおいて長さは 3）。

- 1 番目のトリガ、不連続。シーケンス 1：コンテキスト 1 を消費、1 番目の変換を実行
- 2 番目のトリガ、不連続。シーケンス 1：2 番目の変換
- 3 番目のトリガ、不連続。シーケンス 1：3 番目の変換
- 4 番目のトリガ、不連続。シーケンス 2：コンテキスト 2 を消費、1 番目の変換を実行
- 5 番目のトリガ、不連続。シーケンス 2：2 番目の変換
- 6 番目のトリガ、不連続。シーケンス 2：3 番目の変換

トリガまたはシーケンスのコンテキストを変更する場合の挙動

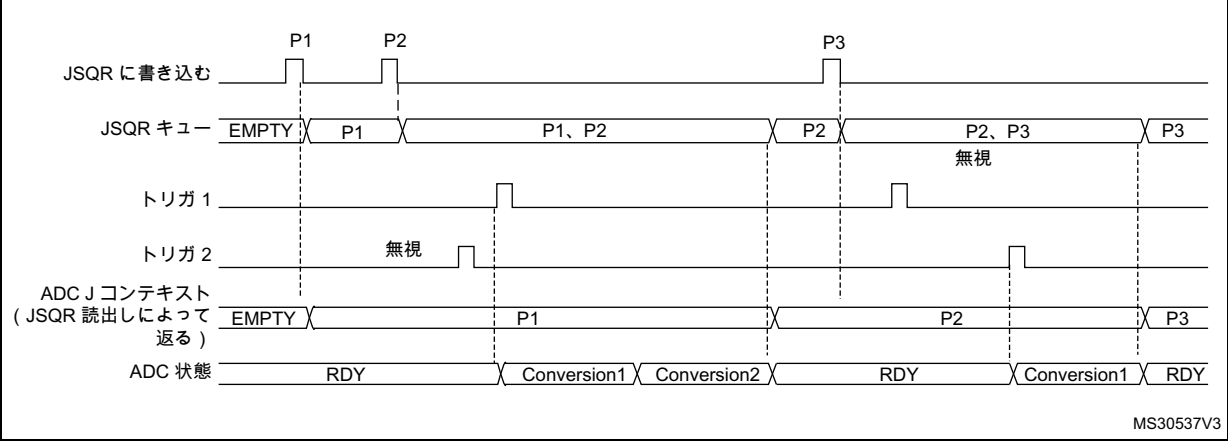
図 201 と 図 202 に、シーケンスまたはトリガを変更する際のコンテキストのキューの挙動を示します。

図 201. コンテキストの JSQR キューの例 (シーケンスの変更)



- 1. パラメータ :
P1 : 3 つの変換のシーケンス、ハードウェアトリガ 1
P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
P3 : 4 つの変換のシーケンス、ハードウェアトリガ 1

図 202. コンテキストの JSQR キューの例 (トリガの変更)

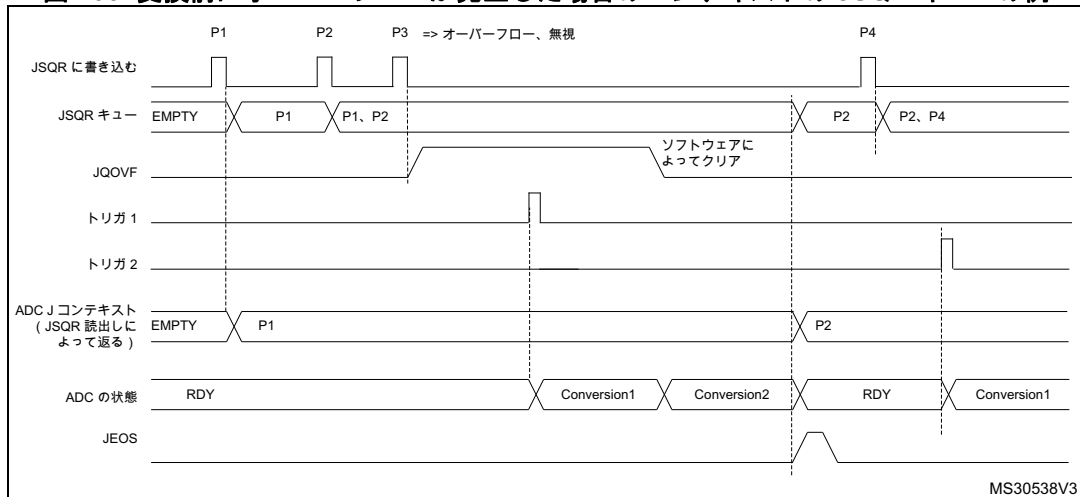


- 1. パラメータ :
P1 : 2 つの変換のシーケンス、ハードウェアトリガ 1
P2 : 1 つの変換のシーケンス、ハードウェアトリガ 2
P3 : 4 つの変換のシーケンス、ハードウェアトリガ 1

コンテキストのキュー：キューのオーバーフローが発生した場合の挙動

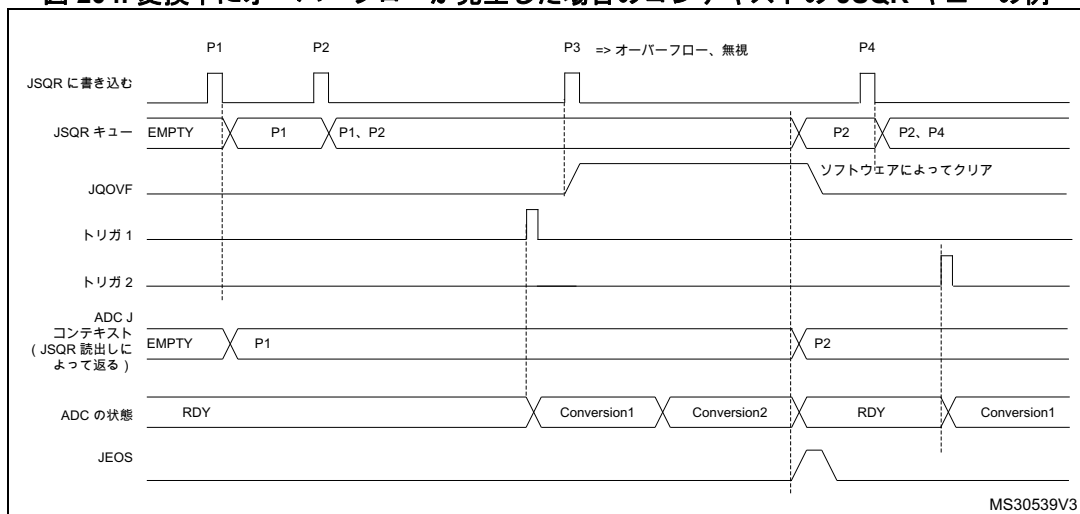
図 203 と 図 204 に、変換前または変換中にオーバーフローが発生した場合のコンテキストのキューの挙動を示します。

図 203. 変換前にオーバーフローが発生した場合のコンテキストの JSQR キューの例



1. パラメータ :
- P1 : 2 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 2
 - P3 : 3 つの変換のシーケンス、ハードウェアトリガ 1
 - P4 : 4 つの変換のシーケンス、ハードウェアトリガ 1

図 204. 変換中にオーバーフローが発生した場合のコンテキストの JSQR キューの例



1. パラメータ :
- P1 : 2 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 2
 - P3 : 3 つの変換のシーケンス、ハードウェアトリガ 1
 - P4 : 4 つの変換のシーケンス、ハードウェアトリガ 1

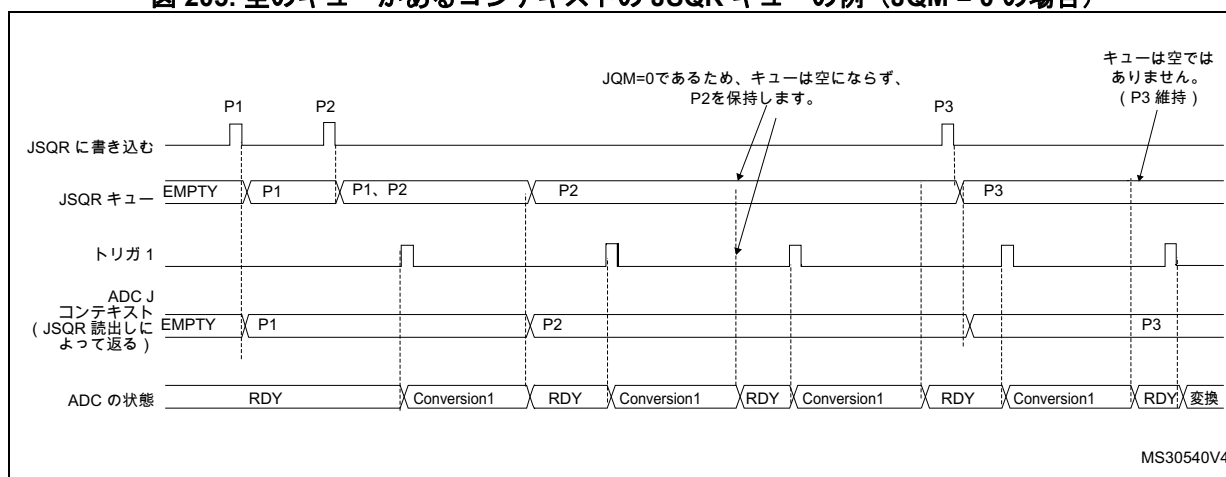
次のようにキューのオーバーフローを管理することを推奨します。

- JSQR レジスタへの各 P コンテキストの書き込み後、JQOVF フラグ は書き込みが無視されたかどうかを示します (割込みを生成することができます)。
- 一度前のコンテキスト P2 のフラグ JEOS がセットされた場合のみ、3 番目のコンテキスト (P3) を書き込んでキューのオーバーフローを避けます。これにより、前のコンテキストが消費され、キューがフル状態ではないことを確認できます。

コンテキストのキュー：キューが空になった場合の挙動

図 205 と 図 206 に、JQM = 0 または 1 の場合にキューが空になる際のコンテキストのキューの挙動を示します。

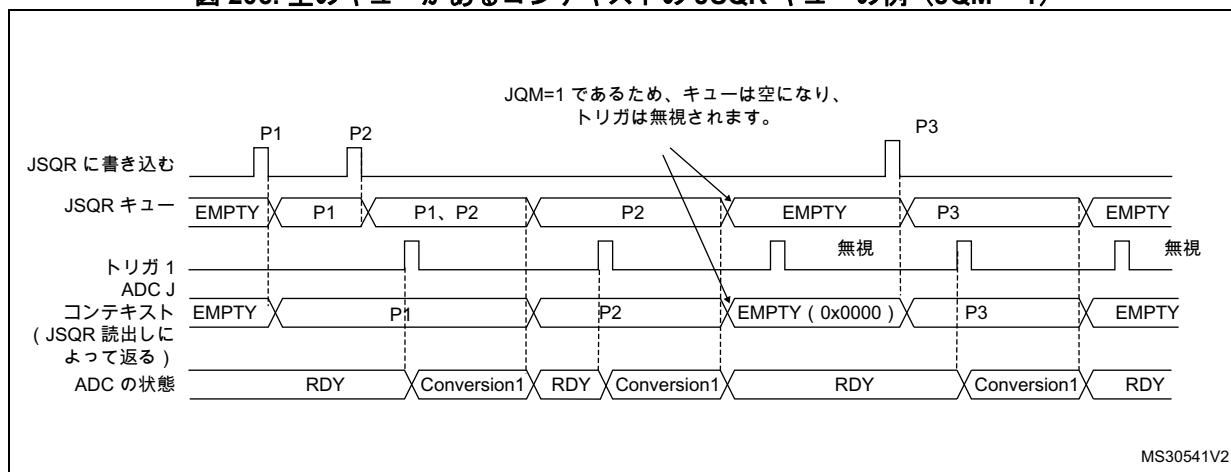
図 205. 空のキューがあるコンテキストの JSQR キューの例 (JQM = 0 の場合)



1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

注 : P3 を書き込む際、コンテキストはすぐに変更されます。ただし、内部での再同期のため、遅延が発生します。また、P3 を書き込んだ後または前にトリガが発生すると、コンテキスト P2 を考慮して変換が発生する可能性があります。この状況を避けるために、ユーザはすぐに適用される新しいコンテキストを書き込む際に ADC トリガが発生していないことを確認する必要があります。

図 206. 空のキューがあるコンテキストの JSQR キューの例 (JQM = 1)

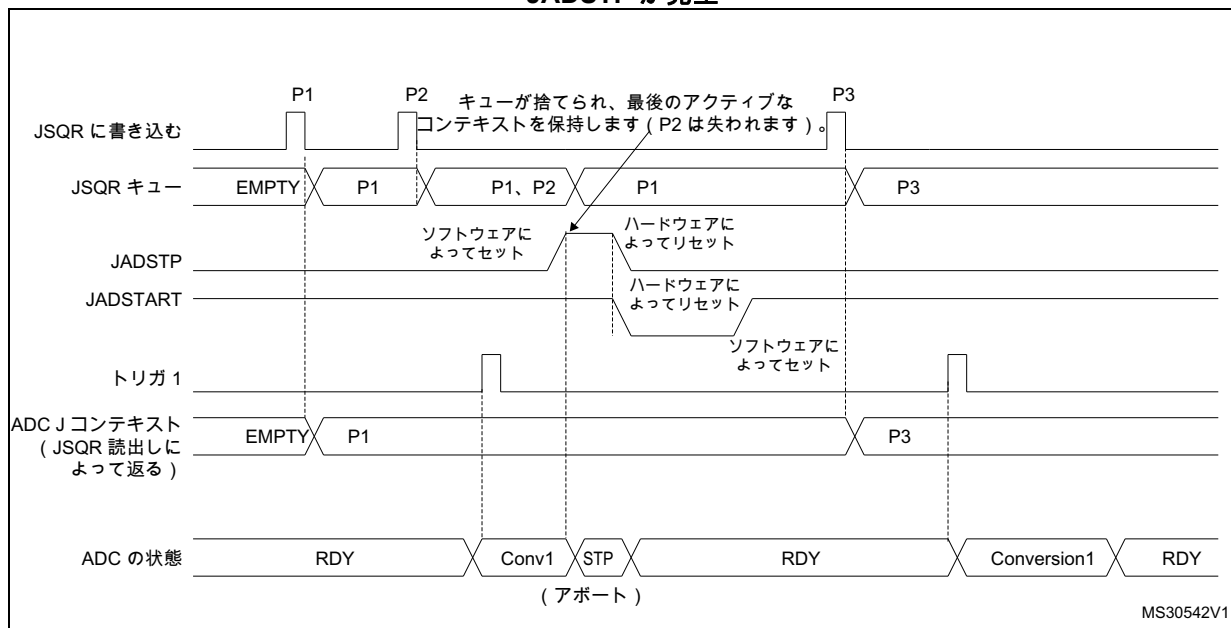


1. パラメータ :
- P1 : 1つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1つの変換のシーケンス、ハードウェアトリガ 1

コンテキストのキューの破棄

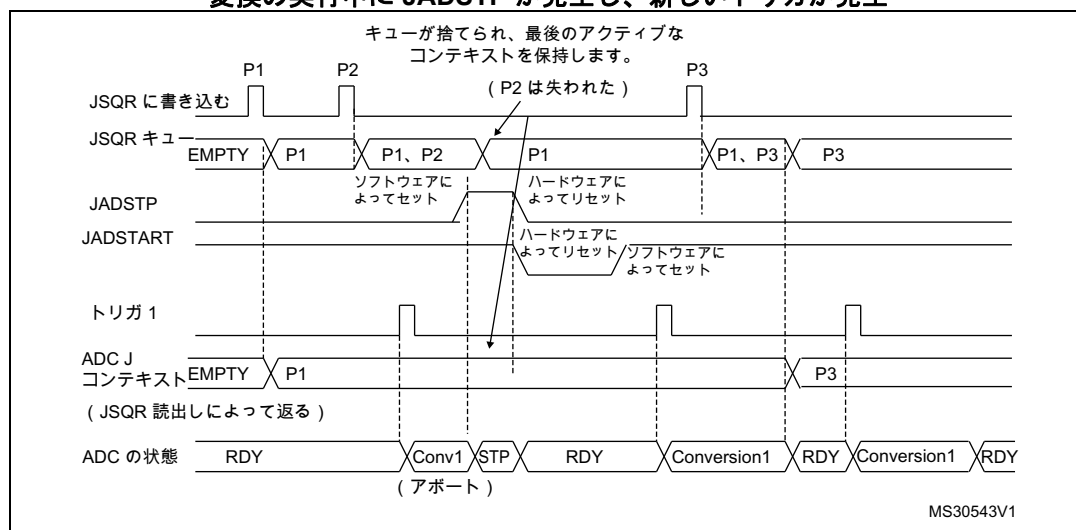
次の図に、キューが破棄された場合のさまざまな状況におけるコンテキストのキューの挙動を示します。

図 207. JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に JADSTP が発生



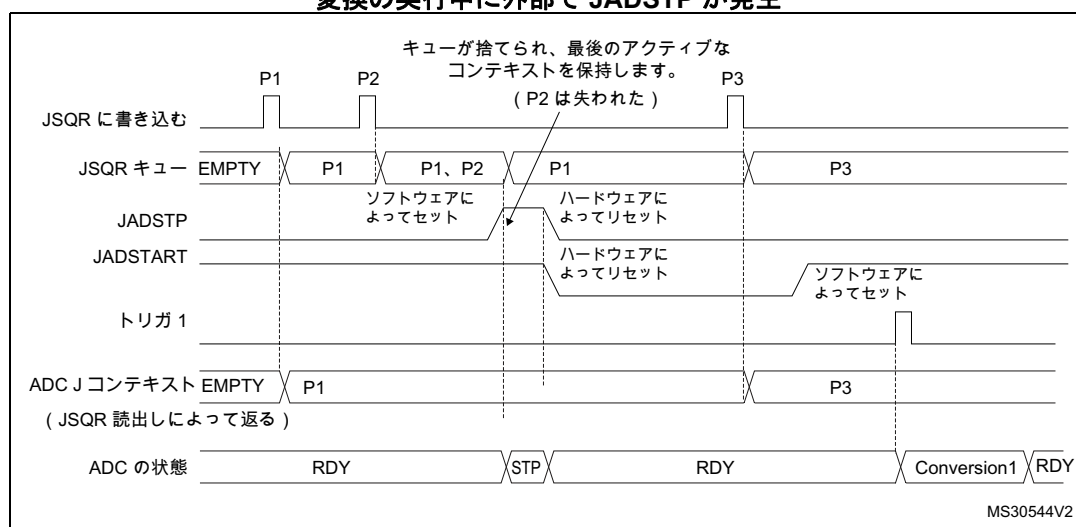
1. パラメータ :
- P1 : 1つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1つの変換のシーケンス、ハードウェアトリガ 1

図 208. JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に JADSTP が発生し、新しいトリガが発生



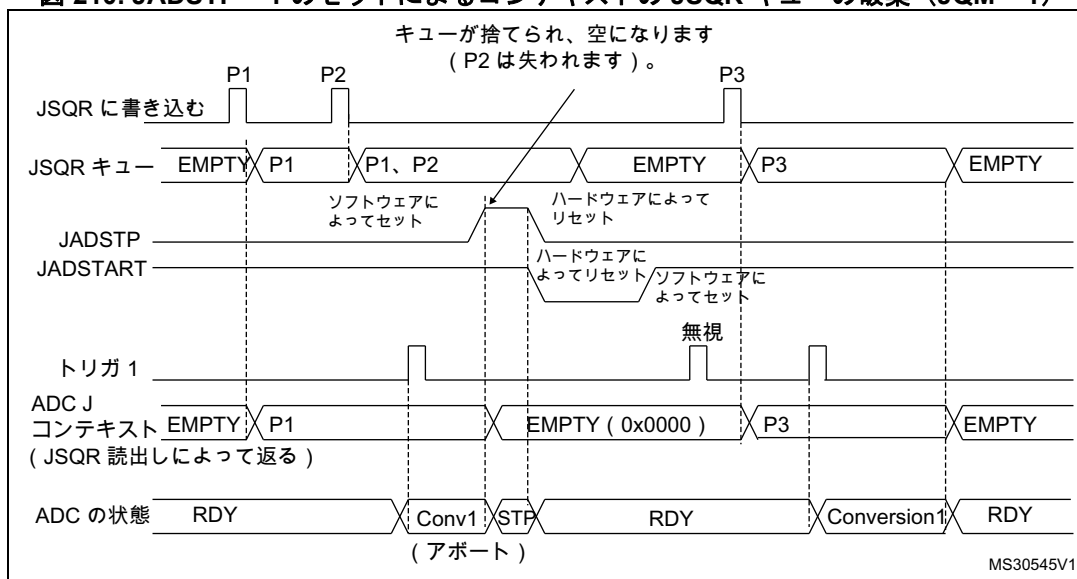
1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

図 209. JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0) - 変換の実行中に外部で JADSTP が発生



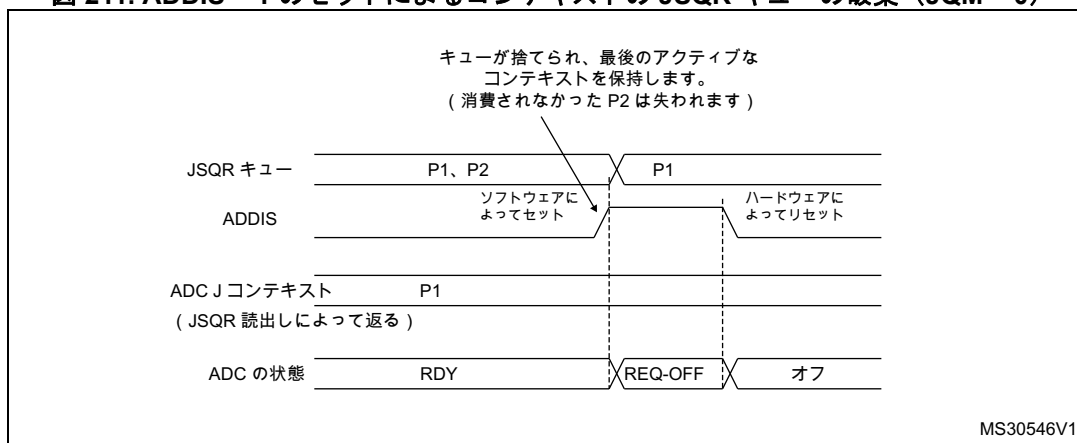
1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

図 210. JADSTP = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 1)



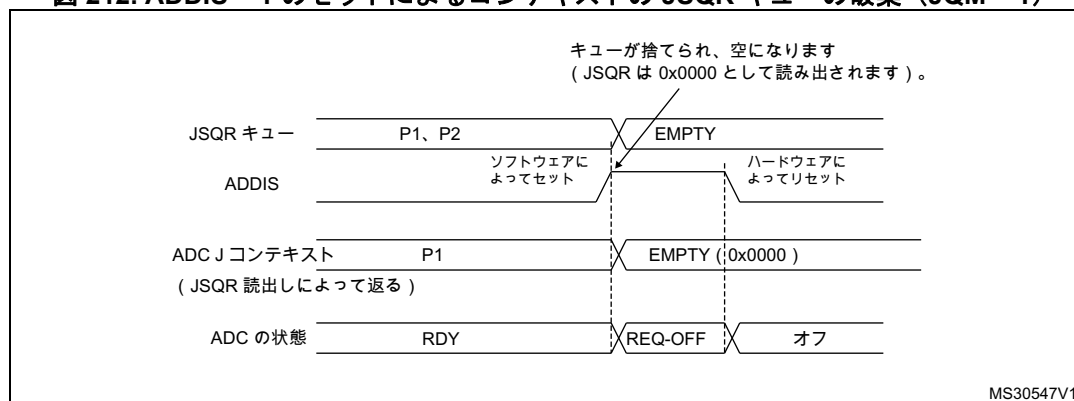
1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

図 211. ADDIS = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 0)



1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

図 212. ADDIS = 1 のセットによるコンテキストの JSQR キューの破棄 (JQM = 1)



1. パラメータ :
 P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
 P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

コンテキストのキュー：空のキューでの ADC 開始

最初のコンテキストが ADC 初期化の時点で不明である場合、空のキューで ADC 操作を開始するには次の手順に従う必要があります。この手順は、JQM ビットがリセットされているときのみ適用可能です。

5. 0 ではない JEXTEN でダミーの JSQR を書き込みます (ソフトウェア変換のトリガを除く)。
6. JADSTART をセットします。
7. JADSTP をセットします。
8. JADSTART がリセットされるまで待ちます。
9. JADSTART をセットします。

キューの無効化

ADC_CFGR レジスタでビット JQDIS = 1 にセットすることによって、キューを無効化できます。

26.4.22 プログラム可能な分解能 (RES) - 高速変換モード

ADC の分解能を下げることによって、高速変換を行うことができます。

分解能は、制御ビット RES[1:0] をプログラムすることによって、12、10、8、または 6 ビットに設定できます。図 217、図 218、図 219、および図 220 に、分解能とデータの整列に対する変換結果のフォーマットを示します。

分解能を下げることによって変換時間を高速にでき、高いデータ精度を必要としないアプリケーションに有効です。表 245 のように、これによって逐次比較ステップの間に変換時間は短縮されます。

表 245. T_{SAR} タイミングは分解能に依存

RES (ビット)	T_{SAR} (ADC クロックサイクル)	T_{SAR} (ns) ($F_{ADC} =$ 30 MHz 時)	T_{CONV} (ADC クロックサイクル) (サンプリング時間 = 2.5 ADC クロックサイクル)	T_{CONV} (ns) ($F_{ADC} = 30$ MHz 時)
12	12.5 ADC クロックサイクル	416.67 ns	15 ADC クロックサイクル	500.0 ns
10	10.5 ADC クロックサイクル	350.0 ns	13 ADC クロックサイクル	433.33 ns
8	8.5 ADC クロックサイクル	203.33 ns	11 ADC クロックサイクル	366.67 ns
6	6.5 ADC クロックサイクル	216.67 ns	9 ADC クロックサイクル	300.0 ns

26.4.23 変換の終了、サンプリングフェーズの終了 (EOC、JEOC、EOSMP)

ADC は各レギュラ変換の終了 (EOC) イベントと各インジェクト変換の終了 (JEOC) イベントをアプリケーションに通知します。

ADC は、新しいレギュラ変換データが ADC_DR レジスタで使用可能になるとすぐに EOC フラグをセットします。EOCIE ビットがセットされている場合は、割込みを生成することができます。EOC フラグは、ソフトウェアによって 1 を書き込むことによって、または ADC_DR を読み出すことによってクリアされます。

ADC は、新しいインジェクト変換データが ADC_JDRy レジスタの 1 つで使用可能になるとすぐに JEOC フラグをセットします。JEOCIE ビットがセットされている場合は、割込みを生成することができます。JEOC フラグは、ソフトウェアによって 1 を書き込むことによって、または対応する ADC_JDRy レジスタを読み出すことによってクリアされます。

さらに、ADC は、ステータスビット EOSMP をセットすることによって、サンプリングフェーズの終了を示します (レギュラ変換のみ)。EOSMP フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。EOSMPIE ビットがセットされている場合は、割込みを生成することができます。

26.4.24 変換シーケンスの終了 (EOS、JEOS)

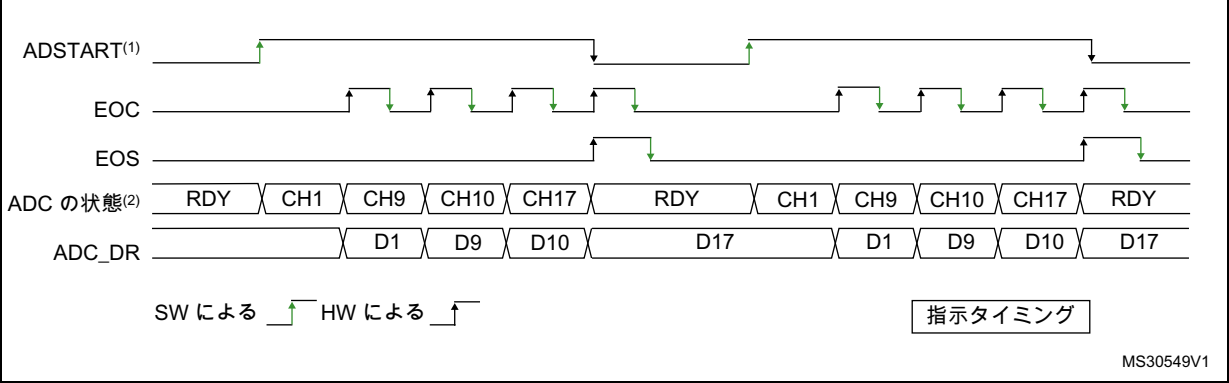
ADC は各レギュラシーケンスの終了 (EOS) と各インジェクトシーケンスの終了 (JEOS) イベントをアプリケーションに通知します。

ADC は、レギュラ変換シーケンスの最後のデータが ADC_DR レジスタで使用可能になるとすぐに、EOS フラグをセットします。EOSIE ビットがセットされている場合は、割込みを生成することができます。EOS フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

ADC は、インジェクト変換シーケンスの最後のデータが完了するとすぐに、JEOS フラグをセットします。JEOSIE ビットがセットされている場合は、割込みを生成することができます。JEOS フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

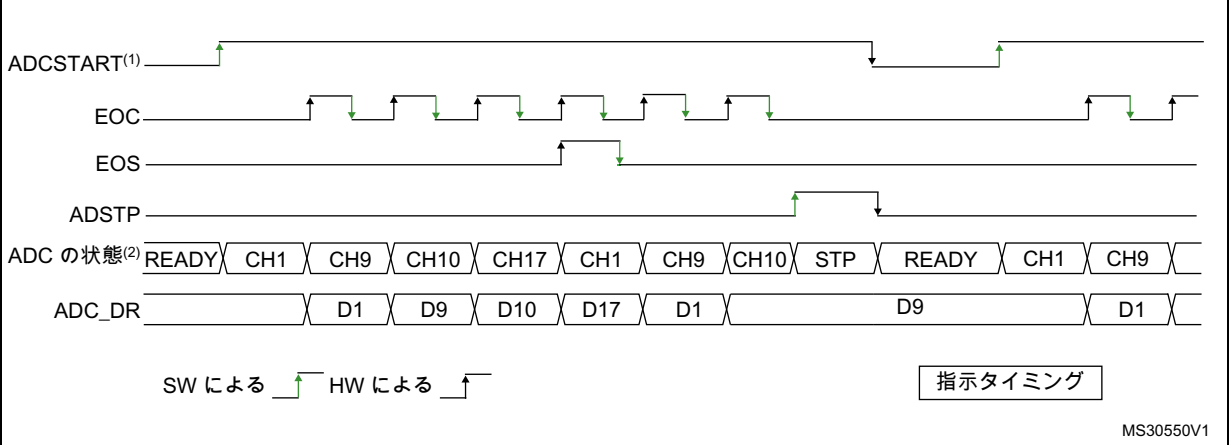
26.4.25 タイミング図の例 (シングル / 連続モード、ハードウェア / ソフトウェアトリガ)

図 213. シーケンスのシングル変換、ソフトウェアトリガ



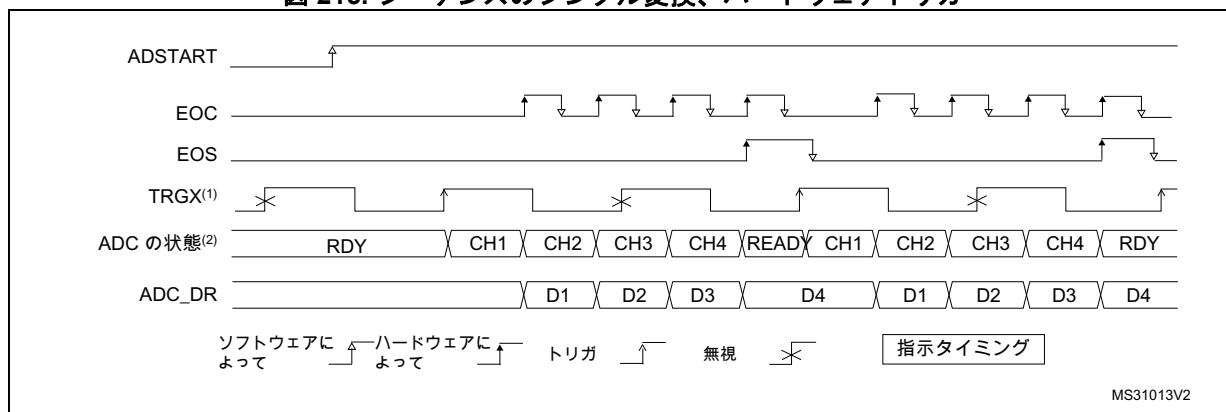
1. EXTEN = 0x0、CONT = 0
2. 選択されたチャネル = 1、9、10、17 ; AUTDLY = 0。

図 214. シーケンスの連続変換、ソフトウェアトリガ



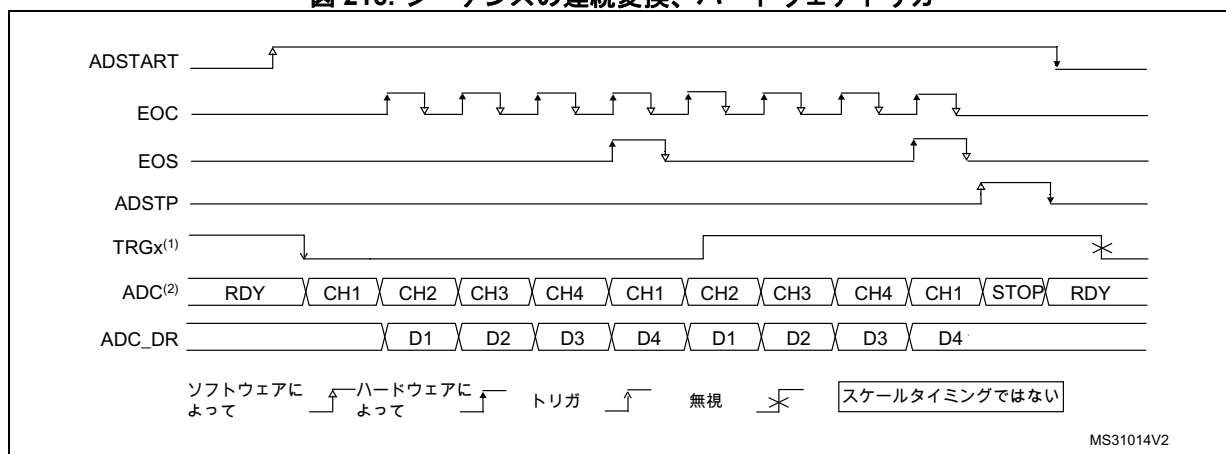
1. EXTEN = 0x0、CONT = 1
2. 選択されたチャネル = 1、9、10、17 ; AUTDLY = 0。

図 215. シーケンスのシングル変換、ハードウェアトリガ



1. TRGX (オーバー周波数) がトリガソースとして選択されます (EXTEN = 01、CONT = 0)。
2. 選択されたチャンネル = 1、2、3、4 ; AUTDLY = 0。

図 216. シーケンスの連続変換、ハードウェアトリガ



1. TRGX がトリガソースとして選択されます (EXTEN = 10、CONT = 1)。
2. 選択されたチャンネル = 1、2、3、4 ; AUTDLY = 0。

26.4.26 データ管理

データレジスタ、データの配置およびデータオフセット (ADC_DR、OFFSET、OFFSET_CH、ALIGN)

データおよび配置

各レギュラ変換チャンネルの終了時 (EOC イベントの発生時)、変換データの結果は 16 ビット幅の ADC_DR データレジスタに格納されます。

各インジェクト変換チャンネルの終了時 (JEOC イベントの発生時)、変換データの結果は対応する 16 ビット幅の ADC_JDRy データレジスタに格納されます。

ADC_CFGR レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。データは、図 217、図 218、図 219、および図 220 に示すように、右詰めまたは左詰めに配置できます。

特殊なケース：左詰めの場合、分解能が 6 ビットに設定されている場合を除き、データはハーフワード単位に配置されます。この場合、図 219 および図 220 に示すように、データはバイト単位に配置されます。

注： 左詰めは、オーバーサンプリングモードでサポートされません。ROVSE ビットおよび/または JOVSE ビットがセットされている場合、ALIGN ビット値は無視され、ADC は右詰めデータのみ提供します。

オフセット

オフセット y ($y = 1, 2, 3, 4$) は、ADC_OFRRy レジスタにビット OFFSET_EN = 1 をセットすることでチャンネルに適用されます。オフセットを適用するチャンネルは、ADC_OFRRy レジスタの OFFSET_CH[4:0] ビットにプログラムされます。この場合、変換された値は OFFSET[11:0] ビットに書き込まれたユーザ定義のオフセットによって減算されます。結果は負の値になることがあり、そのため読出しデータは符号付きで、SEXT ビットは拡張符号値を表します。

注： オフセット補正は、オーバーサンプリングモードでサポートされません。ROVSE および/または JOVSE ビットがセットされている場合、ADC_OFRRy レジスタの OFFSET_EN ビットの内容は無視されます（リセットとみなされます）。

表 248 に、アナログウォッチドッグ 1 に可能なすべての分解能での比較方法を示します。

表 246. オフセット計算対データ分解能

分解能 (ビット RES[1:0])	元の変換データとオフセット間での 減算：		結果	コメント
	元の変換データ、 左詰め	オフセット		
00 : 12 ビット	DATA[11:0]	OFFSET[11:0]	符号付き 12 ビットデータ	-
01 : 10 ビット	DATA[11:2],00	OFFSET[11:0]	符号付き 10 ビットデータ	ユーザは OFFSET[1:0] を "00" に設定する必要があります。
10 : 8 ビット	DATA[11:4],0000	OFFSET[11:0]	符号付き 8 ビットデータ	ユーザは OFFSET[3:0] を "0000" に設定する必要があります。
11 : 6 ビット	DATA[11:6],000000	OFFSET[11:0]	符号付き 6 ビットデータ	ユーザは OFFSET[5:0] を "000000" に設定する必要があります。

チャンネル "i" に対して ADC_DR (レギュラチャンネル) または ADC_JDRy (インジェクトチャンネル、 $y = 1, 2, 3, 4$) からデータを読み出す場合：

- 対応するチャンネルのオフセットの 1 つが有効 (ビット OFFSET_EN = 1) である場合、読出しデータは符号付きです。
- このチャンネルの 4 つのオフセットに有効なものがない場合、読出しデータは符号なしです。

図 217、図 218、図 219、および図 220 に、符号付きおよび符号なしのデータの配置を示します。

図 217. 右詰め (オフセット無効、符号なしの値)

12 ビットデータ															
bit15				bit7								bit0			
0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
10 ビットデータ															
bit15				bit7								bit0			
0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
8 ビットデータ															
bit15				bit7								bit0			
0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
6 ビットデータ															
bit15				bit7								bit0			
0	0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0

MS31015V1

図 218. 右詰め (オフセット有効、符号付きの値)

12 ビットデータ															
bit15				bit7								bit0			
SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
10 ビットデータ															
bit15				bit7								bit0			
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
8 ビットデータ															
bit15				bit7								bit0			
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D7	D6	D5	D4	D3	D2	D1	D0
6 ビットデータ															
bit15				bit7								bit0			
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D5	D4	D3	D2	D1	D0

MS31016V1

図 219. 左詰め (オフセット無効、符号なしの値)

12 ビットデータ															
bit15								bit7							
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
10 ビットデータ															
bit15								bit7							
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
8 ビットデータ															
bit15								bit7							
D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
6 ビットデータ															
bit15								bit7							
0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0

MS31017V1

図 220. 左詰め (オフセット有効、符号付きの値)

12 ビットデータ															
bit15								bit7							
SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
10 ビットデータ															
bit15								bit7							
SEXT	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0
8 ビットデータ															
bit15								bit7							
SEXT	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0
6 ビットデータ															
bit15								bit7							
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D5	D4	D3	D2	D1	D0	0

MS31018V1

オフセット補正

オフセット操作時に ADC_OFR_y レジスタの SATEN ビットがセットされているとき、データは符号なしです。すべてのオフセットデータは 0x000 で飽和されます (12 ビットモードの場合)。OFFSETPOS ビットがセットされている場合、オフセット方向は正で、データは 0xFFFF で飽和されます (12 ビットモードの場合)。8 ビットモードでは、データはそれぞれ 0x00 と 0xFF で飽和されます。

アナログウォッチドッグ比較は、オフセット補正の前に行われます。

ADC オーバーラン (OVR、OVRMOD)

オーバーランフラグ (OVR) は、ADC_DR FIFO (3 ステージ) がオーバーフローする前に、(CPU または DMA によって) レギュラ変換データが読み出されなかったときに通知します。

OVR フラグは、ADC_CR レジスタの FIFO がフルの間に新しい変換が完了すると、セットされます。OVRIE ビットが 1 にセットされると、割込みが生成されます。

オーバーラン条件が発生すると、ADC は動作し続け、ソフトウェアが ADSTP を 1 にセットすることによってシーケンスの停止およびリセットを決めるまで、変換を続行できます。ADC_DR の FIFO には 3 つのステージがあり、FIFO には最大 3 つのデータが保存されます。

OVR フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

制御ビット OVRMOD をプログラムすることによって、オーバーランイベントが発生したときにデータが保存されるか上書きされるかを設定できます。

- OVRMOD = 0 : オーバーランイベントが発生しても、データレジスタは上書きされません。古いデータは ADC_DR FIFO 深さ (3 つのステージ) まで保持され、新しい変換は破棄され、失われます。このモードでは、ADC_DR FIFO が無効になります。FIFO がフルの場合、その後の変換は行われますが、結果データは破棄されます。EOC は、ADC_DR レジスタを読み出すことによってクリアされます。ただし、FIFO にはまだ以前の変換データが含まれている場合があります。
- OVRMOD = 1 : データレジスタは最後の変換結果で上書きされ、以前の未読データは失われます。このモードでは、ADC_DR FIFO が無効になります。OVR が 1 のままの場合、その後の変換は通常どおり行われ、ADC_DR レジスタには常に最新の変換データが含まれます。

図 221. オーバーラン (OVRMOD = 0) の例

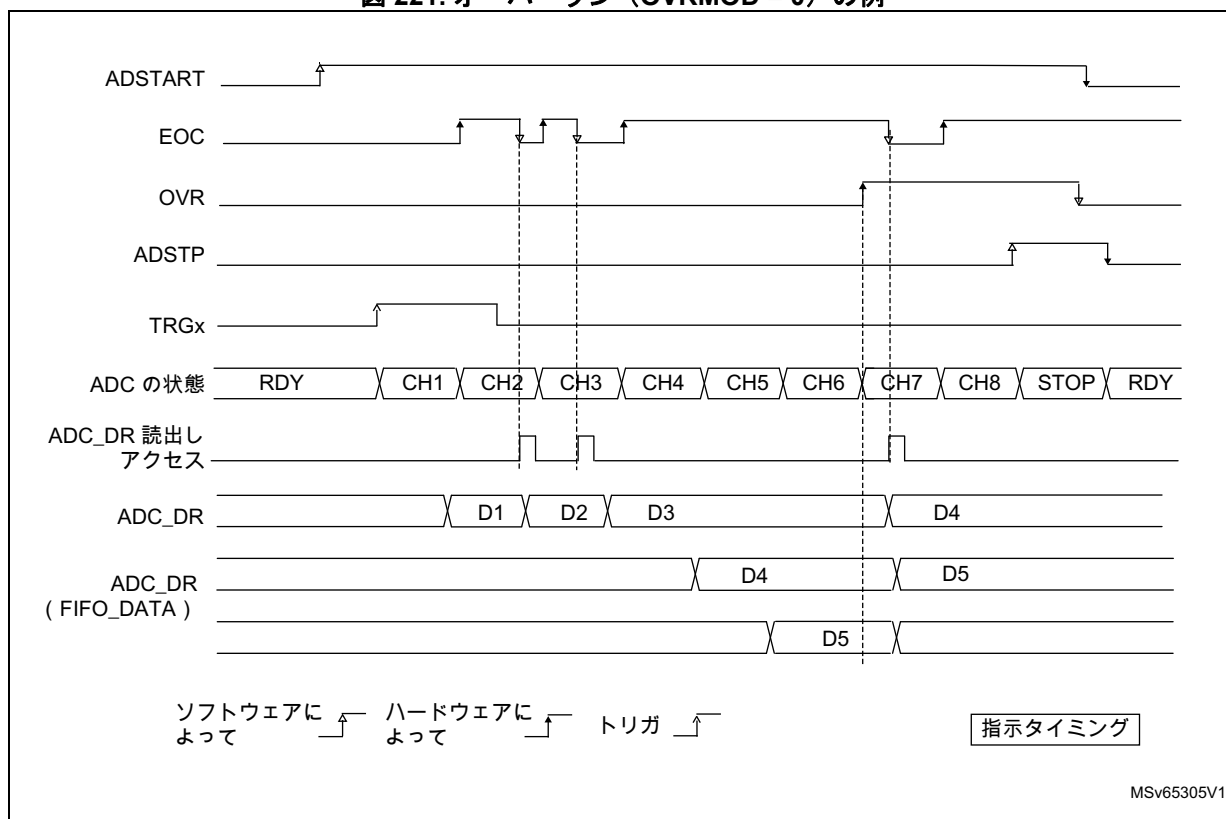
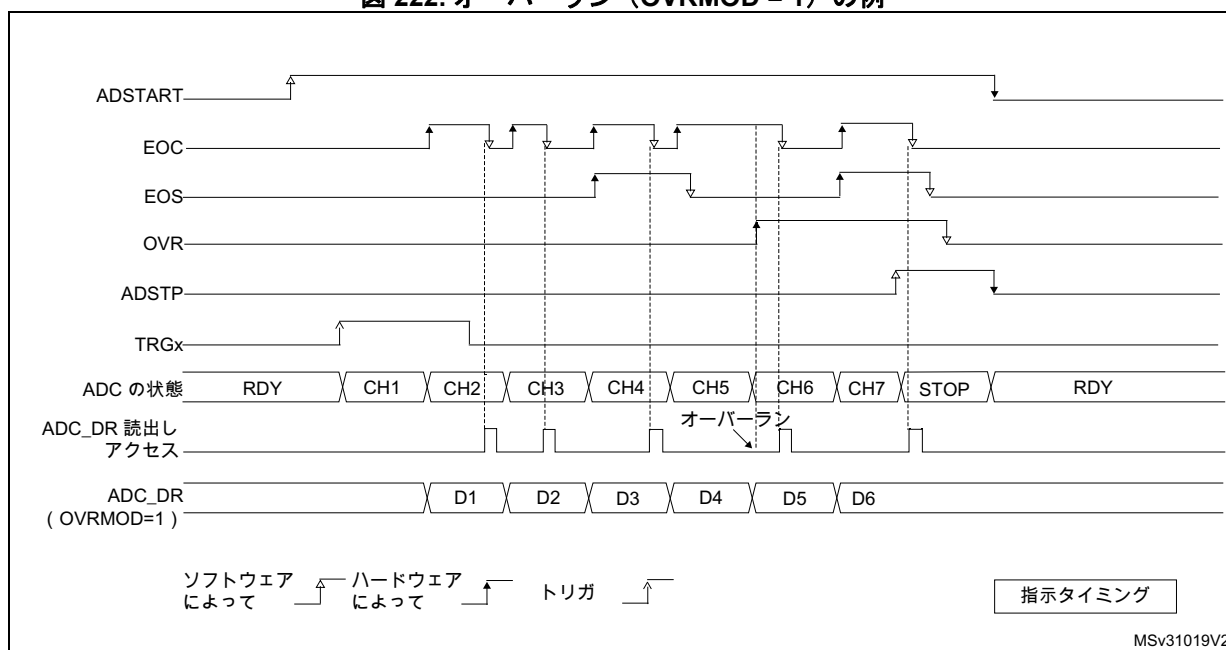


図 222. オーバーラン (OVRMOD = 1) の例



注：インジェクトチャンネルでは、4つのインジェクトチャンネルにそれぞれ専用のデータレジスタがあるため、オーバーランを検出する機能はありません。

DMA を使用しない変換シーケンスの管理

変換が十分遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、ソフトウェアはEOCフラグと関連の割り込みを使用して、各データを処理する必要があります。変換が終了するごとに、EOCはセットされ、ADC_DRレジスタを読み出すことができます。オーバーランイベントまたはFIFOオーバーフローをエラーとして管理するには、OVRMODを0に設定する必要があります。

オーバーランなしでのDMAを使用しない変換の管理

毎回データの読み出しをせずにADCに1つまたは複数のチャンネルを変換させると便利な場合があります（たとえば、アナログウォッチドッグがある場合）。この場合、OVRMODビットを1に設定する必要があります。ソフトウェアはOVRフラグを無視する必要があります。オーバーランイベントが発生してもADCは変換を続行し、ADC_DRレジスタには常に最新の変換が含まれます。

DMA を使用した変換の管理

変換されたチャンネルの値は特定のデータレジスタに格納されるので、複数のチャンネルの変換にはDMAの使用が便利です。これによって、ADC_DRレジスタにすでに格納されているデータの損失を防ぐことができます。

DMAモードが有効なとき（シングルADCモードまたはデュアルADCモードでMDMAが0b00以外に設定されている場合でADC_CFGRレジスタのDMAENビットが1にセットされている場合）、各チャンネルの変換後、DMAリクエストが生成されます。これにより、変換されたデータをADC_DRレジスタからソフトウェアで選択した場所へ転送することができます。

これにもかかわらず、DMA が DMA 転送リクエストを時間内に処理できなかったためにオーバーランが発生した場合 (OVR = 1)、ADC は DMA リクエストの生成を停止し、新しい変換に対応するデータは DMA によって転送されません。これは、RAM に転送されるすべてのデータを有効とみなすことができることを意味します。

OVRMOD ビットの設定に応じて、データは保存または上書きされます ([セクション : ADC オーバーラン \(OVR、OVRMOD\)](#) を参照してください)。

DMA 転送リクエストは、ソフトウェアが OVR ビットをクリアするまでブロックされます。

アプリケーションの用途に応じて 2 つの DMA モードがあり、ADC_CFGR レジスタの DMACFG ビット (シングル ADC モード) で設定され、ADC_CCR レジスタの DMACFG ビット (デュアル ADC モード) で設定されます。

- DMA ワンショットモード (DMACFG = 0)。
このモードは、DMA が固定数のデータを転送するようにプログラムされた場合に適しています。
- DMA サークュラモード (DMACFG = 1)
このモードは、DMA をサーキュラモードでプログラムする場合に適しています。

DMA ワンショットモード (DMACFG = 0)

このモードでは、ADC は新しい変換データが使用可能になるたびに DMA 転送リクエストを生成し、変換が再び開始された場合でも、DMA が最後の DMA 転送に達すると (転送完了割込みが発生すると) (DMA セクションを参照)、DMA リクエストの生成を停止します。

DMA 転送完了後 (DMA コントローラで設定されたすべての転送後) :

- ADC データレジスタの内容が停止されます。
- 実行中の変換は中止され、その部分的な結果は破棄されます。
- DMA コントローラに対する新しい DMA リクエストは発行されません。これによって、開始された変換がある場合のオーバーランエラーの生成を回避します。
- スキャンシーケンスは中止され、リセットされます。
- DMA は停止します。

DMA サークュラモード (DMACFG = 1)

このモードでは、DMA が最後の DMA 転送に達した場合でも、ADC は新しい変換データがデータレジスタで使用可能になるたびに DMA 転送リクエストを生成します。これにより、DMA を連続的なアナログ入力データストリームを処理するようにサーキュラモードに設定できます。

26.4.27 動的低電力機能

自動遅延変換モード (AUTDLY)

ADC は、AUTDLY 設定ビットによって制御されている自動遅延変換モードを実装しています。自動遅延変換は、ソフトウェアを単純化する場合だけでなく、ADC オーバーランが発生するリスクのある低周波数のクロックで動作しているアプリケーションのパフォーマンスを最適化する場合に便利です。

AUTDLY = 1 の場合、新しい変換は同じグループの前のデータがすべて処理された場合にのみ開始できます。

- レギュラ変換の場合 : ADC_DR レジスタの読出し後、または EOC ビットのクリア後 ([図 223](#) を参照)
- インジェクト変換の場合 : JEOS ビットのクリア後 ([図 224](#) を参照)

これは、データを読み出すシステムの速度に ADC の速度を自動的に適応させる方法です。

遅延は、各レギュラ変換後 (DISCEN = 0 か 1 かにかかわらず)、およびインジェクト変換の各シーケンス後 (JDISCEN = 0 か 1 かにかかわらず) に挿入されます。

注： 遅延は、インジェクトシーケンスの各変換の間には挿入されません (最後の変換を除く)。

変換時、この遅延中に発生した (同じ変換グループに対する) ハードウェアトリガイイベントは無視されます。

注： ソフトウェアトリガは、変換を再開するために、この遅延中に ADSTART ビットまたは JADSTART ビットをセットできるようにしている点で、ハードウェアトリガとは異なります。新しい変換を起動する前にデータを読み出すかどうかは、ソフトウェアに依存します。

遅延は、異なるグループの変換 (レギュラ変換に続いてインジェクト変換、またはその逆) の間には挿入されません。

- レギュラ変換の自動遅延中にインジェクトトリガが発生すると、すぐにインジェクト変換が開始されます (図 224 を参照)。
- インジェクトシーケンスが完了すると、新しいレギュラ変換を開始する前に、ADC は前のレギュラ変換の遅延を待ちます (終了していない場合) (図 226 を参照)。

自動インジェクトモード (JAUTO = 1) では挙動が少し異なり、前のインジェクト変換シーケンスの自動遅延が終了した場合 (JEOS がクリアされた場合) にのみ新しいレギュラ変換を開始することができます。これは、新しいシーケンスを開始する前に、ソフトウェアが特定のシーケンスのデータをすべて読み出すことができるようにするためです (図 227 を参照)。

自動遅延モード (JAUTO = 1、CONT = 1、および AUTDLY = 1) と組み合わせられた連続自動インジェクションモードの変換を停止するには、次の手順に従います。

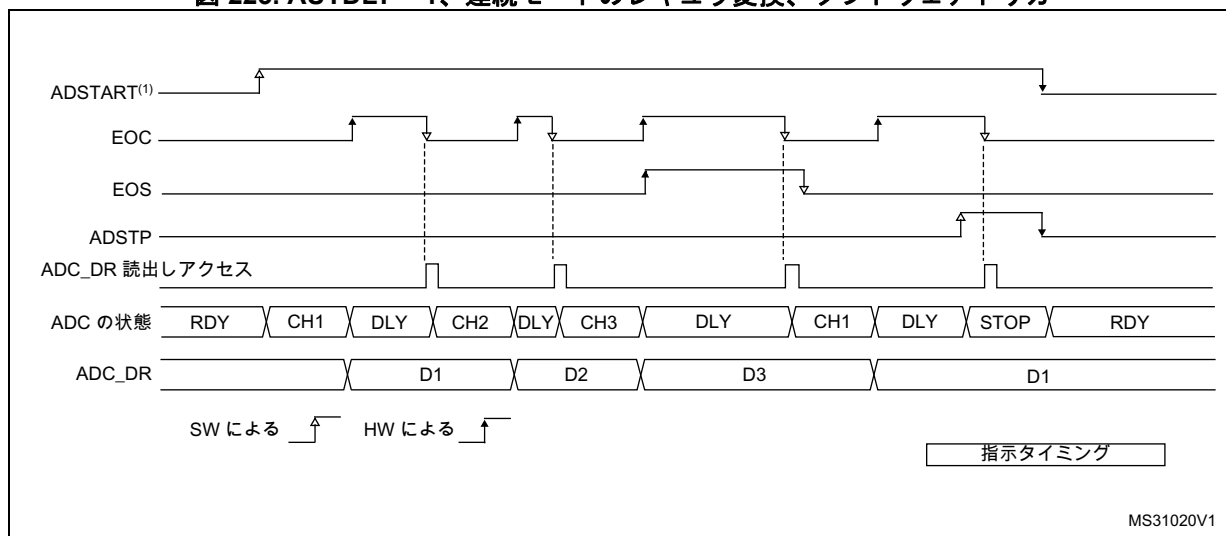
- JEOS = 1 になるまで (リスタートされる変換がなくなるまで) 待ちます。
- JEOS をクリアします。
- ADSTP = 1 にセットします。
- レギュラデータを読み出します。

この手順に従わない場合、ADSTP がセットされた後で JEOS がクリアされると、新しいレギュラシーケンスが再開されます。

AUTDLY モードでは、すでに実行中のレギュラシーケンス中、またはシーケンスの最後のレギュラ変換後の遅延中に発生した場合、ハードウェアレギュラトリガイイベントは無視されます。ただし、この遅延後に発生した場合は、後続の遅延のインジェクトシーケンス中に発生した場合でも、保留とみなされます。その後、インジェクトシーケンスの遅延終了時に変換が開始されます。

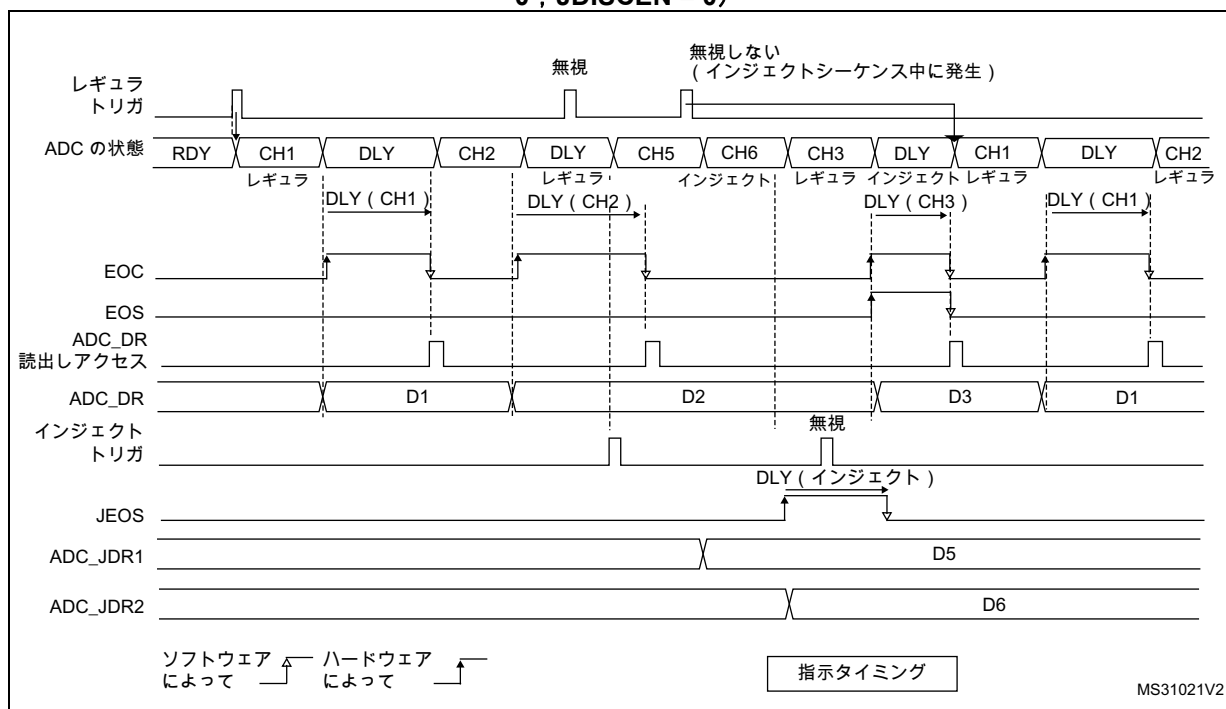
AUTDLY モードでは、すでに実行中のインジェクトシーケンス中、またはシーケンスの最後のインジェクト変換後の遅延中に発生した場合、ハードウェアインジェクトトリガイイベントは無視されます。

図 223. AUTDLY = 1、連続モードのレギュラ変換、ソフトウェアトリガ



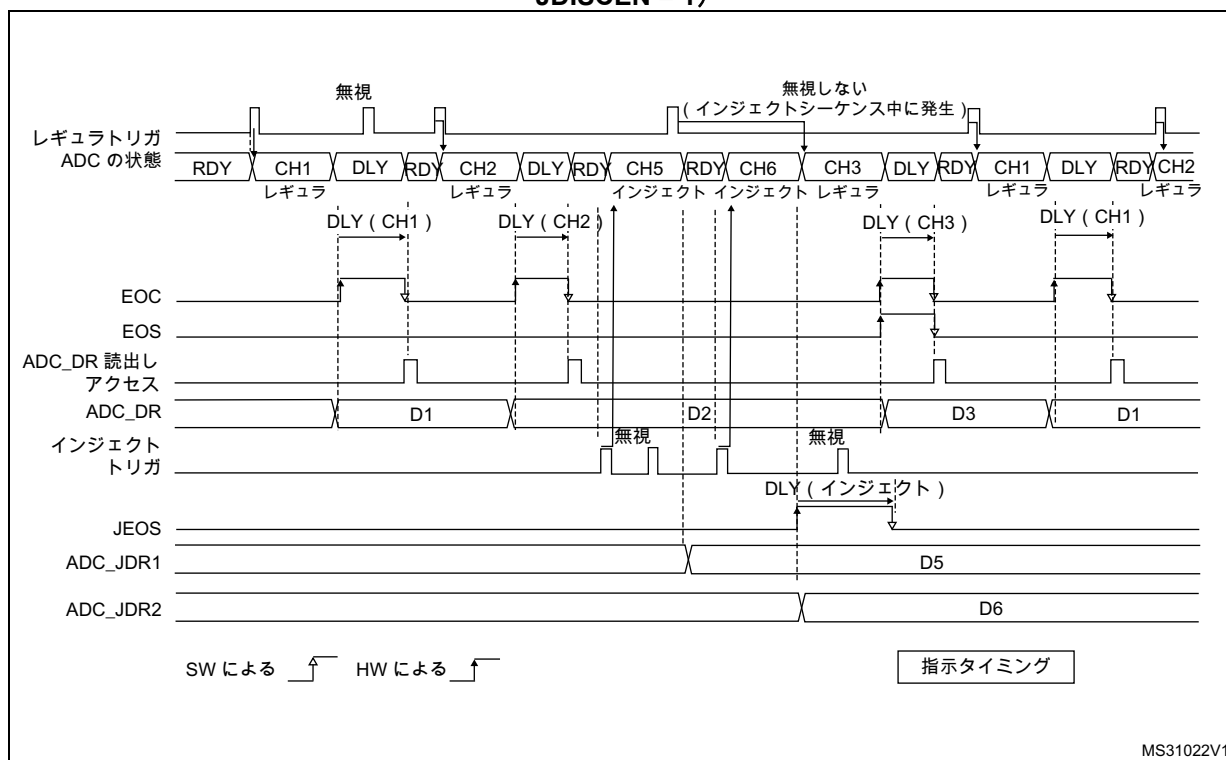
1. AUTDLY = 1
2. レギュラ設定: EXTEN = 0x0 (ソフトウェアトリガ)、CONT = 1、CHANNELS = 1、2、3
3. インジェクト設定: 無効

図 224. AUTDLY = 1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN = 0 ; JDISCEN = 0)



1. AUTDLY = 1
2. レギュラ設定: EXTEN = 0x1 (ハードウェアトリガ)、CONT = 0、DISCEN = 0、CHANNELS = 1、2、3
3. インジェクト設定: JEXTEN = 0x1 (ハードウェアトリガ)、JDISCEN = 0、CHANNELS = 5、6

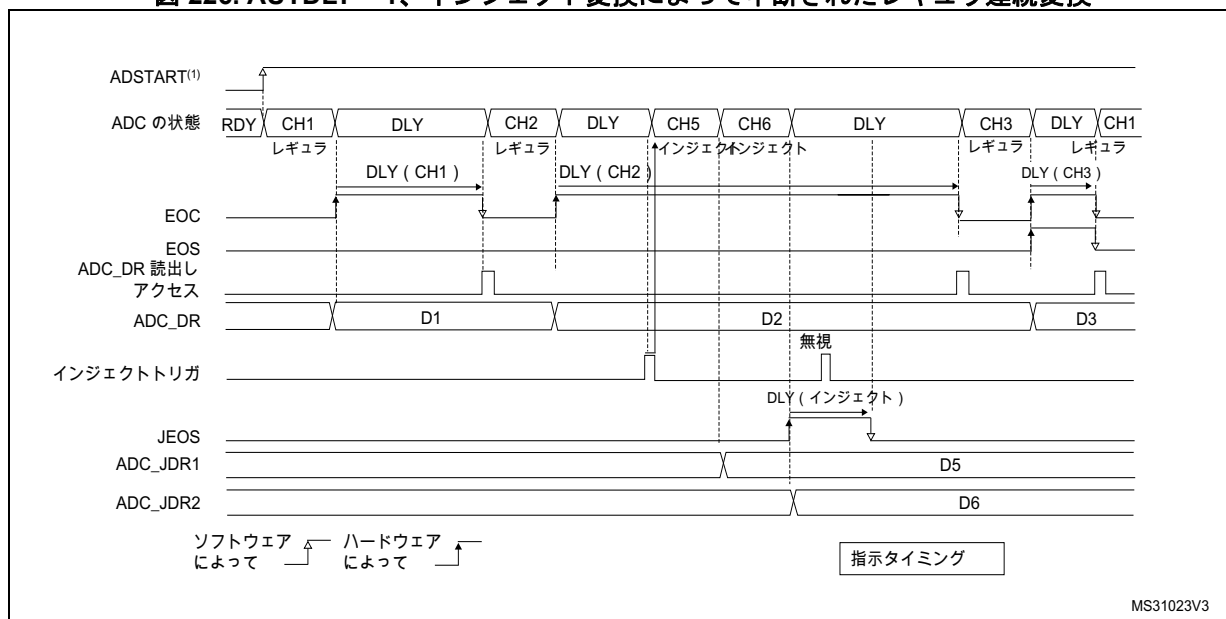
図 225. AUTDLY = 1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN = 1、JDISCEN = 1)



MS31022V1

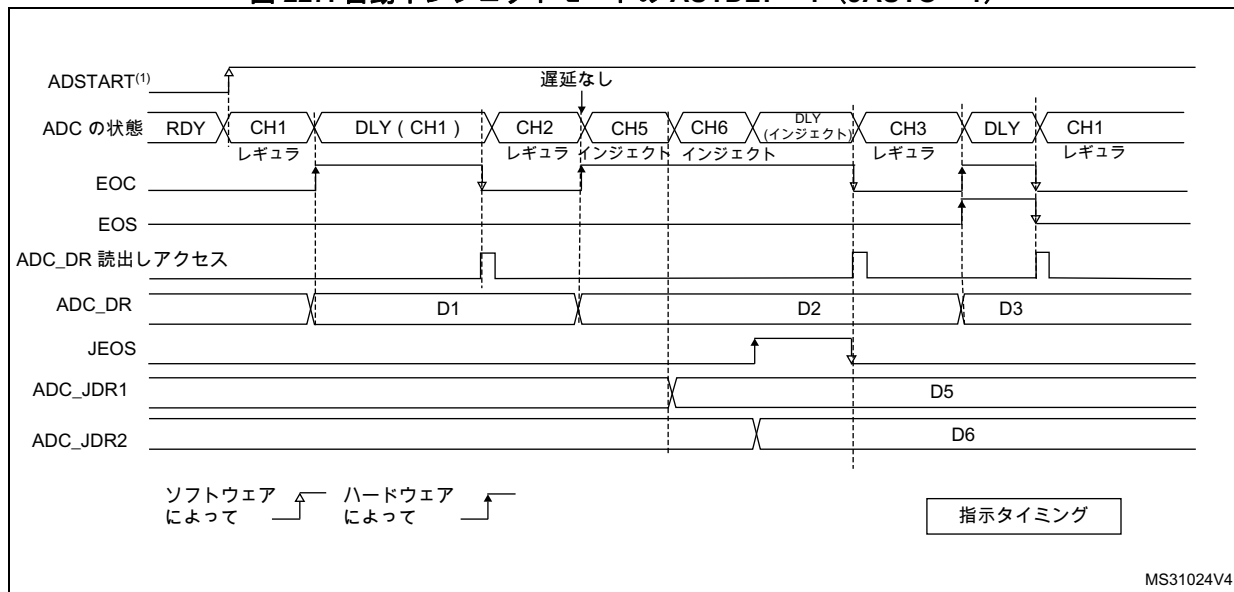
1. AUTDLY = 1
2. レギュラ設定 : EXTEN = 0x1 (ハードウェアトリガ)、CONT = 0、DISCEN = 1、DISCNUM = 1、CHANNELS = 1、2、3
3. インジェクト設定 : JEXTEN = 0x1 (ハードウェアトリガ)、JDISCEN = 1、CHANNELS = 5、6

図 226. AUTDLY = 1、インジェクト変換によって中断されたレギュラ連続変換



1. AUTDLY = 1
2. レギュラ設定 : EXTEN = 0x0 (ソフトウェアトリガ)、CONT = 1、DISCEN = 0、CHANNELS = 1、2、3
3. インジェクト設定 : JEXTEN = 0x1 (ハードウェアトリガ)、JDISCEN = 0、CHANNELS = 5、6

図 227. 自動インジェクトモードの AUTDLY = 1 (JAUTO = 1)

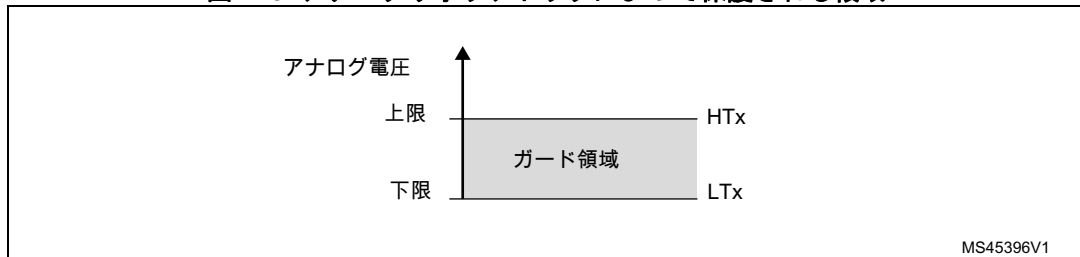


1. AUTDLY = 1
2. レギュラ設定 : EXTEN = 0x0 (ソフトウェアトリガ)、CONT = 1、DISCEN = 0、CHANNELS = 1、2
3. インジェクト設定 : JAUTO = 1、CHANNELS = 5、6

26.4.28 アナログウィンドウウォッチドッグ (AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx)

3 つの AWD アナログウォッチドッグは、設定された電圧範囲（ウィンドウ）内に一部のチャンネルが留まっているかどうかを監視します。

図 228. アナログウォッチドッグによって保護される領域



AWDx フラグと割込み

ADC_IER レジスタの AWDxIE (x = 1, 2, 3) をセットすることによって、3 つのアナログウォッチドッグでそれぞれ割込みを有効にできます。

AWDx (x = 1, 2, 3) フラグは、ソフトウェアで 1 を書き込むことによってクリアされます。

ADC 変換結果は配置される前に低閾値および高閾値と比較されます。

アナログウォッチドッグ 1 の説明

AWD アナログウォッチドッグ 1 は、ADC_CFGR レジスタの AWD1EN ビットをセットすることによって有効になります。ウォッチドッグは、選択された 1 つのチャンネルまたはすべての有効チャンネル⁽¹⁾が設定された電圧範囲（ウィンドウ）内にとどまっているかどうかを監視します。

表 247 に、1 つまたは複数のチャンネル上でアナログウォッチドッグを有効にするためにどのようにして ADC_CFGR レジスタを設定するかを示します。

表 247. アナログウォッチドッグチャンネル選択

アナログウォッチドッグによって保護されるチャンネル	AWD1SGL ビット	AWD1EN ビット	JAWD1EN ビット
なし	x	0	0
すべてのインジェクトチャンネル	0	0	1
すべてのレギュラチャンネル	0	1	0
すべてのレギュラおよびインジェクトチャンネル	0	1	1
1 つの ⁽¹⁾ インジェクトチャンネル	1	0	1
1 つの ⁽¹⁾ レギュラチャンネル	1	1	0
1 つの ⁽¹⁾ レギュラまたはインジェクトチャンネル	1	1	1

1. AWD1CH[4:0] ビットによって選択されます。チャンネルは、適切なレギュラまたはインジェクトシーケンスで変換されるようにプログラムする必要があります。

アナログウォッチドッグ (AWD1) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。

これらの閾値は、アナログウォッチドッグ 1 の ADC_TR1 レジスタの HT1[11:0] ビットおよび LT1[11:0] ビットでプログラムされます。12 ビット未満の分解能でデータを変換するときには

(RES[1:0] ビットに従って)、内部比較はオフセット補正の前に常に 12 ビット全体の元の変換データ (左詰め) に対して実行されるため、プログラムされた閾値の最下位ビットはクリアされたままである必要があります。

表 248 に、アナログウォッチドッグ 1 に可能なすべての分解能での比較方法を示します。

表 248. アナログウォッチドッグ 1 の比較

分解能 (ビット RES[1:0])	アナログウォッチドッグ比較 :		コメント
	元の変換データ、左詰め	閾値	
00 : 12 ビット	DATA[11:0]	LT1[11:0] および HT1[11:0]	-
01 : 10 ビット	DATA[11:2],00	LT1[11:0] および HT1[11:0]	ユーザは LT1[1:0] と HT1[1:0] を 00 に設定する必要があります。
10 : 8 ビット	DATA[11:4],0000	LT1[11:0] および HT1[11:0]	ユーザは LT1[3:0] と HT1[3:0] を 0000 に設定する必要があります。
11 : 6 ビット	DATA[11:6],000000	LT1[11:0] および HT1[11:0]	ユーザは LT1[5:0] と HT1[5:0] を 000000 に設定する必要があります。

ウォッチドッグ 1 のアナログウォッチドッグフィルタ

ADC が 1 つの入力チャネルだけで設定されているとき (スキャンモードで複数のチャネルを選択することはできない)、有効な ADC 変換データ間隔を ADC_TR1 レジスタで設定できます。

- 変換データが ADC_TR1 で定義された間に入ると、DMA リクエストが生成されます。
- そうでない場合、DMA リクエストは発行されません。RDATA レジスタは、変換ごとに更新されます。ADC_TR1 の AWDFLT ビットで指定された値より多い回数だけデータが範囲外であった場合、AWDx フラグがセットされ、対応する割込みが発行されます。

アナログウォッチドッグ 2 および 3 の説明

2 番目と 3 番目のアナログウォッチドッグはより柔軟で、AWDxCH[19:0] (x = 2, 3) の対応するビットをプログラムすることにより、いくつかの選択されたチャネルをガードすることができます。

対応するウォッチドッグは、AWDxCH[19:0] (x = 2, 3) の任意のビットがセットされると有効になります。

これらは 8 ビットの分解能に制限され、閾値の最上位の 8 ビットのみを HTx[7:0] と LTx[7:0] にプログラムすることができます。表 249 に、可能なすべての分解能での比較方法を示します。

表 249. アナログウォッチドッグ 2 および 3 の比較

分解能 (ビット RES[1:0])	アナログウォッチドッグ比較 :		コメント
	元の変換データ、左詰め	閾値	
00 : 12 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	DATA[3:0] はこの比較に適していません
01 : 10 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	DATA[3:2] はこの比較に適していません
10 : 8 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	-
11 : 6 ビット	DATA[11:6],00	LTx[7:0] および HTx[7:0]	ユーザは LTx[1:0] と HTx[1:0] を 00 に設定する必要があります

ADCy_AWDx_OUT 信号出力生成

各アナログウォッチドッグは、一部のオンチップタイマに ETR 入力（外部トリガ）が直接接続された内部ハードウェア信号 ADCy_AWDx_OUT（y = ADC の数、x = ウォッチドッグの数）と関連付けられています。ETR として ADCy_AWDx_OUT 信号を選択する方法を理解するには、オンチップタイマのセクションを参照してください。

ADCy_AWDx_OUT は、関連付けられたアナログウォッチドッグが有効になった時にアクティブ化されます。

- ADCy_AWDx_OUT は、ガードされた変換がプログラムされた閾値の外にある場合にセットされます。
- ADCy_AWDx_OUT は、次のガードされた変換がプログラムされた閾値内にある場合、終了後にリセットされます（次のガードされた変換がプログラムされた閾値の外にある場合は 1 のまま維持されます）。
- ADCy_AWDx_OUT は ADC を無効にする場合にもリセットされます（ADDIS = 1 に設定する場合）。レギュラまたはインジェクト変換の停止（ADSTP = 1 または JADSTP = 1 に設定）は、ADCy_AWDx_OUT の生成にまったく影響しない点に注意してください。

注： AWDx フラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。AWDx フラグは、ADCy_AWDx_OUT の生成にまったく影響しません（ソフトウェアによって AWDx フラグがクリアされず、このフラグが 1 のまま維持される間、ADCy_AWDx_OUT をトグルできます）。

図 229. ADCy_AWDx_OUT 信号生成（すべてのレギュラチャネル）

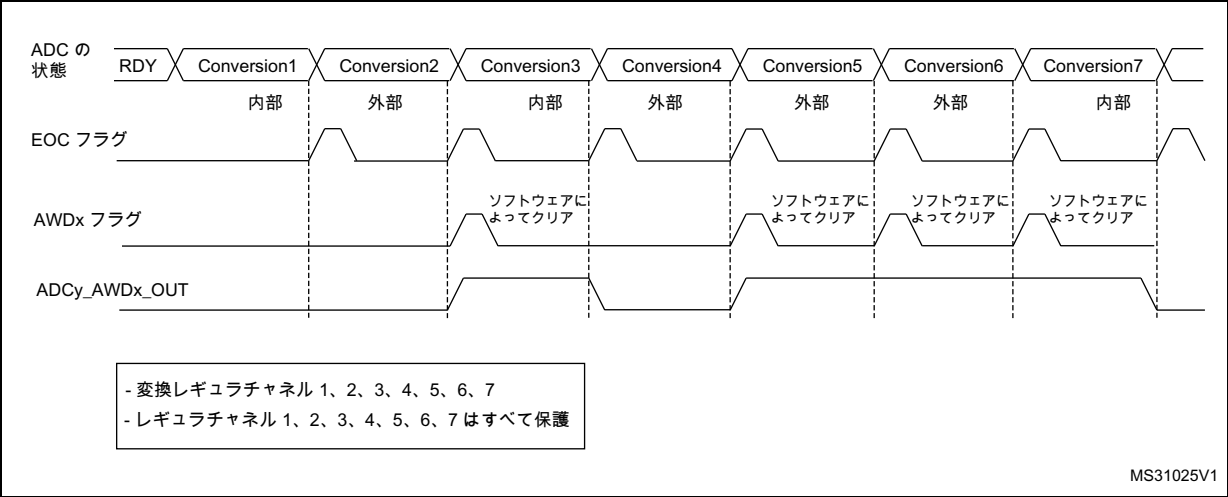


図 230. ADCy_AWDx_OUT 信号生成 (ソフトウェアによって AWDx フラグがクリアされない場合)

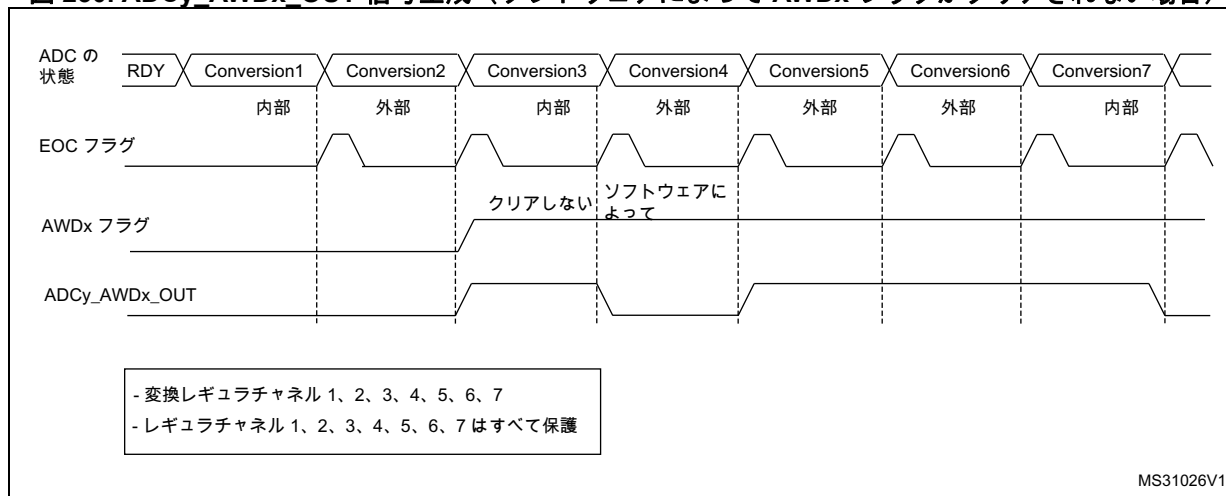


図 231. ADCy_AWDx_OUT 信号生成 (1 つのレギュラチャネル)

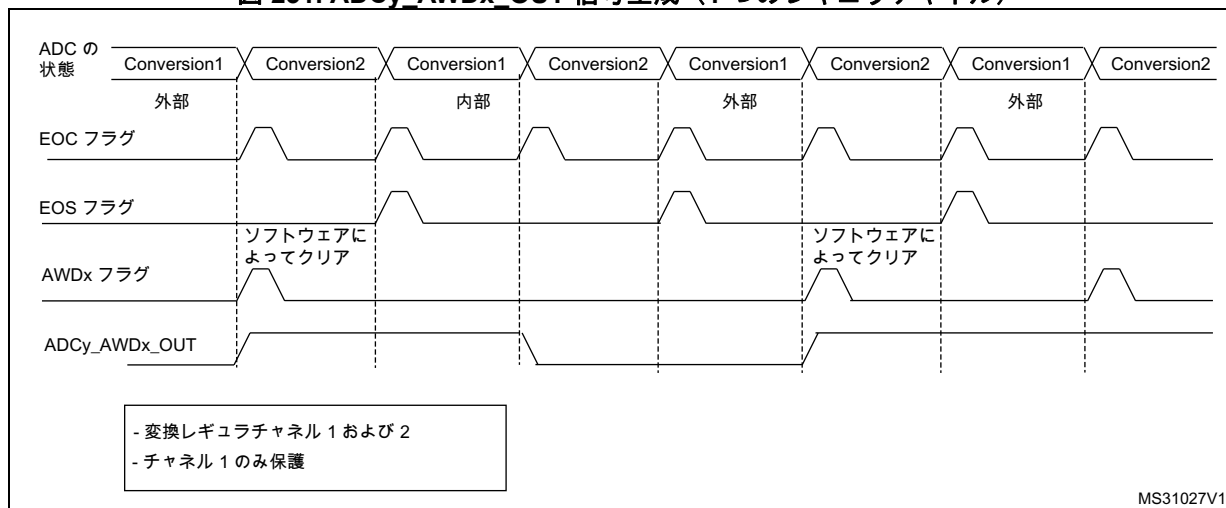
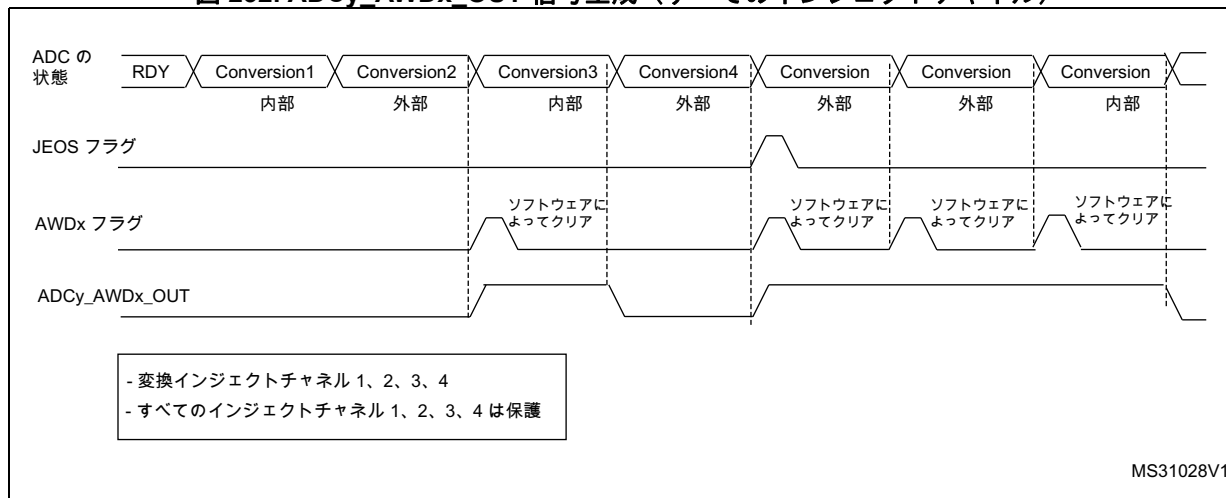


図 232. ADCy_AWDx_OUT 信号生成 (すべてのインジェクトチャネル)



アナログウォッチドッグ閾値制御

LTx[11:0] および HTx[11:0] は、アナログデジタル変換中（ADC 内部状態の変換開始から変換終了後までの間）に変更できます。LTx[11:0] および HTx[11:0] が、ADC ガードチャネルの ADC 変換中に更新された場合、この変換については、ウォッチドッグ機能はマスクされます。このマスクは、次の変換の開始時に除去され、その結果、アナログウォッチドッグ閾値が次の ADC 変換から適用されます。アナログウォッチドッグ比較は、変換が終了するたびに行われます。現在の ADC データが新しい間隔から外れていた場合、割込みまたは ADC_AWDx_OUT 信号は発行されません。割込みおよび AWD 生成は、閾値更新後に開始された変換の終了時にのみ行われます。AWD_xOUT がすでにアサートされていた場合、新しい閾値をプログラムしても、AWDx_OUT 信号はデアサートされません。

オフセット補正とアナログウォッチドッグ

オフセット補正が有効な場合、アナログウォッチドッグはデータ補正の前に閾値を比較します。

26.4.29 オーバーサンプリング回路

オーバーサンプリングユニットは、データの前処理を実行して、CPU の負荷を軽減します。複数の変換を処理して、最大 16 ビット幅の単一データに平均化できます。

以下の形式で結果を提供します。N および M は調整可能です。

$$\text{Result} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{Conversion}(t_n)$$

平均化、データレートの削減、SNR の向上、基本的フィルタリングをハードウェアによって実行できます。

オーバーサンプリング比 N は、ADC_CFGR2 レジスタの OVFS[2:0] ビットによって定義され、その範囲は 2~256倍 です。分周係数 M は、最大 8 ビットの右ビットシフトから成っており、ADC_CFGR2 レジスタの OVSS[3:0] ビットを使用して定義します。

合計ユニットは最大 20 ビット（256x 12 ビットの結果）の結果をもたらし、最初に右へシフトされます。最下位 16 ビットまで切り捨てられ、シフトによって残った最下位ビットを使用して最も近い値に丸められてから ADC_DR データレジスタに転送されます。

注： シフト後の中間結果が 16 ビットを超える場合、結果は飽和なしで単純に切り捨てられます。

図 233. 20 ビットから 16 ビットへの結果の切詰め



図 234 に、元の 20 ビットの累積データから最終的な 16 ビットの結果への処理の数値例を示します。

図 234. 5 ビットシフトと丸めの数値例



表 250 に、元の変換データが 0xFFF の場合のさまざまな N と M の組み合わせでのデータフォーマットを示します。

表 250. 最大出力結果対 N と M (灰色のセルは切詰め)

オーバー サンプリング比	元のデータ (最大)	シフト なし OVSS = 0000	1 ビット シフト OVSS = 0001	2 ビット シフト OVSS = 0010	3 ビット シフト OVSS = 0011	4 ビット シフト OVSS = 0100	5 ビット シフト OVSS = 0101	6 ビット シフト OVSS = 0110	7 ビット シフト OVSS = 0111	8 ビット シフト OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

オーバーサンプリングモードでの変換タイミングに変更はありません。サンプル時間はオーバーサンプリングシーケンス全体を通じて一定に保たれます。新しいデータは N 回の変換ごとに提供され、同等遅延は $N \times T_{CONV} = N \times (t_{SMPL} + t_{SAR})$ に等しくなります。フラグは、次のようにセットされます。

- サンプリングフェーズの終了 (EOSMP) は、各サンプリングフェーズ後にセットされます。
- 変換の終了 (EOC) は、N 回の変換ごとに発生し、オーバーサンプリングされた結果が使用可能になります。
- シーケンスの終了 (EOS) は、オーバーサンプリングされたデータのシーケンスが完了すると発生します (すなわち、N x シーケンス長の変換合計後)。

オーバーサンプリング時の ADC 動作モードのサポート (シングル ADC モード)

オーバーサンプリングモードでは、ほとんどの ADC 動作モードが維持されます。

- シングル/連続変換モード
- ソフトウェアまたはトリガによる ADC 変換の開始
- 変換中の ADC の停止 (中止)
- オーバーラン検出時の CPU または DMA 経由でのデータの読出し
- 低電力モード (AUTDLY)
- プログラム可能な分解能: この場合、削減された変換値 (ADC_CFGR レジスタの RES[1:0] ビットに従って) の累積、切り捨て、丸め、およびシフトは、12ビット変換と同様に行われます。

注： オーバーサンプリングされたデータを操作するときには、配置モードは使用できません。ADC_CFGR の ALIGN ビットは無視され、データは常に右詰めで提供されます。

オフセット補正は、オーバーサンプリングモードでサポートされません。ROVSE および/または JOVSE ビットがセットされている場合、ADC_OFRy レジスタの OFFSET_EN ビットの内容は無視されます（リセットとみなされます）。

アナログウォッチドッグ

アナログウォッチドッグ機能は保持されますが、次のような違いがあります。

- RES[1:0] ビットは無視され、比較は常に AWD1 の場合は HT1[11:0] および LT1[11:0] の 12 ビット値全部、AWD2 および AWD3 の場合は HT2/HT3[7:0] および LT2/LT3[7:0] の 8 MSB 値を使用して行われます。
- 比較は 16 ビットのオーバーサンプリングされた結果の最上位の 12 ビット（AWD1 の場合は ADC_DR[15:4]、AWD2 および AWD3 の場合は ADC_DR[15:8]）に対して行われます。

注： 高いシフト値を使用するときには注意が必要です。これによって比較範囲が小さくなります。たとえば、オーバーサンプリングされた結果が 4 ビットシフトされた場合、12 ビットの右詰めデータになり、有効なアナログウォッチドッグ比較が行われるのは、8 ビットに対してだけです。比較は ADC_DR[11:4] と HTx[7:0] / LTx[7:0]（AWD1/2/3）の間で行われ、HT1[11:8] と LT1[11:8] はリセットのまま保持されます（AWD1 のみ）。

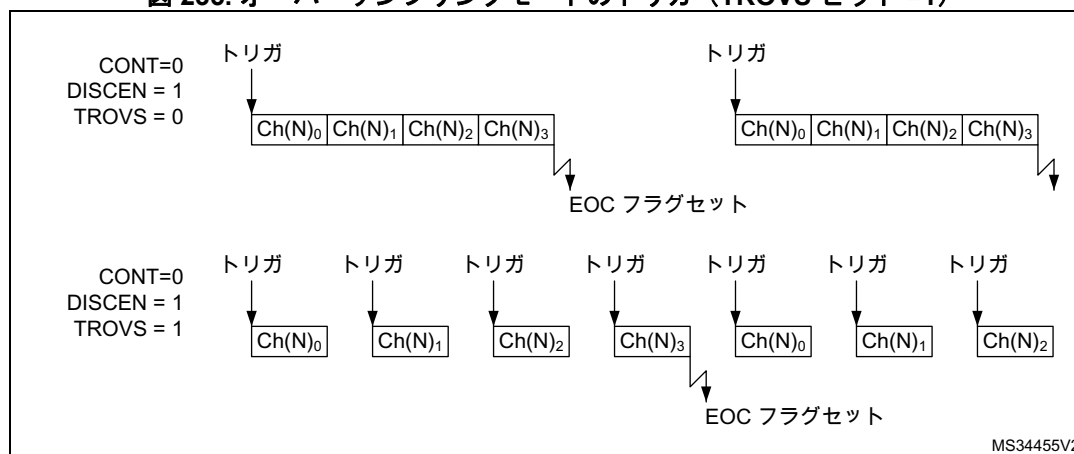
トリガモード

平均化回路は、基本的なフィルタリング目的で使用することもできます。あまり強力なフィルタではありません（ロールオフが低速であり、阻止帯域減衰が限られる）、ノッチフィルタとして使用して、周期的なノイズ周波数を低減できます（一般に主電源またはスイッチモードの電源が原因）。この目的のために、ADC_CFGR2 の TROVS ビットで特定の不連続モードを有効にして、変換時間に依存せずに、オーバーサンプリング周波数をユーザが定義できるようにできます。

下記の図 235 に、不連続モード中にトリガに反応して変換を開始する方法を示します。

TROVS ビットがセットされている場合、DISCEN ビットの内容は無視され、1 とみなされます。

図 235. オーバーサンプリングモードのトリガ (TROVS ビット=1)



オーバーサンプリング時のインジェクトおよびレギュラシーケンサ管理

オーバーサンプリングモードでは、インジェクトおよびレギュラシーケンサに異なる動作を行わせることができます。シーケンサを同時に使用する場合がある場合、それら両方を制限付きでオーバーサンプリングを有効にできます（これは、独自の累算ユニットに関連します）。

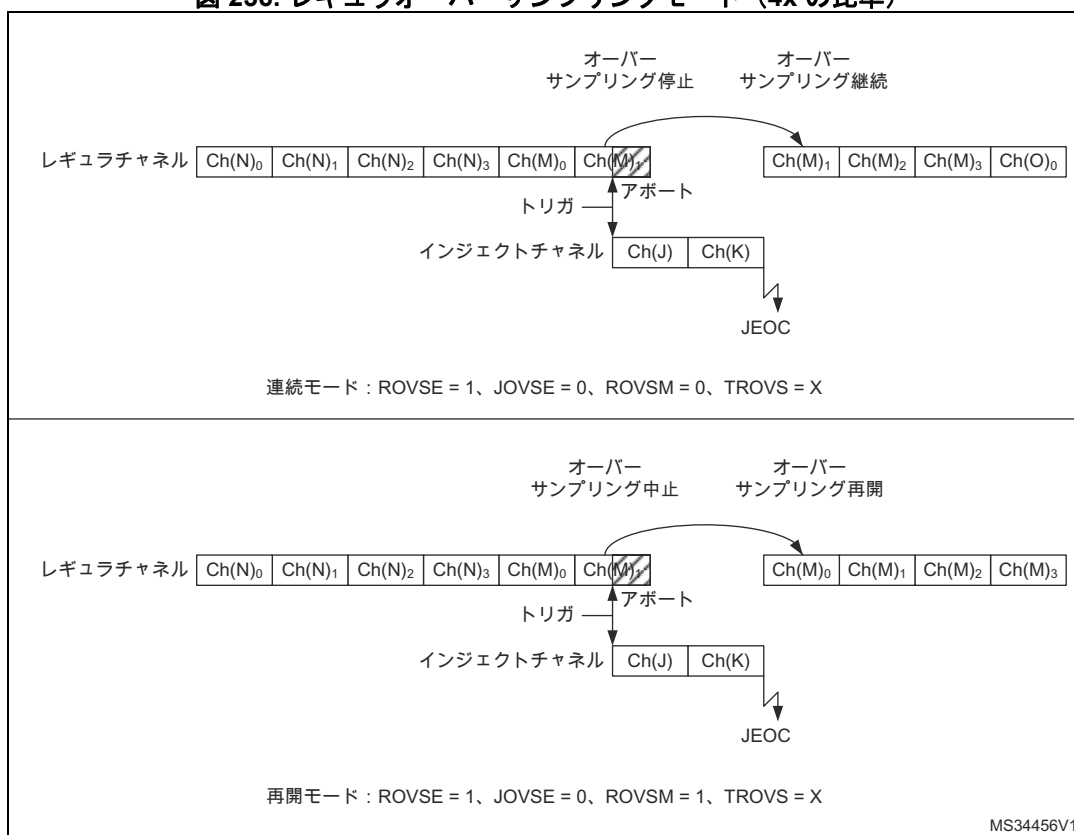
オーバーサンプリングレギュラチャネルのみ

レギュラオーバーサンプリングモードビット ROVSM は、インジェクト変換によって割込みが発生した場合のレギュラサンプリングシーケンスの再開方法を定義します。

- 連続モードでは、最後の有効なデータ（インジェクトトリガによる変換中止リクエストの前）から累算を再開します。これにより、インジェクションの頻度に関係なく、オーバーサンプリングを完了できます（トリガ間で最低 1 回のレギュラ変換を完了できます）。
- 再開モードでは、累算が 0 から再開します（前の変換結果は無視されます）。このモードでは、オーバーサンプリングに使用されるすべてのデータが 1 つの時間スロット内で連続変換されたことを保証できます。オーバーサンプリング時間を超えるインジェクショントリガ時間がかかることに注意が必要です。この条件が満たされない場合、オーバーサンプリングを完了できず、レギュラシーケンスはブロックされます。

図 236 では、4 倍のオーバーサンプリング比の例を示しています。

図 236. レギュラオーバーサンプリングモード (4x の比率)



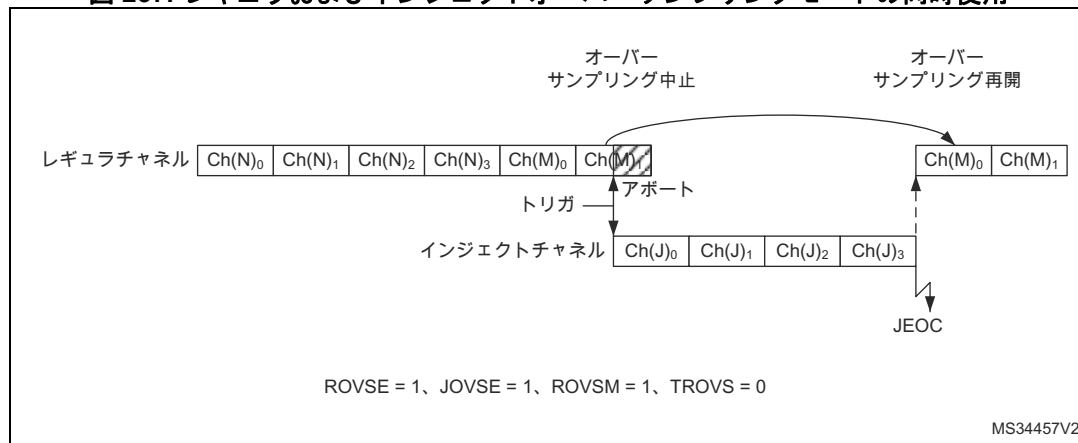
オーバーサンプリングインジェクトチャネルのみ

インジェクトオーバーサンプリングモードビット JOVSE は、インジェクトシーケンサでの変換専用のオーバーサンプリングを有効にします。

オーバーサンプリングレギュラおよびインジェクトチャンネル

ROVSE ビットおよび JOVSE ビットの両方をセットできます。この場合、レギュラオーバーサンプリングモードは以下の図 237 に示すように、強制的に再開モードになります (ROVSM ビット無視)。

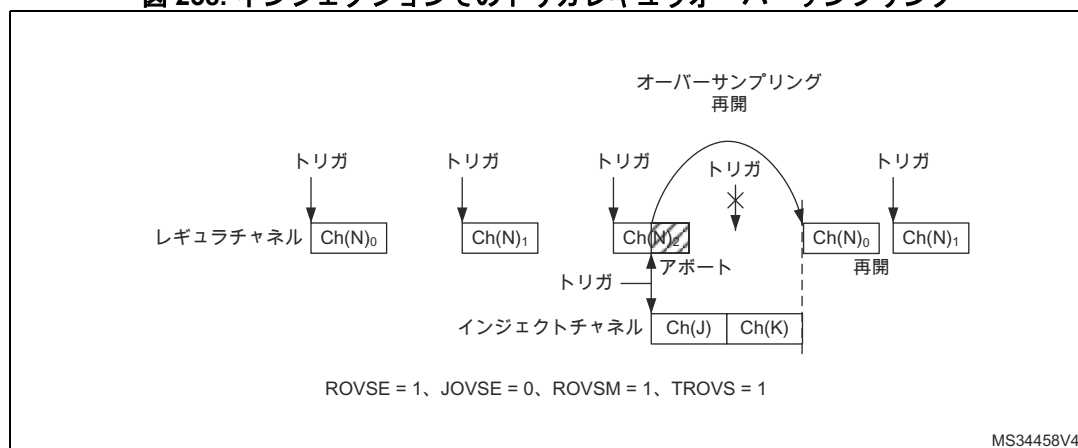
図 237. レギュラおよびインジェクトオーバーサンプリングモードの同時使用



インジェクト変換でのトリガレギュラオーバーサンプリングモード

インジェクト変換でのトリガレギュラオーバーサンプリングモードが可能です。この場合、インジェクトモードオーバーサンプリングモードは無効になり、ROVSM ビットは無視されます (再開モードが強制されます)。JOVSE ビットをリセットする必要があります。動作については、以下の図 238 に示します。

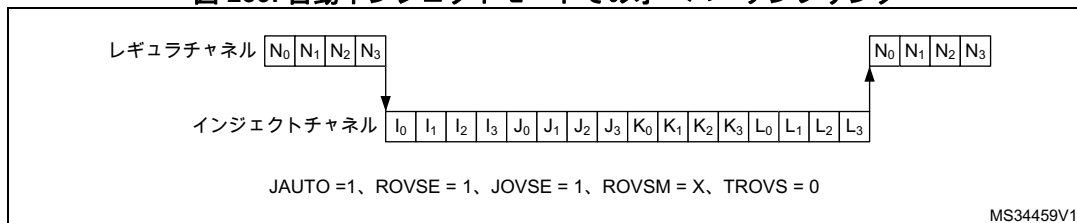
図 238. インジェクションでのトリガレギュラオーバーサンプリング



自動インジェクトモード

自動インジェクトシーケンスをオーバーサンプリングして、レジスタにすべての変換結果を格納し、DMA リソースを保存できます。このモードは、レギュラおよびインジェクトオーバーサンプリング両方がアクティブの状態でのみ使用可能です。JAUTO = 1、ROVSE = 1 および JOVSE = 1 の場合です。その他の組み合わせはサポートされません。ROVSM ビットは自動インジェクトモードでは無視されます。以下の図 239 に、変換シーケンスの方法を示します。

図 239. 自動インジェクトモードでのオーバーサンプリング



TROVS ビットを使用してトリガモードを有効にすることもできます。この場合、ADC は次のように設定する必要があります。JAUTO = 1、DISCEN = 0、JDISCEN = 0、ROVSE = 1、JOVSE = 1 および TROVSE = 1。

オーバーサンプリング時のデュアル ADC モードのサポート

デュアル ADC 設定で動作時、インジェクト同時モードおよびレギュラ同時モードに対してオーバーサンプリングを有効にできます。この場合、2 つの ADC をまったく同じ設定（オーバーサンプリング含む）でプログラムする必要があります。

他のすべてのデュアル ADC モードは、レギュラまたはインジェクトオーバーサンプリングが有効になっている場合（ROVSE = 1 または JOVSE = 1）サポートされません。

組み合わせモードの概要

以下の表 251 に、サポートされていないモードを含むすべての組み合わせの概要を示します。

表 251. オーバーサンプリング動作モードの概要

レギュラ オーバー サンプリング ROVSE	インジェクト オーバー サンプリング JOVSE	オーバー サンプリング モード ROVSM 0 = 連続 1 = 再開	トリガレギュラ モード TROVS	コメント
1	0	0	0	レギュラ連続モード
1	0	0	1	サポートされません
1	0	1	0	レギュラ再開モード
1	0	1	1	トリガレギュラ再開モード
1	1	0	X	サポートされません
1	1	1	0	インジェクトおよびレギュラ再開モード
1	1	1	1	サポートされません
0	1	X	X	インジェクトオーバーサンプリング

26.4.30 デュアル ADC モード

2 つ以上の ADC を持つデバイス（図 240 を参照）ではデュアル ADC モードを使用できます。

デュアル ADC モードでは、変換の開始は ADC_CCR レジスタの DUAL[4:0] ビットによって選択されたモードに応じて、ADCx マスタにより ADC スレーブに交互または同時にトリガされます。

次の 4 つのモードを備えています。

- インジェクト同時モード
- レギュラ同時モード
- インタリーブモード
- オルタネートトリガモード

これらのモードを次のように組み合わせて使用することも可能です。

- インジェクト同時モード + レギュラ同時モード
- レギュラ同時モード + オルタネートトリガモード
- インジェクト同時モード + インタリーブモード

デュアル ADC モードで (ADC_CCR レジスタの DUAL[4:0] ビットがゼロ以外の場合)、ADC_CFGR レジスタの CONT、AUTDLY、DISCEN、DISCNUM[2:0]、JDISCEN、JQM、JAUTO ビットは、マスタおよびスレーブ ADC の間で共有されます。スレーブ ADC のビットは、常にマスタ ADC の対応するビットと等しくなります。

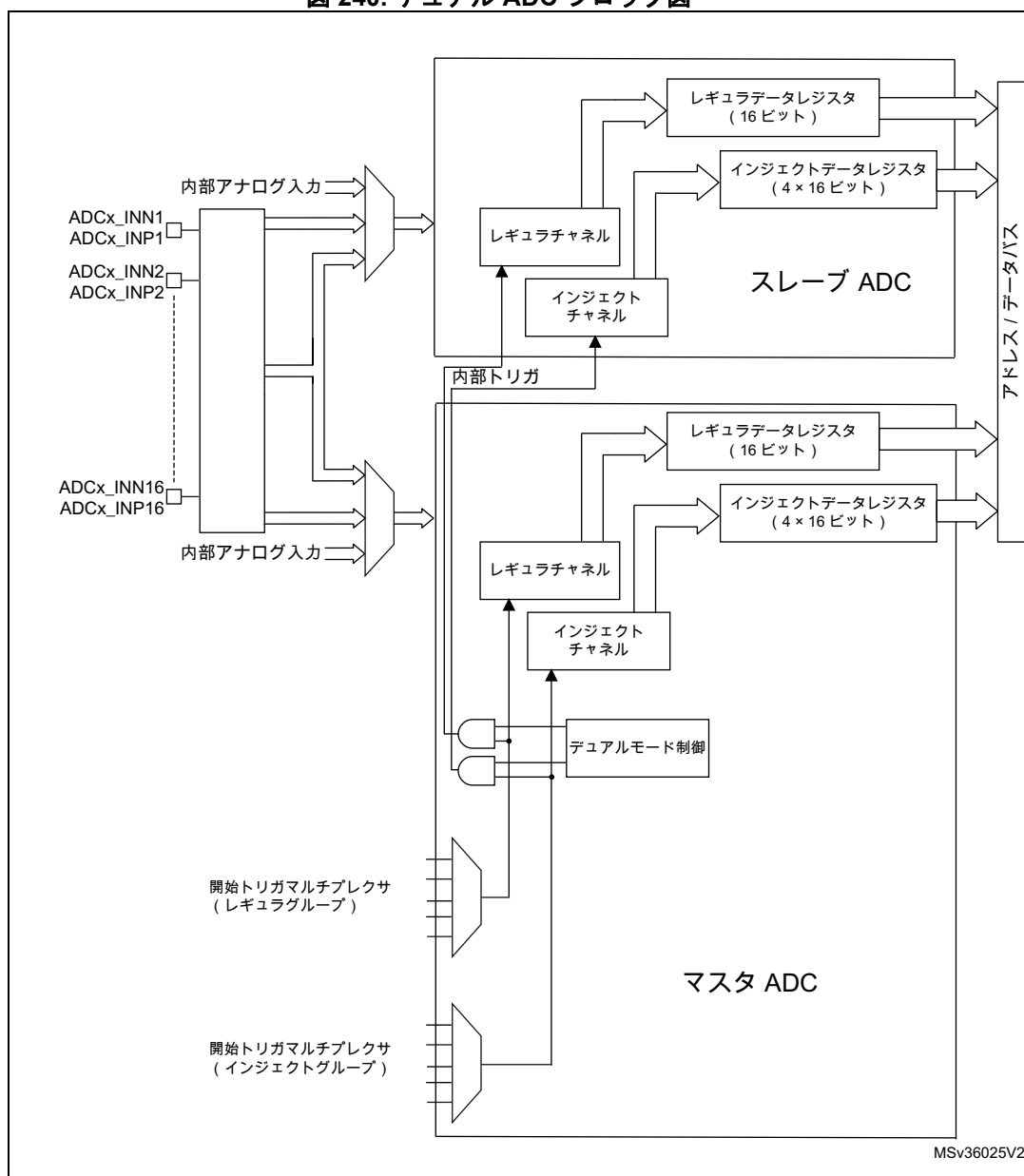
デュアルモードで変換を開始するには、ソフトウェアまたはハードウェアトリガやレギュラまたはインジェクトトリガを設定するために、マスタ ADC の EXTEN、EXTSEL、JEXTEN、JEXTSEL ビットのみをプログラムする必要があります (スレーブ ADC の EXTEN[1:0] および JEXTEN[1:0] ビットは無視されます)。

レギュラ同時モードまたはインタリーブモードでは、マスタ ADC の ADSTART または ADSTP ビットを一度セットすると、スレーブ ADC の対応するビットも自動的にセットされます。ただし、スレーブ ADC の ADSTART または ADSTP ビットは、必ずしもマスタ ADC ビットと同時にクリアされません。

インジェクト同時モードまたはオルタネートトリガモードでは、マスタ ADC の JADSTART または JADSTP ビットを一度セットすると、スレーブ ADC の対応するビットも自動的にセットされます。ただし、スレーブ ADC の JADSTART または JADSTP ビットは、必ずしもマスタ ADC ビットと同時にクリアされません。

デュアル ADC モードでは、ADC 共通データレジスタ (ADC_CDR) を読み出すことで、マスタおよびスレーブ ADC の変換データを並列で読み出すことができます。ステータスビットも、デュアルモードステータスレジスタ (ADC_CSR) を読み出すことで、並列で読み出すことができます。

図 240. デュアル ADC ブロック図 (1)



1. スレーブ ADC にも外部トリガは存在しますが、この図には表示していません。
2. ADC 共通データレジスタ (ADC_CDR) にはマスタとスレーブの両方の ADC レギュラ変換データが格納されます。

インジェクト同時モード

このモードは、ビット DUAL[4:0] = 00101 をプログラムすることによって選択されます。

このモードは、インジェクトチャンネルグループを変換します。外部トリガのソースは、マスタ ADC のインジェクトグループマルチプレクサ (ADC_JSQR レジスタの JEXTSEL ビットによって選択) から供給されます。

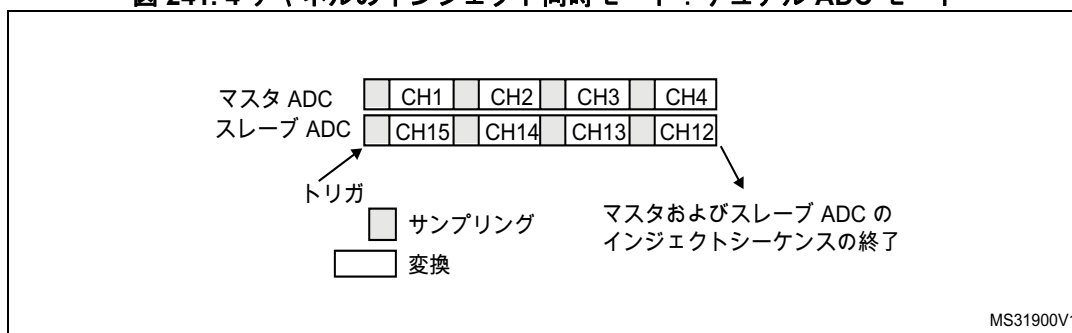
注： 2 つの ADC の同じチャンネルを変換しないでください (同じチャンネルを変換する場合、2 つの ADC のサンプリング時間が重なってはいけません)。

同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

レギュラ変換であれば、1 つまたはすべての ADC で行うことができます。その場合、それらは互いに独立していてインジェクトイベントが発生すると中断されます。それら (レギュラ変換) はインジェクト変換グループ終了時に再開されます。

- マスタ ADC での変換イベントのインジェクトシーケンスの終了時 (JEOS)、変換データはマスタ ADC_JDRy レジスタに格納され、JEOS 割込みが生成されます (有効な場合)。
- スレーブ ADC での変換イベントのインジェクトシーケンスの終了時 (JEOS)、変換データはスレーブ ADC_JDRy レジスタに格納され、JEOS 割込みが生成されます (有効な場合)。
- マスタインジェクトシーケンスの長さがスレーブインジェクトシーケンスの長さと同じ場合 (例：図 241)、ソフトウェアでは 2 つの JEOS 割込みのうちの 1 つのみを有効にすることができます (例：マスタ JEOS)、(マスタ ADC_JDRy とスレーブ ADC_JDRy レジスタの) 両方の変換データを読み出すことができます。

図 241. 4 チャンネルのインジェクト同時モード：デュアル ADC モード



JDISCEN = 1 の場合、インジェクトシーケンスの各同時変換にはインジェクトトリガイイベントが発生する必要があります。

このモードは、AUTDLY モードと組み合わせることができます。

- 変換の同時インジェクトシーケンスの終了後、マスタとスレーブADCの両方の JEOS ビットがクリアされた場合 (遅延フェーズ) にのみ、新しいインジェクトトリガイイベントは受け入れられます。実行中のインジェクトシーケンスで発生したすべての新しいインジェクトトリガイイベントと、これに関連する遅延フェーズは無視されます。
- マスタ ADC のレギュラ変換シーケンスが終了すると、マスタデータレジスタ (ADC_DR) が読み出された場合にのみ、マスタ ADC の新しいレギュラトリガイイベントが受け入れられます。実行中のレギュラシーケンスでマスタ ADC に対して発生したすべての新しいレギュラトリガイイベントと、これに関連する遅延フェーズは無視されます。スレーブ ADC で発生するレギュラシーケンスでも同じ挙動を示します。

独立インジェクト変換に対応したレギュラ同時モード

このモードは、ビット DUAL[4:0]=00110 をプログラムすることによって選択されます。

このモードは、レギュラチャンネルグループに対して行われます。外部トリガのソースは、マスタ ADC のレギュラグループマルチプレクサ (ADC_CFGR レジスタの EXTSEL ビットによって選択) から供給されます。同時トリガは、スレーブ ADC に供給されます。

このモードでは、独立インジェクト変換がサポートされます。(マスタまたはスレーブでの) インジェクションリクエストにより、現在の同時変換がアボートされます。これは、インジェクト変換の完了後に再開されます。

注： 2 つの ADC の同じチャンネルを変換しないでください (同じチャンネルを変換する場合、2 つの ADC のサンプリング時間が重なってはいけません)。

レギュラ同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

データが読み出せるタイミングは、割込みによってソフトウェアに通知されます。

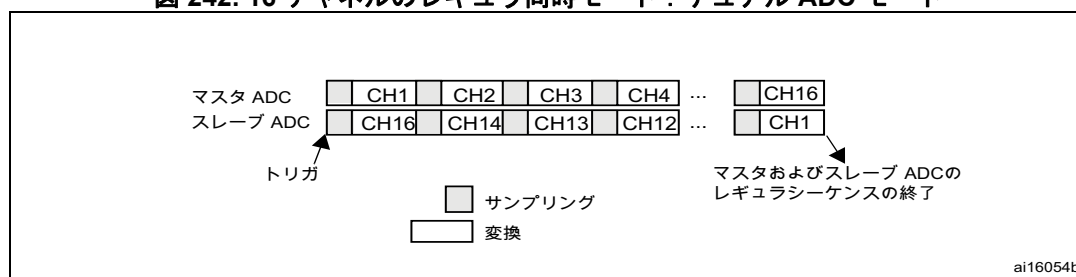
- マスタ ADC での各変換イベント (EOC) の終了時、マスタ EOC 割込みが生成され (EOCIE が有効な場合)、ソフトウェアではマスタ ADC の ADC_DR を読み出すことができます。
- スレーブ ADC での各変換イベント (EOC) の終了時、スレーブ EOC 割込みが生成され (EOCIE が有効な場合)、ソフトウェアではスレーブ ADC の ADC_DR を読み出すことができます。
- マスタレギュラシーケンスの長さがスレーブレギュラシーケンスの長さと同じ場合 (例：[図 242](#))、ソフトウェアでは 2 つの EOC 割込みのうちの 1 つのみを有効にすることができ (例：マスタ EOC)、共通データレジスタ (ADC_CDR) から両方の変換データを読み出すことができます。

DMA を使用してレギュラデータを読み出すことができます。次の 2 つの方法が可能です。

- 2 つの DMA チャンネルを使用します (1 つはマスタ用、もう 1 つはスレーブ用)。この場合、ビット MDMA[1:0] をクリア状態に保つ必要があります。
 - マスタから ADC_DR を読み出すために、DMA のマスタ ADC チャンネルを設定します。DMA リクエストは、マスタ ADC の各 EOC イベントで生成されます。
 - スレーブから ADC_DR を読み出すために、DMA のスレーブ ADC チャンネルを設定します。DMA リクエストは、スレーブ ADC の各 EOC イベントで生成されます。
- MDMA モードを使用して、他の用途のために 1 つの DMA チャンネルを残します。
 - MDMA[1:0] = 0b10 または 0b11 を設定します (分解能による)。
 - DMA チャンネルを 1 つ使用します (マスタのうちの 1 つ)。共通 ADC レジスタ (ADC_CDR) を読み出すために、DMA のマスタ ADC チャンネルを設定します。
 - マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、スレーブ ADC の変換データを ADC_CDR 32 ビットレジスタの上位ハーフワードで使用でき、またマスタ ADC の変換データを ADC_CDR レジスタの下位ハーフワードで使用できます。
 - EOC フラグは、DMA が ADC_CDR レジスタを読み出すと両方ともクリアされます。

注： MDMA モード (MDMA[1:0] = 0b10 または 0b11) で、ユーザはマスタのシーケンスとスレーブのシーケンスに同じ変換数をプログラムする必要があります。そうしないと、残りの変換では DMA リクエストは生成されません。

図 242. 16 チャンネルのレギュラ同時モード：デュアル ADC モード



DISCEN = 1 の場合、レギュラシーケンスの各「n」回の同時変換でレギュラトリガイイベントが発生する必要があります（「n」は DISCNUM で定義される）。

このモードは、AUTDLY モードと組み合わせることができます。

- シーケンスの同時変換の終了後、シーケンスの次の変換は、共通データレジスタ ADC_CDR（またはマスタ ADC のレギュラデータレジスタ）が読み出された場合（遅延フェーズ）にのみ開始されます。
- 変換の同時レギュラシーケンスの終了後、新しいレギュラトリガイイベントは共通データレジスタ（ADC_CDR）が読み出された場合（遅延フェーズ）にのみ受け入れられます。実行中のレギュラシーケンスで発生したすべての新しいレギュラトリガイイベントと、これに関連する遅延フェーズは無視されます。

AUTDLY モードと組み合わせたレギュラ同時モードでデータを処理するために、DMA を使用することができます（このとき、複数の DMA モードが使われるものとします）。MDMA ビットは 0b10 または 0b11 にセットする必要があります。

レギュラ同時モードを AUTDLY モードと組み合わせる場合、ユーザは次の内容を確認する必要があります。

- マスタのシーケンスの変換数が、スレーブの変換数と同じであること。
- シーケンスが同時変換されるたび、スレーブ ADC の変換の長さがマスタ ADC の変換の長さよりも短いこと。シーケンスの長さは、変換するチャンネルの数および各チャンネルのサンプリングタイムと分解能に依存する点に注意してください。

注： このレギュラ同時モードと AUTDLY モードの組み合わせは、レギュラチャンネルのみがプログラムされている場合に制限されます。この組み合わせモードでインジェクトチャンネルをプログラムすることは禁止されています。

独立インジェクト変換に対応したインタリーブモード

このモードは、ビット DUAL[4:0]=00111 をプログラムすることによって選択されます。

このモードは、レギュラグループ（通常は 1 つのチャンネル）でのみ開始できます。外部トリガソースは、マスタ ADC のレギュラチャンネルマルチプレクスから供給されます。

外部トリガが発生した後、

- マスタ ADC はすぐに開始します。
- スレーブ ADC は、マスタ ADC のサンプリングフェーズが完了した後で、数 ADC クロックサイクルの遅延後に開始します。

インタリーブモードにおける 2 つの変換間の最小遅延は ADC_CCR レジスタの DELAY ビットで設定されます。この遅延は、マスタ変換のサンプリングフェーズが終了してから半サイクル後にカウントを開始します。これにより、ADC は相補 ADC が依然として入力をサンプリングしている場合、変換を開始できません（与えられた時間に 1 つの ADC のみ入力信号をサンプリングすることができます）。

- DELAY に設定可能な最小値は 1 です。これにより、マスタ ADC サンプリングフェーズのアナログスイッチを開いてスレーブ ADC サンプリングフェーズのアナログスイッチを閉じるまでに、最低 1 つのサイクル時間を確保できます。
- 最大 DELAY 値は選択した分解能に対するサイクル数と同じです。ただし、ほかの ADC が依然として入力をサンプリングしているときに ADC が変換を開始しないよう、ユーザはこの遅延を適切に計算する必要があります。

マスタとスレーブの両方の ADC で CONT ビットがセットされている場合、両方の ADC の選択されたレギュラチャネルが連続的に変換されます。

スレーブ ADC での各変換の終了イベント (EOC) 時に、データを読み出せるタイミングが、割込みによってソフトウェアに通知されます。スレーブおよびマスタ EOC 割込みが生成され (EOCIE が有効な場合)、ソフトウェアではスレーブ/マスタ ADC の ADC_DR を読み出すことができます。

注： スレーブの EOC 割込みのみを有効にして共通データレジスタ (ADC_CDR) を読み出すことができます。ただしこの場合、ユーザは変換時間が適合しており、それによって新しいマスタ変換をリスタートする前に、シーケンス内では常にマスタ変換に続いてスレーブ変換が実行されることを確認する必要があります。MDMA モードを使用することを推奨します。

DMA で転送されたレギュラデータを持つこともできます。この場合、各 ADC の個別の DMA リクエストを使用できず、次のように MDMA モードの使用が必須になります。

- MDMA[1:0] = 0b10 または 0b11 を設定します (分解能による)。
- DMA チャンネルを 1 つ使用します (マスタのうちの 1 つ)。共通 ADC レジスタ (ADC_CDR) を読み出すために、DMA のマスタ ADC チャンネルを設定します。
- マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、スレーブ ADC の変換データを ADC_CDR 32 ビットレジスタの上位ハーフワードで使用でき、またマスタ ADC の変換データを ADC_CCR レジスタの下位ハーフワードで使用できます。
- EOC フラグは、DMA が ADC_CCR レジスタを読み出すと両方ともクリアされます。

図 243. 連続変換モードにおける 1 チャンルのインタリーブモード：デュアル ADC モード

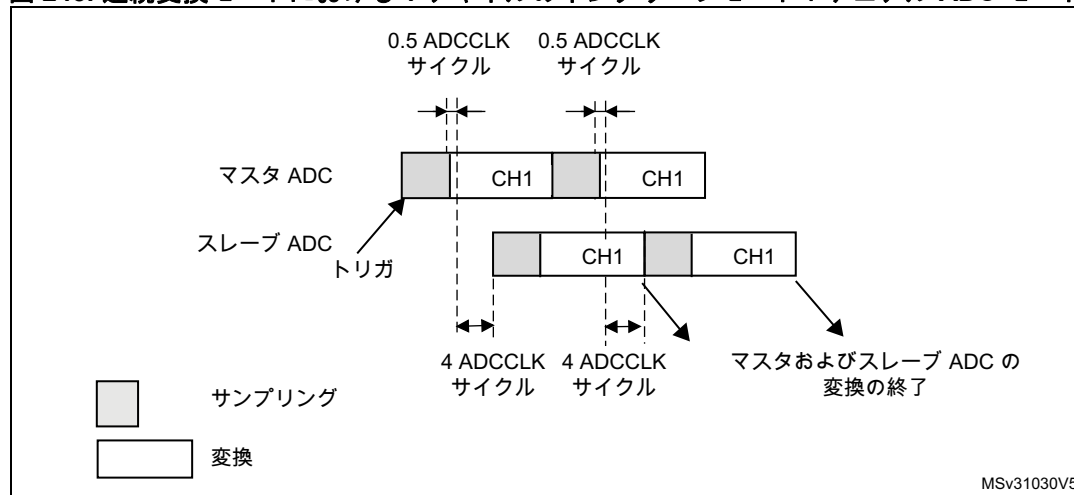
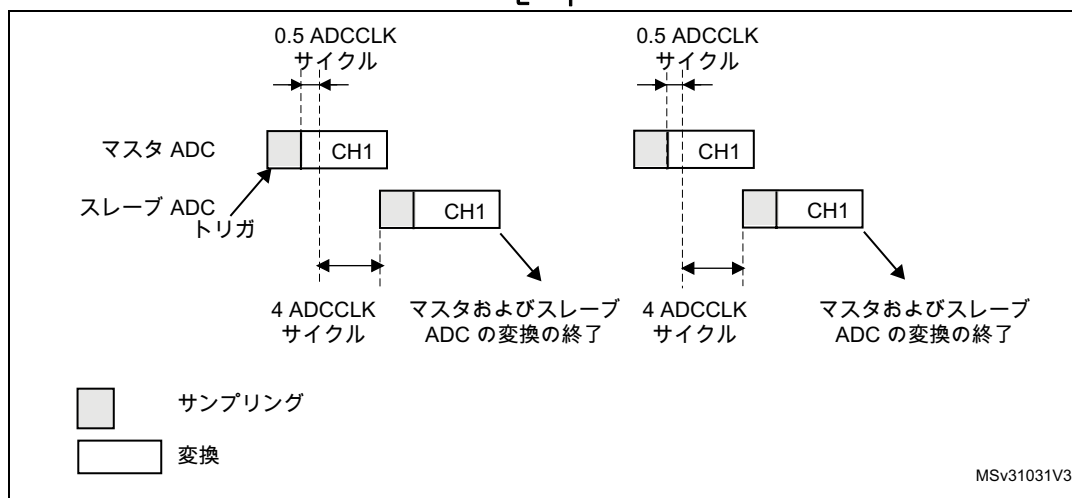


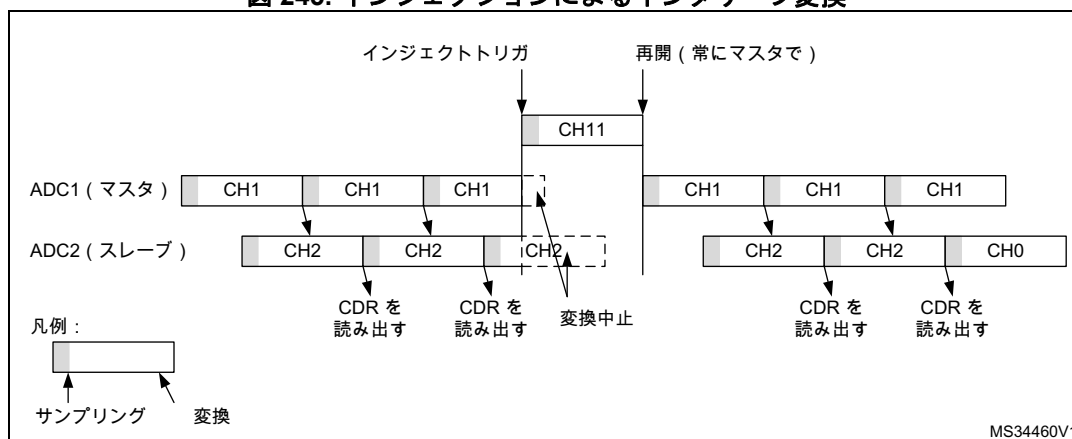
図 244. シングル変換モードにおける 1 チャンネルのインタリーブモード：デュアル ADC モード



DISCEN = 1 の場合、レギュラシーケンスの各「n」回の同時変換（「n」は DISCNUM で定義される）でレギュラトリガイイベントが発生する必要があります。

このモードでは、インジェクト変換がサポートされます。（マスタまたはスレーブでの）インジェクションが完了すると、マスタとスレーブの両方のレギュラ変換がアボートされ、マスタからシーケンスが再開されます（以下の図 245 を参照）。

図 245. インジェクションによるインタリーブ変換



オルタネートトリガモード

このモードは、ビット DUAL[4:0]=01001 をプログラムすることによって選択されます。

このモードは、インジェクトグループでのみ開始できます。外部トリガのソースは、マスタ ADC のインジェクトグループマルチプレクサです。

このモードは、ハードウェアトリガの選択時にのみ使用できます。JEXTEN を 0x0 にセットすることはできません。

インジェクト不連続モードは無効です (両方の ADC で JDISCEN = 0)。

1. 最初のトリガが発生すると、グループ内のすべてのインジェクトされたマスタ ADC チャンネルが変換されます。
2. 2 番目のトリガが発生すると、グループ内のすべてのインジェクトされたスレーブ ADC チャンネルが変換されます。
3. 以下同様です。

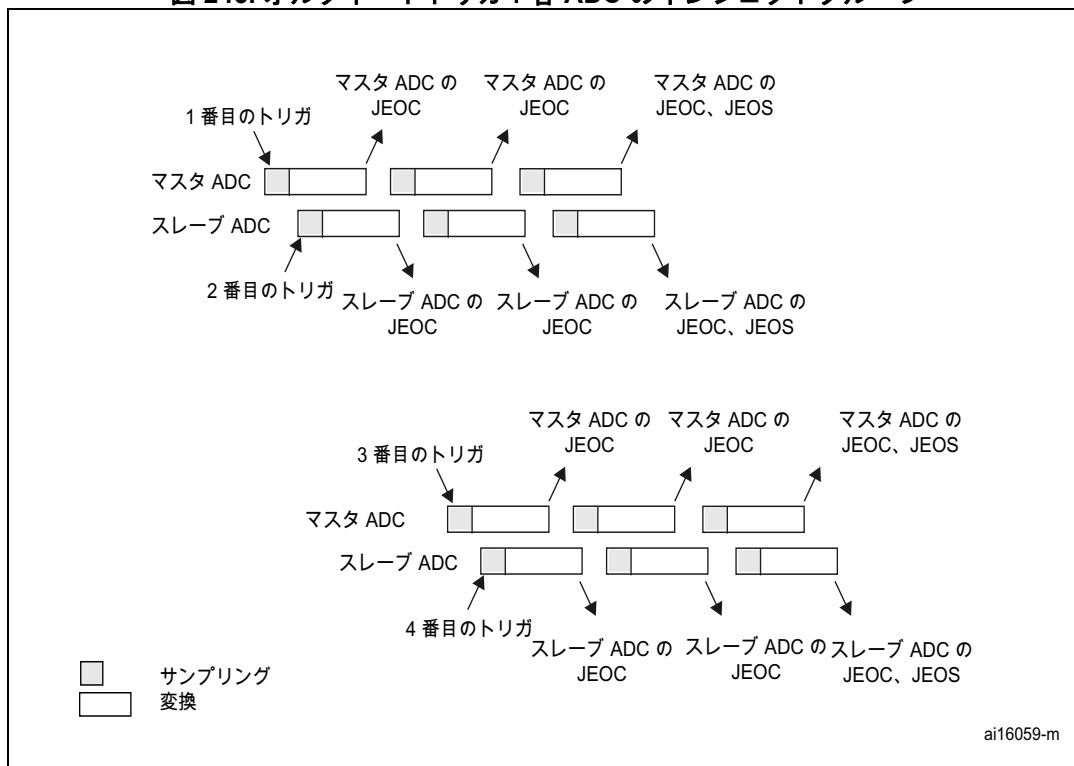
グループ内のマスタ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割込み (有効な場合) が生成されます。

グループ内のスレーブ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割込み (有効な場合) が生成されます。

JEOC 割込み (有効な場合) は、各インジェクト変換後に生成することもできます。

グループ内すべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、グループ内のマスタ ADC のインジェクトチャンネルを変換することによって、オルタネートトリガプロセスがリスタートします。

図 246. オルタネートトリガ：各 ADC のインジェクトグループ



注：

レギュラ変換であれば 1 つまたはすべての ADC で有効にすることができます。この場合、レギュラ変換は互いに独立しています。ADC がインジェクト変換を行う必要があるとき、レギュラ変換は中断されます。インジェクト変換が終了すると(レギュラ変換が)再開されます。

2 つのトリガイベント間の間隔は 1 ADC クロック期間と同等か長い必要があります。同じ ADC で変換を開始する 2 つのトリガイベント間の最小間隔はシングル ADC モードと同じです。

インジェクト不連続モードは有効です (両方の ADC で JDISCEN = 1)。

マスタとスレーブの両方の ADC に対してインジェクト不連続モードが有効な場合、

- 最初のトリガが発生すると、マスタ ADC の最初のインジェクトチャンネルが変換されます。
- 2 番目のトリガが発生すると、スレーブ ADC の最初のインジェクトチャンネルが変換されます。
- 以下同様です。

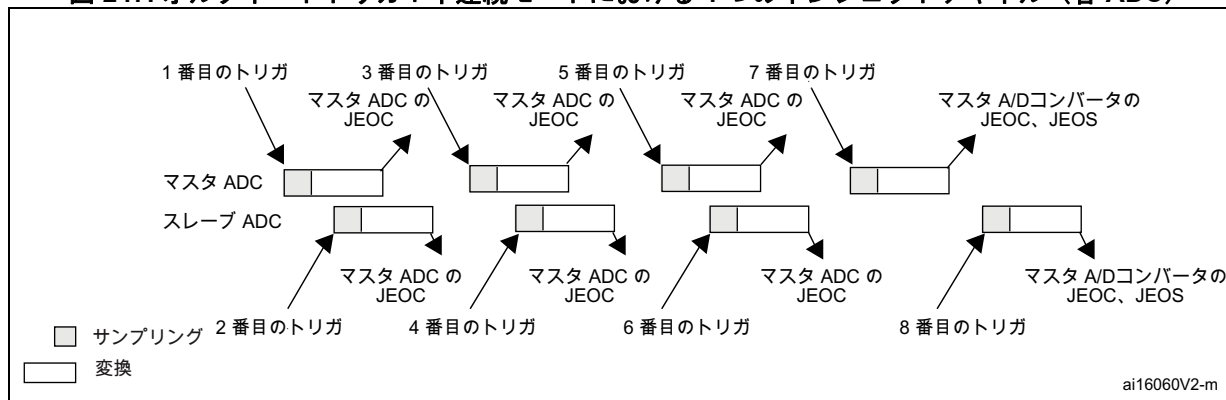
グループ内のマスタ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割込み (有効な場合) が生成されます。

グループ内のスレーブ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割込み (有効な場合) が生成されます。

JEOC 割込み (有効な場合) は、各インジェクト変換後に生成することもできます。

グループ内のすべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、オルタネートトリガプロセスがリスタートします。

図 247. オルタネートトリガ：不連続モードにおける 4 つのインジェクトチャンネル (各 ADC)



レギュラ/インジェクト同時モードの組み合わせ

このモードは、ビット DUAL[4:0]=00001 をプログラムすることによって選択されます。

レギュラグループの同時変換を中断して、インジェクトグループの同時変換を開始することができます。

注： レギュラ/インジェクト同時モードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

レギュラ同時 + オルタネートトリガモードの組み合わせ

このモードは、ビット DUAL[4:0] = 00010 をプログラムすることによって選択されます。

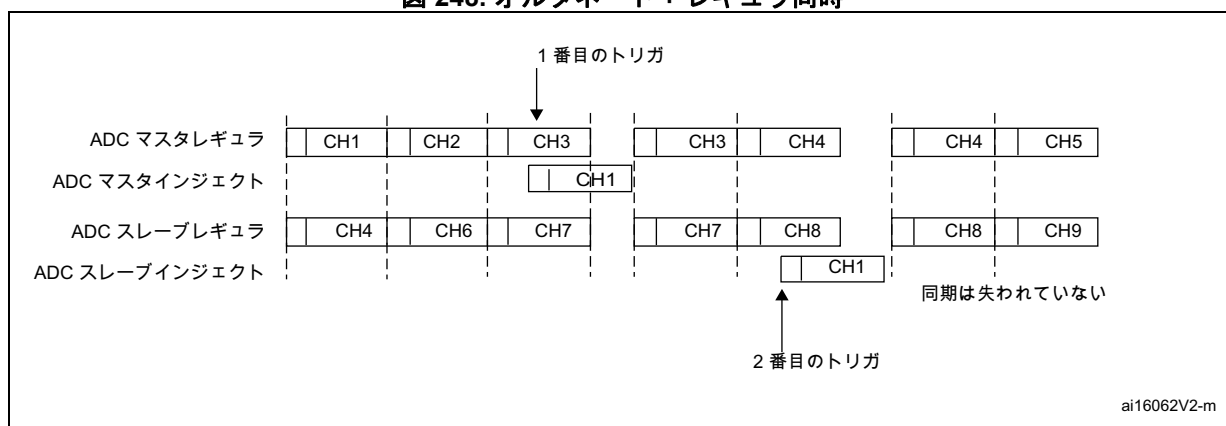
レギュラグループの同時変換を中断して、インジェクトグループのオルタネートトリガ変換を開始することができます。図 248 は、同時レギュラ変換に割り込むオルタネートトリガの動作を示します。

インジェクトオルタネート変換は、インジェクトイベント後、ただちに開始されます。すでにレギュラ変換が実行中の場合、インジェクト変換後の同期を確保するために、すべての (マスタ/スレーブ) ADC のレギュラ変換は停止し、インジェクト変換の終了と同期して再開されます。

注： レギュラ同時 + オルタネートトリガモードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければ

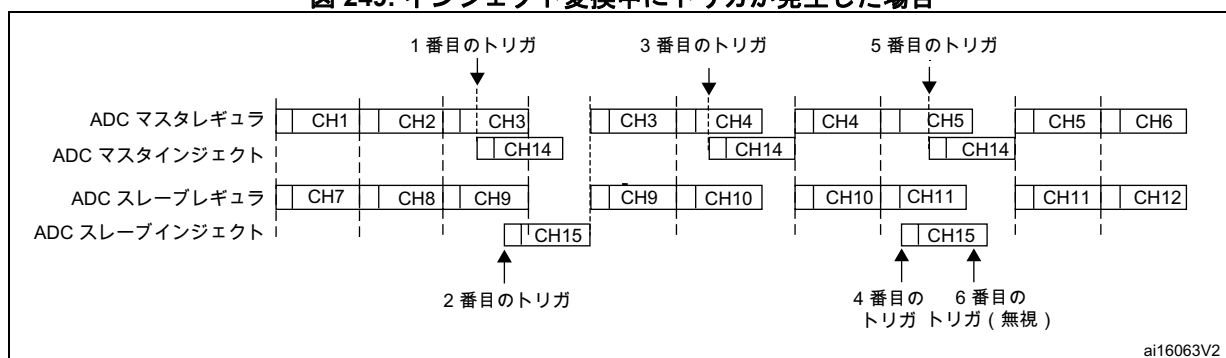
ばなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

図 248. オルタネート + レギュラ同時



レギュラ変換に割り込んだインジェクト変換の最中にトリガが発生した場合はオルタネートトリガが処理されます。図 249 に、この場合の挙動を示します（関連付けられたオルタネート変換は完了していないため、6 番目のトリガは無視されます）。

図 249. インジェクト変換中にトリガが発生した場合



インジェクト同時モードとインタリーブの組み合わせ

このモードは、ビット DUAL[4:0] = 00011 をプログラムすることによって選択されます。

インタリーブ変換に同時インジェクトイベントでの割り込みが可能です。

この場合、インタリーブ変換は即座に割り込まれ、同時インジェクト変換が開始します。インジェクトシーケンス終了時に、インタリーブ変換が再開されます。インタリーブレギュラ変換が再開すると、実行される最初のレギュラ変換は常にマスタのものです。図 250、図 251 および図 252 には、例を使用して動作を示しています。

注意： このモードでは、1 回の読出しアクセスでレギュラデータを読み出すために共通データレジスタを使用する必要があります。逆に、マスタ - スレーブのデータコヒーレンスは保証されません。

図 250. インジェクトシーケンス CH11、CH12 とインタリーブシングルチャネル CH 0

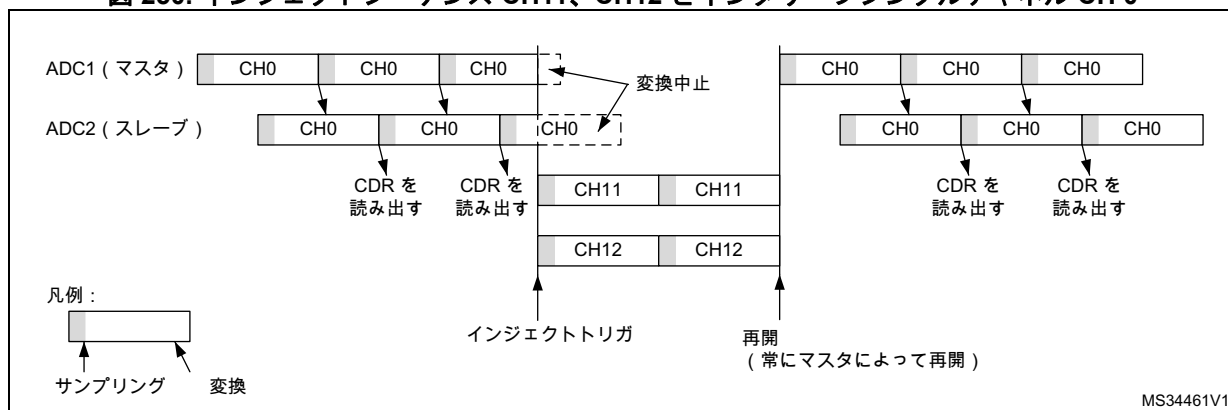


図 251. インジェクトシーケンス CH11、CH12 と 2 つのインタリーブチャネル (CH1、CH2) - ケース 1: マスタ割込み優先

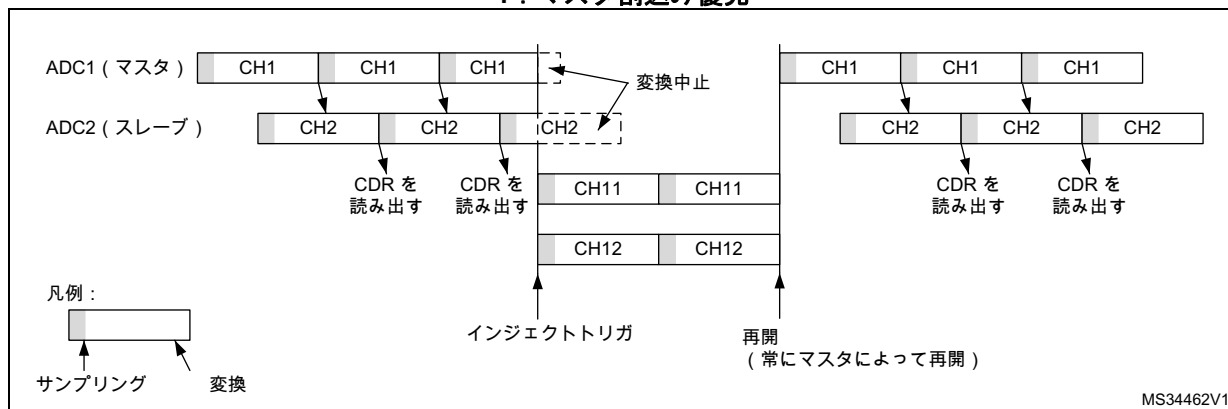
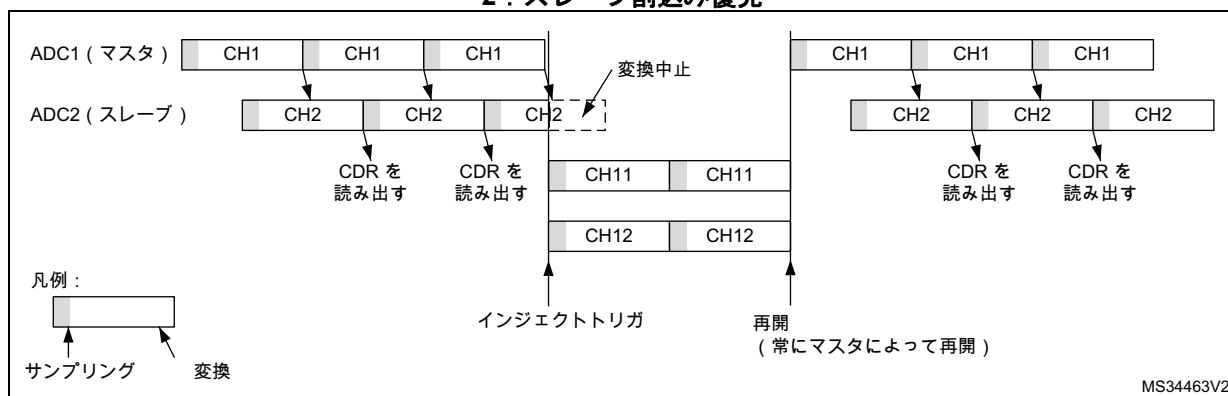


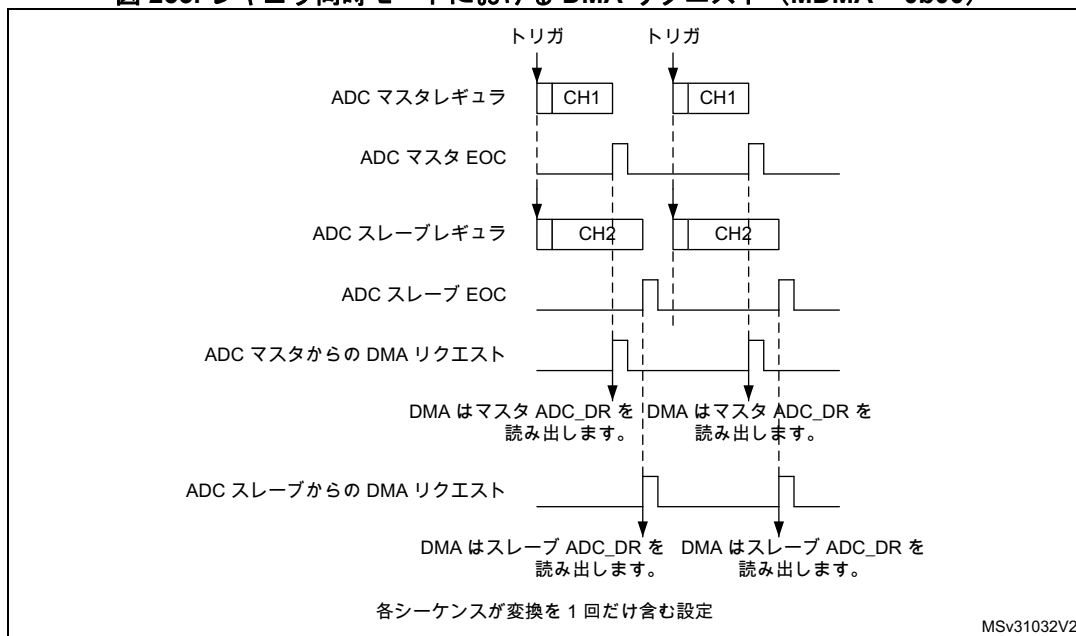
図 252. インジェクトシーケンス CH11、CH12 と 2 つのインタリーブチャネル (CH1、CH2) - ケース 2: スレーブ割込み優先



デュアル ADC モードにおける DMA リクエスト

すべてのデュアル ADC モードでは、シングルモードの場合と同様、データを転送するために 2 つの DMA チャンネル（マスタ用 1 つとスレーブ用 1 つ）を使用することができます（図 253：レギュラ同時モードにおける DMA リクエスト（MDMA = 0b00）を参照）。

図 253. レギュラ同時モードにおける DMA リクエスト（MDMA = 0b00）



レギュラ同時モードおよびインタリーブモードでは、1 つの DMA チャンネルを確保し、シングル DMA チャンネルを使用して両方のデータを転送することもできます。そのためには、ADC_CCR レジスタで MDMA ビットを設定する必要があります。

- MDMA = 0b10:** マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、2 つのデータアイテムが使用可能です。また、32 ビットレジスタ ADC_CDR には、2 つの ADC で変換されたデータアイテムを表す 2 つのハーフワードが含まれます。スレーブ ADC データは上位ハーフワードを使用し、マスタ ADC データは下位ハーフワードを使用します。
 このモードは、分解能が 10 ビットまたは 12 ビットの場合に、インタリーブモードおよびレギュラ同時モードで使用されます。

例：

インタリーブデュアルモード：2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

最初の DMA リクエスト：ADC_CDR[31:0] = SLV_ADC_DR[15:0] | MST_ADC_DR[15:0]

2 番目の DMA リクエスト：ADC_CDR[31:0] = SLV_ADC_DR[15:0] | MST_ADC_DR[15:0]

図 254. レギュラ同時モードにおける DMA リクエスト (MDMA = 0b10)

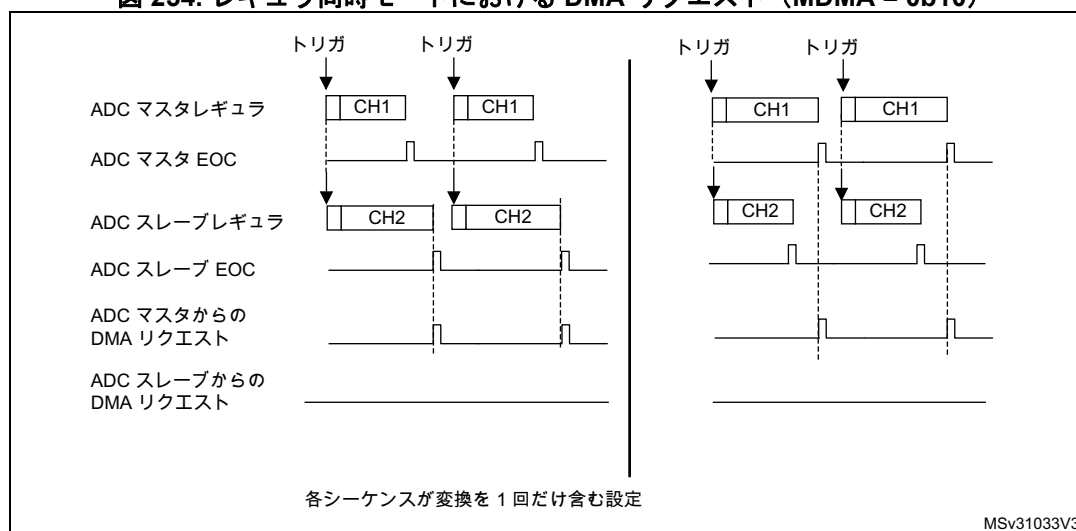
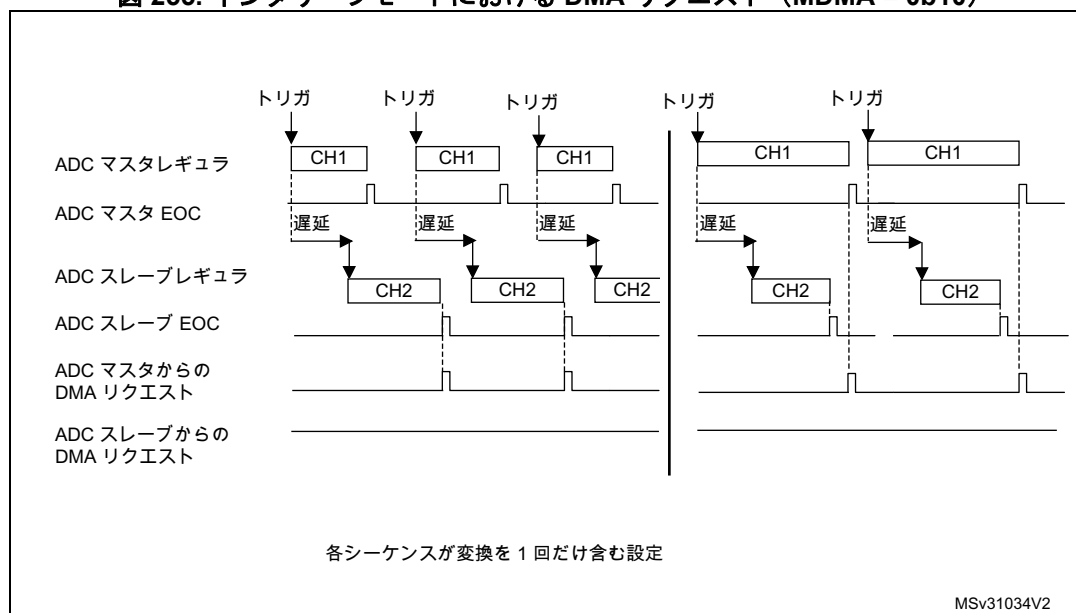


図 255. インタリーブモードにおける DMA リクエスト (MDMA = 0b10)



注: MDMA モードを使用する場合は、マスタおよびスレーブ変換時間を適切に設定するよう注意する必要があります。これにより、新しい変換が使用可能になる前に DMA リクエストを生成して両方のデータ (マスタ + スレーブ) の読出しを処理できます。

- **MDMA = 0b11**: このモードは MDMA = 0b10 に類似しています。唯一の違いは、DMA リクエストごとに (データ項目は 2 つ使用可能)、2 つの ADC 変換データを表す 2 バイトがハーフワードとして転送されます。

このモードは、分解能が 6 ビットまたは 8 ビットでデータが符号なしである場合に、インタリーブモードおよびレギュラ同時モードで使用されます (関連するすべてのチャンネルについて、オフセットは無効である必要があります)。

例:

インタリーブデュアルモード: 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

最初の DMA リクエスト: `ADC_CDR[15:0] = SLV_ADC_DR[7:0] | MST_ADC_DR[7:0]`

2 番目の DMA リクエスト: `ADC_CDR[15:0] = SLV_ADC_DR[7:0] | MST_ADC_DR[7:0]`

オーバーラン検出

デュアル ADC モード (DUAL[4:0] が b00000 以外の場合) で、ADC のうちの 1 つにオーバーランが検出された場合、RAM に転送されたすべてのデータが有効であることを保証するため DMA リクエストの発行を停止します (この動作は MDMA の設定にかかわらず発生します)。ある ADC のデータレジスタに有効なデータが格納されていることによりその ADC に対応する EOC ビットがセットされたままになることがあります。

MDMA モード選択時の DMA ワンショットモード/DMA サーキュラモード

MDMA モードが選択されている場合 (0b10 または 0b11)、[セクション: DMA を使用した変換の管理](#)で説明しているように、DMA ワンショットモード/サーキュラモードのどちらかを選択するために ADC_CCR レジスタの DMACFG ビット も設定する必要があります。(マスタおよびスレーブ ADC_CFGR の DMACFG ビットは無関係です)。

デュアル ADC モードにおける変換の停止

デュアル ADC モードにおける ADC の変換を両方とも停止するには、マスタ ADC の制御ビットに ADSTP/JADSTP をセットする必要があります。デュアル ADC モードには、スレーブ ADC のその他の ADSTP 制御ビットの効果はありません。

両方の ADC が実質的に無効になると、マスタおよびスレーブ ADC の両方の ADSTART/JADSTART ビット がハードウェアによってクリアされます。

26.4.31 温度センサ

温度センサを使用して、デバイスの接合温度 (Tj) を測定できます。

温度センサは、センサの出力電圧をデジタル値に変換するために ADC 入力チャンネルに内部接続されます (詳細については、[セクション 26.4.2: ADC ピンおよび内部信号](#)の ADC 相互接続の表を参照)。使用しないときには、センサをパワーダウンモードにすることができます。サポートする温度範囲は -40 ~ 125 °C です。

[図 256](#) に、温度センサおよび ADC 間の接続のブロック図を示します。

温度センサの出力電圧は、温度に比例して変化します。このラインのオフセットは、プロセスのばらつきにより、チップごとに異なります (チップ間で最大 45 °C)。

較正されていない内部温度センサは、温度の絶対値の代わりに温度変化を検出するアプリケーションに適しています。温度センサの測定精度を高めるために、生産時に ST によって各デバイスの較正値がシステムメモリに格納されています。

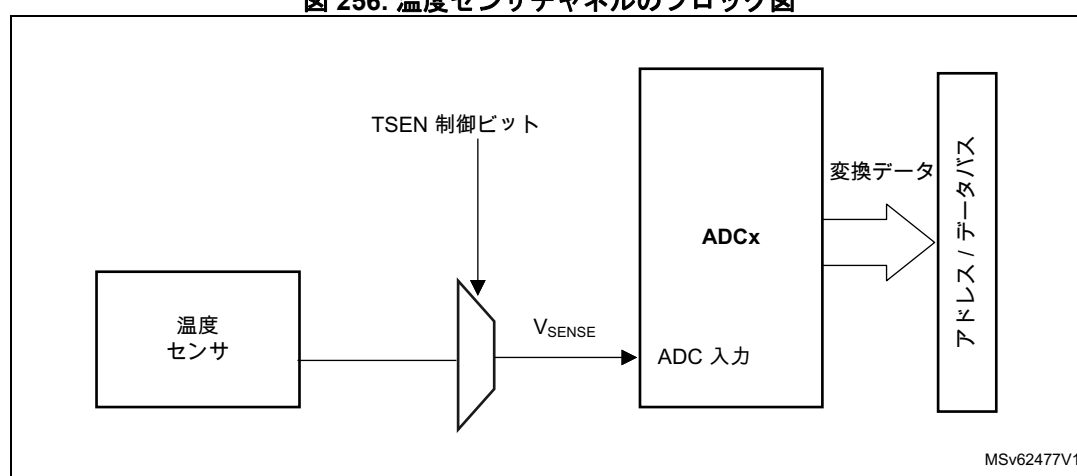
製造プロセス中に、温度センサと内部電圧基準の較正データがシステムメモリ領域に格納されます。ユーザアプリケーションはこれらを読み出して、温度センサまたは内部基準の精度の向上に使用できます（詳しい情報はデータシートを参照してください。）。

温度センサは、センサの出力電圧をデジタル値に変換するために ADC 入力チャネルに内部接続されます。内部温度センサの変換時に適用するサンプリング時間の値については、デバイスデータシートの電気的特性のセクションを参照してください。

使用しないときには、センサをパワーダウンモードにすることができます。

図 256 に、温度センサのブロック図を示します。

図 256. 温度センサチャネルのブロック図



温度の読出し

温度センサを使用するには：

1. V_{SENSE} に接続する ADC 入力チャネルを選択します。
2. 適切なサンプリング時間をプログラムします (デバイスデータシートの電気的特性のセクションを参照)。
3. ADC_CCR レジスタにビットをセットして、温度センサをパワーダウンモードからウェイクアップします。
4. ADC 変換を開始します。
5. A/D コンバータデータレジスタの V_{SENSE} データを読み出します。
6. 次の式を使用して、実際の温度を計算します。

$$\text{Temperature (in } ^\circ\text{C)} = \frac{\text{TS_CAL2_TEMP} - \text{TS_CAL1_TEMP}}{\text{TS_CAL2} - \text{TS_CAL1}} \times (\text{TS_DATA} - \text{TS_CAL1}) + \text{TS_CAL1_TEMP}$$

ここで、

- TS_CAL2 は、TS_CAL2_TEMP で得られた温度センサの較正值です。
- TS_CAL1 は、TS_CAL1_TEMP で得られた温度センサの較正值です。
- TS_DATA は、ADC によって変換された実際の温度センサの出力値です。

TS_CAL1 および TS_CAL2 較正ポイントの詳細については、デバイスデータシートを参照してください。

注： センサがパワーダウンモードからウェイクアップして、正しいレベルで V_{SENSE} を出力できるようになるまでには時間がかかります（スタートアップ時間）。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADEN とビットを同時にセットしてください。

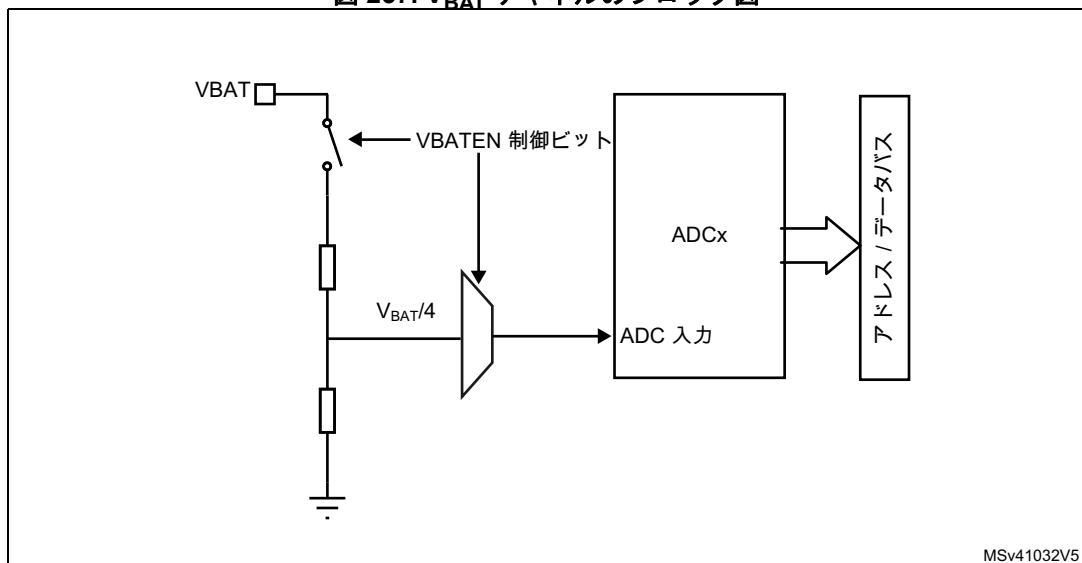
26.4.32 V_{BAT} 電源監視

ADC_CCR レジスタの VBATEN ビットはバッテリー電圧に切り替えるために使用します。 V_{BAT} 電圧は V_{DDA} より高くなることがあるので、ADC の正しい動作を確保するために、 V_{BAT} ピンはブリッジ 4 分圧回路に内部接続されています。このブリッジは、VBATEN ビットがセットされると自動的に有効になり、 $V_{BAT}/4$ を ADC 入力チャンネルに接続します（詳細については、[セクション 26.4.2: ADC ピンおよび内部信号](#)の ADC 相互接続の表を参照）。結果として、変換されたデジタル値は V_{BAT} 電圧の 1/3 です。不要なバッテリー消費を避けるには、ADC 変換を行うときだけ、ブリッジ分圧回路を有効にすることが推奨されます。

$V_{BAT}/4$ 電圧の変換時に適用するサンプリング時間の値については、デバイスデータシートの電気的特性を参照してください。

[図 257](#) に、 V_{BAT} センシング機能のブロック図を示します。

図 257. V_{BAT} チャンネルのブロック図



1. $V_{BAT}/4$ の内部チャンネルの変換を有効にするには、VBATEN ビットをセットする必要があります。

26.4.33 内部電圧基準の監視

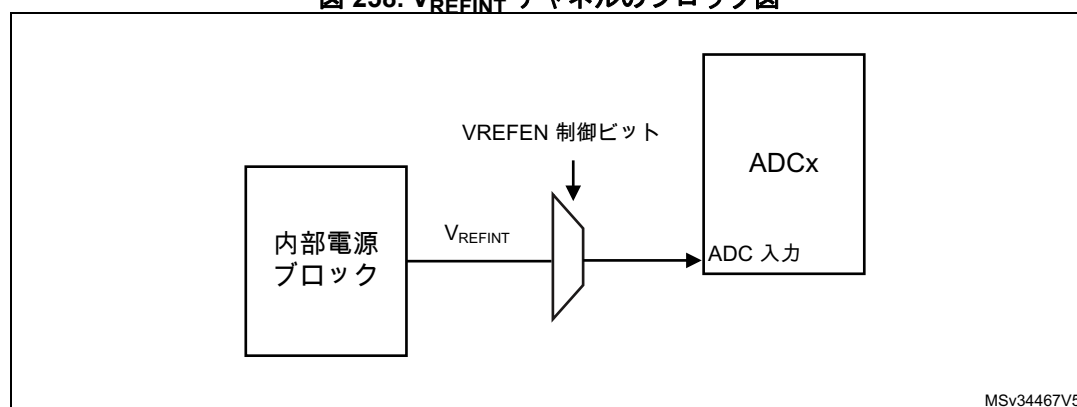
ADC V_{REF+} 電圧レベルを評価するための基準点を得るために、内部電圧基準 (V_{REFINT}) を監視することができます。

内部電圧基準が内部的に接続されている ADC 入力チャネルの詳細については、[セクション 26.4.2 : ADC ピンおよび内部信号](#)の ADC 相互接続の表を参照してください。

内部電圧基準の変換時に適用するサンプリング時間の値については、製品のデータシートの電気的特性のセクションを参照してください。

[図 258](#) に、 V_{REFINT} センシング機能のブロック図を示します。

図 258. V_{REFINT} チャネルのブロック図



1. 内部チャネル (V_{REFINT}) の変換を有効にするには、ADC_CCR レジスタに VREFEN ビットをセットする必要があります。

内部基準電圧を使用した実際の V_{REF+} 電圧の計算

V_{REF+} 電圧は、ばらつきがあり、または正確にはわからないことがあります。内蔵の内部基準電圧 (V_{REFINT}) と、製造プロセス時に V_{REF+_charac} で ADC によって得られた較正データを使用して、実際の V_{REF+} 電圧レベルを評価することができます。

デバイスに印加される実際の V_{REF+} 電圧は、次の式で求められます。

$$V_{REF+} = V_{REF+_Charac} \times VREFINT_CAL / VREFINT_DATA$$

ここで、

- V_{REF+_Charac} は、製造プロセス時に V_{REFINT} で特徴付けられた V_{REF+} 電圧の値です。デバイスのデータシートで指定されています。
- VREFINT_CAL は、VREFINT の較正值です。
- VREFINT_DATA は、ADC によって変換された実際の VREFINT の出力値です。

電源に相対的な ADC 測定値から絶対電圧値への変換

ADC は、 V_{REF+} と、変換されるチャネルに印加される電圧との比に対応するデジタル値を提供するように設計されています。

V_{REF} 値が未知で、ADC によって変換された値が右詰めになっているアプリケーション。この場合、この比を V_{REF} に依存しない電圧に変換する必要があります。

$$V_{CHANNELx} = \frac{V_{REF+}}{FULL_SCALE} \times ADC_DATA$$

V_{REF+} を上記の式で置き換えることによって、絶対電圧値を次の計算式で求めることができます。

$$V_{CHANNELx} = \frac{V_{REF+_Charac} \times VREFINT_CAL \times ADC_DATA}{VREFINT_DATA \times FULL_SCALE}$$

ここで、

- V_{REF+_Charac} は、製造プロセス時に V_{REFINT} で特徴付けられた V_{REF+} 電圧の値です。
- $VREFINT_CAL$ は、 $VREFINT$ の較正值です。
- ADC_DATA は、チャンネル x で ADC によって測定された値です（右詰め）。
- $VREFINT_DATA$ は、ADC によって変換された実際の $VREFINT$ の出力値です。
- $FULL_SCALE$ は、ADC 出力の最大デジタル値です。たとえば、12 ビット分解能では、 $2^{12} - 1 = 4095$ であり、8 ビット分解能では $2^8 - 1 = 255$ です。

注： ADC 測定が 12 ビット右詰め以外の出力形式を使用して行われる場合、計算を行う前に、すべてのパラメータを互換性のある形式に変換しておく必要があります。

26.4.34 電源電圧の監視

ADC2 は内部電源電圧に接続されています。ADC を使用してこの電圧を測定するには、ADC オプションレジスタを使って接続を有効にします。

26.5 ADC 割込み

各 ADC に、割込みを生成することができます。

- ADC パワーアップ後、ADC の準備ができたとき (ADRDY フラグ)
- レギュラグループの任意の変換終了時 (EOC フラグ)
- レギュラグループの任意の変換シーケンス終了時 (EOS フラグ)
- インジェクトグループの任意の変換終了時 (JEOC フラグ)
- インジェクトグループの任意の変換シーケンス終了時 (JEOS フラグ)
- アナログウォッチドッグ検出の発生時 (AWD1、AWD2、および AWD3 フラグ)
- サンプリングフェーズ終了の発生時 (EOSMP フラグ)
- データオーバーランの発生時 (OVR フラグ)
- インジェクトシーケンスのコンテキストキューのオーバーフロー時 (JQOVF フラグ)

高い柔軟性を実現するため、個別の割込みイネーブルビットを使用できます。

表 252. ADC 割込み

割込み ベクタ	割込みイベント	イベント フラグ	イネーブル 制御ビット	割込みの クリア方法	SLEEP モードの 終了	STOP および STANDBY モードの 終了
ADC	ADC レディ	ADRDY	ADRDYIE	ハードウェアに よって セットされ、 ソフトウェアに よってクリア されます。	可能	不可
	レギュラグループの変換終了	EOC	EOCIE			
	レギュラグループの変換シー ケンスの終了	EOS	EOSIE			
	インジェクトグループの変換 終了	JEOC	JEOCIE			
	インジェクトグループの変換 シーケンスの終了	JEOS	JEOSIE			
	アナログウォッチドッグ 1 の ステータスビットのセット	AWD1	AWD1IE			
	アナログウォッチドッグ 2 の ステータスビットのセット	AWD2	AWD2IE			
	アナログウォッチドッグ 3 の ステータスビットのセット	AWD3	AWD3IE			
	サンプリングフェーズの終了	EOSMP	EOSMPIE			
	オーバーラン	OVR	OVRIE			
	インジェクトコンテキスト キューのオーバーフロー	JQOVF	JQOVFIE			

26.6 ADC レジスタ (ADC ごと)

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

26.6.1 ADC 割込みおよびステータスレジスタ (ADC_ISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **JQOVF** : インジェクトコンテキストキューのオーバーフロー

このビットは、コンテキストのインジェクトキューのオーバーフローが発生したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。詳細については、[セクション 26.4.21 : インジェクト変換のコンテキストのキュー](#)を参照してください。

0 : インジェクトコンテキストキューのオーバーフローは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1 : インジェクトコンテキストキューのオーバーフローが発生しました。

ビット 9 **AWD3** : アナログウォッチドッグ 3 フラグ

このビットは、変換された電圧が、ADC_TR3 レジスタのフィールド LT3[7:0] および HT3[7:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : アナログウォッチドッグ 3 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1 : アナログウォッチドッグ 3 イベントが発生しました。

ビット 8 **AWD2** : アナログウォッチドッグ 2 フラグ

このビットは、変換された電圧が、ADC_TR2 レジスタのフィールド LT2[7:0] および HT2[7:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : アナログウォッチドッグ 2 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1 : アナログウォッチドッグ 2 イベントが発生しました。

ビット 7 **AWD1** : アナログウォッチドッグ 1 フラグ

このビットは、変換された電圧が、ADC_TR1 レジスタのフィールド LT1[11:0] および HT1[11:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : アナログウォッチドッグ 1 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1 : アナログウォッチドッグ 1 イベントが発生しました。

ビット 6 JEOS : インジェクトチャネルのシーケンス終了フラグ

このビットは、グループ内のすべてのインジェクトチャネル変換が終了したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : インジェクト変換シーケンスは完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : インジェクト変換シーケンスが完了しました。

ビット 5 JEOC : インジェクトチャネルの変換終了フラグ

このビットは、チャネルの各インジェクト変換の終了時に、新しいデータが対応する ADC_JDRy レジスタで使用可能になったときに、ハードウェアによってセットされます。ソフトウェアによって 1 を書き込むことによって、または対応する ADC_JDRy レジスタを読み出すことによってクリアされます。

0 : インジェクトチャネル変換は完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : インジェクトチャネル変換が完了しました。

ビット 4 OVR : ADC オーバーラン

このビットは、レギュラチャネルでオーバーランが発生したときにハードウェアによってセットされ、EOC フラグがすでにセットされているときに新しい変換が完了したことを意味します。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : オーバーランは発生していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : オーバーランが発生しました。

ビット 3 EOS : レギュラシーケンス終了フラグ

このビットは、レギュラチャネルシーケンスの変換が終了したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : レギュラ変換シーケンスは完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : レギュラ変換シーケンスが完了しました。

ビット 2 EOC : 変換終了フラグ

このビットは、チャネルの各レギュラ変換の終了時に、新しいデータが ADC_DR レジスタで使用可能になったときに、ハードウェアによってセットされます。ソフトウェアによって 1 を書き込むことによって、または ADC_DR レジスタを読み出すことによってクリアされます。

0 : レギュラチャネル変換は完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : レギュラチャネル変換が完了しました。

ビット 1 EOSMP : サンプリング終了フラグ

このビットは、任意のチャネル (レギュラチャネルのみ) の変換中、サンプリングフェーズの終了時にハードウェアによってセットされます。

0 : サンプリングフェーズは終了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : サンプリングフェーズは終了しました。

ビット 0 ADRDY : ADC レディ

このビットは、ADC が有効にされた後 (ADEN = 1)、ADC が変換リクエストを受け入れる準備ができた状態に達したときに、ハードウェアによってセットされます。

ソフトウェアで 1 を書き込むことによってクリアされます。

0 : ADC はまだ変換を開始する準備ができていません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : ADC は変換を開始する準備ができました。

26.6.2 ADC 割込み有効レジスタ (ADC_IER)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **JQOVFIE** : インジェクトコンテキストキューのオーバーフロー割込みイネーブル

このビットは、インジェクトコンテキストキューのオーバーフローの割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : インジェクトコンテキストキューのオーバーフロー割込みは無効です。

1 : インジェクトコンテキストキューのオーバーフロー割込みは有効です。JQOVF ビットがセットされると、割込みが生成されます。

注 : ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 9 **AWD3IE** : アナログウォッチドッグ 3 割込みイネーブル

このビットは、アナログウォッチドッグ 2 割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグ 3 割込みは無効です。

1 : アナログウォッチドッグ 3 割込みは有効です。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 8 **AWD2IE** : アナログウォッチドッグ 2 割込みイネーブル

このビットは、アナログウォッチドッグ 2 割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグ 2 割込みは無効です。

1 : アナログウォッチドッグ 2 割込みは有効です。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 7 **AWD1IE** : アナログウォッチドッグ 1 割込みイネーブル

このビットは、アナログウォッチドッグ 1 割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグ 1 割込みは無効です。

1 : アナログウォッチドッグ 1 割込みは有効です。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 6 JEOSIE : インジェクト変換シーケンス終了割り込みイネーブル

このビットは、インジェクト変換シーケンス終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : JEOS 割り込み無効

1 : JEOS 割り込みは有効です。JEOS ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 5 JEOCIE : インジェクト変換終了割り込みイネーブル

このビットは、インジェクト変換終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : JEOC 割り込みは無効です。

1 : JEOC 割り込みは有効です。JEOC ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 4 OVRIE : オーバーラン割り込みイネーブル

このビットは、レギュラ変換のオーバーラン割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : オーバーラン割り込みは無効です。

1 : オーバーラン割り込みは有効です。OVR ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 3 EOSIE : レギュラ変換シーケンス終了割り込みイネーブル

このビットは、レギュラ変換シーケンス終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOS 割り込み無効

1 : EOS 割り込みは有効です。EOS ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 2 EOCIE : レギュラ変換終了割り込みイネーブル

このビットは、レギュラ変換終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOC 割り込みは無効です。

1 : EOC 割り込みは有効です。EOC ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 1 EOSMPIE : レギュラ変換のサンプリング終了フラグ割り込みイネーブル

このビットは、レギュラ変換のサンプリングフェーズ終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOSMP 割り込みは無効です。

1 : EOSMP 割り込みは有効です。EOSMP ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 0 ADRDYIE : ADC レディ割り込みイネーブル

このビットは、ADC レディ割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : ADRDY 割り込み無効

1 : ADRDY 割り込みは有効です。ADRDY ビットがセットされると、割り込みが生成されます。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

26.6.3 ADC 制御レジスタ (ADC_CR)

アドレスオフセット : 0x08

リセット値 : 0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										rs	rs	rs	rs	rs	rs

ビット 31 ADCAL : ADC 校正

このビットは、ADC の校正を開始するためにソフトウェアによってセットされます。最初に ADCALDIF ビットをプログラムして、この校正がシングルエンドまたは差動入力モードに適用されるかどうかを確認します。

校正の完了後、ハードウェアによってクリアされます。

0 : 校正が完了しました。

1 : ADC を校正するには、1 を書き込みます。1 として読み出されたときには、校正が実行中であることを意味します。

注： ソフトウェアは、ADEN = 0 のときだけ ADCAL をセットして、校正を起動することができます。ソフトウェアは、ADEN = 1 かつ ADSTART = 0 かつ JADSTART = 0 である (ADC が有効であり、変換中でない) のときだけ、ADC_CALFACT を書き込むことによって校正係数を更新できます。

ビット 30 ADCALDIF : 校正の差動モード

このビットは、校正のシングルエンド入力または差動入力モードを設定するために、ソフトウェアによってセット/クリアされます。

0 : ADCAL を書き込むと、シングルエンド入力モードで校正が起動されます。

1 : ADCAL を書き込むと、差動入力モードで校正が起動されます。

注： ソフトウェアは、ADC が無効で校正されていないときだけ (ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、このビットを書き込むことができます。

ビット 29 DEEPPWD : ディープパワーダウンイネーブル

このビットは、ADC をディープパワーダウンモードにするために、ソフトウェアによってセット/クリアされます。

0 : ADC のディープパワーダウンモードは無効です。

1 : ADC のディープパワーダウンモードは有効です (デフォルトのリセット状態)。

注： ソフトウェアは、ADC が無効なときだけ (ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、このビットを書き込むことができます。

ビット 28 **ADVREGEN** : ADC 電圧レギュレータイネーブル

これらのビットは、ADC 電圧レギュレータを有効にするために、ソフトウェアによってセットされます。較正の起動や ADC の有効化などの操作を実行する前に、まずは ADC 電圧レギュレータを有効にし、ソフトウェアはレギュレータの起動時間まで待つ必要があります。

0 : ADC 電圧レギュレータは無効です。

1 : ADC 電圧レギュレータは有効です。

ADC 電圧レギュレータの有効化および無効化シーケンスの詳細については、[セクション 26.4.6 : ADC ディープパワーダウンモード \(DEEPPWD\) および ADC 電圧レギュレータ \(ADVREGEN\)](#) を参照してください。

ソフトウェアは、ADC が無効なときだけ (ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、このビットフィールドをプログラムできます。

ビット 27:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **JADSTP** : ADC インジェクト変換停止コマンド

このビットは、実行中のインジェクト変換を停止および破棄するためにソフトウェアによってセットされます (JADSTP コマンド)。

変換が効果的に破棄され、ADC インジェクトシーケンスとトリガを再設定できる場合に、ハードウェアによってクリアされます。これで、ADC は新しいインジェクト変換の開始 (JADSTART コマンド) を受け入れる準備ができます。

0 : ADC インジェクト変換停止コマンドは実行中ではありません。

1 : 実行中の ADC インジェクト変換を停止するには、1 を書き込みます。1 として読み出されたときには、ADSTP コマンドが実行中であることを意味します。

注 : ソフトウェアは、JADSTART = 1 かつ ADDIS = 0 である (ADC が有効であり、最終的にインジェクト変換が変換され、ADC を無効にする保留中のリクエストがない) ときだけ、JADSTP をセットできます。

自動インジェクションモード (JAUTO = 1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアポートされます (JADSTP は使用しないでください)。

ビット 4 **ADSTP** : ADC レギュラ変換停止コマンド

このビットは、実行中のレギュラ変換を停止および破棄するためにソフトウェアによってセットされます (ADSTP コマンド)。

変換が効果的に破棄され、ADC レギュラシーケンスとトリガを再設定できる場合に、ハードウェアによってクリアされます。これで、ADC は新しいレギュラ変換の開始 (ADSTART コマンド) を受け入れる準備ができます。

0 : ADC レギュラ変換停止コマンドは実行中ではありません。

1 : 実行中の ADC レギュラ変換を停止するには、1 を書き込みます。1 として読み出されたときには、ADSTP コマンドが実行中であることを意味します。

注 : ソフトウェアは、ADSTART = 1 かつ ADDIS = 0 である (ADC が有効であり、最終的にレギュラ変換が変換され、ADC を無効にする保留中のリクエストがない) ときだけ、ADSTP をセットできます。

自動インジェクションモード (JAUTO = 1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアポートされます (JADSTP は使用しないでください)。

ビット 3 JADSTART : ADC インジェクト変換開始

このビットは、インジェクトチャネルの ADC 変換を開始するためにソフトウェアによってセットされます。設定ビット JEXTEN に応じて、変換はただちに開始されるか（ソフトウェアトリガ設定）またはインジェクトハードウェアトリガイイベントが発生（ハードウェアトリガ設定）したときに開始されます。次のときに、ハードウェアによってクリアされます。

- シングル変換モードでは、ソフトウェアトリガが選択されたとき（JEXTSEL = 0x0）：インジェクト変換シーケンス終了（JEOS）フラグのアサート時。
- すべての場合：JADSTP コマンドの実行後、JADSTP ビットがハードウェアによってクリアされると同時に。

0 : ADC インジェクト変換は実行中ではありません。

1 : インジェクト変換を開始するには、1 を書き込みます。1 として読み出されたときには、ADC が動作中であり、最終的にインジェクトチャネルが変換されることを意味します。

注： ソフトウェアは、ADEN = 1 かつ ADDIS = 0 である（ADC が有効であり、ADC を無効にする保留中のリクエストがない）ときだけ、JADSTART をセットできます。

自動インジェクションモード（JAUTO = 1）では、ADSTART ビットのセットによってレギュラおよび自動インジェクト変換を開始します（JADSTART はクリアされたままでなければなりません）。

ビット 2 ADSTART : ADC レギュラ変換開始

このビットは、レギュラチャネルの ADC 変換を開始するためにソフトウェアによってセットされます。設定ビット EXTEN に応じて、変換はただちに開始されるか（ソフトウェアトリガ設定）またはレギュラハードウェアトリガイイベントが発生（ハードウェアトリガ設定）したときに開始されます。次のときに、ハードウェアによってクリアされます。

- シングル変換モードでは、ソフトウェアトリガが選択されたとき（EXTSEL = 0x0）：レギュラ変換シーケンス終了（EOS）フラグのアサート時。
- すべての場合：ADSTP コマンドの実行後、ADSTP ビットがハードウェアによってクリアされると同時に。

0 : ADC レギュラ変換は実行中ではありません。

1 : レギュラ変換を開始するには、1 を書き込みます。1 として読み出されたときには、ADC が動作中であり、最終的にレギュラチャネルが変換されることを意味します。

注： ソフトウェアは、ADEN = 1 かつ ADDIS = 0 である（ADC が有効であり、ADC を無効にする保留中のリクエストがない）ときだけ、ADSTART をセットできます。

自動インジェクションモード（JAUTO = 1）では、ADSTART ビットのセットによってレギュラおよび自動インジェクト変換を開始します（JADSTART はクリアされたままでなければなりません）。

ビット 1 ADDIS : ADC 無効化コマンド

このビットは、ADC を無効にして（ADDIS コマンド）、パワーダウン状態（OFF 状態）にするためにソフトウェアによってセットされます。

ADC が効果的に無効化されると、ハードウェアによってクリアされます（ADEN もこの時点でハードウェアによってクリアされます）。

0 : ADDIS コマンドは実行中ではありません。

1 : ADC を無効にするには、1 を書き込みます。1 として読み出されたときには、ADDIS コマンドが実行中であることを意味します。

注： ソフトウェアは、ADEN = 1 で ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、ADDIS をセットできます。

ビット 0 **ADEN** : ADC イネーブル制御

このビットは、ADC を有効にするために、ソフトウェアによってセットされます。フラグ **ADRDY** がセットされると、ADC は動作する準備ができています。

ADDIS コマンドの実行後、ADC が無効になるとハードウェアによってクリアされます。

0 : ADC は無効です (OFF 状態)。

1 : ADC を有効にするには、1 を書き込みます。

注 : ソフトウェアは、1 にセットする必要がある **ADVREGEN** ビットを除き、**ADC_CR** レジスタのすべてのビットが 0 のとき (**ADCAL** = 0、**JADSTART** = 0、**ADSTART** = 0、**ADSTP** = 0、**ADDIS** = 0、および **ADEN** = 0) だけ、**ADEN** をセットできます (また、ソフトウェアは、電圧レギュレータの起動時間まで待っていなければなりません)。

26.6.4 ADC 設定レジスタ (ADC_CFGR)

アドレスオフセット : 0x0C

リセット値 : 0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JQDIS	AWD1CH[4:0]				JAUTO		JAWDIEN	AWDIEN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]		DISCEN	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIGN	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL4	EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	RES[1:0]		Res.	DMACFG	DMAEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

ビット 31 **JQDIS** : インジェクトキューディセーブル

このビットは、インジェクトキューメカニズムを無効にするために、ソフトウェアによってセット/クリアされます。

0 : インジェクトキューは有効です。

1 : インジェクトキューは無効です。

注 : ソフトウェアは、**ADSTART** = 0 かつ **JADSTART** = 0 である (レギュラ変換とインジェクト変換がどちらも実行中でない) ときのみ、このビットに書き込むことができます。

JQDIS ビットをセットまたはリセットすることで、インジェクトキューは破棄され、**JSQR** レジスタはクリアされます。

ビット 30:26 **AWD1CH[4:0]** : アナログウォッチドッグ 1 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグによって保護される入力チャンネルを選択します。

00000 : AWD1 によって監視されている ADC アナログ入力チャンネル 0 (ADC1 のみ使用可能)

00001 : AWD1 によって監視されている ADC アナログ入力チャンネル 1

.....

10011 : AWD1 によって監視されている ADC アナログ入力チャンネル 19

その他。予約済み。使用できません。

注 : 一部のチャンネルは物理的に接続されていません。対応する **AWD1CH[4:0]** 設定をリセット値で保持します。**AWD1CH** で選択したチャンネルは、**SQRi** または **JSQRi** レジスタでも選択する必要があります。

ソフトウェアは、**ADSTART** = 0 かつ **JADSTART** = 0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 25 JAUTO : 自動インジェクトグループ変換

このビットは、レギュラグループ変換の後、自動インジェクトグループ変換を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 自動インジェクトグループ変換は無効です。

1 : 自動インジェクトグループ変換は有効です。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（レギュラ変換とインジェクト変換がどちらも実行中でない）ときのみ、このビットに書き込むことができます。

デュアルモードが有効（ADC_CCR レジスタの DUAL ビットが 0 以外）である場合、スレーブ ADC の JAUTO ビットへの書き込みができなくなり、その内容はマスタ ADC の JAUTO ビットと等しくなります。

ビット 24 JAWD1EN : アナログウォッチドッグ 1 有効（インジェクトチャネル）

このビットは、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルのアナログウォッチドッグ 1 は無効です。

1 : インジェクトチャネルのアナログウォッチドッグ 1 は有効です。

注： ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 23 AWD1EN : アナログウォッチドッグ 1 有効（レギュラチャネル）

このビットは、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルのアナログウォッチドッグ 1 は無効です。

1 : レギュラチャネルのアナログウォッチドッグ 1 は有効です。

注： ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 22 AWD1SGL : ウォッチドッグ 1 を単一チャネルまたはすべてのチャネルで有効にする。

このビットは、AWD1CH[4:0] ビットによって指定されたチャネルまたはすべてのチャネルに対するアナログウォッチドッグを有効にするために、ソフトウェアによってセット／クリアされます。

0 : すべてのチャネルでアナログウォッチドッグ 1 は有効です。

1 : 単一チャネルでアナログウォッチドッグ 1 は有効です。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 21 JQM : JSQR キューモード

このビットは、ソフトウェアによってセット／クリアされます。

空のキューの管理方法について定義します。

0 : JSQR モード 0 : キューは空にならず、JSQR に最後に書き込んだ設定を保持します。

1 : JSQR モード 1 : キューは空である場合もありますが、このとき、インジェクトシーケンスのソフトウェア／ハードウェアトリガは、最後の有効なインジェクトシーケンスの完了後、すぐに両方とも内部的に無効化されます。

詳細については、[セクション 26.4.21 : インジェクト変換のコンテキストのキュー](#)を参照してください。

注： ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアルモードが有効（ADC_CCR レジスタの DUAL ビットが 0 以外）である場合、スレーブ ADC の JQM ビットへの書き込みができなくなり、その内容はマスタ ADC の JQM ビットと等しくなります。

ビット 20 **JDISCEN** : インジェクトチャネルの不連続モード

このビットは、グループのインジェクトチャネルに対する不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルの不連続モードは無効です。

1 : インジェクトチャネルの不連続モードは有効です。

注 : ソフトウェアは、JADSTART = 0 である（インジェクト変換が実行中でない）ときのみ、このビットに書き込むことができます。

自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO のセット時には、ビット DISCEN と JDISCEN はクリアされたままでなければなりません。

デュアルモードが有効 (ADC_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の JDISCEN ビットへの書き込みができなくなり、その内容はマスタ ADC の JDISCEN ビットと等しくなります。

ビット 19:17 **DISCNUM[2:0]** : 不連続モードチャネルカウント

これらのビットは、外部トリガの受信後に、不連続モードで変換されるレギュラチャネルの数を定義するために、ソフトウェアによって書き込まれます。

000 : 1 チャネル

001 : 2 チャネル

.....

111 : 8 チャネル

注 : ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

デュアルモードが有効 (ADC_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の DISCNUM[2:0] ビットへの書き込みができなくなり、その内容はマスタ ADC の DISCNUM[2:0] ビットと等しくなります。

ビット 16 **DISCEN** : レギュラチャネルの不連続モード

このビットは、レギュラチャネルの不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルの不連続モードは無効です。

1 : レギュラチャネルの不連続モードは有効です。

注 : 不連続モードと連続モードの両方を有効にすることはできません。DISCEN = 1 と CONT = 1 の両方をセットすることは禁じられています。

自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO のセット時には、ビット DISCEN と JDISCEN はクリアされたままでなければなりません。

ソフトウェアは、ADSTART = 0 である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアルモードが有効 (ADC_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の DISCEN ビットへの書き込みができなくなり、その内容はマスタ ADC の DISCEN ビットと等しくなります。

ビット 15 **ALIGN** : データの配置

このビットは、右詰めまたは左詰めを選択するために、ソフトウェアによってセット／クリアされます。[セクション : データレジスタ、データの配置およびデータオフセット \(ADC_DR, OFFSET, OFFSET_CH, ALIGN\)](#) を参照してください。

0 : 右詰め

1 : 左詰め

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 14 **AUTDLY** : 遅延変換モード

このビットは、自動遅延変換モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 自動遅延変換モードはオフです。

1 : 自動遅延変換モードはオンです。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアルモードが有効（**ADC_CCR** レジスタの **DUAL** ビットが 0 以外）である場合、スレーブ ADC の **AUTDLY** ビットへの書き込みができなくなり、その内容はマスタ ADC の **AUTDLY** ビットと等しくなります。

ビット 13 **CONT** : レギュラ変換のシングル／連続変換モード

このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされた場合、それがクリアされるまで連続的にレギュラ変換が行われます。

0 : シングル変換モード

1 : 連続変換モード

注 : 不連続モードと連続モードの両方を有効にすることはできません。 **DISCEN = 1** と **CONT = 1** の両方をセットすることは禁じられています。

ソフトウェアは、**ADSTART = 0** である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアルモードが有効（**ADC_CCR** レジスタの **DUAL** ビットが 0 以外）である場合、スレーブ ADC の **CONT** ビットへの書き込みができなくなり、その内容はマスタ ADC の **CONT** ビットと等しくなります。

ビット 12 **OVRMOD** : オーバーランモード

このビットは、ソフトウェアによってセット／クリアされ、データオーバーランを管理する方法を設定します。

0 : オーバーランが検出されたとき、**ADC_DR** レジスタの古いデータが保存されます。

1 : オーバーランが検出されたとき、**ADC_DR** レジスタは最後の変換結果で上書きされます。

注 : ソフトウェアは、**ADSTART = 0** である（レギュラ変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 11:10 **EXTEN[1:0]** : レギュラチャネルの外部トリガ有効および極性選択

これらのビットは、外部トリガ極性を選択しレギュラグループのトリガを有効にするために、ソフトウェアによってセット／クリアされます。

00 : ハードウェアトリガ検出は無効です（変換はソフトウェアによって起動できます）。

01 : 立ち上がりエッジでハードウェアトリガを検出します。

10 : 立ち下がりエッジでハードウェアトリガを検出します。

11 : 立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

注 : ソフトウェアは、**ADSTART = 0** である（レギュラ変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 9:5 **EXTSEL[4:0]** : レギュラグループの外部トリガ選択

これらのビットは、レギュラグループの変換の開始をトリガするために使用される外部イベントを選択します。

00000 : **adc_ext_trg0**

00001 : **adc_ext_trg1**

00010 : **adc_ext_trg2**

00011 : **adc_ext_trg3**

00100 : **adc_ext_trg4**

00101 : **adc_ext_trg5**

00110 : **adc_ext_trg6**

00111 : **adc_ext_trg7**

.....

11111 : **adc_ext_trg31**

注 : ソフトウェアは、**ADSTART = 0** である（レギュラ変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 4:3 **RES[1:0]** : データ分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

00 : 12 ビット

01 : 10 ビット

10 : 8 ビット

11 : 6 ビット

注： ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DMACFG** : ダイレクトメモリアクセス設定

このビットは、2 つの DMA 動作モードを選択するためにソフトウェアによってセット／クリアされ、**DMAEN = 1** のときのみ有効です。

0 : DMA ワンショットモードが選択されています。

1 : DMA サーキュラモードが選択されています。

詳細については、[セクション : DMA を使用した変換の管理](#)を参照してください。

注： ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアル ADC モードでは、このビットは無関係で、**ADC_CCR** レジスタの制御ビット **DMACFG** に置き換えられます。

ビット 0 **DMAEN** : ダイレクトメモリアクセス有効

このビットは、DMA リクエストの生成を有効にするために、ソフトウェアによってセット／クリアされます。これにより、DMA を使用して変換データを自動的に管理できます。詳細については、[セクション : DMA を使用した変換の管理](#)を参照してください。

0 : DMA は無効です。

1 : DMA は有効です。

注： ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

デュアル ADC モードでは、このビットは無関係で、**ADC_CCR** レジスタの制御ビット **MDMA[1:0]** に置き換えられます。

26.6.5 ADC 設定レジスタ 2 (ADC_CFGR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SMPTRIG	BULB	SWTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rW	rW	rW									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR[2:0]				JOVSE
					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **SMPTRIG** : サンプリング時間制御トリガモード

このビットは、サンプリング時間制御トリガモードを有効にするために、ソフトウェアによってセット/クリアされます。

0 : サンプリング時間制御トリガモードは無効です。

1 : サンプリング時間制御トリガモードは有効です。

サンプリング時間はトリガの立ち上がりエッジで開始し、変換はトリガの立ち下がりエッジで開始します。

EXTEN ビットは 01 にセットしなければなりません。SMPTRIG ビットがセットされている場合、BULB ビットをセットすることはできません。

EXTEN ビットが 00 にセットされているとき、サンプリングを開始するには SWTRIG ビットをセットし、変換を開始するには SWTRIG ビットをクリアします。

注 : ソフトウェアは、ADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 26 **BULB** : パルプサンプリングモード

このビットは、パルプサンプリングモードを有効にするために、ソフトウェアによってセット/クリアされます。

0 : パルプサンプリングモードは無効です。

1 : パルプサンプリングモードは有効です。サンプリング期間は前の変換終了直後に開始されます。

BULB ビットがセットされている場合、SAMPTRIG ビットをセットすることはできません。

最初の ADC 変換は SMPx ビットで指定されたサンプリング時間で実行されます。

注 : ソフトウェアは、ADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 25 **SWTRIG** : サンプリング時間制御トリガモードのソフトウェアトリガビット

このビットは、パルプサンプリングモードを有効にするために、ソフトウェアによってセット/クリアされます。

0 : ソフトウェアトリガはサンプリング時間制御トリガモードの変換を開始します。

1 : ソフトウェアトリガはサンプリング時間制御トリガモードのサンプリングを開始します。

注 : ソフトウェアは、ADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 24:17 予約済みであり、リセット値に保持する必要があります。

ビット 16:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **ROVSM** : レギュラオーバーサンプリングモード

このビットは、レギュラオーバーサンプリングモードを選択するために、ソフトウェアによってセット/クリアされます。

0 : 連続モード : インジェクト変換がトリガされると、オーバーサンプリングが一時的に停止され、インジェクションシーケンス後に続行されます (オーバーサンプリングバッファはインジェクションシーケンス中、維持されます)。

1 : 再開モード : インジェクト変換がトリガされると、現在のオーバーサンプリングが中止され、インジェクションシーケンス後に最初から再開されます (オーバーサンプリングバッファはインジェクションシーケンス開始によって 0 になります)。

注 : ソフトウェアは、**ADSTART = 0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 9 **TROVS** : トリガレギュラオーバーサンプリング

このビットは、トリガオーバーサンプリングを有効にするために、ソフトウェアによってセット/クリアされます。

0 : チャネルのオーバーサンプリング変換はすべて、トリガ後に連続的に行われます。

1 : チャネルのオーバーサンプリング変換ごとに新しいトリガが必要です。

注 : ソフトウェアは、**ADSTART = 0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 8:5 **OVSS[3:0]** : オーバーサンプリングシフト

このビットフィールドは、そのままのオーバーサンプリング結果に適用する右シフトを定義するために、ソフトウェアによってセット/クリアされます。

0000 : シフトなし

0001 : 1 ビットシフト

0010 : 2 ビットシフト

0011 : 3 ビットシフト

0100 : 4 ビットシフト

0101 : 5 ビットシフト

0110 : 6 ビットシフト

0111 : 7 ビットシフト

1000 : 8 ビットシフト

その他のコードは予約済み

注 : ソフトウェアは、**ADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 4:2 **OVSRR[2:0]** : オーバーサンプリング比

このビットフィールドは、オーバーサンプリング比を定義するために、ソフトウェアによってセット/クリアされます。

000 : 2x

001 : 4x

010 : 8x

011 : 16x

100 : 32x

101 : 64x

110 : 128x

111 : 256x

注 : ソフトウェアは、**ADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 1 JOVSE : インジェクトオーバーサンプリング有効化

このビットは、インジェクトオーバーサンプリングを有効にするために、ソフトウェアによってセット/クリアされます。

0 : インジェクトオーバーサンプリングは無効です。

1 : インジェクトオーバーサンプリングは有効です。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 0 ROVSE : レギュラオーバーサンプリング有効化

このビットは、レギュラオーバーサンプリングを有効にするために、ソフトウェアによってセット/クリアされます。

0 : レギュラオーバーサンプリングは無効です。

1 : レギュラオーバーサンプリングは有効です。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

26.6.6 ADC サンプル時間レジスタ 1 (ADC_SMPR1)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMP PLUS	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:1]	
r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]	SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 SMPPLUS : サンプリング時間への 1クロックサイクルの追加

1: 2.5 ADC クロックサイクルのサンプリング時間が ADC_SMPR1 および ADC_SMPR2 レジスタに対して 3.5 ADC クロックサイクルになります。

0 : サンプリング時間が、2.5 ADC クロックサイクルのままにセットされます。

進行中の変換がないことを確認するために、ソフトウェアは ADSTART = 0 および JADSTART = 0 の場合のみ、このビットに書き込みます。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:0 SMPx[2:0] : チャネル x サンプリング時間選択 (x = 9~0)

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプルサイクル中は、チャネル選択ビットを変更することはできません。

000 : 2.5 ADC クロックサイクル

001 : 6.5 ADC クロックサイクル

010 : 12.5 ADC クロックサイクル

011 : 24.5 ADC クロックサイクル

100 : 47.5 ADC クロックサイクル

101 : 92.5 ADC クロックサイクル

110 : 247.5 ADC クロックサイクル

111 : 640.5 ADC クロックサイクル

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

一部のチャネルは物理的に接続されていません。対応する SMPx[2:0] 設定をリセット値で保持します。

26.6.7 ADC サンプル時間レジスタ 2 (ADC_SMPR2)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP19 [2:0]			SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:1]	
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15[0]		SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]	
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:0 **SMPx[2:0]** : チャネル x サンプリング時間選択 (x = 19~10)

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャネル選択ビットを変更することはできません。

000 : 2.5 ADC クロックサイクル

001 : 6.5 ADC クロックサイクル

010 : 12.5 ADC クロックサイクル

011 : 24.5 ADC クロックサイクル

100 : 47.5 ADC クロックサイクル

101 : 92.5 ADC クロックサイクル

110 : 247.5 ADC クロックサイクル

111 : 640.5 ADC クロックサイクル

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

一部のチャネルは物理的に接続されていません。対応する SMPx[2:0] 設定をリセット値で保持します。

26.6.8 ADC ウォッチドッグ閾値レジスタ 1 (ADC_TR1)

アドレスオフセット : 0x20

リセット値 : 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1 [11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AWDFILT[2:0]			LT1 [11:0]											
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **HT1[11:0]** : アナログウォッチドッグ 1 高閾値

これらのビットは、アナログウォッチドッグ 1 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD_HTx, AWD_LTx, AWDx\)](#) を参照してください。

注 : ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **AWDFILT[2:0]** : アナログウォッチドッグフィルタパラメータ

このビットは、ソフトウェアによってセット/クリアされます。

000 : フィルタリングなし

001 : 2 個の連続した検出によって AWDx フラグまたは割込みが生成されます。

.....

111 : 8 個の連続した検出によって AWDx フラグまたは割込みが生成されます。

注 : ソフトウェアは、**ADSTART = 0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 11:0 **LT1[11:0]** : アナログウォッチドッグ 1 低閾値

これらのビットは、アナログウォッチドッグ 1 の低閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx\)](#) を参照してください。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

26.6.9 ADC ウォッチドッグ閾値レジスタ 2 (ADC_TR2)

アドレスオフセット : 0x24

リセット値 : 0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2 [7:0]							
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2 [7:0]							
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **HT2[7:0]** : アナログウォッチドッグ 2 高閾値

これらのビットは、アナログウォッチドッグ 2 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx\)](#) を参照してください。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **LT2[7:0]** : アナログウォッチドッグ 2 低閾値

これらのビットは、アナログウォッチドッグ 2 の低閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx\)](#) を参照してください。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

26.6.10 ADC ウォッチドッグ閾値レジスタ 3 (ADC_TR3)

アドレスオフセット : 0x28

リセット値 : 0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3 [7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3 [7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **HT3[7:0]** : アナログウォッチドッグ 3 高閾値

これらのビットは、アナログウォッチドッグ 3 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 26.4.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx\)](#) を参照してください。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **LT3[7:0]** : アナログウォッチドッグ 3 低閾値

これらのビットは、アナログウォッチドッグ 3 の低閾値を定義するために、ソフトウェアによって書き込まれます。

このウォッチドッグでは、LT3 の 8 ビットを変換データの 最上位の8ビットと比較します。

注 : ソフトウェアは、**ADSTART = 0** かつ **JADSTART = 0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

26.6.11 ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4 [4:0]					Res.	SQ3 [4:0]					Res.	SQ2[4]
			rW	rW	rW	rW	rW		rW	rW	rW	rW	rW		rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2 [3:0]				Res.	SQ1 [4:0]					Res.	Res.	L[3:0]			
rW	rW	rW	rW		rW	rW	rW	rW	rW			rW	rW	rW	rW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ4[4:0]** : レギュラシーケンスの 4 番目の変換

これらのビットには、レギュラ変換シーケンス内の 4 番目として割り当てられたチャネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ3[4:0]** : レギュラシーケンスの 3 番目の変換

これらのビットには、レギュラ変換シーケンス内の 3 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ2[4:0]** : レギュラシーケンスの 2 番目の変換

これらのビットには、レギュラ変換シーケンス内の 2 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ1[4:0]** : レギュラシーケンスの 1 番目の変換

これらのビットには、レギュラ変換シーケンス内の 1 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **L[3:0]** : レギュラチャンネルシーケンス長

これらのビットは、レギュラチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

0000 : 1 変換

0001 : 2 変換

.....

1111 : 16 変換

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

注 : 一部のチャンネルは物理的に接続されていないため、変換対象として選択することはできません。

26.6.12 ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9 [4:0]				Res.	SQ8 [4:0]				Res.	SQ7[4]		
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ7 [3:0]			Res.	SQ6 [4:0]				Res.	SQ5 [4:0]						
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ9[4:0]** : レギュラシーケンスの 9 番目の変換

これらのビットには、レギュラ変換シーケンス内の 9 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ8[4:0]** : レギュラシーケンスの 8 番目の変換

これらのビットには、レギュラ変換シーケンス内の 8 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ7[4:0]** : レギュラシーケンスの 7 番目の変換

これらのビットには、レギュラ変換シーケンス内の 7 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ6[4:0]** : レギュラシーケンスの 6 番目の変換

これらのビットには、レギュラ変換シーケンス内の 6 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ5[4:0]** : レギュラシーケンスの 5 番目の変換

これらのビットには、レギュラ変換シーケンス内の 5 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

注 : 一部のチャンネルは物理的に接続されていないため、変換対象として選択することはできません。

26.6.13 ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14 [4:0]					Res.	SQ13 [4:0]					Res.	SQ12[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ12 [3:0]				Res.	SQ11 [4:0]					Res.	SQ10 [4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ14[4:0]** : レギュラシーケンスの 14 番目の変換

これらのビットには、レギュラ変換シーケンス内の 14 番目として割り当てられたチャンネル番号 (0~19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ13[4:0]** : レギュラシーケンスの 13 番目の変換

これらのビットには、レギュラ変換シーケンス内の 13 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ12[4:0]** : レギュラシーケンスの 12 番目の変換

これらのビットには、レギュラ変換シーケンス内の 12 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ11[4:0]** : レギュラシーケンスの 11 番目の変換

これらのビットには、レギュラ変換シーケンス内の 11 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ10[4:0]** : レギュラシーケンスの 10 番目の変換

これらのビットには、レギュラ変換シーケンス内の 10 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

注 : 一部のチャンネルは物理的に接続されていないため、変換対象として選択することはできません。

26.6.14 ADC レギュラシーケンスレジスタ 4 (ADC_SQR4)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SQ16 [4:0]					Res.	SQ15 [4:0]				
					rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ16[4:0]** : レギュラシーケンスの 16 番目の変換

これらのビットには、レギュラ変換シーケンス内の 16 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注 : ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ15[4:0]** : レギュラシーケンスの 15 番目の変換

これらのビットには、レギュラ変換シーケンス内の 15 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注: ソフトウェアは、**ADSTART = 0** である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

注: 一部のチャンネルは物理的に接続されていないため、変換対象として選択することはできません。

26.6.15 ADC レギュラデータレジスタ (ADC_DR)

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RDATA[15:0]** : レギュラデータ変換

これらのビットは読み出し専用です。これらは、最後の変換されたレギュラチャンネルの変換結果を含んでいます。
[セクション 26.4.26 : データ管理](#)に示すように、データは左詰めまたは右詰めされています。

26.6.16 ADC インジェクトシーケンスレジスタ (ADC_JSQR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JSQ4 [4:0]					Res.	JSQ3 [4:0]					Res.	JSQ2 [4:1]			
r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ2[0]	Res.	JSQ1 [4:0]				JEXTEN[1:0]		JEXTSEL[4:0]				JL[1:0]			
r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:27 **JSQ4[4:0]** : インジェクトシーケンスの 4 番目の変換

これらのビットには、インジェクト変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注: ソフトウェアは、**JADSTART = 0** である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25:21 **JSQ3[4:0]** : インジェクトシーケンスの 3 番目の変換

これらのビットには、インジェクト変換シーケンス内の 3 番目として割り当てられたチャンネル番号 (0～19) がソフトウェアによって書き込まれます。

注: ソフトウェアは、**JADSTART = 0** である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19:15 JSQ2[4:0] : インジェクトシーケンスの 2 番目の変換

これらのビットには、インジェクト変換シーケンス内の 2 番目として割り当てられたチャンネル番号 (0 ~ 19) がソフトウェアによって書き込まれます。

注： ソフトウェアは、JADSTART = 0 である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:9 JSQ1[4:0] : インジェクトシーケンスの 1 番目の変換

これらのビットには、インジェクト変換シーケンス内の 1 番目として割り当てられたチャンネル番号 (0 ~ 19) がソフトウェアによって書き込まれます。

注： ソフトウェアは、JADSTART = 0 である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 8:7 JEXTEN[1:0] : インジェクトチャンネルの外部トリガ有効および極性選択

これらのビットは、外部トリガ極性を選択しインジェクトグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : JQDIS = 0 (キュー有効) の場合、ハードウェアおよびソフトウェアトリガ検出は無効になります。それ以外の場合、キューもハードウェアトリガ検出も無効です (変換はソフトウェアによって起動できます)。

01 : 立ち上がりエッジでハードウェアトリガを検出します。

10 : 立ち下がりエッジでハードウェアトリガを検出します。

11 : 立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

注： ソフトウェアは、JADSTART = 0 である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

JQM = 1 で、コンテキストのキューが空である場合、インジェクトシーケンスのソフトウェアおよびハードウェアトリガは両方とも内部的に無効化されます ([セクション 26.4.21 : インジェクト変換のコンテキストのキュー](#)を参照)。

ビット 6:2 JEXTSEL[4:0] : インジェクトグループの外部トリガ選択

これらのビットは、インジェクトグループの変換の開始をトリガするために使用される外部イベントを選択します。

00000 : adc_jext_trg0

00001 : adc_jext_trg1

00010 : adc_jext_trg2

00011 : adc_jext_trg3

00100 : adc_jext_trg4

00101 : adc_jext_trg5

00110 : adc_jext_trg6

00111 : adc_jext_trg7

.....

11111 : adc_jext_trg31

注： ソフトウェアは、JADSTART = 0 である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 1:0 JL[1:0] : インジェクトチャンネルシーケンス長

これらのビットは、インジェクトチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

00 : 1 変換

01 : 2 変換

10 : 3 変換

11 : 4 変換

注： ソフトウェアは、JADSTART = 0 である (インジェクト変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

注： 一部のチャンネルは物理的に接続されていないため、変換対象として選択することはできません。

26.6.17 ADC オフセット y レジスタ (ADC_OFRy)

アドレスオフセット : $0x60 + 0x04 * (y - 1)$ 、(y = 1 から 4)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSET_EN	OFFSET_CH[4:0]					SATEN	OFFSET_POS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	OFFSET[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **OFFSET_EN** : オフセット y イネーブル

このビットは、OFFSET[11:0] ビットにプログラムされたオフセットを有効または無効にするために、ソフトウェアによって書き込まれます。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 30:26 **OFFSET_CH[4:0]** : データオフセット y のチャンネル選択

このビットは、OFFSET[11:0] ビットにプログラムされたオフセットが適用されるチャンネルを定義するために、ソフトウェアによって書き込まれます。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

一部のチャンネルは物理的に接続されていないため、データオフセット y として選択することはできません。

OFFSET_EN がセットされている場合、同じチャンネルを別の ADC_OFRy レジスタに選択することはできません。

ビット 25 **SATEN** : 飽和イネーブル

このビットは、オフセット機能のために 0x000 および 0xFFFF で飽和を有効にするために、ソフトウェアによってセット/クリアされます。

0 : 飽和制御はなしで、オフセット結果は符号付きにできます。

1 : 飽和は有効であり、オフセット結果は符号なしで、0x000 および 0xFFFF で飽和します。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 24 **OFFSETPOS** : 正のオフセット

このビットは、正のオフセットを有効にするために、ソフトウェアによってセット/クリアされます。

0 : 負のオフセット

1 : 正のオフセット

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

ビット 23:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **OFFSET[11:0]** : OFFSET_CH[4:0] ビットにプログラムされたチャンネルのデータオフセット y

これらのビットは、チャンネル（レギュラまたはインジェクトのどちらか）を変換するとき、元の変換データから引かれるオフセットを定義するために、ソフトウェアによって書き込まれます。データオフセットを適用するチャンネルは、ビット OFFSET_CH[4:0] にプログラムする必要があります。変換結果は、ADC_DR レジスタ（レギュラ変換）または ADC_JDRy レジスタ（インジェクト変換）から読み出すことができます。

注： ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である（変換が実行中でない）ときのみ、これらのビットに書き込むことができます。

複数のオフセット（OFFSET）が同じチャンネルを指している場合、減算時には最小の x 値を持つオフセットのみが考慮されます。

例：OFFSET1_CH[4:0] = 4 かつ OFFSET2_CH[4:0] = 4 である場合、これはチャンネル 4 を変換するときに減算される OFFSET1[11:0] です。

26.6.18 ADC インジェクトチャンネル y データレジスタ (ADC_JDRy)

アドレスオフセット : $0x80 + 0x04 * (y - 1)$ 、(y = 1 から 4)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **JDATA[15:0]** : インジェクトデータ

これらのビットは読み出し専用です。これらのビットには、インジェクトチャンネル y の変換結果が格納されます。[セクション 26.4.26: データ管理](#)に示すように、データは左詰めまたは右詰めされています。

26.6.19 ADC アナログウォッチドッグ 2 設定レジスタ (ADC_AWD2CR)

アドレスオフセット : 0xA0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **AWD2CH[19:0]** : アナログウォッチドッグ 2 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグ 2 によって保護される入力チャンネルを有効化して選択します。

AWD2CH[i] = 0 : ADC アナログ入力チャンネル i は AWD2 によって監視されません。

AWD2CH[i] = 1 : ADC アナログ入力チャンネル i は AWD2 によって監視されます。

AWD2CH[19:0] = 000..0 である場合、アナログウォッチドッグ 2 は無効になります。

注 : AWD2CH で選択したチャンネルは、SQRI または JSQRI レジスタでも選択する必要があります。

ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

一部のチャンネルは物理的に接続されていないため、アナログウォッチドッグとして選択することはできません。

26.6.20 ADC アナログウォッチドッグ 3 設定レジスタ (ADC_AWD3CR)

アドレスオフセット : 0xA4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **AWD3CH[19:0]** : アナログウォッチドッグ 3 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグ 3 によって保護される入力チャンネルを有効化して選択します。

AWD3CH[i] = 0 : ADC アナログ入力チャンネル i は AWD3 によって監視されません。

AWD3CH[i] = 1 : ADC アナログ入力チャンネル i は AWD3 によって監視されます。

AWD3CH[19:0] = 000..0 である場合、アナログウォッチドッグ 3 は無効になります。

注 : AWD3CH で選択したチャンネルは、SQRI または JSQRI レジスタでも選択する必要があります。

ソフトウェアは、ADSTART = 0 かつ JADSTART = 0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

一部のチャンネルは物理的に接続されていないため、アナログウォッチドッグとして選択することはできません。

26.6.21 ADC 差動モード選択レジスタ (ADC_DIFSEL)

アドレスオフセット : 0xB0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFSEL[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	r

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **DIFSEL[19:0]** : チャンネル (19~0) の差動モード

これらのビットは、ソフトウェアによってセット/クリアされます。これにより、チャンネルの設定をシングルエンドモードまたは差動モードから選択できます。

DIFSEL[i] = 0 : ADC アナログ入力チャンネルはシングルエンドモードで設定されます。

DIFSEL[i] = 1 : ADC アナログ入力チャンネルは差動モードで設定されます。

注 : シングルエンド I/O ポートまたは内部チャンネルのどちらかに接続されたチャンネルに対応する DIFSEL ビットは、リセット値に保持する必要があります (シングルエンド入力モード)。

ソフトウェアは、ADC が無効なときだけ (ADCAL = 0、JADSTART = 0、JADSTP = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、これらのビットを書き込むことができます。

26.6.22 ADC 較正係数 (ADC_CALFACT)

アドレスオフセット : 0xB4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]						
									rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]						
									rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **CALFACT_D[6:0]** : 差動モードでの較正係数

これらのビットは、ハードウェアまたはソフトウェアによって書き込まれます。

差動入力較正が完了すると、ハードウェアによって較正係数で更新されます。

ソフトウェアは、これらのビットに新しい較正係数を書き込むことができます。新しい較正係数がアナログ ADC に格納されている現在のものと異なる場合は、新しい差動較正の起動時に適用されます。

注 : ソフトウェアは、ADEN = 1、ADSTART = 0、および JADSTART = 0 のときのみ (ADC が有効であり、較正中でなく、変換中でない)、これらのビットに書き込むことができます。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **CALFACT_S[6:0]** : シングルエンドモードの較正係数

これらのビットは、ハードウェアまたはソフトウェアによって書き込まれます。

シングルエンド入力較正が完了すると、ハードウェアによって較正係数で更新されます。

ソフトウェアは、これらのビットに新しい較正係数を書き込むことができます。新しい較正係数がアナログ ADC に格納されている現在のものと異なる場合は、新しいシングルエンド較正の起動時に適用されます。

注 : ソフトウェアは、**ADEN = 1**、**ADSTART = 0**、および **JADSTART = 0** のときのみ (ADC が有効であり、較正中でなく、変換中でない)、これらのビットに書き込むことができます。

26.6.23 ADC オプションレジスタ (ADC_OR)

アドレスオフセット : 0xC8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP0
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **OP0** : オプションビット 0

ADC1 の場合 :

0 : INP0/INN1 GPIO スイッチ制御が無効です。

1 : INP0/INN1 GPIO スイッチ制御が有効です。

注 : このオプションビットは、ADCx_INP0 または ADCx_INN1 チャンネルが選択されている場合、1 にセットする必要があります。

ADC2 の場合 :

0 : V_{DDCORE} チャンネルは無効です。

1 : V_{DDCORE} チャンネルは有効です。

注 : **ADC_OR** レジスタは、一部の ADC インスタンスで予約済みである場合があります。[セクション 26.3: ADC の実装](#) を参照してください。

26.7 ADC 共通レジスタ

これらのレジスタは、マスタおよびスレーブ ADC に共通の制御およびステータスレジスタを定義します。

26.7.1 ADC 共通ステータスレジスタ (ADC_CSR)

アドレスオフセット : 0x300

リセット値 : 0x0000 0000

このレジスタにより、別の ADC のステータスビットの状態を確認することができます。ただし、これは読み出し専用で、別の ADC のステータスビットをクリアすることはできません。その代わり、対応する ADC_ISR レジスタに 0 を書き込むことによって各ステータスビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	JQOVF_S LV	AWD3_S LV	AWD2_S LV	AWD1_S LV	JEOS_S LV	JEOC_S LV	OVR_S LV	EOS_S LV	EOC_S LV	EOSMP_S LV	ADRDY_S LV
					r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF_M ST	AWD3_M ST	AWD2_M ST	AWD1_M ST	JEOS_M ST	JEOC_M ST	OVR_M MST	EOS_M MST	EOC_M MST	EOSMP_M ST	ADRDY_M ST
					r	r	r	r	r	r	r	r	r	r	r

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **JQOVF_SLV** : スレーブ ADC のインジェクトコンテキストキューのオーバーフローフラグ

このビットは対応する ADC_ISR レジスタの JQOVF ビットのコピーです。

ビット 25 **AWD3_SLV** : スレーブ ADC のアナログウォッチドッグ 3 フラグ

このビットは対応する ADC_ISR レジスタの AWD3 ビットのコピーです。

ビット 24 **AWD2_SLV** : スレーブ ADC のアナログウォッチドッグ 2 フラグ

このビットは対応する ADC_ISR レジスタの AWD2 ビットのコピーです。

ビット 23 **AWD1_SLV** : スレーブ ADC のアナログウォッチドッグ 1 フラグ

このビットは対応する ADC_ISR レジスタの AWD1 ビットのコピーです。

ビット 22 **JEOS_SLV** : スレーブ ADC のインジェクトシーケンスの終了フラグ

このビットは対応する ADC_ISR レジスタの JEOS ビットのコピーです。

ビット 21 **JEOC_SLV** : スレーブ ADC のインジェクト変換の終了フラグ

このビットは対応する ADC_ISR レジスタの JEOC ビットのコピーです。

ビット 20 **OVR_SLV** : スレーブ ADC のオーバーランフラグ

このビットは対応する ADC_ISR レジスタの OVR ビットのコピーです。

ビット 19 **EOS_SLV** : スレーブ ADC のレギュラシーケンスの終了フラグこのビットは対応する ADC_ISR レジスタの EOS ビットのコピーです。

ビット 18 **EOC_SLV** : スレーブ ADC のレギュラ変換の終了フラグ

このビットは対応する ADC_ISR レジスタの EOC ビットのコピーです。

ビット 17 **EOSMP_SLV** : スレーブ ADC のサンプリングフェーズの終了フラグ

このビットは対応する ADC_ISR レジスタの EOSMP2 ビットのコピーです。

ビット 16 **ADRDY_SLV** : スレーブ ADC レディ

このビットは対応する ADC_ISR レジスタの ADRDY ビットのコピーです。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **JQOVF_MST** : マスタ ADC のインジェクトコンテキストキューのオーバーフローフラグ
このビットは対応する ADC_ISR レジスタの JQOVF ビットのコピーです。

ビット 9 **AWD3_MST** : マスタ ADC のアナログウォッチドッグ 3 フラグ
このビットは対応する ADC_ISR レジスタの AWD3 ビットのコピーです。

ビット 8 **AWD2_MST** : マスタ ADC のアナログウォッチドッグ 2 フラグ
このビットは対応する ADC_ISR レジスタの AWD2 ビットのコピーです。

ビット 7 **AWD1_MST** : マスタ ADC のアナログウォッチドッグ 1 フラグ
このビットは対応する ADC_ISR レジスタの AWD1 ビットのコピーです。

ビット 6 **JEOS_MST** : マスタ ADC のインジェクトシーケンスの終了フラグ
このビットは対応する ADC_ISR レジスタの JEOS ビットのコピーです。

ビット 5 **JEOC_MST** : マスタ ADC のインジェクト変換の終了フラグ
このビットは対応する ADC_ISR レジスタの JEOC ビットのコピーです。

ビット 4 **OVR_MST** : マスタ ADC のオーバーランフラグ
このビットは対応する ADC_ISR レジスタの OVR ビットのコピーです。

ビット 3 **EOS_MST** : マスタ ADC のレギュラシーケンスの終了フラグ
このビットは対応する ADC_ISR レジスタの EOS ビットのコピーです。

ビット 2 **EOC_MST** : マスタ ADC のレギュラ変換の終了フラグ
このビットは対応する ADC_ISR レジスタの EOC ビットのコピーです。

ビット 1 **EOSMP_MST** : マスタ ADC のサンプリングフェーズの終了フラグ
このビットは対応する ADC_ISR レジスタの EOSMP ビットのコピーです。

ビット 0 **ADRDY_MST** : マスタ ADC レディ
このビットは対応する ADC_ISR レジスタの ADRDY ビットのコピーです。

26.7.2 ADC 共通制御レジスタ (ADC_CCR)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREF EN	PRESC[3:0]				CKMODE[1:0]	
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDMA[1:0]		DMA CFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
r/w	r/w	r/w		r/w	r/w	r/w	r/w				r/w	r/w	r/w	r/w	r/w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **VBATEN** : VBAT 有効化

このビットは、制御するために、ソフトウェアによってセット/クリアされます。

0 : V_{BAT} チャンネルは無効です。

1 : V_{BAT} チャンネルは有効です。

ビット 23 **TSEN** : V_{SENSE} 有効化

このビットは、V_{SENSE} を制御するために、ソフトウェアによってセット/クリアされます。

0 : 温度センサチャンネルは無効です。

1 : 温度センサチャンネルは有効です。

ビット 22 **VREFEN** : V_{REFINT} 有効化

このビットは、 V_{REFINT} チャネルを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : V_{REFINT} チャネルは無効です。

1 : V_{REFINT} チャネルは有効です。

ビット 21:18 **PRESC[3:0]** : ADC プリスケアラ

これらのビットは、ADC へのクロックの周波数を選択するために、ソフトウェアによってセット／クリアされます。クロックはすべての ADC に共通です。

0000 : ADC 入力クロックは分周されません。

0001 : ADC 入力クロックが 2 分周されます。

0010 : ADC 入力クロックが 4 分周されます。

0011 : ADC 入力クロックが 6 分周されます。

0100 : ADC 入力クロックが 8 分周されます。

0101 : ADC 入力クロックが 10 分周されます。

0110 : ADC 入力クロックが 12 分周されます。

0111 : ADC 入力クロックが 16 分周されます。

1000 : ADC 入力クロックが 32 分周されます。

1001 : ADC 入力クロックが 64 分周されます。

1010 : ADC 入力クロックが 128 分周されます。

1011 : ADC 入力クロックが 256 分周されます。

その他 : 予約済み

注 : ソフトウェアは、ADC が無効なときだけ ($ADCAL = 0$ 、 $JADSTART = 0$ 、 $ADSTART = 0$ 、 $ADSTP = 0$ 、 $ADDIS = 0$ 、および $ADEN = 0$)、これらのビットに書き込むことができます。 $CKMODE[1:0] = 0b00$ である場合のみ、ADC プリスケアラ値が適用されます。

ビット 17:16 **CKMODE[1:0]** : ADC クロックモード

これらのビットは、ADC クロックスキーム（マスタとスレーブの両方の ADC に共通）を定義するために、ソフトウェアによってセット／クリアされます。

00 : adc_ker_ck ($x = 1/2$) (非同期クロックモード)。製品レベルで生成されます (セクション 6 : リセットおよびクロック制御 (RCC) を参照)。

01 : $adc_hclk/1$ (同期クロックモード)。この設定は、AHB クロックプリスケアラが 1 にセットされた場合で (RCC_CFGR レジスタの $HPRE[3:0] = 0XXX$)、システムクロックが 50% のデューティサイクルを持つ場合にのみ有効にする必要があります。

10 : $adc_hclk/2$ (同期クロックモード)。

11 : $adc_hclk/4$ (同期クロックモード)。

すべての同期クロックモードにおいて、タイマトリガから変換開始までの遅延にジッタはありません。

注 : ソフトウェアは、ADC が無効なときだけ ($ADCAL = 0$ 、 $JADSTART = 0$ 、 $ADSTART = 0$ 、 $ADSTP = 0$ 、 $ADDIS = 0$ 、および $ADEN = 0$)、これらのビットに書き込むことができます。

ビット 15:14 **MDMA[1:0]** : デュアル ADC モードのためのダイレクトメモリアクセスモード

このビットフィールドは、ソフトウェアによってセット／クリアされます。詳細については、DMA コントローラのセクションを参照してください。

00 : MDMA モードが無効の場合

01 : 予約済み

10 : MDMA モードは、12 ビットおよび 10 ビットの分解能で有効です。

11 : MDMA モードは、8 ビットおよび 6 ビットの分解能で有効です。

注 : ソフトウェアは、 $ADSTART = 0$ である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 13 **DMACFG** : DMA 設定 (デュアル ADC モード)

このビットは、2 つの DMA 動作モードを選択するためにソフトウェアによってセット/クリアされ、DMAEN = 1 のときのみ有効です。

0 : DMA ワンショットモードが選択されています。

1 : DMA サーキュラモードが選択されています。

詳細については、[セクション : DMA を使用した変換の管理](#)を参照してください。

注 : ソフトウェアは、ADSTART = 0 である (レギュラ変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **DELAY[3:0]** : 2 つのサンプリングフェーズ間の遅延

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットはデュアルインターリーブモードで使用します。ADC 分解能の値対 DELAY ビット値については、[表 253](#)を参照してください。

注 : ソフトウェアは、ADC が無効なときだけ (ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、これらのビットに書き込むことができます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DUAL[4:0]** : デュアル ADC モード選択

これらのビットは、動作モードを選択するために、ソフトウェアによって書き込まれます。00000 は独立モードに対応しています。値 00001~01001 は、協調して動作するデュアルモード、マスタおよびスレーブ ADC に対応しています。

00000 : 独立モード

00001 : レギュラ同時 + インジェクト同時モードの組み合わせ

00010 : レギュラ同時 + オルタネートトリガモードの組み合わせ

00011 : インタリーブモード + インジェクト同時モードの組み合わせ

00100 : 予約済み

00101 : インジェクト同時モードのみ

00110 : レギュラ同時モードのみ

00111 : インタリーブモードのみ

01001 : オルタネートトリガモードのみ

その他 : 予約済み。使用できません。

注 : ソフトウェアは、ADC が無効なときだけ (ADCAL = 0、JADSTART = 0、ADSTART = 0、ADSTP = 0、ADDIS = 0、および ADEN = 0)、これらのビットに書き込むことができます。

表 253. DELAY ビット対 ADC 分解能

DELAY ビット	12 ビット分解能	10 ビット分解能	8 ビット分解能	6 ビット分解能
0000	$1 * T_{\text{adc_ker_ck}}$	$1 * T_{\text{adc_ker_ck}}$	$1 * T_{\text{adc_ker_ck}}$	$1 * T_{\text{adc_ker_ck}}$
0001	$2 * T_{\text{adc_ker_ck}}$	$2 * T_{\text{adc_ker_ck}}$	$2 * T_{\text{adc_ker_ck}}$	$2 * T_{\text{adc_ker_ck}}$
0010	$3 * T_{\text{adc_ker_ck}}$	$3 * T_{\text{adc_ker_ck}}$	$3 * T_{\text{adc_ker_ck}}$	$3 * T_{\text{adc_ker_ck}}$
0011	$4 * T_{\text{adc_ker_ck}}$	$4 * T_{\text{adc_ker_ck}}$	$4 * T_{\text{adc_ker_ck}}$	$4 * T_{\text{adc_ker_ck}}$
0100	$5 * T_{\text{adc_ker_ck}}$	$5 * T_{\text{adc_ker_ck}}$	$5 * T_{\text{adc_ker_ck}}$	$5 * T_{\text{adc_ker_ck}}$
0101	$6 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
0110	$7 * T_{\text{adc_ker_ck}}$	$7 * T_{\text{adc_ker_ck}}$	$7 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
0111	$8 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
1000	$9 * T_{\text{adc_ker_ck}}$	$9 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
1001	$10 * T_{\text{adc_ker_ck}}$	$10 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$

表 253. DELAY ビット対 ADC 分解能 (続き)

DELAY ビット	12 ビット分解能	10 ビット分解能	8 ビット分解能	6 ビット分解能
1010	$11 * T_{\text{adc_ker_ck}}$	$10 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
1011	$12 * T_{\text{adc_ker_ck}}$	$10 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$
その他	$12 * T_{\text{adc_ker_ck}}$	$10 * T_{\text{adc_ker_ck}}$	$8 * T_{\text{adc_ker_ck}}$	$6 * T_{\text{adc_ker_ck}}$

26.7.3 デュアルモード用 ADC 共通レギュラータレジスタ (ADC_CDR)

アドレスオフセット : 0x30C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA_SLV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA_MST[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **RDATA_SLV[15:0]** : スレーブ ADC のレギュラータデータ

デュアルモードでは、これらのビットにはスレーブ ADC のレギュラータデータが格納されます。[セクション 26.4.30 : デュアル ADC モード](#) を参照してください。

データの配置は、[セクション : データレジスタ、データの配置およびデータオフセット \(ADC_DR、OFFSET、OFFSET_CH、ALIGN\)](#) に示すように適用されています。

ビット 15:0 **RDATA_MST[15:0]** : マスタ ADC のレギュラータデータ

デュアルモードでは、これらのビットにはマスタ ADC のレギュラータデータが格納されます。[セクション 26.4.30 : デュアル ADC モード](#) を参照してください。

データの配置は、[セクション : データレジスタ、データの配置およびデータオフセット \(ADC_DR、OFFSET、OFFSET_CH、ALIGN\)](#) に示すように適用されています。

MDMA = 0b11 モードでは、ビット 15:8 には SLV_ADC_DR[7:0] が、ビット 7:0 には MST_ADC_DR[7:0] が含まれています。

26.7.4 ADC ハードウェア設定レジスタ (ADC_HWCFGR0)

アドレスオフセット : 0x3F0

リセット値 : 0x0000 1212

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLEVALUE[3:0]				OPBITS[3:0]				MULPIPE[3:0]				ADCNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IDLEVALUE[3:0]** : 非選択チャンネルのアイドル値

0000 : ダミーチャンネル選択は 0x13 です。

0001 : ダミーチャンネル選択は 0x1F です。

ビット 11:8 **OPBITS[3:0]** : オプションビット数

0000 : オプションレジスタが実装されていません。

0001 : アドレスオフセット 0xC8 の ADC オプションレジスタ (ADC_OR) に 1 オプションビットが実装されています。

ビット 7:4 **MULPIPE[3:0]** : パイプラインの段数

0001 : 1 段パイプライン

ビット 3:0 **ADCNUM[3:0]** : 実装される ADC の数

0001 : 1 つの ADC インスタンスが実装されます。

0010 : 2 つの ADC インスタンスが実装されます。

0011 : 3 つの ADC インスタンスが実装されます。

26.7.5 ADC バージョンレジスタ (ADC_VERR)

アドレスオフセット : 0x3F4

リセット値 : 0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **MAJREV[3:0]** : メジャーリビジョン

このビットは ADC IP のメジャーリビジョンを返します。

0001 : メジャーリビジョン = 1.X

ビット 3:0 **MINREV[3:0]** : マイナーリビジョン

このビットは ADC IP のマイナーリビジョンを返します。

0001 : マイナーリビジョン = X.1

0002 : マイナーリビジョン = X.2

0003 : マイナーリビジョン = X.3

26.7.6 ADC 識別レジスタ (ADC_IPDR)

アドレスオフセット : 0x3F8

リセット値 : 0x0011 0006

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **ID[31:0]** : ペリフェラル識別子

これらのビットは ADC 識別子を返します。

ID[31:0] = 0x0011 0006 : c7amba_aditf5_90_v1

26.7.7 ADC サイズ識別レジスタ (ADC_SIDR)

アドレスオフセット : 0x3FC

リセット値 : 0xA3C5 : DD01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **SID[31:0]** : サイズ識別

SID[31:8]: ADC_SIDR レジスタを特徴付ける固定コード。このフィールドは常に 0xA3C5DD で読み出されます。

SID[7:0] : IP ベースアドレスから識別レジスタのアドレスオフセット (KB 単位) を返す読出し専用数値フィールド :

0x01 : 1 KB のアドレスオフセット

0x02 : 2 KB のアドレスオフセット

0x04 : 4 KB のアドレスオフセット

0x08 : 8 KB のアドレスオフセット

26.8 ADC レジスタマップ

次の表に ADC レジスタの一覧を示します。

表 254. ADC グローバルレジスタマップ

オフセット	レジスタ
0x000 - 0x0B4	マスタ ADC1
0x0B8 - 0x0FC	予約済み
0x100 - 0x1B4	スレーブ ADC2
0x1B8 - 0x2FC	予約済み
0x300 - 0x30C	マスタおよびスレーブ ADC に共通のレジスタ

表 255. ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100)

オフ セット	レジスタ名とリ セット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	ADC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY		
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0		
0x04	ADC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE		
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0		
0x08	ADC_CR	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN																							JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN			
	リセット値	0	0	1	0																							0	0	0	0	0	0	0		
0x0C	ADC_CFGR	JQDIS	AWD1CH [4:0]				JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]				DISCEN	ALIGN	AUTDLY	CONT	OVRRMOD	EXTEN[1:0]	EXTSEL4	EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	RES[1:0]		Res.	DMACFG	DMAEN				
	リセット値	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0			
0x0C	ADC_CFGR2	Res.	Res.	Res.	Res.	SMPTRIG	BULB	SWTRIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]			OVSR[2:0]		JOVSE	ROVSE					
	リセット値					0	0	0															0	0	0	0	0	0	0	0	0	0	0			
0x14	ADC_SMPR1	SMPPLUS	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:0]			SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]					
	リセット値	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x18	ADC_SMPR2	Res.	Res.	SMP19 [2:0]			SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]					
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x1C	予約済み	Res.																																		
0x20	ADC_TR1	Res.	Res.	Res.	Res.	HT1 [11:0]											Res.	AWDFILT [2:0]			LT1 [11:0]															
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x24	ADC_TR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2 [7:0]										
	リセット値									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0	0		
0x28	ADC_TR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3 [7:0]									
	リセット値									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0	0		

表 255. ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100) (続き)

オフ セット	レジスタ名とリ セット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x2C	予約済み	Res.																																	
0x30	ADC_SQR1	Res.	Res.	Res.	SQ4 [4:0]				Res.	SQ3 [4:0]				Res.	SQ2 [4:0]				Res.	SQ1 [4:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L[3:0]			
	リセット値				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0			0	0	0	0		
0x34	ADC_SQR2	Res.	Res.	Res.	SQ9 [4:0]				Res.	SQ8 [4:0]				Res.	SQ7 [4:0]				Res.	SQ6 [4:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ5 [4:0]			
	リセット値				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0			0	0	0	0		
0x38	ADC_SQR3	Res.	Res.	Res.	SQ14 [4:0]				Res.	SQ13 [4:0]				Res.	SQ12 [4:0]				Res.	SQ11 [4:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ10 [4:0]			
	リセット値				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0			0	0	0	0		
0x3C	ADC_SQR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16 [4:0]				Res.	SQ15 [4:0]							
	リセット値																						0	0	0	0	0			0	0	0	0		
0x40	ADC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	レギュラ RDATA[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44- 0x48	予約済み	Res.																																	
0x4C	ADC_JSQR	JSQ4 [4:0]				Res.	JSQ3 [4:0]				Res.	JSQ2 [4:0]				Res.	JSQ1 [4:0]				JEXTEN[1:0]	JEXTSEL [4:0]				JL[1:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x50- 0x5C	予約済み	Res.																																	
0x60	ADC_OFR1	OFFSET_EN	OFFSET_CH[4:0]				SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]													
	リセット値	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0			
0x64	ADC_OFR2	OFFSET_EN	OFFSET_CH[4:0]				SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]													
	リセット値	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0			
0x68	ADC_OFR3	OFFSET_EN	OFFSET_CH[4:0]				SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]													
	リセット値	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0			
0x6C	ADC_OFR4	OFFSET_EN	OFFSET_CH[4:0]				SATEN	OFFSETPOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET[11:0]													
	リセット値	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0			
0x70- 0x7C	予約済み	Res.																																	
0x80	ADC_JDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA1 [15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x84	ADC_JDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA2 [15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x88	ADC_JDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA3 [15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x8C	ADC_JDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA4 [15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x90- 0x9C	予約済み	Res.																																	

表 255. ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100) (続き)

オフ セット	レジスタ名とリ セット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA0	ADC_AWD2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH [19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xA4	ADC_AWD3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH [19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xA8- 0xAC	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0xB0	ADC_DIFSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xB4	ADC_CALFACT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]									
	リセット値										0	0	0	0	0	0	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0xB8- 0xC4	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0xC8	ADC_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																															0	
0xCC- 0xFC	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	

表 256. ADC レジスタマップとリセット値 (マスタおよびスレーブ ADC 共通レジスタ)

オフ セット	レジスタ名とリ セット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x300	ADC_CSR	Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
								スレーブ ADC2																マスタ ADC1										
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x304	予約済み	Res.																																
0x308	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	PRESC[3:0]				CKMODE[1:0]		MDMA[1:0]		DMACFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]					
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0				0	0	0	0
0x30C	ADC_CDR	RDATA_SLV[15:0]															RDATA_MST[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x310- 0x3EC	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x3F0	ADC_HWCFGR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLEVALUE [3:0]			OPBITS[3:0]			MULTIPIPE[3:0]			ADCNUM[3:0]							
	リセット値	0x0000 1212																																
0x3F4	ADC_VERR		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]			MINREV[3:0]			
	リセット値	0x0000 0012																																
0x3F8	ADC_IPDR	ID[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0x3FC	ADC_SIDR	SID[31:0]																																
	リセット値	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

27 デジタル温度センサ (DTS)

27.1 概要

このデバイスは、周波数が温度に比例する矩形波に温度を変換するセンサを内蔵しています。周波数は PCLK または LSE クロックのいずれかで測定されます。

27.2 DTS の主な機能

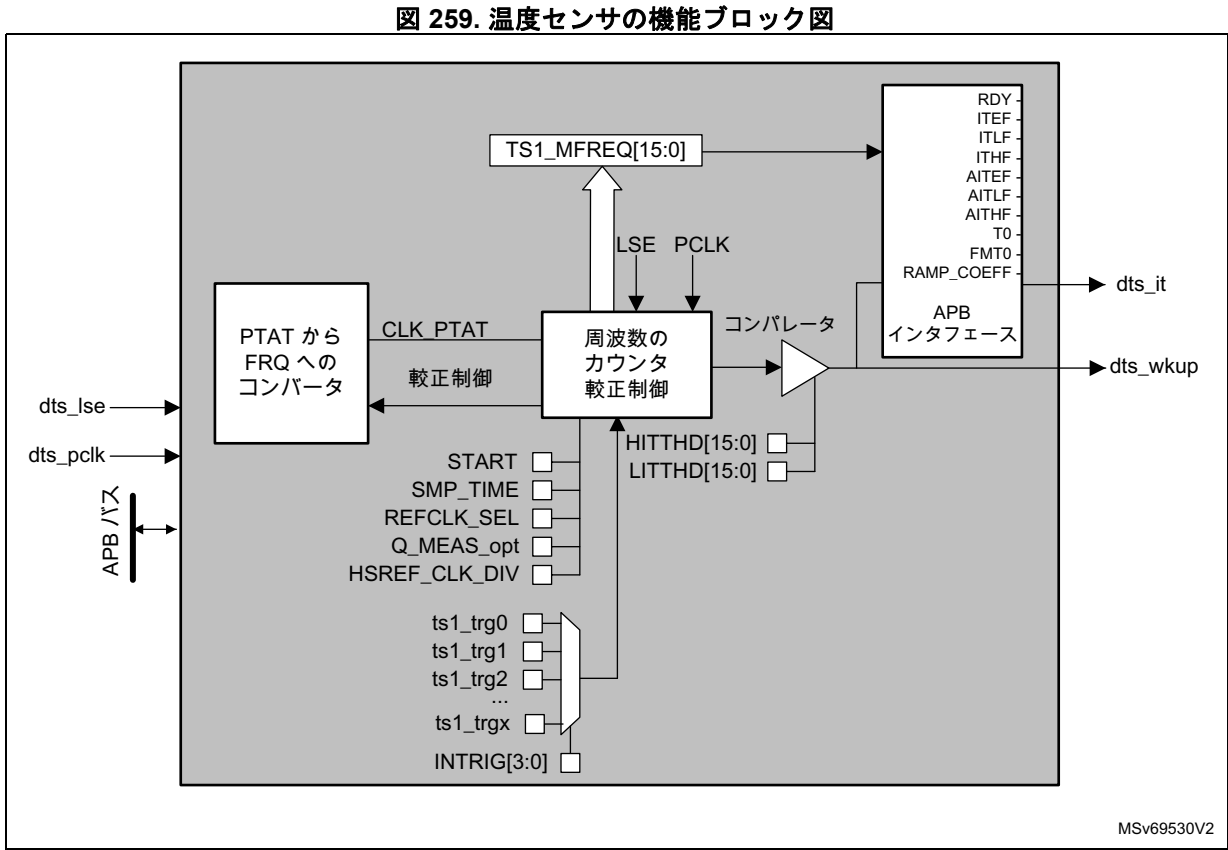
温度センサブロックの主な機能を次に示します。

- 測定の開始は、ソフトウェアまたは 4つのハードウェアソースによってトリガされます。
- 温度測定精度を向上させるプログラム可能なサンプリング時間
- LSE または PCLK クロック上で同期するカウンタ
- 低閾値／高閾値の温度ウォッチドッグ
- 事前定義されている閾値よりも温度が低いまたは高い場合、または測定終了時の割込み生成
- 事前定義されている閾値よりも温度が低いまたは高い場合の非同期ウェイクアップ信号生成 (LSE モードのみ)
- LSE クロックを使用したクイック測定

27.3 DTS の機能詳細

27.3.1 DTS ブロック図

図 259 に温度センサブロック図を示します。



27.3.2 DTS 内部信号

表 257. DTS 内部入力／出力信号

信号名	信号タイプ	説明
dts_lse	デジタル入力	LSE クロック
dts_pclk	デジタル入力	APB クロック
dts_it	デジタル出力	温度センサ割込み
dts_wkup	デジタル出力	温度センサウェイクアップ

27.3.3 DTS ブロックの動作

温度センサのアナログ部分は、絶対温度 (CLK_PTAT) に比例する周波数を出力します。周波数測定は PCLK または LSE クロックを採用しています。

各測定の前に、温度センサは周波数生成ブロックの較正を行います。

27.3.4 動作モード

温度センサ設定レジスタ 1 (DTS_CFGR1) の REFCLK_SEL ビットをセットすることで、いくつかの動作モードを選択できます。

- PCLK のみ (REFCLK_SEL = 0)
温度センサレジスタにアクセスできます。インタフェースは結果的に再設定することができ、測定シーケンスは PCLK クロックを使用して実行されます。
- PCLK および LSE (REFCLK_SEL = 1)
温度センサレジスタにアクセスできます。インタフェースは結果的に再設定することができ、測定シーケンスは LSE クロックを使用して実行されます。
- LSE のみ (REFCLK_SEL = 1) および PCLK OFF
レジスタにはアクセスできません。測定は LSE クロックを使用して実行できます。このモードは、ハードウェアトリガおよび非同期割込みラインを使用して、SLEEP モードの終了に使用されます。

27.3.5 較正

温度センサは、周波数測定の前に、較正を実行する必要があります。較正は、クイック測定モード (DTS_CFGR1 で Q_MEAS_OPT を 1 にセット) の場合を除き、温度測定がトリガされると自動的に実行されます。

27.3.6 プリスケアラ

較正の実行中、カウンタクロックは 1MHz より遅くなければなりません。これは、温度センサに内蔵された PCLK クロックプリスケアラによって成されます。

温度の測定周期の間、プリスケアラはバイパスされます。

- PCLK をリファレンスクロックとして使用する場合は (DTS_CFGR1 の REFCLK_SEL を 0 にセット)、プリスケアラを使用します。分周比は最大 127 に設定する必要があります (分周器の設定については、HSREF_CLK_DIV[6:0] レジスタの定義を参照)。
- LSE をリファレンスクロックとして使用する場合 (DTS_CFGR1 の REFCLK_SEL を 1 にセット)、タイムベースは 2 LSE 周期と等しくなります。この場合、プリスケアラは使用されません。

27.3.7 温度測定の方法

温度センサのアナログ部分は、FM(T) 周波数が温度によって変わる信号 (CLK_PTAT) を出力します。

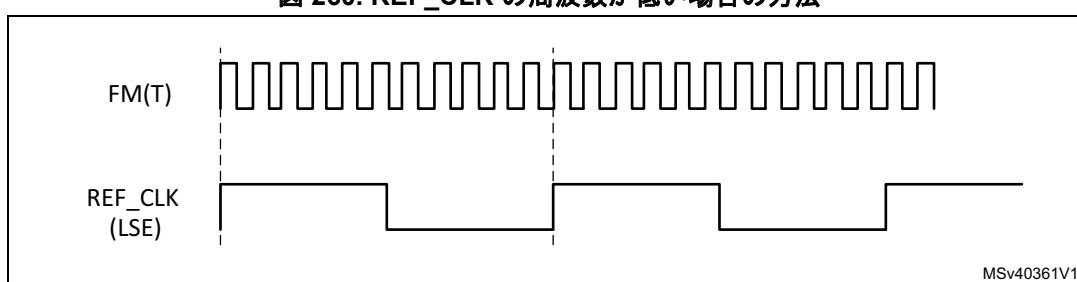
DTS_CFGR1 の REFCLK_SEL ビットによって、PCLK または LSE をリファレンスクロック (REF_CLK) として選択できます。

カウント手法は REF_CLK 周波数に依存します。これは、温度センサブロックに 2つのカウンタが実装されているためです。

- REF_CLK の周波数が低い場合、FM(T) サイクルのカウントは 1回以上の REF_CLK サイクル中に実行されます。
- REF_CLK の周波数が高い場合、REF_CLK サイクルのカウントは 1回以上の FM(T) サイクル中に実行されます。

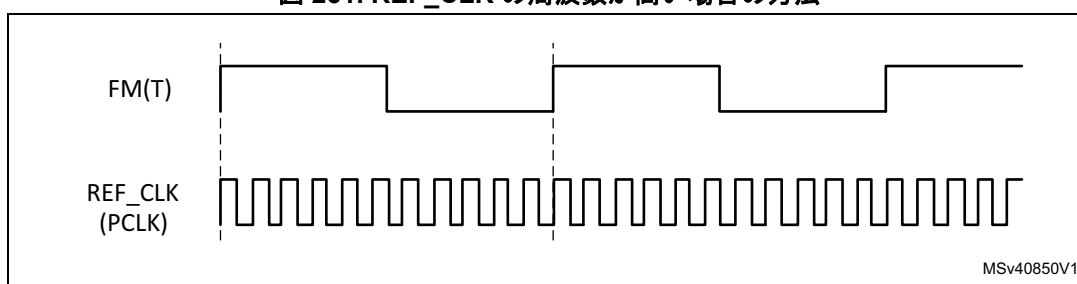
このカウンタの動作を図 260 および図 261 に示します。

図 260. REF_CLK の周波数が低い場合の方法



1. 精度向上のため、FM(T) 測定をいくつかの LSE 周期で実行できます。

図 261. REF_CLK の周波数が高い場合の方法



1. 精度向上のため、PCLK 測定をいくつかの FM(T) 周期で実行できます。

カウント結果は DTS_DR レジスタに格納されます (温度センサデータレジスタ (DTS_DR) を参照)。

一度 FM(T) 周波数を取得すると、ソフトウェアで次の公式を使用して対応する温度を計算できます。

- PCLK を使用する場合：

$$T = T_0 + ((F_{PCLK} / TS1_MFREQ) \times TS1_SMP_TIME - 100 \times TS1_FMT0) / TS1_RAMP_COEFF$$

ここで、

T_0 (工場校正温度) は 30°C。

TS1_FMT0 は DTS_T0VALR1 レジスタで測定され、保管されます。数百 Hz 単位で表されます。

TS1_RAMP_COEFF は工場でのテスト中に測定され、DTS_RAMPVALR レジスタに格納されます。この値は、Hz/°C 単位で表されます。

- LSE クロックを使用する場合：

$$T = T_0 + ((F_{LSE} \times TS1_MFREQ / TS1_SMP_TIME) - (100 \times TS1_FMT0)) / TS1_RAMP_COEFF$$

27.3.8 サンプルング時間

サンプルング期間は、測定精度を向上させるために増やすことができます。これは、リファレンス周波数 (REF_CLK) が FM(T) 周波数に近い場合に有効です。デフォルト値は、LSE モードで 1 REF_CLK サイクル、PCLK モードで 1 FM(T) サイクルです。

サンプルング時間は、DTS_CFGR1 レジスタの TS1_SMP_TIME ビットで設定します(表 258 を参照)

表 258. サンプルング時間の設定

TS1_SMP_TIME[3:0]	LSE または FM(T) クロックサイクル
0000	1
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

27.3.9 クイック測定モード

高精度が必要ない場合は、DTS_CFGR1 レジスタの Q_MEAS_OPT を 1 に設定することで、各測定シーケンスに含まれる較正ステップをスキップできます。この方法は、LSE をリファレンスクロックとして選択する (LSREF_CLK を 1 にセットする) 場合にのみ使用する必要があります。このモードでは測定時間が削減できます。

27.3.10 トリガ入力

温度測定は、ソフトウェアによって、または外部イベントによってトリガできます。トリガソースは、DTS_CFGR1の TS1_INTRIG[3:0] ビットによって選択できます。

表 259. トリガの設定

名前	TS1_INTRIG [3:0]				コメント
N/A	0	0	0	0	ハードウェアトリガなし
ts1_trg0	0	0	0	1	lptim1_ch1
ts1_trg1	0	0	1	0	lptim2_ch1
ts1_trg2	0	0	1	1	lptim3_ch1
ts1_trg3	0	1	0	0	exti13
ts1_trg4	0	1	0	1	予約済み
ts1_trg5	0	1	1	0	
ts1_trg6	0	1	1	1	
ts1_trg7	1	0	0	0	
ts1_trg8	1	0	0	1	
ts1_trg9	1	0	1	0	
ts1_trg10	1	0	1	1	
ts1_trg11	1	1	0	0	
ts1_trg12	1	1	0	1	
ts1_trg13	1	1	1	0	
ts1_trg14	1	1	1	1	

注： ハードウェアトリガは立上がりエッジでのみアクティブです。

温度センサは、TS1_RDY ビットがセットされている場合は、ハードウェアトリガの立上がりエッジのみをキャプチャできます（[セクション 27.3.11：オン／オフ制御およびレディフラグ](#)を参照。それ以外の場合、トリガは無視されます）。

トリガソースが動作中に変更される場合、新しいトリガソース信号はローでなければなりません。新しいソース信号がハイである場合、温度センサは立上がりエッジを検出し、測定シーケンスを開始します。

27.3.11 オン／オフ制御およびレディフラグ

DTS ブロックは DTS_CFGR1 レジスタの TS1_EN ビットをセットすることによって有効にできます。[温度センサステータスレジスタ \(DTS_SR\)](#) の TS1_RDY フラグは、DTS ブロックの温度測定準備ができていないことを示します。TS1_RDY ビットを 1 にセットすると、測定を開始できます。測定が開始されると、TS1_RDY ビットがリセットされます。これにより、新しい測定リクエストは無視されます。測定終了後、TS1_RDY ビットは再セットされ、センサで新しい測定を開始する準備ができたことを示します。

27.3.12 温度測定シーケンス

測定の開始は、ソフトウェアまたはハードウェアによってトリガできます。

ソフトウェアトリガ

ソフトウェアトリガは、DTS_CFGR1の TS1_INTRIG_SEL[3:0] を「0000」にセットすることによって選択されます。

TS1_RDY が 1 にセットされている場合、DTS_CFGR1の TS1_START ビットに 1 を書き込むことによって測定が開始されます。

TS1_RDY が 0 と等しい場合、ソフトウェアトリガは TS1_RDY がセットされるまで開始されません。

測定終了後に TS1_START ビットが 1 で保持される場合、TS1_RDY フラグは 1 になり、測定が再開されます。

ハードウェアトリガ

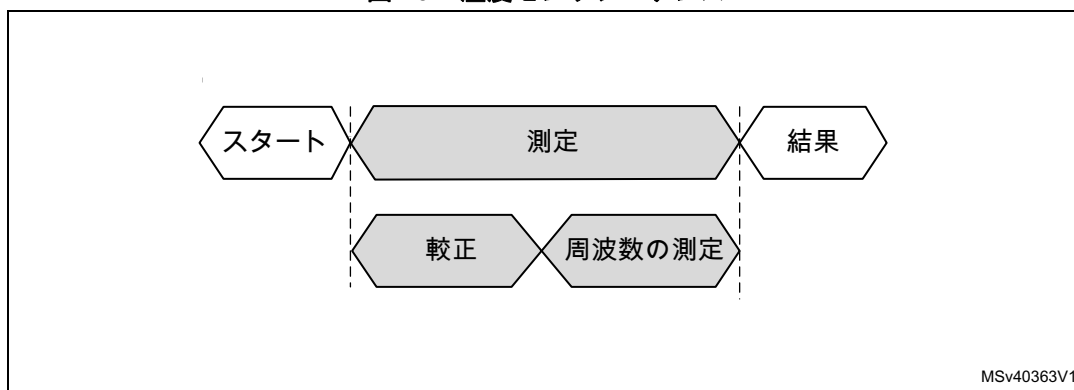
TS1_INTRIG_SEL[3:0] ビットにより、4つのハードウェアトリガのうちの 1つを選択できます。TS1_RDY が 1 にセットされている場合、トリガ信号の立上がりエッジにより測定が開始されます。TS1_RDY が 0 の場合、立上がりエッジは無視されます。

温度測定シーケンス

1つの測定は 2つのステップから成ります。アナログブロックの較正と測定です。較正は、測定がトリガされると自動的に実行されます（[セクション 27.3.5 : 較正](#)を参照）。測定周期は DTS_CFGR1 の次のビットによって決まります。

- REFCLK_SEL ビットによって選択されたリファレンスクロック
- HSREF_CLK_DIV ビットによって設定された分周比
- TS1_SMP_TIME ビットによって定義されたサンプリング時間

図 262. 温度センサシーケンス



27.4 DTS 低消費電力モード

表 260. 低消費電力モードでの温度センサの動作

モード	説明
SLEEP	LSE モードでのみ動作します。 DTS 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	LSE モードでのみ動作します。 DTS 割込みによって、デバイスは STOP モードから復帰します。

27.5 DTS 割込み

DTS ブロックは、2 つの方法で割込みソースとして使用されます。DTS 割込みラインは、EXTI コントローラ ([セクション 27.5.3: 非同期ウェイクアップ](#)を参照) または CPU NVIC ([セクション 27.5.2: 同期割込み](#)を参照) に接続することができます。

27.5.1 温度ウィンドウコンパレータ

DTS_ITR1 レジスタによって、温度の比較に使用される高閾値と低閾値が定義できます。温度データが TS1_HITTHD 以上、または TS1_LITTHD ビット以下である場合、割込みが生成され、対応するフラグ TS1_ITLF、TS1_ITHF、TS1_AITLF、および TS1_AITHF が DTS_SR レジスタにセットされます ([セクション 27.6.6](#)を参照)。

27.5.2 同期割込み

グローバル割込み出力ラインは、DTS ブロック上で使用可能です。割込みは、測定終了時、および/または測定結果が事前定義されている閾値以上または以下である場合に生成することができます ([セクション 27.5.1: 温度ウィンドウコンパレータ](#)を参照)。

DTS_ITENR レジスタの 3 つのビットで、3 つの割込みイベントを選択することができます ([セクション 27.6.7](#)を参照)。あらゆる組み合わせの割込みを使用できます。

DTS_SR レジスタの TS1_ITEF、TS1_ITLF、および TS1_ITHF フラグには、割込みイベントが反映されます。これらは、DTS_ICIFR レジスタの対応するビットによってリセットすることができます ([セクション 27.6.8](#)を参照)。

27.5.3 非同期ウェイクアップ

DTS ブロックでは、非同期割込みラインも提供しています。LSE をリファレンスクロックとして選択する場合にのみ使用されます (REFCLK_SEL=1)。

このラインは、測定終了時、および/または測定結果が事前定義されている閾値以上または以下である場合に、システムを SLEEP モードからウェイクアップする信号を生成することができます ([セクション 27.5.1: 温度ウィンドウコンパレータ](#)を参照)。

DTS_ITENR レジスタの 3 つのビットで、3 つの非同期ウェイクアップイベントを選択することができます。あらゆる組み合わせの割込みを使用できます。

DTS_SR レジスタの TS1_AITEF、TS1_AITLF、および TS1_AITHF フラグには、割込みステータスが反映されます。これらは、DTS_ICIFR レジスタの対応するビットによってリセットすることができます。

次の表に、割込みビットとその説明を示します。

表 261. 割込み制御ビット

割込みイベント	割込みフラグ	有効制御ビット	割込みクリアビット	SLEEP モードの終了	同期/非同期
測定終了時	DTS_SR の TS1_ITEF	DTS_ITENR の TS1_ITEEN	DTS_ICIFR の TS1_CITEF	不可	PCLK で同期
測定値が低閾値 以下である場合	DTS_SR の TS1_ITLF	DTS_ITENR の TS1_ITLEN	DTS_ICIFR の TS1_CITLF	不可	
測定値が高閾値 以上である場合	DTS_SR の TS1_ITHF	DTS_ITENR の TS1_ITHEN	DTS_ICIFR の TS1_CITHF	不可	
測定終了時	DTS_SR の TS1_AITEF	DTS_ITENR の TS1_AITEEN	DTS_ICIFR の TS1_CAITEF	はい	非同期
測定値が低閾値 以下である場合	DTS_SR の TS1_AITLF	DTS_ITENR の TS1_AITLEN	DTS_ICIFR の TS1_CAITLF	はい	
測定値が高閾値 以上である場合	DTS_SR の TS1_AITHF	DTS_ITENR の TS1_AITHEN	DTS_ICIFR の TS1_CAITHF	はい	

27.6 DTS レジスタ

このペリフェラルのレジスタには、ワード（32 ビット）単位でのみアクセスすることができます。

27.6.1 温度センサ設定レジスタ 1 (DTS_CFGR1)

DTS_CFGR1は、温度センサ 1に対する設定レジスタです。

アドレスオフセット：0x00

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	HSREF_CLK_DIV[6:0]							Res.	Res.	Q_MEAS_OPT	REFCLK_SEL	TS1_SMP_TIME [3:0]			
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TS1_INTRIG_SEL [3:0]				Res.	Res.	Res.	TS1_START	Res.	Res.	Res.	TS1_EN
				rw	rw	rw	rw				rw				rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **HSREF_CLK_DIV[6:0]**：高速クロック分周比

これらのビットは、ソフトウェアによってセット／クリアされます。これらは、較正に必要な 1MHz 未満の内部周波数を取得するための、メインクロックの分周比の定義に使用できます。PCLK がリファレンスクロックとして選択された場合の較正にのみ適用されます (REFCLK_SEL=0)。

0000000：分周器なし

0000001：分周器なし

0000010：1/2の分周比

.....

1111111：1/127の分周比

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **Q_MEAS_OPT** : クイック測定オプションビット

このビットは、ソフトウェアによってセット／クリアされます。校正ステップを抑制することにより測定速度を上げるために使用できます。LSE クロックをリファレンスクロックとして使用する場合にはのみ有効です (REFCLK_SEL=1)。

0 : 校正を伴う測定

1 : 校正を伴わない測定

ビット 20 **REFCLK_SEL** : リファレンスクロック選択ビット

このビットは、ソフトウェアによってセット／クリアされます。リファレンスクロックが高速クロック (PCLK) または低速クロック (LSE) であることを示します。

0 : 高速リファレンスクロック (PCLK)

1 : 低速リファレンスクロック (LSE)

ビット 19:16 **TS1_SMP_TIME[3:0]** : 温度センサ 1 のサンプリング時間

これらのビットによって、測定の精度を向上させるためにサンプリング時間を増加することができます。PCLK クロックがリファレンスクロックとして選択された場合 (REFCLK_SEL = 0)、測定は CLK_PTAT の TS1_SMP_TIME 周期で実行されます。

LSE がリファレンスクロックとして選択された場合 (REFCLK_SEL = 1)、測定は LSE の TS1_SMP_TIME 周期で実行されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TS1_INTRIG_SEL[3:0]** : 温度センサ 1 の入力トリガ選択ビット

これらのビットは、ソフトウェアによってセット／クリアされます。温度測定をトリガする入力を選択します。[セクション 27.3.10 : トリガ入力](#)を参照してください。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **TS1_START** : 温度センサ 1 の周波数測定の開始

このビットは、ソフトウェアによってセット／クリアされます。

0 : ソフトウェアトリガなし

1 : 周波数測定用のソフトウェアトリガあり (TS1 の準備ができている場合のみ)

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **TS1_EN** : 温度センサ 1 イネーブルビット

このビットは、ソフトウェアによってセット／クリアされます。

0 : 温度センサ 1 は無効です。

1 : 温度センサ 1 は有効です。

注 : 有効にすると、温度センサは特定の遅延時間経過後にアクティブになります。センサの準備ができると、TS1_RDY フラグがセットされます。

27.6.2 温度センサ T0 値レジスタ 1 (DTS_T0VALR1)

DTS_T0VALR1には、温度センサ 1の工場較正温度の値 (T0) が格納されています。リセット値は出荷時に調整されます。

アドレスオフセット : 0x08

リセット値 : 0x000X XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_T0 [1:0]	
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_FMT0 [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **TS1_T0[1:0]** : 温度センサ 1の T0温度の工学値。

00 : 30 °C

01 : 130 °C

その他 : 予約済み。使用できません。

ビット 15:0 **TS1_FMT0[15:0]** : 温度センサ 1の T0で測定された周波数の工学値

この値は、0.1kHz 単位で表されます。

27.6.3 温度センサのランプ値レジスタ (DTS_RAMPVALR)

DTS_RAMPVALR は、温度センサのランプ係数です。リセット値は出荷時に調整されます。

アドレスオフセット : 0x10

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_RAMP_COEFF [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TS1_RAMP_COEFF[15:0]** : 温度センサ 1のランプ係数の出荷時設定値

この値は、Hz/°C 単位で表されます。

27.6.4 温度センサ割込み閾値レジスタ 1 (DTS_ITR1)

DTS_ITR1には、センサ 1に対する閾値が格納されています。

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TS1_HITTHD [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_LITTHD [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **TS1_HITTHD[15:0]** : 温度センサ 1のロー割込み閾値

これらのビットは、ソフトウェアによってセット／クリアされます。割込み信号を生成する前には達成できない最高値を示します。

ビット 15:0 **TS1_LITTHD[15:0]** : 温度センサ 1の割込み低閾値

これらのビットは、ソフトウェアによってセット／クリアされます。割込み信号を生成する前には達成できない最低値を示します。

27.6.5 温度センサデータレジスタ (DTS_DR)

DTS_DR には、FM(T) 周波数の計算に使用する REF_CLK サイクルの数が格納されています。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_MFREQ [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TS1_MFREQ[15:0]** : 温度センサ 1のカウンタ出力値の値

27.6.6 温度センサステータスレジスタ (DTS_SR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1_RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHF	TS1_AITLF	TS1_AITEF	Res.	TS1_ITHF	TS1_ITLF	TS1_ITEF
r									r	r	r		r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TS1_RDY** : 温度センサ 1 のレディフラグ

このビットは、ハードウェアによってセット／リセットされます。

測定が実行中であることを示します。

0 : 温度センサ 1 はビジーです。

1 : 温度センサ 1 はレディ状態です。

ビット 14:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TS1_AITHF** : 温度センサ 1 の高閾値用の非同期割込みフラグ

このビットは、高閾値に達したときに、ハードウェアによってセットされます。

DTS_ICIFR レジスタの TS1_CAITHF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で高閾値に達しません。

1 : 温度センサ 1 で高閾値に達します。

注 : このビットは、TS1_AITHFEN ビットがセットされたときにのみアクティブとなります。

ビット 5 **TS1_AITLF** : 温度センサ 1 の低閾値用の非同期割込みフラグ

このビットは、低閾値に達したときに、ハードウェアによってセットされます。

DTS_ICIFR レジスタの TS1_CAITLF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で低閾値に達しません。

1 : 温度センサ 1 で低閾値に達します。

注 : このビットは、TS1_AITLFEN ビットがセットされたときにのみアクティブとなります。

ビット 4 **TS1_AITEF** : 温度センサ 1 の測定終了時用の非同期割込みフラグ

このビットは、温度測定の終了時に、ハードウェアによってセットされます。

DTS_ICIFR レジスタの TS1_CAITEF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で測定終了が検出されません。

1 : 温度センサ 1 で測定終了が検出されました。

注 : このビットは、TS1_AITEFEN ビットがセットされたときにのみアクティブとなります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TS1_ITHF** : PCLK 上で同期された温度センサ 1 の高閾値用の割込みフラグ

このビットは、高閾値がセットされ、達成されたときに、ハードウェアによってセットされます。
DTS_ICIFR レジスタの TS1_CITHF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で高閾値に達しません。

1 : 温度センサ 1 で高閾値に達します。

注 : このビットは、TS1_ITHFEN ビットがセットされたときにのみアクティブとなります。

ビット 1 **TS1_ITLF** : PCLK 上で同期された温度センサ 1 の低閾値用の割込みフラグ。

このビットは、低閾値がセットされ、達成されたときに、ハードウェアによってセットされます。
DTS_ICIFR レジスタの TS1_CITLF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で低閾値に達しません。

1 : 温度センサ 1 で低閾値に達します。

注 : このビットは、TS1_ITLFEN ビットがセットされたときにのみアクティブとなります。

ビット 0 **TS1_ITEF** : PCLK 上で同期された温度センサ 1 の測定終了時用の割込みフラグ。

このビットは、温度測定の終了時に、ハードウェアによってセットされます。

DTS_ICIFR レジスタの TS2_CITEF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : 温度センサ 1 で測定終了が検出されません。

1 : 温度センサ 1 で測定終了が検出されました。

注 : このビットは、TS1_ITEFEN ビットがセットされたときにのみアクティブとなります。

27.6.7 温度センサ割込み有効レジスタ (DTS_ITENR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_	TS1_	TS1_	Res.	TS1_	TS1_	TS1_
									AITHEN	AITLEN	AITEEN		ITHEN	ITLEN	ITEEN
									rw	rw	rw		rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TS1_AITHEN** : 温度センサ 1 の高閾値の非同期割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

温度が高閾値を超える場合に、非同期割込みを有効にします (REFCLK_SEL = 1 の場合のみ)。

0 : 温度センサ 1 のために無効化された高閾値での非同期割込み

1 : 温度センサ 1 のために有効化された高閾値での非同期割込み

ビット 5 **TS1_AITLEN** : 温度センサ 1 の低閾値用の非同期割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

温度が低閾値未満の場合に、非同期割込みを有効にします (REFCLK_SEL = 1 の場合のみ)。

0 : 温度センサ 1 のために無効化された低閾値での非同期割込み

1 : 温度センサ 1 のために有効化された低閾値での非同期割込み

ビット 4 **TS1_AITEEN** : 温度センサ 1 の測定終了時用の非同期割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

測定終了時用の非同期割込みを有効にします (REFCLK_SEL = 1 の場合のみ)。

0 : 温度センサ 1 で無効化された測定終了時用の非同期割込み

1 : 温度センサ 1 で有効化された測定終了時用の非同期割込み

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TS1_ITHEN** : PCLK 上で同期された温度センサ 1 の高閾値用の割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

測定が高閾値に達した場合、または超えた場合に、割込みを有効にします。

0 : 温度センサ 1 で無効化された高閾値用の同期割込み

1 : 温度センサ 1 で有効化された高閾値用の同期割込み

ビット 1 **TS1_ITLEN** : PCLK 上で同期された温度センサ 1 の低閾値用の割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

測定が低閾値に達した場合、または低閾値未満の場合に、同期割込みを有効にします。

0 : 温度センサ 1 で無効化された低閾値用の同期割込み

1 : 温度センサ 1 で有効化された低閾値用の同期割込み

ビット 0 **TS1_ITEEN** : PCLK 上で同期された温度センサ 1 の測定終了時用の割込みイネーブルフラグ

このビットは、ソフトウェアによってセット／クリアされます。

測定終了時用の同期割込みを有効にします。

0 : 温度センサ 1 で無効化された測定終了時用の同期割込み

1 : 温度センサ 1 で有効化された測定終了時用の同期割込み

27.6.8 温度センサ割込みクリアフラグレジスタ (DTS_ICIFR)

DTS_ICIFR は、割込みフラグの制御レジスタです。

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_	TS1_	TS1_	Res.	TS1_	TS1_	TS1_
									CAITHF	CAITLF	CAITEF		CITHF	CITLTF	CITEF
									rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TS1_CAITHF** : 温度センサ 1 の高閾値用の非同期割込みクリアフラグ

このビットに 1 を書き込むと、DTS_SR レジスタの TS1_AITHF フラグがクリアされます。

ビット 5 **TS1_CAITLF** : 温度センサ 1 の低閾値用の非同期割込みクリアフラグ

このビットに 1 を書き込むと、DTS_SR レジスタの TS1_AITLF フラグがクリアされます。

ビット 4 **TS1_CAITEF** : 1 回だけ書き込み可能なビット。温度センサ 1 の測定終了時用の非同期 IT フラグをクリアします。

このビットに 1 を書き込むと、DTS_SR レジスタの TS1_AITEF フラグがクリアされます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TS1_CITHF** : 温度センサ 1 の高閾値用の割込みクリアフラグ

このビットに 1 を書き込むと DTS_SR レジスタの TS1_ITHF フラグがクリアされます。

ビット 1 **TS1_CITLF** : 温度センサ 1 の低閾値用の割込みクリアフラグ

このビットに 1 を書き込むと、DTS_SR レジスタの TS1_ITLF フラグがクリアされます。

ビット 0 **TS1_CITEF** : 温度センサ 1 の測定終了時用の割込みクリアフラグ

このビットに 1 を書き込むと、DTS_SR レジスタの TS1_ITEF フラグがクリアされます。

27.6.9 温度センサ・オプション・レジスタ (DTS_OR)

DTS_OR には、汎用オプションビットが格納されます。

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TS_OP3 1	TS_OP3 0	TS_OP2 9	TS_OP2 8	TS_OP2 7	TS_OP2 6	TS_OP2 5	TS_OP2 4	TS_OP2 3	TS_OP2 2	TS_OP2 1	TS_OP2 0	TS_OP1 9	TS_OP1 8	TS_OP1 7	TS_OP1 6
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_OP1 5	TS_OP1 4	TS_OP1 3	TS_OP1 2	TS_OP1 1	TS_OP1 0	TS_OP9	TS_OP8	TS_OP7	TS_OP6	TS_OP5	TS_OP4	TS_OP3	TS_OP2	TS_OP1	TS_OP0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **TS_OP[31:0]** : 汎用オプションビット

27.6.10 DTS レジスタマップ

次の表に温度センサレジスタの一覧を示します。

表 262. DTS レジスタのマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	DTS_CFGR1	Res.	HSREF_CLK_DIV [6:0]								Res.	Res.	Q_MEAS_OPT	REFCLK_SEL	TS1_SMP_TIME [3:0]				Res.	Res.	Res.	Res.	TS1_INTRIG_SEL [3:0]				Res.	Res.	Res.	TS1_START	Res.	Res.	Res.	TS1_EN
	リセット値		0	0	0	0	0	0	0			0	0	0	0	0	0						0	0	0	0				0				0
0x04	予約済みです。																																	
0x08	DTS_T0VALR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_T0 [1:0]	TS1_FMT0 [15:0]																
	リセット値																X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x0C	予約済みです。																																	
0x10	DTS_RAMPVALR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_RAMP_COEFF [15:0]																
	リセット値																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x14	DTS_ITR1	TS1_HITTHD [15:0]																TS1_LITTHD [15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	予約済みです。																																	
0x1C	DTS_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_MFREQ [15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	DTS_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_RDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHF	TS1_AITLF	TS1_AITEF	Res.	TS1_ITHF	TS1_ITLF	TS1_ITEF
	リセット値																	0										0	0	0		0	0	0
0x24	DTS_ITENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_AITHEN	TS1_AITLEN	TS1_AITEEN	Res.	TS1_ITHEN	TS1_ITLEN	TS1_ITEEN
	リセット値																											0	0	0		0	0	0
0x28	DTS_ICIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1_CAITHF	TS1_CAITLF	TS1_CAITEF	Res.	TS1_CITHF	TS1_CITLF	TS1_CITEF
	リセット値																											0	0	0		0	0	0
0x2C	DTS_OR	TS_OP31	TS_OP30	TS_OP29	TS_OP28	TS_OP27	TS_OP26	TS_OP25	TS_OP24	TS_OP23	TS_OP22	TS_OP21	TS_OP20	TS_OP19	TS_OP18	TS_OP17	TS_OP16	TS_OP15	TS_OP14	TS_OP13	TS_OP12	TS_OP11	TS_OP10	TS_OP9	TS_OP8	TS_OP7	TS_OP6	TS_OP5	TS_OP4	TS_OP3	TS_OP2	TS_OP1	TS_OP0	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

28 DAC (DAC)

28.1 概要

DAC モジュールは、12 ビットの電圧出力デジタルアナログコンバータです。DAC は、8 または 12 ビットモードで設定でき、DMA コントローラと組み合わせて使用することもできます。12 ビットモードでは、データを左詰めまたは右詰めにできます。DAC には2つの出力チャンネルがあり、それぞれがコンバータを搭載しています。デュアル DAC チャンネルモードでは、変換は独立して行うか、両方のチャンネルが同期更新操作のためにグループ化されているときには同時に行うことができます。精度を高めるために、入力基準ピン V_{REF+} (その他のアナログペリフェラルと共用) を使用することができます。内部基準も、同じ入力でセットできます。電圧基準バッファ (VREFBUF) セクションを参照してください。

駆動出力電流を上げるために、オプションで DAC 出力バッファを有効にできます。個別の較正が各 DAC 出力チャンネルに適用可能です。DAC出力チャンネルは低消費電力モード (サンプルおよびホールドモード) をサポートしています。

28.2 DAC の主な機能

DAC の主な機能は以下のとおりです (図 263 : デュアルチャンネル DAC ブロック図を参照)。

- 1つの DAC インタフェースで、最大2つの出力チャンネル
- 12 ビットモードでのデータの左詰めまたは右詰め
- 同期更新機能
- ノイズ波および三角波生成
- デュアル DAC チャンネルの独立または同時変換
- DMA アンダーランエラー検出を含む各チャンネルの DMA 機能
- バスへの負担を軽減するダブルデータ DMA 機能
- 変換外部トリガ
- DAC 出力チャンネルバッファ/バッファ無モード
- バッファオフセット較正
- STOP モードの低消費電力動作でのサンプルおよびホールドモード
- V_{REF+} ピンまたは内部 VREFBUF 基準からの入力電圧基準

図 263 は DAC チャンネルのブロック図を、表 264 はピンの概要を示します。

28.3 DAC の実装

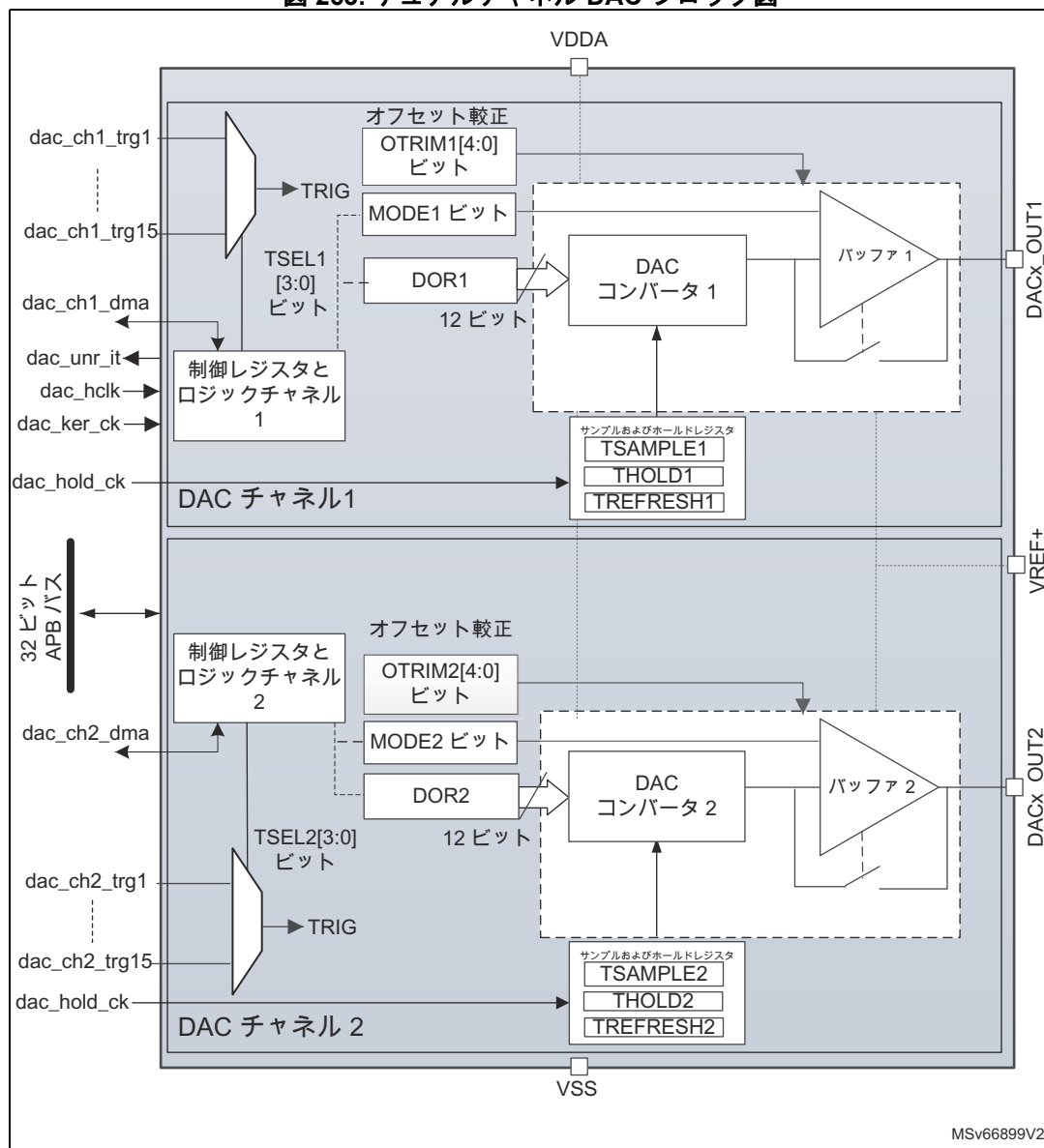
表 263. DAC の機能

DAC の機能	DAC1
デュアルチャネル	X
出力バッファ	X
I/O 接続	PA4 へ DAC1_OUT1、PA5 へ DAC1_OUT2
最大サンプリング時間	1 Msps
自律モード	-
VREF+ ピン	X

28.4 DAC の機能説明

28.4.1 DAC ブロック図

図 263. デュアルチャネル DAC ブロック図



MSv66899V2

1. DAC_MCR の MODEx ビットによって、出力モードが制御され、バッファ有/バッファ無し設定の通常モードとサンプルおよびホールドモードを切り替えることができます。
2. チャンネル2の有無については、[セクション 28.3 : DAC の実装](#)を参照してください。

28.4.2 DAC ピンおよび内部信号

DAC には次の要素が含まれます。

- 最大 2 つの出力チャンネル
- DAC 出力チャンネルのバッファ有またはバッファ無し
- スタティック変換用 LSI/LSE クロックソース (dac_hold_ck) を使用した、STOP モードで動作可能なサンプルおよびホールドブロック、レジスタ。

DAC には最大 2 つの個別出力チャンネルが搭載されています。各出力チャンネルは、コンパレータ、オペアンプ、および ADC（使用できる場合）などの内部ペリフェラルに接続できます。この場合、DAC 出力チャンネルは DACx_OUTy 出力ピンから切断でき、対応する GPIO を別の目的に使用できます。

DAC 出力はバッファの有無を設定できます。サンプルホールドブロックと、それに関連するレジスタは、LSI/LSE クロックソース (dac_hold_ck) を使用して STOP モードで動作可能です。

表 264. DAC の入出力ピン

ピン名	信号タイプ	説明
VREF+	入力、アナログ基準電圧正	DAC のハイレベル/正基準電圧、 $V_{REF+} \leq V_{DDAmax}$ （データシートを参照）
VDDA	入力、アナログ電源供給	アナログ電源供給
VSSA	入力、アナログ供給グラウンド	アナログ電源供給のグラウンド
DACx_OUTy	アナログ出力信号	DACx チャンネル y アナログ出力

表 265. DAC 内部入力／出力信号

内部信号名	信号タイプ	説明
dac_ch1_dma	双方向	DACチャンネル 1 DMA リクエスト／確認応答
dac_ch2_dma	双方向	DACチャンネル 2 DMA リクエスト／確認応答
dac_ch1_trgx (x = 1~15)	入力	DAC channel1 トリガ入力
dac_ch2_trgx (x = 1~15)	入力	DAC channel2 トリガ入力
dac_unr_it	出力	DAC アンダーラン割込み
dac_hclk	入力	DAC ペリフェラルクロック
dac_ker_ck	入力	DAC カーネルクロック
dac_hold_ck	入力	サンプルおよびホールドモードで使用される DAC低消費電力クロック

表 266. DAC の相互接続

信号名	転送元	ソースタイプ
dac_hold_ck	ck_lsi または ck_lse	RCC で選択された LSI または LSE クロック
dac_chx_trg1 (x = 1、2)	tim1_trgo	内部タイマからの内部信号
dac_chx_trg2 (x = 1、2)	tim2_trgo	内部タイマからの内部信号
dac_chx_trg3 (x = 1、2)	tim4_trgo	内部タイマからの内部信号
dac_chx_trg4 (x = 1、2)	tim5_trgo	内部タイマからの内部信号
dac_chx_trg5 (x = 1、2)	tim6_trgo	内部タイマからの内部信号

表 266. DAC の相互接続 (続き)

信号名	転送元	ソースタイプ
dac_chx_trg6 (x = 1, 2)	tim7_trgo	内部タイマからの内部信号
dac_chx_trg7 (x = 1, 2)	tim8_trgo	内部タイマからの内部信号
dac_chx_trg8 (x = 1, 2)	tim15_trgo	内部タイマからの内部信号
dac_chx_trg11 (x = 1, 2)	lptim1_ch1	内部タイマからの内部信号
dac_chx_trg12 (x = 1, 2)	lptim2_ch1	内部タイマからの内部信号
dac_chx_trg13 (x = 1, 2)	exti9	外部ピン

28.4.3 DAC クロック

2 つのクロックソースを使用して、DAC を更新できます。

- dac_hclk : DAC ペリフェラルクロック (AHB クロック)
- dac_ker_ck : DAC カーネルクロック : このクロックを使用して DAC と ADC を同期させることができます。
- dac_hold_ck : サンプルおよびホールドモードで使用される低消費電力クロック

DAC クロックは RCC で選択されます。

28.4.4 DAC チャネルイネーブル

各 DAC チャネルは、DAC_CR レジスタの対応する ENx ビットをセットすることによって起動できます。DAC チャネルは、t_{WAKEUP} のスタートアップ時間後に有効になります。

DAC インタフェースがデータを受信する準備ができており、DAC_SR レジスタの DAC x RDY ビットがセットされます。DACxRDY 信号がリセットされているときに ENx ビットがセットされている場合、新しいデータの書き込みまたはトリガのアサートはできません。

注 : ENx ビットは、アナログ DAC チャネル x のみを有効にします。DAC チャネル x デジタルインタフェースは、ENx ビットがリセットされた場合でも有効になります。

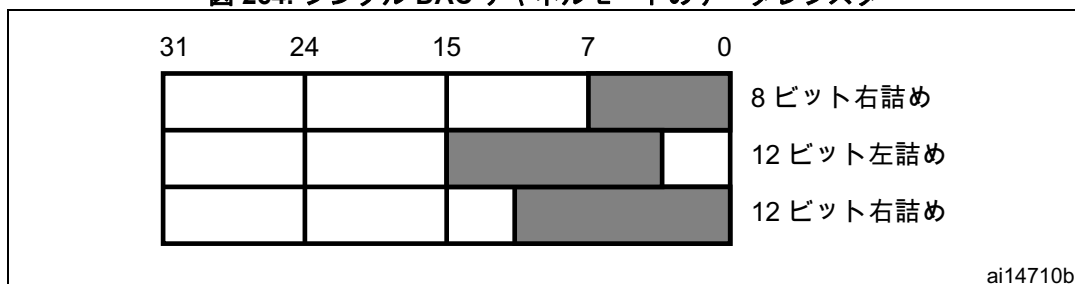
28.4.5 DAC データフォーマット

以下に示すように、選択された設定モードに応じて、指定されたレジスタにデータを書き込む必要があります。

- シングル DAC チャネル
この場合、次の 3 つの設定が可能です。
 - 8 ビット右詰め : ソフトウェアはデータを DAC_DHR8Rx[7:0] ビットにロードする必要があります (DHRx[11:4] ビットに格納)。
 - 12 ビット左詰め : ソフトウェアはデータを DAC_DHR12Lx [15:4] ビットにロードする必要があります (DHRx[11:0] ビットに格納)。
 - 12 ビット右詰め : ソフトウェアはデータを DAC_DHR12Rx [11:0] ビットにロードする必要があります (DHRx [11:0] ビットに格納)。

ユーザによって書き込まれたデータは、ロードされた DAC_DHRyyyx レジスタに応じて、シフトされてから、対応する DHRx (メモリマップされない内部レジスタであるデータ保持レジスタ x) に格納されます。その後、DHRx レジスタは自動的に、ソフトウェアトリガまたは外部イベントトリガによって、DORx レジスタにロードされます。

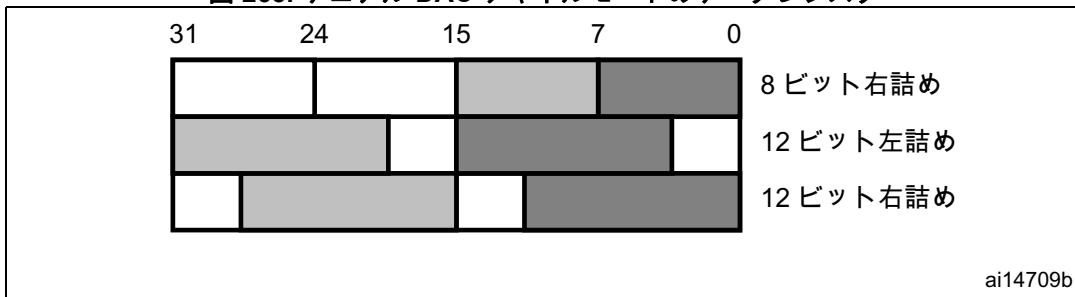
図 264. シングル DAC チャンネルモードのデータレジスタ



- デュアル DAC チャンネル（使用できる場合）
この場合、次の 3 つの設定が可能です。
 - 8 ビット右詰め: DAC チャンネル 1 のデータは、DAC_DHR8RD [7:0] ビット (DHR1[11:4] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR8RD [15:8] ビット (DHR2[11:4] ビットに格納) にロードされます。
 - 12 ビット左詰め: DAC チャンネル 1 のデータは、DAC_DHR12LD [15:4] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR12LD [31:20] ビット (DHR2[11:0] ビットに格納) にロードされます。
 - 12 ビット右詰め: DAC チャンネル 1 のデータは、DAC_DHR12RD [11:0] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR12RD [27:16] ビット (DHR2[11:0] ビットに格納) にロードされます。

ロードされた DAC_DHRyyyD レジスタに応じて、ユーザによって書き込まれたデータは、シフトされてから、DHR1 および DHR2 (Data Holding Register、メモリマップされない内部レジスタ) に格納されます。その後、DHR1 および DHR2 レジスタは、自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、それぞれ DAC_DOR1 および DOR2 レジスタにロードされます。

図 265. デュアル DAC チャンネルモードのデータレジスタ



符号付き/符号なしデータ

DAC 入力データは符号なし: 12 ビットモードでは、0x000 は最小値に対応し、0xFFFF は最大値に対応します。

DAC は 2 の補数形式の符号付き入力データも扱うことができます。このためには、DAC_MCR レジスタの SINFORMATx ビットをセットします。

SINFORMATx ビットがセットされると、DHRx レジスタに書き込まれたデータの MSB ビットは、DAC_DORx レジスタにコピーされるときに反転され、DAC インタフェースは符号付データを受け入れることができます (Q1.15、Q1.11、または Q1.7 形式)。DAC_DHR12Lx レジスタを使用して、16 ビット符号付きデータをデータ保持レジスタに格納することができます。16 ビットデータの 12 MSB は、DAC 出力データに使用され、MSB ビットは反転されます。4 つの LSB は単純に無視されます。

表 267. データフォーマット (12 ビットデータの場合)

SINFORMATx ビット	DHRx レジスタに書き込まれるデータ	DORx レジスタに転送されるデータ
0	0x000	0x000
0	0xFFFF	0xFFFF
1	0x7FF	0xFFFF
1	0x000	0x800
1	0xFFFF	0x7FF
1	0x800	0x000

28.4.6 DAC 変換

DAC_DORx に直接書き込むことはできず、DAC_DHRx レジスタをロードすることによって (DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12RD、または DAC_DHR12LD への書き込み操作)、DAC チャンネル x へのデータ転送を行う必要があります。

DAC_DHRx レジスタに格納されたデータは、ハードウェアトリガが選択されていない (DAC_CR レジスタの TENx ビットがリセットされている) 場合に、1 dac_hclk クロックサイクル後に DAC_DORx レジスタに自動的に転送されます。ただし、ハードウェアトリガが選択されている (DAC_CR レジスタの TENx ビットがセットされている) ときには、トリガが発生すると、転送はトリガ信号の 3 dac_hclk クロックサイクル後に行われます。

DAC_DORx に DAC_DHRx の内容がロードされると、電源電圧とアナログ出力負荷に応じて決定される t_{SETTLING} 時間後にアナログ出力電圧が使用可能になります。

DAC と ADC を同期させるために、両ペリフェラルに同じクロックソースを使用することができます。これは、RCC で dac_hclk クロック (AHB クロック) の代わりに dac_ker_ck クロックを選択することによって行われます。

dac_hclk または dac_ker_ck クロック速度が 80 MHz を超える場合は、DAC_MCR の HFSEL ビットをセットする必要があります。それによって、DAC_DHRx レジスタから DAC_DORx レジスタへの転送に余分の遅延が追加されます。

HFSEL ビットと dac_hclk クロック周波数に応じた DAC_DORx の更新速度の制限については、以下の表 HFSEL の説明 を参照してください。

許されない期間にデータが更新された場合、またはソフトウェア/ハードウェアトリガイベントが発生した場合、ペリフェラルの動作は予期できません。

上記のタイミングは、DAC インタフェースの制限にのみ関係します。製品データシートの t_{SETTLING} パラメータ値も参照してください。

表 268. HFSEL の説明

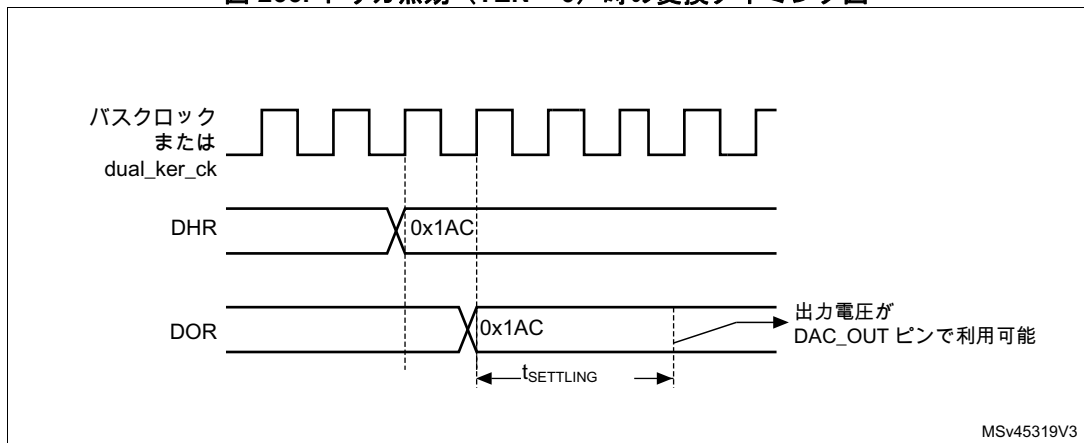
HFSEL [1:0]	AHB 周波数	AHB クロック (dac_hclk) を使用したレイテンシ	dac_ker_ck クロックを使用したレイテンシ	機能
00	< 80 MHz	3	4	DAC_DOR の更新速度は最大 3 AHB クロックサイクルまたは 4 dac_ker_ck サイクル。
01	$\geq 80 \text{ MHz}^{(1)}$	5	5	DAC_DOR の更新速度は最大 5 AHB クロックまたは 5 dac_ker_ck サイクル。

表 268. HFSEL の説明 (続き)

HFSEL [1:0]	AHB 周波数	AHB クロック (dac_hclk) を 使用したレイテンシ	dac_ker_ck クロックを使用した レイテンシ	機能
10	≥ 160 MHz	7	6	DAC_DOR の更新速度は最大 7 AHB クロックサイクルまたは 6 dac_ker_ck サイクル。
11	予約済み	-	-	-

1. 最大の AHB 周波数の値については、デバイスデータシートを参照してください。

図 266. トリガ無効 (TEN = 0) 時の変換タイミング図



MSv45319V3

28.4.7 DAC 出力電圧

デジタル入力は、0 から V_{REF+} までのリニア変換で出力電圧に変換されます。

各 DAC チャンネルピンのアナログ出力電圧は、次の式によって求められます。

$$DAC_{output} = V_{REF} \times \frac{DOR}{4096}$$

28.4.8 DAC トリガ選択

TENx 制御ビットがセットされている場合、外部イベント (タイマカウンタ、外部割込みラインなど) によって変換をトリガできます。DAC_CR レジスタの TSELx[3:0] ビットに示すように、TSELx[3:0] 制御ビットは、変換をトリガするイベントを 16 のイベントから決定します。これらのイベントは、ソフトウェアトリガまたはハードウェアトリガのいずれかです。[セクション 28.4.2 : DAC ピンおよび内部信号](#)の相互接続の表を参照してください。

DAC インタフェースが選択されたトリガソース (次の表を参照) で立上がりエッジを検出するたびに、DAC_DHRx レジスタに最後に格納されたデータが DAC_DORx レジスタに転送されます。DAC_DORx レジスタは、トリガが発生してから 3 dac_hclk サイクル後に更新されます。

ソフトウェアトリガが選択されている場合、変換は、SWTRIG ビットがセットされると開始されます。SWTRIG ビットは、DAC_DORx レジスタの内容が DAC_DHRx レジスタにロードされると、ハードウェアによってリセットされます。

注 : ENx ビットがセットされているときには、TSELx[3:0] ビットを変更することはできません。
ソフトウェアトリガが選択されているときには、DAC_DHRx レジスタから DAC_DORx レジスタへの転送は、わずか 1 dac_hclk クロックサイクルで行われます。

28.4.9 DMA リクエスト

各 DAC チャンネルは、DMA 機能を備えています。DAC チャンネルの DMA リクエストは、2 つの DMA チャンネルを使用して処理されます。

DMAENx ビットがセットされているときに、外部トリガ（ソフトウェアトリガ以外）が発生した場合、DAC_DHRx レジスタの値は DAC_DORx レジスタに転送され、転送が終了すると、DMA リクエストが生成されます。

デュアルモードでは、両方の DMAENx ビットがセットされている場合、2 つの DMA リクエストが生成されます。1 つの DMA リクエストしか必要ない場合には、対応する DMAENx ビットだけをセットする必要があります。これにより、アプリケーションは 1 つの DMA リクエストと一意な DMA チャンネルを使用して、両方の DAC チャンネルをデュアルモードで管理することができます。

DAC_DHRx から DAC_DORx へのデータ転送は DMA リクエストの前に起きるので、最初のトリガイベントが発生する前に最初のデータを DAC_DHRx に書き込む必要があります。

DMA アンダーラン

DAC DMA リクエストは待ち行列で管理されないため、最初の外部トリガに対する確認応答が受信される（最初のリクエスト）前に 2 つ目の外部トリガが発生すると、新しいリクエストは発行されず、DAC_SR レジスタの DMA チャンネル x アンダーランフラグ DMAUDRx がセットされてエラー条件を報告します。DAC チャンネル x は、古いデータを変換し続けます。

ソフトウェアは、1 を書き込むことによって DMAUDRx フラグをクリアし、使用された DMA ストリームの DMAEN ビットをクリアし、DMA と DAC の両方のチャンネル x を再初期化して転送を正しくリスタートさせる必要があります。また、DAC トリガ変換周波数を変更するか DMA の負荷を軽減して、新しい DMA アンダーランを回避する必要があります。最後に、DMA データ転送と変換トリガを有効にすることによって DAC 変換を再開することができます。

各 DAC チャンネル x では、DAC_CR レジスタの対応する DMAUDRIEx ビットが有効にされている場合、割り込みも生成されます。

DMA ダブルデータモード

DMA コントローラが通常モードで使用されるときには、DMA リクエストによって 12 ビット（または 8 ビット）データのみが転送されます。AHB 幅は 32 ビットなので、2 つの 12 ビットデータを同時に転送できます。このモードを使用するには、DAC_MCR レジスタの DMADOUBLEx ビットをセットします。

DAC DMA リクエストは、DMAENx ビットがセットされているとき、外部トリガ（ソフトウェアトリガでなく）が 2 つ発生するたびに生成されます。

1. 最初のトリガが検出されると、DAC_DHRx および DAC_DHRBx レジスタの値が DAC_DORx および DAC_DORBx レジスタに転送されます。実際の DAC データが DAC_DORx レジスタにロードされます。その後、DMA リクエストが生成されます。DMA は、新しいデータを DAC_DHRx および DAC_DHRBx データレジスタに書き込みます。
2. 次のトリガが検出されると、実際の DAC データが DAC_DHRBx レジスタにロードされます。この 2 つ目のトリガでは、DMA リクエストは生成されません。DORSTATx ビットは、アナログ DAC 入力に実際にロードされる DOR データを示します。

DMA ダブルデータモードでは、DMA アンダーラン機能もサポートされます。

DMA ダブルモードでは、DMA リクエストは 1 つの DAC チャンネルのみを処理できます。DMA ダブルモードで 2 つのチャンネル出力を使用するには、各 DMA チャンネルを個別に設定する必要があります。

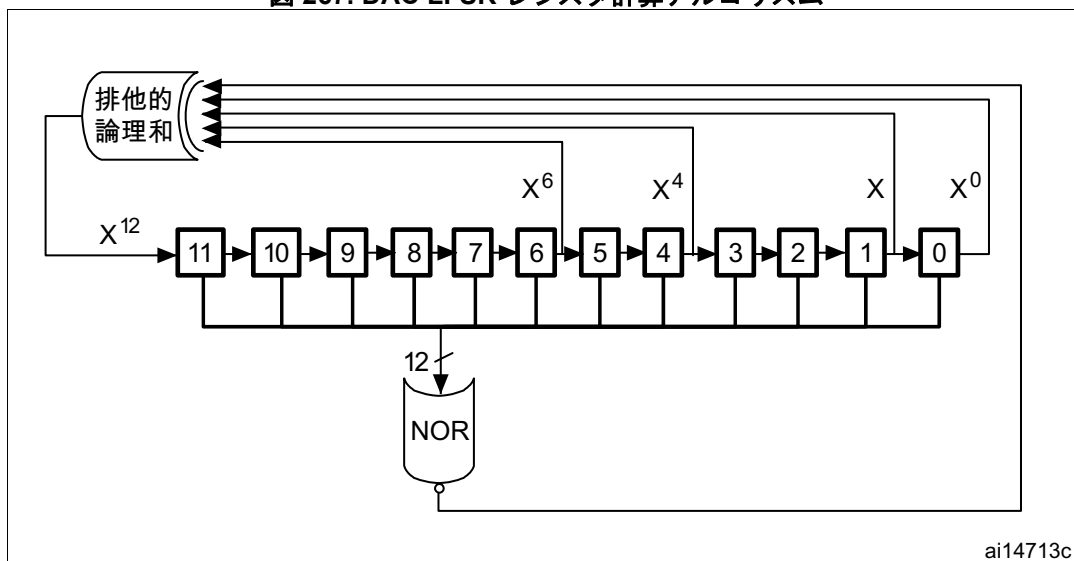
ダブルデータモードからシングルデータモードに、またはその逆に変更するには、以下の条件が満たされている必要があります。

- DAC を無効にする必要があります。
- DMAEN ビットをクリアする必要があります (ENx = 0 かつ DMAEN = 0)。

28.4.10 ノイズ生成

リニアフィードバックシフトレジスタ (LFSR) を使用して、可変振幅の擬似ノイズを生成することができます。DAC ノイズ生成を選択するには、WAVEx[1:0] を 01 にセットします。LFSR にプリロードされる値は 0xAAA です。このレジスタは、各トリガイイベントの 3 dac_hclk クロックサイクル後に、特定の計算アルゴリズムに従って更新されます。

図 267. DAC LFSR レジスタ計算アルゴリズム

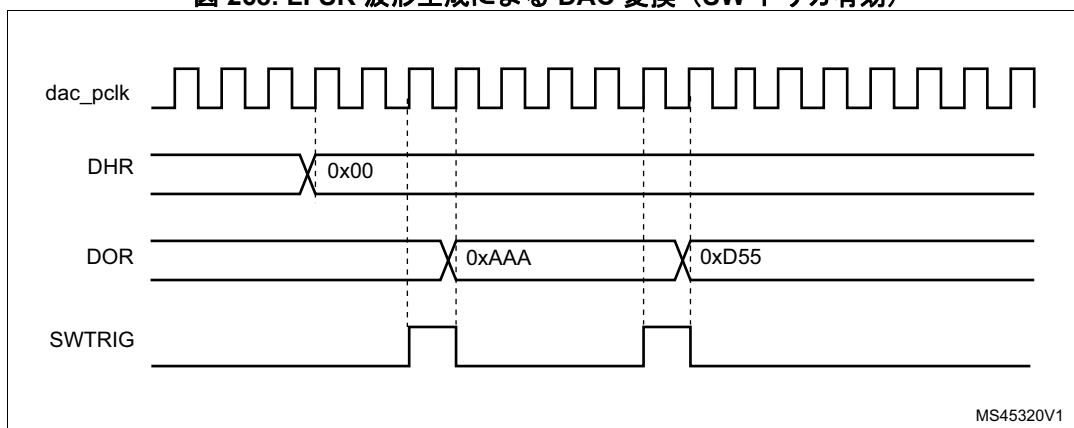


LFSR 値は、DAC_CR レジスタの MAMPx[3:0] ビットによって部分的または全体的にマスクでき、オーバーフローなしに DAC_DHRx の内容に加算され、DAC_DORx レジスタに転送されます。

LFSR が 0x0000 の場合、“1”がインジェクトされます (アンチロックアップメカニズム)。

WAVEx[1:0] ビットをリセットすることによって、LFSR 波形生成をリセットできます。

図 268. LFSR 波形生成による DAC 変換 (SW トリガ有効)



注： ノイズ生成のためには、DAC_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

28.4.11 三角波生成

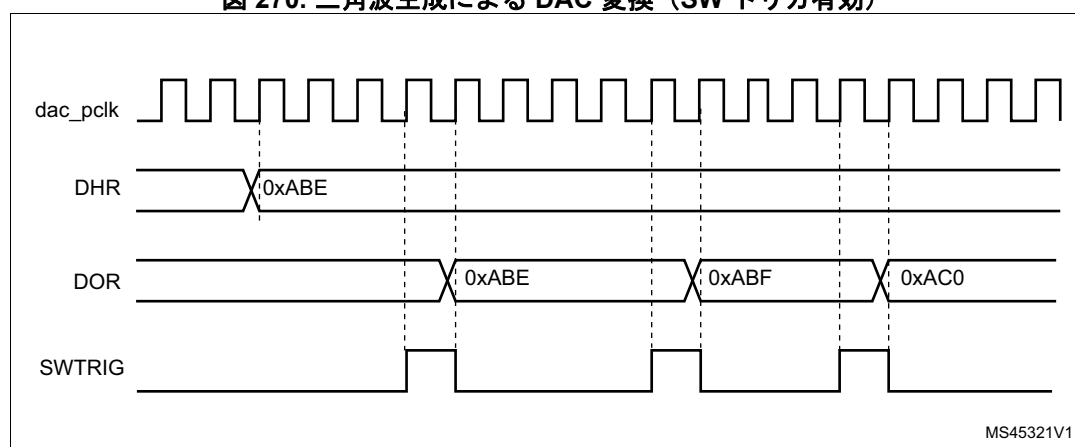
DC または低周波数信号上に、小さな振幅の三角波を追加することが可能です。DAC 三角波生成を選択するには、WAVEx[1:0] を 10 にセットします。振幅は、DAC_CR レジスタの MAMPx[3:0] ビットによって設定されます。内部三角波カウンタは、各トリガイベントの 3 dac_hclk クロックサイクル後にインクリメントされます。このカウンタの値は、オーバーフローなしに DAC_DHRx レジスタに加えられ、合計は DAC_DORx レジスタに転送されます。三角波カウンタは、MAMPx[3:0] ビットによって定義された最大振幅以上になるまでインクリメントされます。設定された振幅に達すると、カウンタは 0 にデクリメントされ、再びインクリメントが開始されます。

WAVEx[1:0] ビットをリセットすることによって、三角波生成をリセットできます。

図 269. DAC 三角波生成



図 270. 三角波生成による DAC 変換 (SW トリガ有効)



注： 三角波生成のためには、DAC_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

DAC を有効にする前に、MAMPx[3:0] ビットを設定する必要があります。そうしないと、これらのビットは変更できません。

28.4.12 DACチャネルモード

各 DACチャネルは、通常モードまたはサンプルおよびホールドモードのいずれかに設定できます。高駆動能力を得るために、出力バッファを有効にすることができます。出力バッファを有効にする前に、電圧オフセットを校正する必要があります。この校正は出荷時に実施され（リセット後にロードされ）、アプリケーション動作中にソフトウェアで調整できます。

通常モード

通常モードでは、バッファ状態の変更と DACx_OUTy ピン相互接続の変更による 4 つの組み合わせがあります。

出力バッファを有効にするには、DAC_MCR レジスタの MODEx[2:0] ビットを次のようにセットする必要があります。

- 000 : DAC を外部ピンに接続

出力バッファを無効にするには、DAC_MCR レジスタの MODEx[2:0] ビットを次のようにセットする必要があります。

- 010 : DAC を外部ピンに接続

サンプルおよびホールドモード

サンプルおよびホールドモードでは、DACコアがトリガ変換でデータを変換してから、コンデンサで変換された電圧を保持します。変換しない場合、DAC コアおよびバッファはサンプル間で完全にオフになり、DAC 出力はトライステートになるため、全体の消費電力を低減します。新しい変換を行う前ごとに、安定時間（その値はバッファ状態に応じて変わります）が必要になります。

このモードでは、DACコアならびにすべての対応するロジックとレジスタは、dac_hclk クロックに加えて LSI/LSE ロースピードクロック（dac_hold_ck）によっても駆動されますので、STOP モードなどの超低消費電力モードで DACチャネルを使用できます。

LSI/LSE ロースピードクロック（dac_hold_ck）は、サンプルおよびホールドモードが有効なときに停止しないでください。

サンプル/ホールドモードの動作は、3 つのフェーズに分けられます。

1. サンプルフェーズ : サンプル/ホールド要素が対象の電圧までチャージされます。チャージ時間はコンデンサ値（ユーザによって選択された内部または外部）によって異なります。サンプリング時間は、DAC_SHSRx レジスタの TSAMPLEx[9:0] ビットで設定されます。TSAMPLEx[9:0] ビットの書込み中、両方のクロックドメイン（AHB および低速クロック）を同期するために DAC_SR レジスタの BWSTx ビットが 1 にセットされ、DACチャネル動作中にソフトウェアによってサンプルフェーズの値を変更できるようになります。
2. ホールドフェーズ : DAC 出力チャネルはトライステートになり、DAC コアおよびバッファはオフになって消費電流を低減します。ホールド時間は、DAC_SHHR レジスタの THOLDx[9:0] ビットで設定されます。
3. リフレッシュフェーズ : リフレッシュ時間は、DAC_SHRR レジスタの TREFRESHx[7:0] ビットで設定されます。

上記 3 つのフェーズのタイミングは、LSI/LSE クロック周期の単位で行われます。たとえば、350 μ s のサンプル時間、2 ms のホールド時間、100 μ s のリフレッシュ時間を設定するために、LSI/LSE ~ 32 kHz を選択するとします。

12 サイクルがサンプルフェーズで必要になります。TSAMPLEx[9:0] = 11、

62 サイクルがホールドフェーズで必要になります。THOLDx[9:0] = 62、

そして、4 サイクルがリフレッシュ時間で必要になります。TREFRESHx[7:0] = 4。

この例では、消費電力が通常モードに対してほぼ 15 分の 1 に低減されます。

適切なサンプルおよびリフレッシュタイミングを計算する公式は、以下の表に記載しており、ホールド時間はリーク電流によって異なります。

表 269. サンプルおよびリフレッシュタイミング

バッファ状態	$t_{\text{SAMP}}^{(1)(2)}$	$t_{\text{REFRESH}}^{(2)(3)}$
イネーブル	$7 \mu\text{s} + (10 \cdot R_{\text{BON}} \cdot C_{\text{SH}})$	$7 \mu\text{s} + (R_{\text{BON}} \cdot C_{\text{SH}}) \cdot \ln(2 \cdot N_{\text{LSB}})$
無効化	$3 \mu\text{s} + (10 \cdot R_{\text{BOFF}} \cdot C_{\text{SH}})$	$3 \mu\text{s} + (R_{\text{BOFF}} \cdot C_{\text{SH}}) \cdot \ln(2 \cdot N_{\text{LSB}})$

- 上記の公式で、 $\frac{1}{2}$ LSBエラー精度での目標のコード値までの整定には、12 ビットの分解能の場合、10 定数時間が必要です。8 ビット分解能の場合、整定時間は7 定数時間です。
- C_{SH} はサンプルおよびホールドモードでのコンデンサ容量です。
- ホールドフェーズ中に許容される電圧低下「 V_d 」は、コンデンサが出力リーク電流で放電した後の LSB の数で表されます。 $\frac{1}{2}$ LSB エラー精度で目標のコード値まで整定し直すには、DAC の $\ln(2 \cdot N_{\text{LSB}})$ 定数時間が必要です。

出力バッファがオンの場合のサンプルおよびリフレッシュ時間の計算例

以下に使用される値は、例示のみを目的としています。製品データについては、製品データシートを参照してください。

$$C_{\text{SH}} = 100 \text{ nF}$$

$$V_{\text{REF+}} = 3.0 \text{ V}$$

サンプリングフェーズ：

$$t_{\text{SAMP}} = 7 \mu\text{s} + (10 \cdot 2000 \cdot 100 \cdot 10^{-9}) = 2.007 \text{ ms}$$

(ここで、 $R_{\text{BON}} = 2 \text{ k}\Omega$)

リフレッシュフェーズ：

$$t_{\text{REFRESH}} = 7 \mu\text{s} + (2000 \cdot 100 \cdot 10^{-9}) \cdot \ln(2 \cdot 10) = 606.1 \mu\text{s}$$

(ここで、 $N_{\text{LSB}} = 10$ (ホールドフェーズ中は 10 LSB 低下))

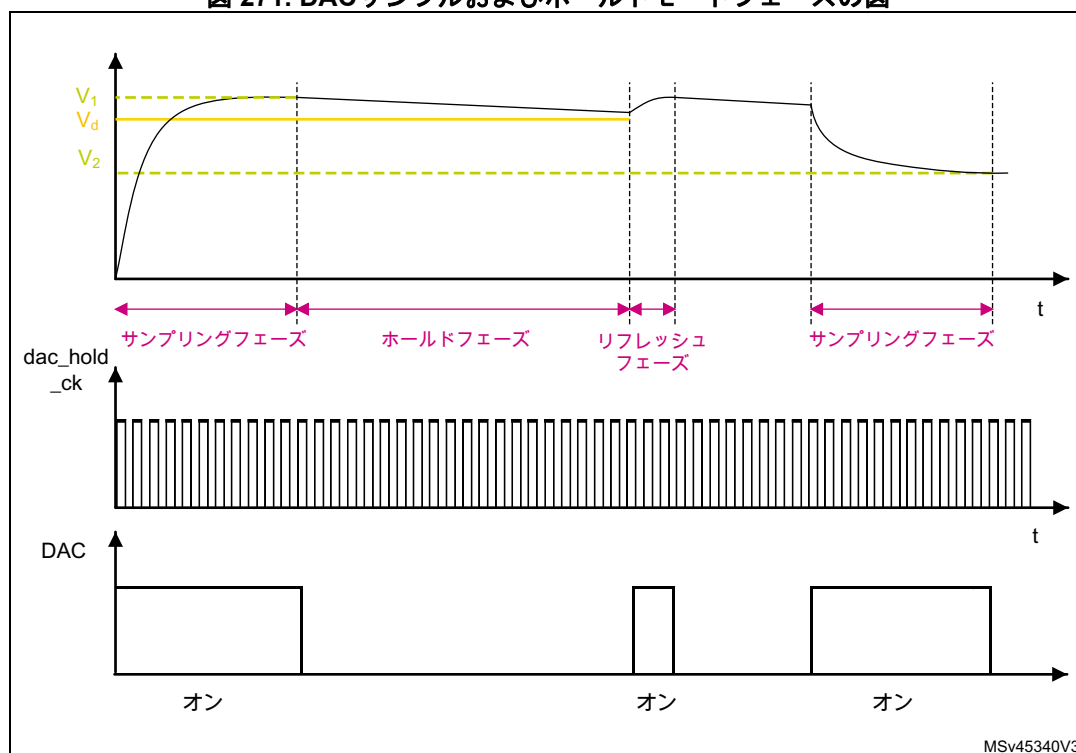
ホールドフェーズ：

$$D_v = i_{\text{leak}} \cdot t_{\text{hold}} / C_{\text{SH}} = 0.0073 \text{ V (3 V で 12 ビットの 10 LSB)}$$

$$i_{\text{leak}} = 150 \text{ nA (すべての温度範囲の IO リークで最悪のケース)}$$

$$t_{\text{hold}} = 0.0073 \cdot 100 \cdot 10^{-9} / (150 \cdot 10^{-9}) = 4.867 \text{ ms}$$

図 271. DACサンプルおよびホールドモードフェーズの図



通常モードと同様に、サンプルおよびホールドモードには個別の設定があります。

出力バッファを有効にするには、DAC_MCR レジスタの MODEx[2:0] ビットを次のようにセットする必要があります。

- 100 : DAC を外部ピンに接続

出力バッファを無効にするには、DAC_MCR レジスタの MODEx[2:0] ビットを次のようにセットする必要があります。

- 110 : DAC を外部ピンに接続

MODEx[2:0] ビットが 111 と等しい場合、内部コンデンサ C_{Lint} は、DACコアの電圧出力を保持し、それによって内部ペリフェラルを駆動します。

すべてのサンプルおよびホールドフェーズは割込み可能であり、DAC_DHRx のあらゆる変更は新しいサンプルフェーズを即座にトリガします。

表 270. チャネル出力モードの概要

MODEx[2:0]			モード	バッファ	出力接続
0	0	0	通常モード	有効	外部ピンに接続
0	1	0		無効	外部ピンに接続
1	0	0	サンプルおよびホールドモード	有効	外部ピンに接続
1	1	0		無効	外部ピンと内部ペリフェラルに接続 (コンパレータなど)

28.4.13 DAC チャネルバッファの較正

N ビットのデジタルアナログコンバータ (DAC) の転送関数は次のとおりです。

$$V_{out} = ((D/2^N) \times G \times V_{ref}) + V_{OS}$$

ここで、VOUT はアナログ出力、D はデジタル入力、G はゲイン、Vref は公称フルスケール電圧、vos はオフセット電圧です。理想的な dac チャネルでは、g = 1、Vos = 0 となります。

出力バッファの特性によって、オフセット電圧が部分的に異なり、アナログ出力で絶対的なオフセットエラーをもたらす場合があります。Vos を補正するには、トリミング技法による較正が必要です。

この較正は、DAC チャネル x がバッファ有効状態で動作している場合 (MODEx[2:0] = 0b000 または 0b001 または 0b100 または 0b101) のみ有効です。バッファがオフのときに他のモードで適用される場合は影響しません。較正中は次のようになります。

- バッファ出力がピンの内部／外部接続から切断され、トライステートモード (HiZ) になります。
- バッファがミドルコード値 0x800 を検出して、内部ブリッジによる VREF+/2 信号と比較するためにコンパレータとして機能し、比較結果 (CAL_FLAGx ビット) に応じて出力信号を 0 または 1 にトグルします。

次の 2 つの較正技法があります。

- 出荷時トリミング (デフォルト設定)
DAC バッファオフセットが出荷時にトリミングされます。DAC_CCR レジスタの OTRIMx[4:0] ビットのデフォルト値は出荷時トリミング値となり、DAC デジタルインタフェースがリセットされるとロードされます。
- ユーザトリミング
動作条件が出荷時トリミング条件と異なる場合、特に VDDA 電圧、温度、VREF+ の値が変化する場合に、ユーザトリミングはソフトウェアによって適用時にいつでも実行できます。

注： 出荷時トリミング条件の詳細については、データシートを参照してください。

また、VDD が削除される場合 (デバイスが STANDBY モードや VBAT モードに移行する場合など)、較正が必要になります。

ユーザトリミング較正を実行する手順は次のとおりです。

1. DAC チャネルがアクティブな場合、DAC_CR の ENx ビットに 0 を書き込み、チャネルを無効にします。
2. DAC_MCR レジスタに MODEx[2:0] = 0b000 または 0b001 または 0b100 または 0b101 を書き込んで、バッファが有効なモードを選択します。
3. DAC_CR レジスタの CENx ビットを 1 にセットして、DAC チャネル x の較正を開始します。
4. トリミングアルゴリズムを適用します。
 - a) OTRIMx[4:0] ビットにコード (0b000000 で開始する) を書き込みます。
 - b) tTRIM の遅延を待ちます。
 - c) DAC_SR の CAL_FLAGx ビットが 1 にセットされているか確認します。
 - d) CAL_FLAGx が 1 にセットされている場合、OTRIMx[4:0] トリミングコードが検索され、これを出力値を補正するデバイスの操作中に使用できます。それ以外は OTRIMx[4:0] をインクリメントし、(a) から (d) のサブステップを再び繰り返します。

ソフトウェアアルゴリズムによって、逐次近似や二分法を使用して、より早く OTRIMx[4:0] ビットの内容を計算してセットできます。

CAL_FLAGx ビットの転流／トグルは、オフセットが正しく補正されたことを示し、対応するトリミングコードを DAC_CCR レジスタの OTRIMx[4:0] ビットに保持しておく必要があります。

注： tTRIM 遅延が、OTRIMx[4:0] ビットへの書込みと DAC_SR レジスタの CAL_FLAGx ビットの読出しの間で守られなければ、正しい値を取得できません。このパラメータは、データシートの電気的特性のセクションで指定されています。

V_{DDA}、VREF+、および温度条件がデバイス動作中に変化せず、STANDBY および VBAT モードに頻繁に移行する場合、ソフトウェアは最初のユーザ較正時に見つかった OTRIMx[4:0] ビットを Flash またはバックアップレジスタに格納して、後でデバイスの電源が復帰したときに直接ロード／書き込むことができ、新しい較正にかかる時間を待たなくて済むようになります。

CENx ビットがセットされているとき、ENx ビットをセットすることはできません。

28.4.14 デュアル DAC チャンネル変換モード（デュアルチャンネルを使用できる場合）

同時に 2 つの DAC チャンネルを必要とするアプリケーションで、バスのバンド幅を有効に使用するために、DHR8RD、DHR12RD、および DHR12LD の 3 つのデュアルレジスタが搭載されています。両方の DAC チャンネルを同時に駆動するには、一意なレジスタアクセスが必要です。波形の生成の場合は、DHRxxxD レジスタへのアクセスは必要ありません。したがって、2 つの出力チャンネルは独立に、または同時に使用できます。

2 つの DAC チャンネルとこれらのデュアルレジスタを使用することで、15 の変換モードが使用可能です。すべての変換モードは、必要な場合には、別の DHRx レジスタを使っても利用できます。

すべてのモードについて、以下の節で説明します。

波形生成なしの独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに異なる値をセットすることによって、異なるトリガソースを設定します。
3. 目的の DHR レジスタ（DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD）にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、DHR1 レジスタが DAC_DOR1 に転送されます（3 dac_hclk クロックサイクル後）。

DAC チャンネル 2 トリガが発生すると、DHR2 レジスタが DAC_DOR2 に転送されます（3 dac_hclk クロックサイクル後）。

1 つの LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに異なる値をセットすることによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを 01 に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DHR レジスタ（DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD）にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます（3 dac_hclk クロックサイクル後）。その後、LFSR1 カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

異なる LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに異なる値をセットすることによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを 01 に設定し、MAMP1[3:0] ビットと MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、MAMP1[3:0] によって設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、MAMP2[3:0] によって設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

1 つの三角波生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに異なる値をセットすることによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネルの WAVEx[1:0] ビットを 1x に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル 1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル 2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

異なる三角波生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットに異なる値をセットすることによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを 1x に、異なる最大振幅値を MAMP1[3:0] と MAMP2[3:0] ビットに設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、MAMP1[3:0] によって設定された三角波振幅を持つ DAC チャンネル 1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、MAMP2[3:0] によって設定された三角波振幅を持つ DAC チャンネル 2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

同時ソフトウェア開始

DAC をこの変換モードに設定するには、次の手順が必要です。

- デュアル DAC チャンネルデータを目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にロードします。

この設定では、1 dac_hclk クロックサイクル後に、DHR1 および DHR2 レジスタが DAC_DOR1 と DAC_DOR2 にそれぞれ転送されます。

波形生成なしの同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1 および TSEL2 ビットフィールドに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- デュアル DAC チャンネルデータを目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にロードします。

トリガが発生すると、DHR1 および DHR2 レジスタが DAC_DOR1 と DAC_DOR2 にそれぞれ転送されます (3 dac_hclk クロックサイクル後)。

1 つの LFSR 生成による同時トリガ

- DAC をこの変換モードに設定するには、次の手順が必要です。
- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1 および TSEL2 ビットフィールドに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを 01 に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
- デュアル DAC チャンネルデータを目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) にロードします。

トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR1 カウンタが更新されます。同時に、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

異なる LFSR 生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1 および TSEL2 ビットフィールドに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネルの WAVEx[1:0] ビットを 01 に設定し、MAMP1[3:0] と MAMP2[3:0] ビットで異なる LFSR マスク値をセットします。

4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、MAMP1[3:0] で設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

同時に、MAMP2[3:0] で設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

1 つの三角波生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネルの WAVEx[1:0] ビットを 1x に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

同時に、同じ三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

異なる三角波生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1 および TSEL2 ビットフィールドに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを 1x に、異なる最大振幅値を MAMP1[3:0] と MAMP2[3:0] ビットに設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、MAMP1[3:0] によって設定された三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 AHB クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

同時に、MAMP2[3:0] で設定された三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 dac_hclk クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

28.5 低消費電力モードでの DAC

表 271. 低消費電力モードが DAC に与える影響

モード	説明
SLEEP	影響はありません。DAC は DMA と共に使用されます。
STOP ⁽¹⁾	DACはスタティック値でアクティブなままになります。サンプルおよびホールドモードは LSE/LSI クロックを使って選択できます。
STANDBY	DAC ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. DAC ペリフェラルでサポートされる STOP モードについては、[セクション 28.3 : DAC の実装](#) を参照してください。

28.6 DAC 割込み

表 272. DAC 割込み

項目 (割込みの略称)	割込み イベント	イベント フラグ	有効制御 ビット	割込みの クリア方法	SLEEP モードの終了	STOP モードの終了	STANDBY モードの終了
DAC	DMA アンダー ラン	DMAUDRX	DMAUDRIEX	DMAUDRx = 1を 書き込む	可	不可	不可

28.7 DAC レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1](#)を参照してください。

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。

28.7.1 DAC 制御レジスタ (DAC_CR)

アドレスオフセット：0x00

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2 [3:0]				WAVE2 [1:0]		TSEL2[3]	TSEL2[2]	TSEL2[1]	TSEL2[0]	TEN2	EN2
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CEN1	DMAUDRIE1	DMAEN1	MAMP1 [3:0]				WAVE1 [1:0]		TSEL1[3]	TSEL1[2]	TSEL1[1]	TSEL1[0]	TEN1	EN1
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **CEN2** : DAC チャンネル 2 較正イネーブル

このビットは、DAC チャンネル 2 の較正を有効化/無効化するためにソフトウェアによってセット/クリアされます。DAC_CR で EN2 ビットが 0 にセットされている場合のみ書き込み可能で（DAC チャンネルが無効化された場合のみ較正モードに移行／終了できます）、それ以外の書き込み操作は無視されます。

0 : DAC チャンネル 2 は通常動作モードです。

1 : DAC チャンネル 2 は較正モードです。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 29 **DMAUDRIE2** : DAC チャンネル 2 DMA アンダーラン割込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 2 DMA アンダーラン割込みは無効です。

1 : DAC チャンネル 2 DMA アンダーラン割込みは有効です。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 28 **DMAEN2** : DAC チャンネル 2 DMA イネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 2 DMA モードは無効です。

1 : DAC チャンネル 2 DMA モードは有効です。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 27:24 **MAMP2[3:0]** : DAC チャンネル2 マスク／振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

0000 : LFSR／三角波振幅の非マスクビット 0 は 1 に等しい。
 0001 : LFSR／三角波振幅の非マスクビット [1:0] は 3 に等しい。
 0010 : LFSR／三角波振幅の非マスクビット [2:0] は 7 に等しい。
 0011 : LFSR／三角波振幅の非マスクビット [3:0] は 15 に等しい。
 0100 : LFSR／三角波振幅の非マスクビット [4:0] は 31 に等しい。
 0101 : LFSR／三角波振幅の非マスクビット [5:0] は 63 に等しい。
 0110 : LFSR／三角波振幅の非マスクビット [6:0] は 127 に等しい。
 0111 : LFSR／三角波振幅の非マスクビット [7:0] は 255 に等しい。
 1000 : LFSR／三角波振幅の非マスクビット [8:0] は 511 に等しい。
 1001 : LFSR／三角波振幅の非マスクビット [9:0] は 1023 に等しい。
 1010 : LFSR／三角波振幅の非マスクビット [10:0] は 2047 に等しい。
 ≥ 1011 : LFSR／三角波振幅の非マスクビット [11:0] は 4095 に等しい。

注： これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 23:22 **WAVE2[1:0]** : DAC チャンネル2 ノイズ／三角波生成イネーブル

これらのビットは、ソフトウェアによってセット／リセットされます。

00 : 波形生成は無効です。
 01 : ノイズ波生成は有効です。
 1x : 三角波生成は有効です。

注： ビット **TEN2 = 1** (DAC チャンネル 2 トリガ有効) の場合のみ使用されます。
 これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 21:18 **TSEL2[3:0]** : DAC チャンネル2 トリガ選択

これらのビットは、DAC チャンネル2 をトリガするために使用される外部イベントを選択します。

0000 : SWTRIG2
 0001 : dac_ch2_trg1
 0010 : dac_ch2_trg2

 1111 : dac_ch2_trg15

トリガの設定とマッピングの詳細については、[セクション 28.4.2 : DAC ピンおよび内部信号のトリガ選択表](#)を参照してください。

注： ビット **TEN2 = 1** (DAC チャンネル 2 トリガ有効) の場合のみ使用されます。
 これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 17 **TEN2** : DAC チャンネル2 トリガイネーブル

このビットは、DAC チャンネル2 トリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 2 トリガは無効であり、DAC_DHR2 レジスタに書き込まれたデータは、1 dac_hclk クロックサイクル後に DAC_DOR2 レジスタに転送されます。
 1 : DAC チャンネル 2 トリガは有効であり、DAC_DHR2 レジスタからのデータは、3 dac_hclk クロックサイクル後に DAC_DOR2 レジスタに転送されます。

注： ソフトウェアトリガが選択されているときには、DAC_DHR2 から DAC_DOR2 レジスタへの転送は、わずか 1 dac_hclk クロックサイクルで行われます。
 これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 16 **EN2** : DAC チャンネル2 イネーブル

このビットは、DAC チャンネル2 を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル2 は無効です。

1 : DAC チャンネル2 は有効です。

注： これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **CEN1** : DAC チャンネル 1 較正イネーブル

このビットは、DAC チャンネル 1 較正を有効化／無効化するためにソフトウェアによってセット／クリアされます。これは、ビット EN1 = 0 の場合のみ DAC_CR に書き込み可能で (DAC チャンネルが無効化されている場合のみ較正モードに移行／終了できます)、それ以外の場合、書き込み操作は無視されます。

0 : DAC チャンネル 1 は通常動作モードです。

1 : DAC チャンネル 1 は較正モードです。

ビット 13 **DMAUDRIE1** : DAC チャンネル 1 DMA アンダーラン割込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 DMA アンダーラン割込みは無効です。

1 : DAC チャンネル 1 DMA アンダーラン割込みは有効です。

ビット 12 **DMAEN1** : DAC チャンネル 1 DMA イネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 DMA モードは無効です。

1 : DAC チャンネル 1 DMA モードは有効です。

ビット 11:8 **MAMP1[3:0]** : DAC チャンネル 1 マスク／振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

0000 : LFSR／三角波振幅の非マスクビット 0 は 1 に等しい。

0001 : LFSR／三角波振幅の非マスクビット [1:0] は 3 に等しい。

0010 : LFSR／三角波振幅の非マスクビット [2:0] は 7 に等しい。

0011 : LFSR／三角波振幅の非マスクビット [3:0] は 15 に等しい。

0100 : LFSR／三角波振幅の非マスクビット [4:0] は 31 に等しい。

0101 : LFSR／三角波振幅の非マスクビット [5:0] は 63 に等しい。

0110 : LFSR／三角波振幅の非マスクビット [6:0] は 127 に等しい。

0111 : LFSR／三角波振幅の非マスクビット [7:0] は 255 に等しい。

1000 : LFSR／三角波振幅の非マスクビット [8:0] は 511 に等しい。

1001 : LFSR／三角波振幅の非マスクビット [9:0] は 1023 に等しい。

1010 : LFSR／三角波振幅の非マスクビット [10:0] は 2047 に等しい。

≥ 1011: LFSR／三角波振幅の非マスクビット [11:0] は 4095 に等しい。

ビット 7:6 **WAVE1[1:0]** : DAC チャンネル 1 ノイズ／三角波生成イネーブル

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : 波形生成は無効です。

01 : ノイズ波生成は有効です。

1x : 三角波生成は有効です。

ビット TEN1 = 1 (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

ビット 5:2 **TSEL1[3:0]** : DAC チャンネル 1 トリガ選択

これらのビットは、DAC チャンネル 1 をトリガするために使用される外部イベントを選択します。

0000 : SWTRIG1

0001 : dac_ch1_trg1

0010 : dac_ch1_trg2

.....

1111 : dac_ch1_trg15

トリガの設定とマッピングの詳細については、[セクション 28.4.2 : DAC ピンおよび内部信号のトリガ選択表](#)を参照してください。

注： ビット **TEN1 = 1** (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

ビット 1 **TEN1** : DAC チャンネル 1 トリガイネーブル

このビットは、DAC チャンネル 1 トリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 トリガは無効であり、DAC_DHR1 レジスタに書き込まれたデータは、1 dac_hclk クロックサイクル後に DAC_DOR1 レジスタに転送されます。

1 : DAC チャンネル 1 トリガは有効であり、DAC_DHR1 レジスタからのデータは、3 dac_hclk クロックサイクル後に DAC_DOR1 レジスタに転送されます。

注： ソフトウェアトリガが選択されているときには、DAC_DHR1 レジスタから DAC_DOR1 レジスタへの転送は、わずか 1 dac_hclk クロックサイクルで行われます。

ビット 0 **EN1** : DAC チャンネル 1 イネーブル

このビットは、DAC チャンネル 1 を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 は無効です。

1 : DAC チャンネル 1 は有効です。

28.7.2 DAC ソフトウェアトリガレジスタ (DAC_SWTRGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SWTRIG2** : DAC チャンネル2 ソフトウェアトリガ

このビットは、ソフトウェアトリガモードで DAC をトリガするために、ソフトウェアによってセットされます。

0 : トリガなし

1 : トリガ

注： このビットは、DAC_DHR2 レジスタの値が DAC_DOR2 レジスタにロードされると、ハードウェアによってクリアされます (1 dac_hclk クロックサイクル後)。

このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 0 **SWTRIG1** : DAC チャンネル 1 ソフトウェアトリガ

このビットは、ソフトウェアトリガモードで DAC をトリガするために、ソフトウェアによってセットされます。

0 : トリガなし

1 : トリガ

注： このビットは、DAC_DHR1 レジスタの値が DAC_DOR1 レジスタにロードされると、ハードウェアによってクリアされます (1 dac_hclk クロックサイクル後)。

28.7.3 DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC1DHRB [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC1DHRB[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ B

これらのビットは、ソフトウェアによって書き込まれます。DAC がダブルデータモードで動作しているとき、これらには DAC チャンネル 1 の 12 ビットのデータを指定します。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、ソフトウェアによって書き込まれます。これらには、DAC チャンネル 1 の 12 ビットデータを指定します。

28.7.4 DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC1DHRB [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:20 **DACC1DHRB[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ B

これらのビットは、ソフトウェアによって書き込まれます。DAC がダブルデータモードで動作しているとき、これらには DAC チャンネル 1 の 12 ビットのデータを指定します。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ
 これらのビットは、ソフトウェアによって書き込まれます。
 これらには、DAC チャンネル 1 の 12 ビットデータを指定します。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

28.7.5 DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHRB [7:0]								DACC1DHR [7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC1DHRB[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ
 これらのビットは、ソフトウェアによって書き込まれます。DAC がダブルデータモードで動作しているとき、これらには DAC チャンネル 1 の 8 ビットのデータを指定します。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ
 これらのビットは、ソフトウェアによって書き込まれます。これらには、DAC チャンネル 1 の 8 ビットデータを指定します。

28.7.6 DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)

このレジスタは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3: DAC の実装](#)を参照してください。

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	DACC2DHRB [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	DACC2DHR [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DHRB[11:0]** : DAC チャンネル 2 の 12 ビット右詰めデータ
 これらのビットは、ソフトウェアによって書き込まれます。DAC が DMA ダブルデータモードで動作しているとき、これらには DAC チャンネル 2 の 12 ビットのデータを指定します。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DHR[11:0]** : DAC チャンネル2 の12ビット右詰めデータ

これらのビットは、ソフトウェアによって書き込まれます。これらには、DAC チャンネル 2 の 12 ビットデータを指定します。

28.7.7 DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2)

このレジスタは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHRB [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:20 **DACC2DHRB[11:0]** : DAC チャンネル 2 の 12 ビット左詰めデータ B

これらのビットは、ソフトウェアによって書き込まれます。DAC がダブルデータモードで動作しているとき、これらには DAC チャンネル 2 の 12 ビットのデータを指定します。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC2DHR[11:0]** : DAC チャンネル2 の12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

28.7.8 DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2)

このレジスタは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHRB [7:0]								DACC2DHR [7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC2DHRB[7:0]** : DAC チャンネル2 の8ビット右詰めデータ

これらのビットは、ソフトウェアによって書き込まれます。DAC がダブルデータモードで動作しているとき、これらには DAC チャンネル 2 の 8 ビットのデータを指定します。

ビット 7:0 **DACC2DHR[7:0]** : DAC チャンネル2 の8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

28.7.9 デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR [11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DHR[11:0]** : DAC チャンネル2 の12ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

28.7.10 デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR [11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:20 **DACC2DHR[11:0]** : DAC チャンネル2 の12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

28.7.11 デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR [7:0]								DACC1DHR [7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC2DHR[7:0]** : DAC チャンネル2 の8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

28.7.12 DAC チャンネル 1 データ出力レジスタ (DAC_DOR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC1DORB [11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DOR [11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC1DORB[11:0]** : DAC チャンネル 1 データ出力

これらのビットは読み出し専用です。これらには、DAC チャンネル 1 B のデータ出力が含まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DOR[11:0]** : DAC チャンネル 1 データ出力

これらのビットは読み出し専用であり、DAC チャンネル 1 のデータ出力を含みます。

28.7.13 DAC チャンネル 2 データ出力レジスタ (DAC_DOR2)

このレジスタは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DORB [11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DOR [11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DORB[11:0]** : DAC チャンネル2 データ出力

これらのビットは読み出し専用です。これらには、DAC チャンネル 2 B のデータ出力が含まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DOR[11:0]** : DAC チャンネル2 データ出力

これらのビットは読み出し専用であり、DAC チャンネル 2 のデータ出力を含みます。

28.7.14 DAC ステータスレジスタ (DAC_SR)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWST2	CAL_FLAG2	DMAUD_R2	DORSTA_T2	DAC2RD_Y	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1	r	r											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BWST1	CAL_FLAG1	DMAUD_R1	DORSTA_T1	DAC1RD_Y	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1	r	r											

ビット 31 **BWST2** : DAC チャンネル 2 ビジー書き込みサンプル時間フラグ

このビットは、サンプルおよびホールドモードが有効になった直後にシステムによってセットされます。このビットは、ソフトウェアが DAC_SHSR2 レジスタに書き込むたびにセットされ、DAC_SHSR2 の書き込み操作が完了するとハードウェアによってクリアされます（同期に約 3 LSI/LSE 周期かかります）。

0 : 進行中の DAC_SHSR2 の書き込み操作はありません。DAC_SHSR2 は書き込み可能です。

1 : 進行中の DAC_SHSR2 の書き込み操作があります。DAC_SHSR2 は書き込みできません。

注 : このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 30 **CAL_FLAG2** : DAC チャンネル 2 較正オフセットステータス

このビットは、ハードウェアによってセット／クリアされます。

0 : 較正トリミング値はオフセット較正值未満です。

1 : 較正トリミング値がオフセット較正值以上です。

注 : このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 29 DMAUDR2 : DAC チャンネル2 DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル2 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル 2 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル 2 変換を駆動しています)。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 28 DORSTAT2 : DAC チャンネル 2 出力レジスタステータスビット

このビットは、ハードウェアによってセット/クリアされます。これは、DAC がダブルデータモードで動作している場合のみ適用されます。

0 : DOR[11:0] が実際の DAC 出力に使用されます。

1 : DORB[11:0] が実際の DAC 出力に使用されます。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 27 DAC2RDY : DAC チャンネル 2 レディステータスビット

このビットは、ハードウェアによってセット/クリアされます。

0 : DAC チャンネル 2 は、まだトリガも出力データも受け入れる準備ができていません。

1 : DAC チャンネル 2 は、トリガまたは出力データを受け入れる準備ができています。

注： このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 26:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 BWST1 : DAC チャンネル 1 ビジー書き込みサンプル時間フラグ

このビットは、サンプルおよびホールドモードが有効になった直後にシステムによって、およびソフトウェアが DAC_SHSR1 レジスタに書き込むたびにセットされます。DAC_SHSR1 の書き込み操作が完了するとハードウェアによってクリアされます (同期に約 3 LSI/LSE 周期かかります)。

0 : 進行中の DAC_SHSR1 の書き込み操作はありません。DAC_SHSR1 は書き込み可能です。

1 : 進行中の DAC_SHSR1 の書き込み操作があります。DAC_SHSR1 は書き込みできません。

ビット 14 CAL_FLAG1 : DAC チャンネル 1 較正オフセットステータス

このビットは、ハードウェアによってセット/クリアされます。

0 : 較正トリミング値はオフセット較正值未満です。

1 : 較正トリミング値がオフセット較正值以上です。

ビット 13 DMAUDR1 : DAC チャンネル 1 の DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル 1 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル 1 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル 1 変換を駆動しています)。

ビット 12 DORSTAT1 : DAC チャンネル 1 出力レジスタステータスビット

このビットは、ハードウェアによってセット/クリアされます。これは、DAC がダブルデータモードで動作している場合のみ適用されます。

0 : DOR[11:0] が実際の DAC 出力に使用されます。

1 : DORB[11:0] が実際の DAC 出力に使用されます。

ビット 11 DAC1RDY : DAC チャンネル 1 レディステータスビット

このビットは、ハードウェアによってセット/クリアされます。

0 : DAC チャンネル 1 は、まだトリガも出力データも受け入れる準備ができていません。

1 : DAC チャンネル 1 は、トリガまたは出力データを受け入れる準備ができています。

ビット 10:0 予約済みであり、リセット値に保持する必要があります。

28.7.15 DAC 較正制御レジスタ (DAC_CCR)

アドレスオフセット : 0x38

リセット値 : 0x00XX 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM2 [4:0]				
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM1 [4:0]				
											rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **OTRIM2[4:0]** : DAC チャンネル 2 オフセットトリミング値

これらのビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 15:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **OTRIM1[4:0]** : DAC チャンネル 1 オフセットトリミング値

28.7.16 DAC モード制御レジスタ (DAC_MCR)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SINFOR MAT2	DMA DOUBLE2	Res.	Res.	Res.	Res.	Res.	MODE2 [2:0]		
						rw	rw						rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HFSEL[1]	HFSEL[0]	Res.	Res.	Res.	Res.	SINFOR MAT1	DMA DOUBLE1	Res.	Res.	Res.	Res.	Res.	MODE1 [2:0]		
rw	rw					rw	rw						rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **SINFORMAT2** : DAC チャンネル 2 の符号付きフォーマットの有効化

このビットは、ソフトウェアによってセット／クリアされます。

0 : 入力データは符号なしフォーマットです。

1 : 入力データは符号付きフォーマット (2 の補数) です。MSB ビットは、符号を表します。

注 : このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 24 **DMADouble2** : DAC チャンネル 2 DMA ダブルデータモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : DMA 通常モードが選択されています。

1 : DMA ダブルデータモードが選択されています。

注 : このビットは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 23:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **MODE2[2:0]** : DAC チャンネル 2 モード

これらのビットは、DAC が無効になっており、較正モードでない場合 (DAC_CR レジスタでビット EN2 = 0 およびビット CEN2 = 0 の場合) のみ書込み可能です。EN2 = 1 または CEN2 = 1 の場合、書込み操作は無視されます。

これらは、DAC チャンネル 2 モードを選択するために、ソフトウェアによってセット/クリアされます。

- DAC チャンネル 2 は通常動作モードです。
 - 000 : DAC チャンネル 2 はバッファ有効で外部ピンに接続されています。
 - 001 : 予約済み
 - 010 : DAC チャンネル 2 はバッファ無効で外部ピンに接続されています。
 - 011 : 予約済み
- DAC チャンネル 2 はサンプルおよびホールドモードです。
 - 100 : DAC チャンネル 2 はバッファ有効で外部ピンに接続されています。
 - 101 : 予約済み
 - 110 : DAC チャンネル 2 はバッファ無効で外部ピンに接続されています。
 - 111 : 予約済み

注： このレジスタは、EN2 = 0 のときにのみ修正できます。

DAC チャンネル 2 の有無については、[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 15:14 **HFSEL[1:0]** : 高周波数インタフェースモードの選択

- 00 : 高周波数インタフェースモードは無効です。
- 01 : AHB クロック周波数 > 80 MHz に対して高周波数インタフェースモードは有効です。
- 10 : AHB クロック周波数 > 160 MHz に対して高周波数インタフェースモードは有効です。
- 11 : 予約済み

ビット 13:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **SINFORMAT1** : DAC チャンネル 1 の符号付きフォーマットの有効化

- このビットは、ソフトウェアによってセット/クリアされます。
- 0 : 入力データは符号なしフォーマットです。
 - 1 : 入力データは符号付きフォーマット (2 の補数) です。MSB ビットは、符号を表します。

ビット 8 **DMADouble1** : DAC チャンネル 1 DMA ダブルデータモード

- このビットは、ソフトウェアによってセット/クリアされます。
- 0 : DMA 通常モードが選択されています。
 - 1 : DMA ダブルデータモードが選択されています。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **MODE1[2:0]** : DAC チャンネル 1 モード

これらのビットは、DAC が無効になっており、較正モードでない場合 (DAC_CR レジスタでビット EN1 = 0 およびビット CEN1 = 0 の場合) のみ書込み可能です。EN1 = 1 または CEN1 = 1 の場合、書込み操作は無視されます。

これらは、DAC チャンネル 1 モードを選択するために、ソフトウェアによってセット/クリアされます。

- DAC チャンネル 1 は通常動作モードです。
 - 000 : DAC チャンネル 1 はバッファ有効で外部ピンに接続されています。
 - 001 : 予約済み
 - 010 : DAC チャンネル 1 はバッファ無効で外部ピンに接続されています。
 - 011 : 予約済み
- DAC チャンネル 1 はサンプルおよびホールドモードです。
 - 100 : DAC チャンネル 1 はバッファ有効で外部ピンに接続されています。
 - 101 : 予約済み
 - 110 : DAC チャンネル 1 はバッファ無効で外部ピンに接続されています。
 - 111 : 予約済み

注： このレジスタは、EN1 = 0 のときにのみ修正できます。

28.7.17 DAC チャンネル 1 サンプルおよびホールドのサンプル時間レジスタ (DAC_SHSR1)

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE1 [9:0]									
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **TSAMPLE1[9:0]** : DACチャンネル 1 サンプル時間 (サンプルおよびホールドモードでのみ有効)

これらのビットは DAC チャンネル 1 が無効である場合や通常動作中にも書込み可能です。後者の場合、DAC_SR レジスタの BWST1 がローのときのみ書込み可能であり、BWST1 = 1 の場合、書込み操作は無視されます。

注 : これは、サンプルフェーズを実行する LSI/LSE クロックの数を表します。
 サンプリング時間 = (TSAMPLE1[9:0] + 1) x LSI/LSE クロック周期です。

28.7.18 DAC チャンネル 2 サンプルおよびホールドのサンプル時間レジスタ (DAC_SHSR2)

このレジスタは、デュアルチャンネル DAC でのみ使用できます。[セクション 28.3: DAC の実装](#)を参照してください。

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE2 [9:0]									
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **TSAMPLE2[9:0]** : DACチャンネル 2 サンプル時間 (サンプルおよびホールドモードでのみ有効)

これらのビットは DAC チャンネル 2 が無効である場合や通常動作中にも書込み可能です。後者の場合、DAC_SR レジスタの BWST2 がローのときのみ書込み可能であり、BWST2 = 1 の場合、書込み操作は無視されます。

注 : これは、サンプルフェーズを実行する LSI/LSE クロックの数を表します。
 サンプリング時間 = (TSAMPLE2[9:0] + 1) x LSI/LSE クロック周期です。

28.7.19 DAC サンプルおよびホールド時間レジスタ (DAC_SHHR)

アドレスオフセット : 0x48

リセット値 : 0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	THOLD2 [9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	THOLD1 [9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:16 **THOLD2[9:0]** : DACチャネル 2ホールド時間 (サンプルおよびホールドモードでのみ有効)

ホールド時間 = (THOLD[9:0]) × LSI/LSE クロック周期

注: このレジスタは、EN2 = 0 のときにのみ修正できます。

これらのビットは、デュアルチャネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **THOLD1[9:0]** : DACチャネル 1ホールド時間 (サンプルおよびホールドモードでのみ有効)

ホールド時間 = (THOLD[9:0]) × LSI/LSE クロック周期

注: このレジスタは、EN1 = 0 のときにのみ修正できます。

注: これらのビットは、DAC チャネルが無効になっていて、通常動作モードの場合 (DAC_CR レジスタのビット ENx = 0 およびビット CENx = 0 の場合) のみ書き込み可能です。ENx = 1 または CENx = 1 の場合、書き込み操作は無視されます。

28.7.20 DAC サンプルおよびホールドのリフレッシュ時間レジスタ (DAC_SHRR)

アドレスオフセット : 0x4C

リセット値 : 0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH2 [7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH1 [7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **TREFRESH2[7:0]** : DACチャネル 2リフレッシュ時間 (サンプルおよびホールドモードでのみ有効)

リフレッシュ時間 = (TREFRESH[7:0]) × LSI/LSE クロック周期

注: このレジスタは、EN2 = 0 のときにのみ修正できます。

これらのビットは、デュアルチャネル DAC でのみ使用できます。[セクション 28.3 : DAC の実装](#)を参照してください。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TREFRESH1[7:0]** : DACチャネル 1リフレッシュ時間（サンプルおよびホールドモードでのみ有効）

リフレッシュ時間 = (TREFRESH[7:0]) x LSI/LSE クロック周期

注： このレジスタは、EN1 = 0 のときにのみ修正できます。

注： これらのビットは、DAC チャネルが無効になっていて、通常動作モードの場合（DAC_CR レジスタのビット ENx = 0 およびビット CENx = 0 の場合）のみ書き込み可能です。ENx = 1 または CENx = 1 の場合、書き込み操作は無視されます。

28.7.21 DAC レジスタマップ

表 273 に DAC レジスタの要約を示します。

表 273. DAC レジスタのマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DAC_CR	Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2 [3:0]				WAVE2 [2:0]		TSEL2 [3:1]		TSEL2[0]	TEN2	EN2	Res.	CEN1	DMAUDRIE1	DMAEN1	MAMP1 [3:0]				WAVE1 [1:0]		TSEL1 [3:1]		TSEL1[0]	TEN1	EN1		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	DAC_SWTRGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
	リセット値																		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	
0x08	DAC_DHR12R1	Res.	Res.	Res.	Res.	DACC1DHRB [11:0]											Res.	Res.	Res.	Res.	DACC1DHR [11:0]												
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0
0x0C	DAC_DHR12L1	DACC1DHRB [11:0]											Res.	Res.	Res.	Res.	DACC1DHR [11:0]											Res.	Res.	Res.	Res.		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0				
0x10	DAC_DHR8R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHRB [7:0]							DACC1DHR [7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DAC_DHR12R2	Res.	Res.	Res.	Res.	DACC2DHRB [11:0]											Res.	Res.	Res.	Res.	DACC2DHR [11:0]												
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0
0x18	DAC_DHR12L2	DACC2DHRB [11:0]											Res.	Res.	Res.	Res.	DACC2DHR [11:0]											Res.	Res.	Res.	Res.		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0				
0x1C	DAC_DHR8R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHRB [7:0]							DACC2DHR [7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DAC_DHR12RD	Res.	Res.	Res.	Res.	DACC2DHR [11:0]											Res.	Res.	Res.	Res.	DACC1DHR [11:0]												
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0
0x24	DAC_DHR12LD	DACC2DHR [11:0]											Res.	Res.	Res.	Res.	DACC1DHR [11:0]											Res.	Res.	Res.	Res.		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0				

表 273. DAC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x28	DAC_ DHR8RD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DACC2DHR [7:0]				DACC1DHR [7:0]												
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	DAC_ DOR1	Res	Res	Res	Res	DACC1DORB [11:0]											Res	Res	Res	Res	DACC1DOR [11:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	DAC_ DOR2	Res	Res	Res	Res	DACC2DORB [11:0]											Res	Res	Res	Res	DACC2DOR [11:0]													
	リセット値					0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x34	DAC_SR	BWST2	CAL_FLAG2	DMAUDR2	DORSTAT2	DAC2RDY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BWST1	CAL_FLAG1	DMAUDR1	DORSTAT1	DAC1RDY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値	0	0	0	0	0												0	0	0	0	0												
0x38	DAC_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OTRIM[24]	OTRIM[23]	OTRIM[22]	OTRIM[21]	OTRIM[20]	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OTRIM[14]	OTRIM[13]	OTRIM[12]	OTRIM[11]	OTRIM[10]
	リセット値												X	X	X	X	X													X	X	X	X	X
0x3C	DAC_MCR	Res	Res	Res	Res	Res	Res	SINFORMAT2	DMADDOUBLE2	Res	Res	Res	Res	Res	MODE2 [2:0]		HFSEL[1]	HFSEL[0]	Res	Res	Res	Res	Res	SINFORMAT1	DMADDOUBLE1	Res	Res	Res	Res	Res	Res	MODE1 [2:0]		
	リセット値							0	0								0	0	0				0	0							0	0	0	
0x40	DAC_ SHSR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TSAMPLE1 [9:0]										
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	
0x44	DAC_ SHSR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TSAMPLE2 [9:0]										
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	
0x48	DAC_ SHHR	Res	Res	Res	Res	Res	THOLD2 [9:0]										Res	Res	Res	Res	Res	Res	THOLD1 [9:0]											
	リセット値						0	0	0	0	0	0	0	0	0	0	1						0	0	0	0	0	0	0	0	0	0	1	
0x4C	DAC_ SHRR	Res	Res	Res	Res	Res	Res	Res	TREFRESH2 [7:0]								Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TREFRESH1 [7:0]							
	リセット値								0	0	0	0	0	0	0	0	1									0	0	0	0	0	0	0	0	1
0x50 - 0x54	予約済み	Res.																																

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

29 基準電圧バッファ (VREFBUF)

29.1 概要

デバイスには基準電圧バッファが搭載されており、ADC の基準電圧や、VREF+ ピンを介した外部コンポーネントの基準電圧としても使用できます。

29.2 VREFBUF の実装

以下の表に VREFBUF 電圧の標準値を示します。

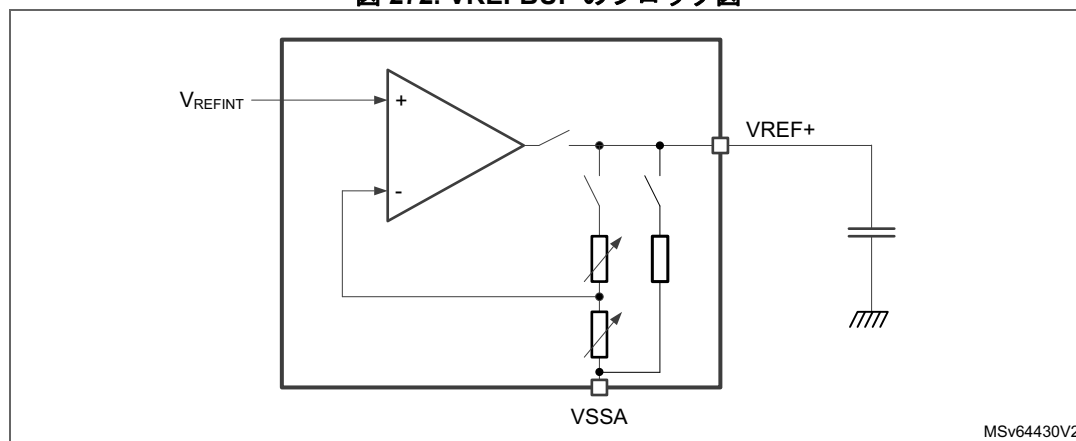
表 274. VREFBUF の標準値

記号	値
VREFBUF0	2.5 V
VREFBUF1	2.048 V
VREFBUF2	1.8 V

注： 詳細については、製品データシートを参照してください。

29.3 VREFBUF の機能説明

図 272. VREFBUF のブロック図



内部基準電圧バッファは、ゲインをプログラムできるオペアンプです。アンプの入力は内部基準電圧 V_{REFINT} に接続されています。VREFBUF は次の 4 つの電圧^(a)をサポートしており、VREFBUF_CSR レジスタの VRS ビットで設定します。

- VRS = 000 : VREFBUF0 電圧選択
- VRS = 001 : VREFBUF1 電圧選択
- VRS = 010 : VREFBUF2 電圧選択

a. V_{DDA} の最小電圧は VRS の設定に依存します。製品データシートを参照してください。

内部基準電圧は ENVR および HIZ ビットの設定に応じて 4 つの個別モードに設定できます。これらのモードを以下の表に示します。

表 275. VREF バッファモード

ENVR	HIZ	VREF バッファ設定
0	0	VREFBUF バッファオフモード : – V_{REF+} ピンが V_{SSA} にプルダウン
0	1	外部基準電圧モード (デフォルト値) : – VREFBUF バッファオフ – V_{REF+} ピンは入力モード
1	0	内部基準電圧モード – VREFBUF バッファオン – V_{REF+} ピンは VREFBUF バッファ出力に接続
1	1	ホールドモード : – VREF は出力バッファなしで有効、VREF+ ピン電圧は外部コンデンサで保持 – VRR の検出は無効で、VRR ビットは最後の状態を保持

VREFBUF_CSR レジスタの ENVR ビットをセットし、HIZ ビットをクリアすることで VREFBUF を有効化した後、ユーザは基準電圧出力が期待値に達したことを示す VRR ビットがセットされるまで待つ必要があります。

29.4 VREFBUF のトリミング

VREFBUF の出力電圧は ST によって工場で較正されています。リセット時、および VRS の設定を変更するたびに、較正データが自動的に TRIM レジスタにロードされます。

また、オプションで TRIM レジスタのビットを直接変更することによって、出力電圧を調整することができます。この場合、デバイスをリセットするまでは、VRS の設定が TRIM レジスタに影響することはありません。

29.5 VREFBUF レジスタ

29.5.1 VREFBUF 制御およびステータスレジスタ (VREFBUF_CSR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRS[2:0]			VRR	Res.	HIZ	ENVR
									rW	rW	rW	r		rW	rW

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **VRS[2:0]** : 基準電圧スケール

これらのビットは、基準電圧バッファによって生成される値を選択します。

VRS = 000 : VREFBUF0 電圧選択

VRS = 001 : VREFBUF1 電圧選択

VRS = 010 : VREFBUF2 電圧選択

その他 : 予約済み

注 : 各 VREFBUFx 電圧の設定値については、製品データシートを参照してください。
ソフトウェアは、VREFBUF が無効な (ENVR=0) のときだけこのビットフィールドをプログラムできます。

ビット 3 **VRR** : 基準電圧バッファ レディ

0 : 基準電圧バッファ出力がレディ状態ではありません。

1 : 基準電圧バッファ出力がリクエストされたレベルに達しました。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **HIZ** : ハイインピーダンスモード

このビットによって、VREF+ ピンを接続/切断するためのアナログスイッチを制御します。

0 : VREF+ ピンが 基準電圧バッファ出力に内部接続されます。

1 : VREF+ ピンをハイインピーダンスにします。

ENVR ビットの設定に応じたモード説明については、表 275 : VREF バッファモードを参照してください。

ビット 0 **ENVR** : 基準電圧バッファモード有効化

このビットは、基準電圧バッファモードの有効化に使用されます。

0 : 内部基準電圧モード無効化 (外部基準電圧モード)

1 : 内部基準電圧モード (基準バッファ有効化またはホールドモード) 有効化

29.5.2 VREFBUF 較正制御レジスタ (VREFBUF_CCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 : 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
										RW	RW	RW	RW	RW	RW

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **TRIM[5:0]** : トリミングコード

TRIM コードは、次に示す機構に従ってセットおよび更新された 6 ビットの符号なしデータ（最小 000000、最大 111111）です。

リセット :

TRIM[5:0] は、生産試験中に Flash メモリに格納されたトリミング値 VRS = 0 で自動的に初期化されます。

VRS の変化 :

TRIM[5:0] は、生産試験中に Flash メモリに格納された（VRS 設定に対応する）トリミング値で自動的に初期化されます。

TRIM[5:0] への書き込み :

ユーザは TRIM[5:0] を任意の値で変更できます。これにより、VRS によるトリミング値の制御は恒久的に無効化されます（デバイスがリセットされるまで）。

注 : ユーザアプリケーションがトリミングを行う場合、トリミングコードは、000000 から 111111 まで昇順で開始する必要があります。

29.5.3 VREFBUF レジスタマップ

次の表に、VREFBUF レジスタマップとリセット値を示します。

表 276. VREFBUF レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	VREFBUF_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRS[2:0]		VRR		Res.	HIZ	ENVR				
	リセット値																									0	0	0	0		1	0					
0x04	VREFBUF_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]										
	リセット値																											x	x	x	x	x	x				

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

30 デジタルカメラインタフェース (DCMI)

30.1 概要

このデジタルカメラインタフェースは、外部 8、10、12、または 14 ビット CMOS カメラモジュールからハイスピードデータフローを受信できる同期型パラレルインタフェースです。YCbCr4:2:2/RGB565 プログレッシブビデオおよび圧縮データ (JPEG) など、さまざまなデータフォーマットをサポートします。

30.2 DCMI の主な機能

- 8、10、12、または 14 ビットのパラレルインタフェース
- 埋め込み／外部ラインおよびフレーム同期
- 連続モードまたはスナップショットモード
- クロップ機能
- 以下のデータフォーマットに対応：
 - 8/10/12/14 ビットのプログレッシブビデオ：モノクロまたは Raw ベイヤー
 - YCbCr 4:2:2 プログレッシブビデオ
 - RGB 565 プログレッシブビデオ
 - 圧縮データ：JPEG

30.3 DCMI の機能説明

デジタルカメラインタフェースは、高速のデータフローを受信できる同期型パラレルインタフェースです。最大 14 本のデータライン (DCMI_D[13:0]) と 1 本のピクセルクロックライン (DCMI_PIXCLK) で構成されています。ピクセルクロックの極性はプログラム可能であるため、データはピクセルクロックの立上がりエッジまたは立下がりエッジのどちらでもキャプチャすることができます。

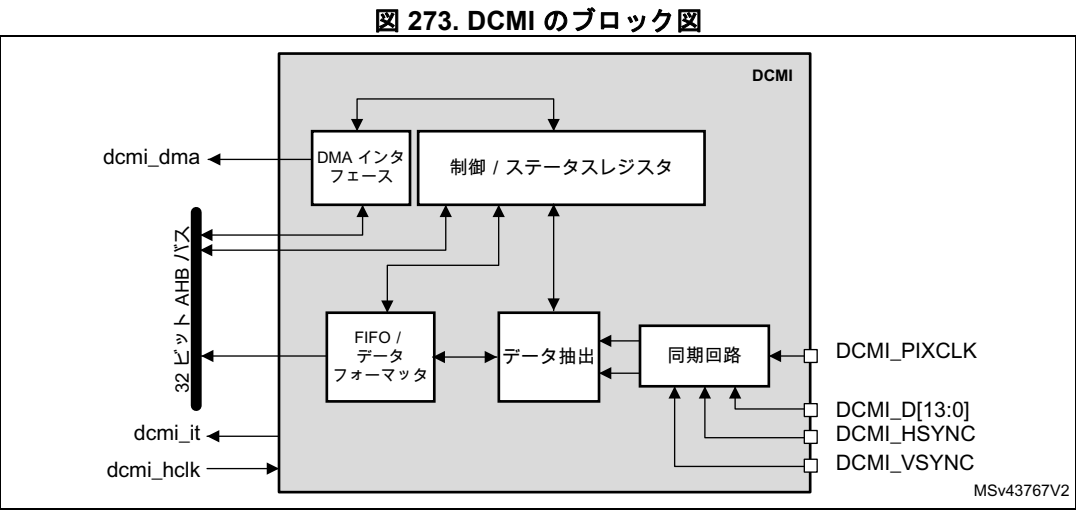
データは 32 ビットのデータレジスタ (DCMI_DR) に格納され、汎用 DMA チャンネルを介して転送されます。画像バッファはカメラインタフェースではなく DMA が管理します。

カメラから受信したデータは、ライン/フレーム (Raw YUV/RGB/Bayer モード) で構成したり、一連の JPEG 画像にすることができます。JPEG 画像を受信できるようにするには、JPEG ビット (DCMI_CR レジスタのビット 3) をセットする必要があります。

データフローは、オプションの DCMI_HSYNC (水平同期) 信号および DCMI_VSYNC (垂直同期) 信号を用いてハードウェアで同期をとることもできれば、データフローに埋め込まれた同期コードで同期をとることもできます。

30.3.1 DCMI ブロック図

図 273 に DCMI ブロック図を示します。



30.3.2 DCMI ピンおよび内部信号

次の表に、DCMI のピンを示します。

表 277. DCMI の入出力ピン			
モード	ピン名	信号タイプ	説明
8 ビット	DCMI_D[7:0]	入力	DCMI データ
10 ビット	DCMI_D[9:0]		
12 ビット	DCMI_D[11:0]		
14 ビット	DCMI_D[13:0]		
DCMI_PIXCLK		入力	ピクセルクロック
DCMI_HSYNC		入力	水平同期/データ有効
DCMI_VSYNC		入力	垂直同期

次の表に、DCMI の内部信号を示します。

表 278. DCMI 内部入力／出力信号		
内部信号名	信号タイプ	説明
dcmi_dma	出力	DCMI DMA リクエスト
dcmi_it	出力	DCMI 割込みリクエスト
dcmi_hclk	入力	DCMI インタフェースクロック

30.3.3 DCMI クロック

デジタルカメラインタフェースでは、DCMI_PIXCLK および HCLK の 2 つのクロックドメインを使用します。DCMI_PIXCLK と HCLK が安定した後、DCMI_PIXCLK で生成された信号は、HCLK の立上がりエッジでサンプリングされます。カメラからのデータが安定し、サンプリング可能状態であることを示すため、イネーブル信号が HCLK ドメインで生成されます。最大 DCMI_PIXCLK 周期は、HCLK 2.5 周期分よりも長い必要があります。

30.3.4 DCMI DMA インタフェース

DCMI_CR レジスタの CAPTURE ビットがセットされているときには、DMA インタフェースがアクティブです。カメラインタフェースがレジスタに 32 ビットデータブロックを受信するたびに、DMA リクエストが生成されます。

30.3.5 DCMI 物理インタフェース

インタフェースは、11/13/15/17 個の入力で構成されています。スレーブモードにのみ対応しています。

カメラインタフェースは、DCMI_CR レジスタの EDM[1:0] ビットに従って、8、10、12、または 14 ビットのデータをキャプチャすることができます。使用されるのが 14 ビット未満である場合、使用されない入力ピンはグラウンドに接続する必要があります。

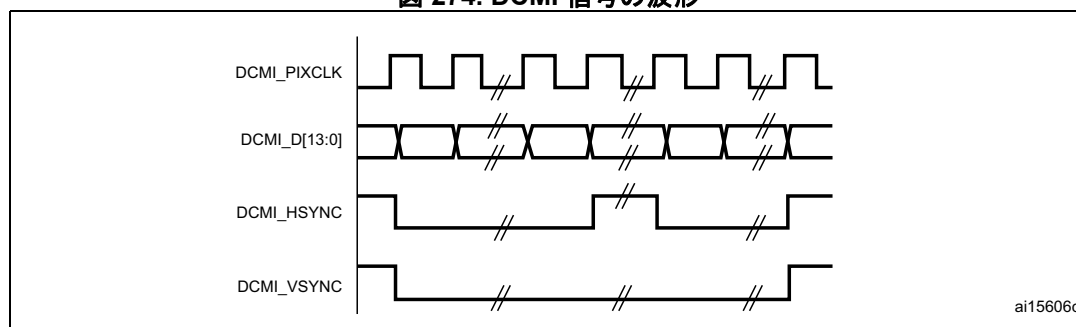
DCMI のピンを表 277 に示します。

データは DCMI_PIXCLK と同期していますが、ピクセルクロックの立上がり/立下がりエッジにおけるデータは極性によって変わります。

DCMI_HSYNC 信号は、ラインの開始/終了を示します。

DCMI_VSYNC 信号は、フレームの開始/終了を示します。

図 274. DCMI 信号の波形



1. DCMI_PIXCLK のキャプチャエッジは立下がりエッジであり、DCMI_HSYNC と DCMI_VSYNC のアクティブ状態は 1 です。
2. DCMI_HSYNC と DCMI_VSYNC の状態は、同時に変化することがあります。

8 ビットデータ

DCMI_CR で EDM[1:0] = 00 の場合、このインタフェースは入力 (DCMI_D[7:0]) で LSB の 8 ビットをキャプチャし、それを 8 ビットデータとして記憶します。DCMI_D[13:8] 入力は無視されます。この場合、32 ビットワードをキャプチャするには、カメラインタフェースでピクセルクロック 4 サイクルが必要です。

最初にキャプチャされたデータバイトは 32 ビットワードの LSB 位置に置かれ、4 番目にキャプチャされたデータバイトは 32 ビットワードの MSB 位置に置かれます。次の表に、2 つの 32 ビットワードでキャプチャされたデータバイトの配置例を示します。

表 279. 32 ビットワード (8 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:24	23:16	15:8	7:0
0	$D_{n+3}[7:0]$	$D_{n+2}[7:0]$	$D_{n+1}[7:0]$	$D_n[7:0]$
4	$D_{n+7}[7:0]$	$D_{n+6}[7:0]$	$D_{n+5}[7:0]$	$D_{n+4}[7:0]$

10 ビットデータ

DCMI_CR で EDM[1:0] = 01 の場合、カメラインタフェースは入力 (DCMI_D[9:0]) で 10 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 10 ビットとして記憶します。DCMI_DR レジスタの残りの MSB (ビット 11 から 15) はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

次の表に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの MSB 側に置かれます。

表 280. 32 ビットワード (10 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:26	25:16	15:10	9:0
0	0	$D_{n+1}[9:0]$	0	$D_n[9:0]$
4	0	$D_{n+3}[9:0]$	0	$D_{n+2}[9:0]$

12 ビットデータ

DCMI_CR で EDM[1:0] = 10 の場合、カメラインタフェースは入力 (DCMI_D[11:0]) で 12 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 12 ビットとして記憶します。残りの MSB はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

次の表に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの MSB 側に置かれます。

表 281. 32 ビットワード (12 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:28	27:16	15:12	11:0
0	0	$D_{n+1}[11:0]$	0	$D_n[11:0]$
4	0	$D_{n+3}[11:0]$	0	$D_{n+2}[11:0]$

14 ビットデータ

DCMI_CR で EDM[1:0] = 11 の場合、カメラインタフェースは入力 (DCMI_D[13:0]) で 14 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 14 ビットとして記憶します。残りの MSB はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

次の表に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの MSB 側に置かれます。

表 282. 32 ビットワード (14 ビット幅) でキャプチャされたデータバイトの配置

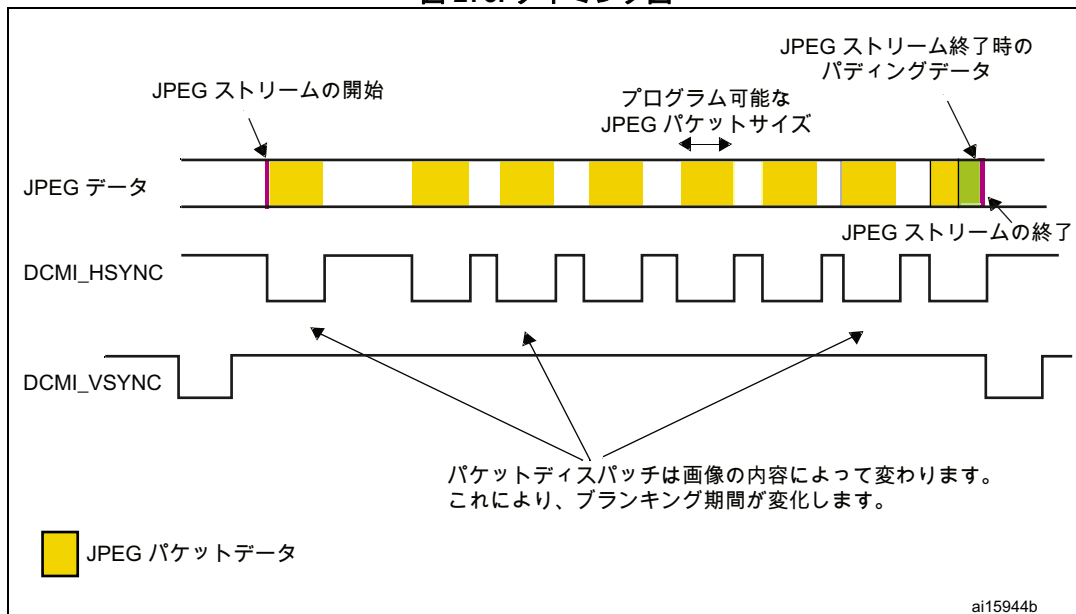
バイトアドレス	31:30	29:16	15:14	13:0
0	0	$D_{n+1}[13:0]$	0	$D_n[13:0]$
4	0	$D_{n+3}[13:0]$	0	$D_{n+2}[13:0]$

30.3.6 DCMI の同期

このデジタルカメラインタフェースは、データ同期またはハードウェア (DCMI_HSYNC および DCMI_VSYNC) 同期に対応します。埋め込み同期を使用する場合、0x00 および 0xFF の値が (有効画像データではなく) 同期のためにのみ使用されているのを確認するのはデジタルカメラモジュールの仕様に依存します。埋め込み同期コードは、8 ビットパラレルデータインタフェース幅 (すなわち DCMI_CR レジスタの EDM[1:0] ビットをクリア) に対してのみサポートされています。

圧縮データに対しては、DCMI はハードウェア同期モードにのみ対応します。この場合、DCMI_VSYNC は画像の開始/終了として使用し、DCMI_HSYNC はデータ有効信号として使用されます。図 275 に対応するタイミング図を示します。

図 275. タイミング図



ハードウェア同期モード

ハードウェア同期モードでは、2 つの同期信号 (DCMI_HSYNC/DCMI_VSYNC) を使用します。

カメラモジュール/モードによっては、データが水平/垂直同期期間中に転送されることがあります。DCMI_HSYNC/DCMI_VSYNC のアクティブ期間に受信されるデータはすべて無視されるため、これらの信号はブランキング信号のように振る舞います。

また、画像を正しく DMA/RAM バッファに転送するため、データ転送は DCMI_VSYNC 信号と同期します。ハードウェア同期モードを選択し、キャプチャを有効 (DCMI_CR の CAPTURE ビットをセットする) にすると、データ転送は DCMI_VSYNC 信号の無効化 (フレームの次の開始) と同期します。

その後、連続したバッファまたは同じ/サーキュラバッファへ DMA によって転送された連続フレームにより、連続的な転送ができるようになります。連続したフレームの DMA 管理を可能にするため、各フレームの終了時には VSIF (垂直同期割込みフラグ) がアクティブになります。

データ同期モード

この同期モードでは、データフローに埋め込まれている 32 ビットコードを使用してデータフローを同期させます。このコードには、有効画像データでは使用されていない 0x00/0xFF 値を使用します。コードには 4 種類あり、いずれも 0xFF0000XY フォーマットを用います。埋め込み同期コードは、8 ビットパラレルデータ幅キャプチャ (DCMI_CR レジスタの EDM[1:0] ビットをクリア) においてのみサポートされています。他のデータ幅に対しては、このモードは予期しない動作となるため、使用しないでください。

注： カメラモジュールでは、このようなコードが 8 つ使用可能です (インタリーブモード)。このため、インタリーブモードはカメラインタフェースではサポートされません (ハーフフレームごとに破棄されてしまうため)。

- モード 2：

4 種類の同期コードにより、次のイベントが通知されます。

- フレーム開始 (FS)
- フレーム終了 (FE)
- ライン開始 (LS)
- ライン終了 (LE)

4 種類のコードの 0xFF0000XY フォーマットにおける XY 値は、プログラム可能です (セクション 30.5.7：DCMI 埋め込み同期コードレジスタ (DCMI_ESCR) を参照)。

「フレーム終了」としてプログラムされた値“0xFF”は、使用されていないすべてのコードが有効なフレーム終了コードとして解釈されることを意味します。

このモードでは、カメラインタフェースが有効になると、最初に発生したフレーム終了 (FE) コードの後にフレーム開始 (FS) コードが続くと、フレームのキャプチャを開始します。

- モード 1：

もう一方のデータ同期モードとして、カメラモード 1 が使用できます。このモードは ITU656 対応です。

以下の同期コードにより次の各イベントが通知されます。

- SAV (アクティブライン) - ライン開始
- EAV (アクティブライン) - ライン終了
- SAV (ブランキング) - インタフレームブランキング期間中のライン終了
- EAV (ブランキング) - インタフレームブランキング期間中のライン終了

このモードは、次のコードをプログラムすることにより対応できます。

- $FS \leq 0xFF$
- $FE \leq 0xFF$
- $LS \leq SAV$ (アクティブ)
- $LS \leq EAV$ (アクティブ)

埋め込み型マスク解除コードもまた、フレーム／ラインの開始コードおよびラインの終了コード用に準備されています。これを使用すれば、プログラムされたコードに対して選択したマスク解除ビットのみを比較することができます。したがって、1 つのビットを選択して埋め込みコード内で比較し、フレーム／ラインの開始や終了を検出することができます。つまり、マスク解除ビットの位置が同じでも、フレーム／ラインの開始や終了のコードは異なる可能性があるということです。

例

FS = 0xA5

FS のマスク解除コード = 0x10

この場合、フレーム開始コードがフレーム開始コードのビット 4 に埋め込まれています。

30.3.7 DCMI キャプチャモード

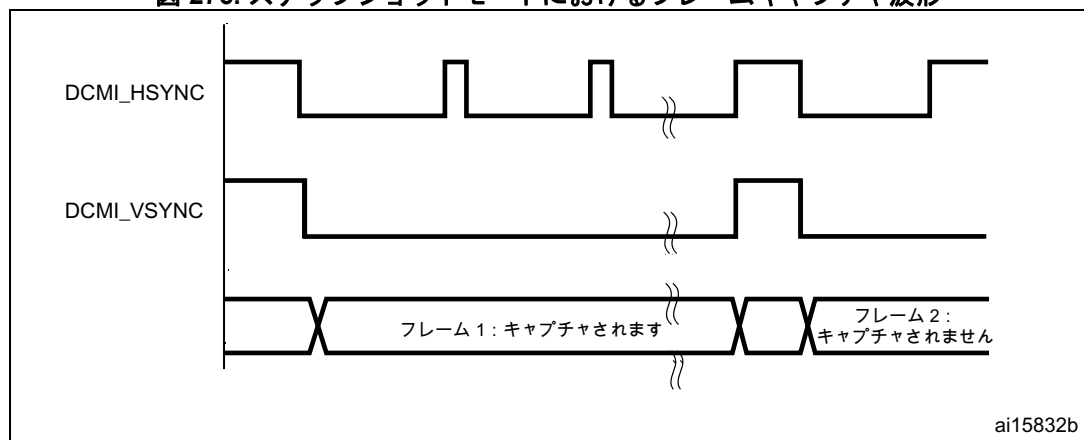
このインタフェースは、スナップショット（単一フレーム）、連続グラブの2種類のキャプチャをサポートしています。

スナップショットモード（単一フレーム）

このモードでは、1つのフレームをキャプチャします（DCMI_CR レジスタの CM = 1）。DCMI_CR で CAPTURE ビットがセットされると、インタフェースはフレームの開始が検出されるのを待ってからデータをサンプリングします。最初の完全なフレームを受信した後、カメラインタフェースは自動的に無効になります（DCMI_CR の CAPTURE ビットがクリアされます）。割込みが有効になっている場合、割込み（IT_FRAME）が生成されます。

オーバーランの場合、フレームが失われ、CAPTURE ビットがクリアされます。

図 276. スナップショットモードにおけるフレームキャプチャ波形

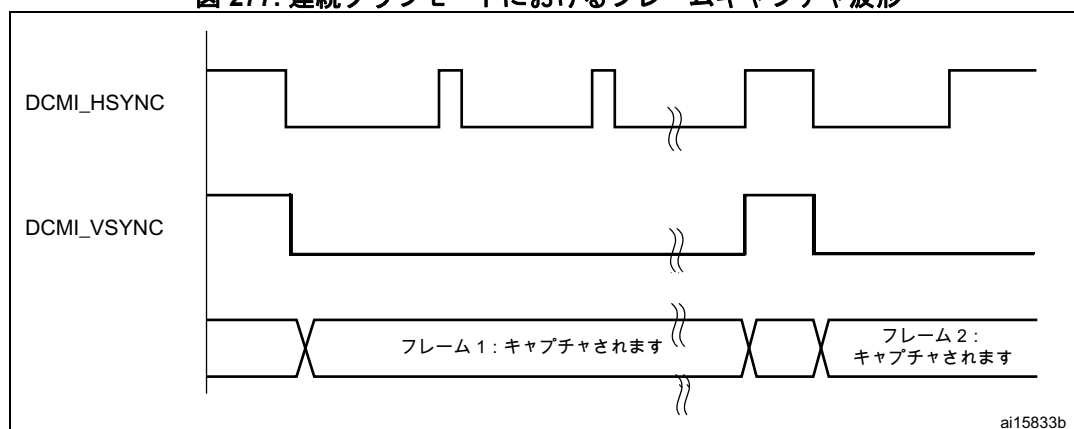


1. ここで、DCMI_HSYNC と DCMI_VSYNC のアクティブ状態は 1 です。
2. DCMI_HSYNC と DCMI_VSYNC の状態は、同時に変化することがあります。

連続グラブモード

このモード（DCMI_CR の CM ビット = 0）では、DCMI_CR の CAPTURE ビットがセットされると、モードによっては次の DCMI_VSYNC または埋め込みフレーム開始でグラブ処理が開始されます。この処理は、DCMI_CR の CAPTURE ビットがクリアされるまで続きます。いったん CAPTURE ビットがクリアされると、現在のフレームの終了までグラブ処理が続きます。

図 277. 連続グラブモードにおけるフレームキャプチャ波形



1. ここで、DCMI_HSYNC と DCMI_VSYNC のアクティブ状態は 1 です。
2. DCMI_HSYNC と DCMI_VSYNC の状態は、同時に変化することがあります。

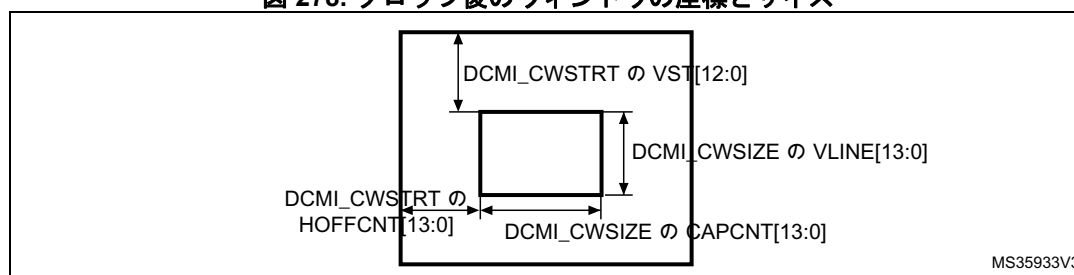
連続グラブモードでは、DCMI_CR の FCRC[1:0] ビットを設定して、すべての画像をグラブ、1 枚おきにグラブ、または 4 枚おきにグラブしてフレームキャプチャ速度を下げるすることができます。

注： ハードウェア同期モード (DCMI_CR の ESS = 0) では、IT_VSYNC 割込みが有効になっていれば DCMI_CR で CAPTURE = 0 になっていても IT_VSYNC 割込みが発生しますので、スナップショットモードと併せて IT_VSYNC 割込みを使用し、2 つのキャプチャ間でのフレーム数を数えることにより、さらにフレームキャプチャレートを下げることができます。これは埋め込みデータ同期モードでは行うことができません。

30.3.8 DCMI クロップ機能

クロップ機能を用いて、カメラインタフェースが受信した画像から長方形のウィンドウを選択することができます。2 つの 32 ビットレジスタ (DCMI_CWSTRT および DCMI_CWSIZE) を使用して開始 (左上の角) 座標とサイズ (水平方向はピクセルクロック数、垂直方向はライン数) を指定します。ウィンドウのサイズは、ピクセルクロック数 (水平方向) とライン数 (垂直方向) で指定します。

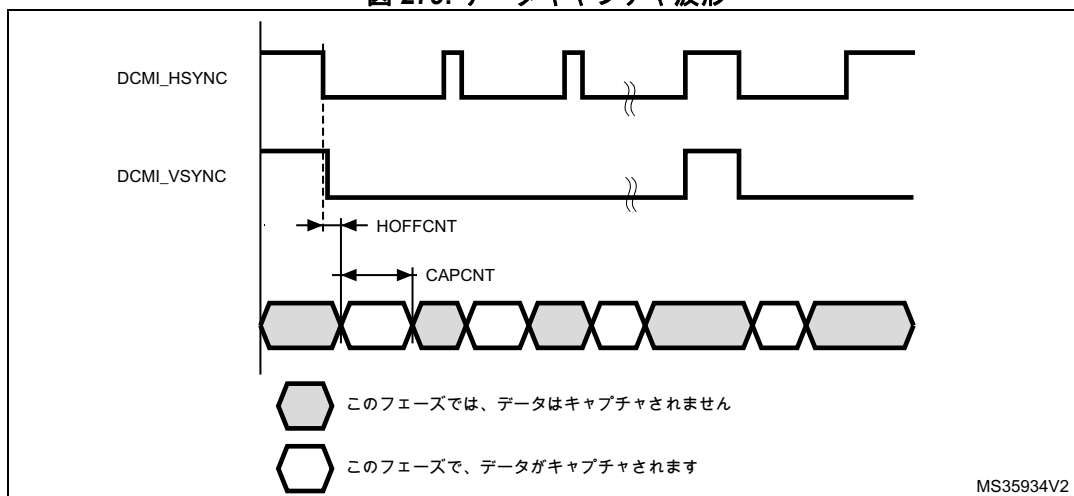
図 278. クロップ後のウィンドウの座標とサイズ



これらのレジスタを用いて、キャプチャウィンドウの開始点の座標をライン番号 (フレーム内、0 から開始) とピクセルクロック数 (ライン上、0 から開始) で指定し、ウィンドウのサイズをライン番号とピクセルクロック数で指定します。CAPCNT 値は、DMA を介してデータを正しく転送できるよう、必ず 4 の倍数となります (LSB の 2 ビットは強制的に 0)。

DCMI_CWSIZE レジスタでライン番号が指定される前に DCMI_VSYNC 信号がアクティブになると、キャプチャが停止し、IT_FRAME 割込みが有効になっていれば IT_FRAME 割込みが生成されます。

図 279. データキャプチャ波形



1. ここで、DCMI_HSYNC と DCMI_VSYNC のアクティブ状態は 1 です。
2. DCMI_HSYNC と DCMI_VSYNC の状態は、同時に変化することがあります。

30.3.9 DCMI JPEG フォーマット

JPEG 画像を受信できるようにするには、DCMI_CR レジスタの JPEG ビットをセットする必要があります。JPEG 画像はラインやフレームとして保存されないで、DCMI_HSYNC がデータイネーブル信号として使用されている間に、DCMI_VSYNC 信号を使用してキャプチャを開始します。1 ライン内のバイト数は 4 の倍数でないこともあります。キャプチャされたデータから完全な 32 ビットワードが構成されるたびに DMA リクエストが生成されるため、この場合のデータ操作には注意が必要です。フレームの終了が検出され、転送される 32 ビットワードが完全に受信されていない場合、残りのデータはゼロでパディングされ、DMA リクエストが生成されます。

クロップ機能および埋め込み同期コードは、JPEG 形式では使用できません。

30.3.10 DCMI の FIFO

AHB 上でのデータ転送レートを管理するため、8 ワードの FIFO が搭載されています。この DCMI には、カメラインタフェースが AHb から読出しを行うたびにインクリメントされる読出しポイントと、FIFO に書込みを行うたびにインクリメントされる書込みポイントを持った単純な FIFO コントローラを備えています。AHb インタフェースがデータ転送速度を維持できない場合にデータの上書きを防ぐオーバーラン保護はありません。

オーバーランや同期信号でのエラーの場合、FIFO はリセットされ、DCMI インタフェースは新規フレームの開始を待ちます。

30.3.11 DCMI データフォーマットの説明

データフォーマット

サポートされるデータは次の 3 種類です。

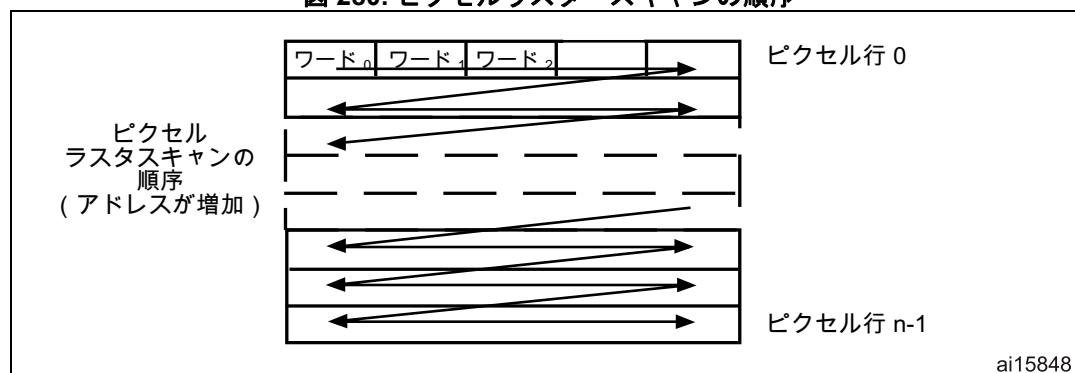
- 8/10/12/14 ビットのプログレッシブビデオ：モノクロまたは Raw ベイヤーフォーマット
- YCbCr 4:2:2 プログレッシブビデオ
- RGB565 プログレッシブビデオ。16 ビット（青が 5 ビット、赤が 5 ビット、緑が 6 ビット）で符号化されたピクセルの転送には、クロック 2 サイクルが必要です。

圧縮データ : JPEG

モノクロ (白黒)、YCbCr、RGB データの場合、最大入力サイズは 2048×2048 ピクセルです。JPEG 圧縮モードでは制限はありません。

モノクロ、RGB、YCbCr の場合、フレームバッファはラスタモードで保存されます。32 ビットワードが使用されます。リトルエンディアン形式のみをサポートします。

図 280. ピクセルラスタスキャンの順序



モノクロフォーマット

特徴 :

- ラスタフォーマット
- ピクセル当たり 8 ビット

次の表に、データ保存方法を示します。

表 283. モノクロプログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:24	23:16	15:8	7:0
0	n + 3	n + 2	n + 1	n
4	n + 7	n + 6	n + 5	n + 4

RGB フォーマット

特徴 :

- ラスタフォーマット
- RGB
- インタリーブ : バッファ 1 つ : R、G、B をインタリーブ (BRGBRGRG のように)
- ディスプレイ出力に対して最適

RGB プレーナフォーマットは、標準 OS フレームバッファディスプレイフォーマットと互換性があります。

サポートされるのは 16 BPP (bits per pixel) : RGB565 (32 ビットワード当たり 2 ピクセル) だけです。

24 BPP (パレットフォーマット) とグレースケールフォーマットは、サポートされません。ピクセルはラスタスキャン順、つまりピクセル行に対して上から下、ピクセル行内では左から右の順で保存されます。ピクセルコンポーネントは R (赤)、G (緑)、B (青) です。すべてのコンポーネントの空間分解能は同じ (4 : 4 : 4 フォーマット) です。ピクセル単位でインタリーブされたコンポーネントと共に、フレームは単一パートで保存されます。

次の表に、データ保存方法を示します。

表 284. RGB プログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:27	26:21	20:16	15:11	10:5	4:0
0	赤 $n+1$	緑 $n+1$	青 $n+1$	赤 n	緑 n	青 n
4	赤 $n+4$	緑 $n+3$	青 $n+3$	赤 $n+2$	緑 $n+2$	青 $n+2$

YCbCr フォーマット

特徴：

- ラスタフォーマット
- YCbCr 4:2:2
- インタリーブ：バッファ 1 つ：Y、Cb、Cr をインタリーブ (CbYCrYCbYCr のように)

ピクセルコンポーネントは Y (輝度または「ルマ」)、Cb、Cr (色度または「クロマ」青および「クロマ」赤) です。各コンポーネントは 8 ビットで符号化されます。輝度と色度は、次の表に示すように、(インタリーブして) 一緒に保存されます。

表 285. YCbCr プログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:24	23:16	15:8	7:0
0	Y $n+1$	Cr n	Y n	Cb n
4	Y $n+3$	Cr $n+2$	Y $n+2$	Cb $n+2$

YCbCr フォーマット - Y のみ

特徴：

- ラスタフォーマット
- YCbCr 4:2:2
- バッファには Y 情報のみが含まれる - モノクロ画像

ピクセルコンポーネントは Y (輝度または「ルマ」)、Cb、Cr (色度または「クロマ」青および「クロマ」赤) です。このモードでは、色度情報がドロップされます。8 ビットでエンコードされた各ピクセルのルマコンポーネントのみが表 286 に示すように保存されます。

結果として、元の YCbCr データと同じ分解能を持つモノクロ画像が得られます。

表 286. YCbCr プログレッシブビデオフォーマットでのデータ保存 - Y 抽出モード

バイトアドレス	31:24	23:16	15:8	7:0
0	Y $n+3$	Y $n+2$	Y $n+1$	Y n
4	Y $n+7$	Y $n+6$	Y $n+5$	Y $n+4$

ハーフ解像度画像抽出

以前の受信モードを変更したもので、モノクロ、RGB、または Y 抽出モードに適用可能です。

このモードは、ハーフ解像度の画像のみを保存するために使用します。OELS および LSM 制御ビットを通して選択されます。

30.4 DCMI 割込み

5 種類の割込みが生成されます。すべての割込みは、ソフトウェアでマスク可能です。グローバル割込み (dcmi_it) は、個別割込みすべての論理和 (OR) です。次の表に、すべての割込みの一覧を示します。

表 287. DCMI 割込み

項目 (割込みの略称)	割込みイベント	イベントフラグ	有効制御ビット	割込みのクリア方法	SLEEP モードの終了	STOP および STANDBY モードの終了
dcmi_it	ラインの終了	LINE_RIS	LINE_IE	LINE_ISC のセット	可	不可
	フレームキャプチャ終了	FRAME_RIS	FRAME_IE	FRAME_ISC のセット	可	不可
	データ受信のオーバーラン	OVR_RIS	OVR_IE	OVR_ISC のセット	可	不可
	同期フレーム	VSYNC_RIS	VSYNC_IE	VSYNC_ISC のセット	可	不可
	埋め込み同期フレーム検出 におけるエラーの検出	ERR_RIS	ERR_IE	ERR_ISC のセット	可	不可

30.5 DCMI レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。すべての DCMI レジスタは、32 ビットワードとしてアクセスする必要があり、そうしないとバスエラーが発生します。

30.5.1 DCMI 制御レジスタ (DCMI_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM[1:0]	
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ENABLE	Res.	Res.	EDM[1:0]		FCRC[1:0]		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OELS** : 奇数／偶数ライン選択 (ライン選択開始)

このビットは LSM フィールドと連携して動作します (LSM = 1)。

0 : インタフェースはフレーム開始後に最初のラインをキャプチャします (2 番目のラインはドロップされる)。

1 : インタフェースはフレーム開始から 2 番目のラインをキャプチャします (最初のラインはドロップされる)。

ビット 19 **LSM** : ライン選択モード

0 : インタフェースはすべての受信したラインをキャプチャします。

1 : インタフェースは 2 つのラインのうちの 1 つをキャプチャします。

ビット 18 **OEBS** : 奇数/偶数バイト選択 (バイト選択開始)

このビットは BSM フィールドと連携して動作します (BSM ≠ 00)。

0 : インタフェースはフレーム/ライン開始から最初のデータ (バイトまたはダブルバイト) をキャプチャします (2 番目のデータはドロップされる)。

1 : インタフェースはフレーム/ライン開始から 2 番目のデータ (バイトまたはダブルバイト) をキャプチャします (最初のデータはドロップされる)。

ビット 17:16 **BSM[1:0]** : バイト選択モード

00 : インタフェースはすべての受信したデータをキャプチャします。

01 : インタフェースは受信したデータから 1 バイトおきにキャプチャします。

10 : インタフェースは 4 つのバイトのうちの 1 つをキャプチャします。

11 : インタフェースは 4 つのバイトのうちの 2 つをキャプチャします。

注 : このモードは EDM[1:0]=00 のみ動作します。その他すべての EDM 値に対し、このフィールドをリセット値にプログラムする必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **ENABLE** : DCMI 有効化

0 : DCMI は無効です。

1 : DCMI は有効です。

注 : このビットを有効にする前に、DCMI 設定レジスタを正しくプログラムする必要があります。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **EDM[1:0]** : 拡張データモード

00 : インタフェースはピクセルクロックごとに 8 ビットデータをキャプチャします。

01 : インタフェースはピクセルクロックごとに 10 ビットデータをキャプチャします。

10 : インタフェースはピクセルクロックごとに 12 ビットデータをキャプチャします。

11 : インタフェースはピクセルクロックごとに 14 ビットデータをキャプチャします。

ビット 9:8 **FCRC[1:0]** : フレームキャプチャレート制御

このビットによって、フレームキャプチャの頻度を規定します。これらのビットは、連続グラブモードでなければ意味を持ちません。スナップショットモードでは無視されます。

00 : 全フレームをキャプチャ。

01 : フレームを 1 つおきにキャプチャ (帯域幅 50 % 減少)

10 : 4 フレームごとに 1 フレームをキャプチャ (帯域幅 75 % 減少)

11 : 予約済み

ビット 7 **VSPOL** : 垂直同期極性

このビットは、データがパラレルインタフェース上で有効でないときの DCMI_VSYNC ピン上でのレベルを示します。

0 : DCMI_VSYNC はアクティブローです。

1 : DCMI_VSYNC はアクティブハイです。

ビット 6 **HSPOL** : 水平同期極性

このビットは、データがパラレルインタフェース上で有効でないときの DCMI_HSYNC ピン上でのレベルを示します。

0 : DCMI_HSYNC はアクティブローです。

1 : DCMI_HSYNC はアクティブハイです。

ビット 5 **PCKPOL** : ピクセルクロック極性

このビットによってピクセルクロックのキャプチャエッジを設定します。

0 : 立下がりエッジでアクティブ

1 : 立上がりエッジでアクティブ

ビット 4 **ESS** : 埋め込み同期選択

0 : ハードウェア同期。データキャプチャ（フレーム／ラインの開始／終了）は DCMI_HSYNC/DCMI_VSYNC 信号と同期します。

1 : 埋め込み同期。データキャプチャはデータフローに埋め込まれた同期コードと同期します。

注 : 8 ビットのパラレルデータに対してのみ有効です。ESS ビットがセットされると、HSPOL/VSPOL は無視されます。

このビットは、JPEG モードでは無効になります。

ビット 3 **JPEG** : JPEG フォーマット

0 : 非圧縮ビデオフォーマット

1 : このビットは JPEG データ転送に使用します。DCMI_HSYNC 信号はデータイネーブルとして使用します。このモードでは、クロップおよび埋め込み同期機能（ESS ビット）は使用できません。

ビット 2 **CROP** : クロップ機能

0 : 画像全体がキャプチャされます。この場合、1 枚の画像フレーム内の総バイト数は 4 の倍数でなければなりません。

1 : クロップレジスタで指定されたウィンドウ内のデータのみがキャプチャされます。クロップウィンドウのサイズがピクチャサイズを超える場合は、そのピクチャサイズのみがキャプチャされます。

ビット 1 **CM** : キャプチャモード

0 : 連続グラブモード - 受信したデータは DMA を通じて転送先メモリに転送されます。バッファの場所やモード（リニアバッファまたはサーキュラバッファ）は、システム DMA を通じて制御されます。

1 : スナップショットモード（単一フレーム） - アクティブになると、インタフェースはフレーム開始を待ってから DMA を通じて単一フレームを転送します。フレーム終了時、CAPTURE ビットは、自動的にリセットされます。

ビット 0 **CAPTURE** : キャプチャイネーブル

0 : キャプチャは無効です。

1 : キャプチャは有効です。

カメラインタフェースは最初のフレーム開始を待ち、その後 DMA リクエストが生成されて受信したデータが転送先メモリに転送されます。

スナップショットモードでは、受信した最初のフレームの終了時に CAPTURE ビットが自動的にクリアされます。

連続グラブモードにおいて、キャプチャ進行中にソフトウェアがこのビットをクリアすると、そのビットは実質的にフレーム終了時にクリアされます。

注 : DMA コントローラとすべての DCMI 設定レジスタを正しくプログラムしてから、このビットを有効にする必要があります。

30.5.2 DCMI ステータスレジスタ (DCMI_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNC	HSYNC
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FNE** : FIFO は空ではありません。

このビットは、FIFO のステータスを示します。

1 : FIFO に有効なデータあり。

0 : FIFO エンプティ

ビット 1 **VSYNC** : 垂直同期

このビットは、極性が正しくプログラムされた DCMI_VSYNC ピンの状態を示します。埋め込み同期コードを使用する場合、このビットは次のような意味を持ちます。

0 : アクティブフレーム

1 : フレーム間同期

埋め込み同期の場合、このビットは DCMI_CR の CAPTURE ビットがセットされている場合にのみ意味を持ちます。

ビット 0 **HSYNC** : 水平同期

このビットは、極性が正しくプログラムされた DCMI_HSYNC ピンの状態を示します。埋め込み同期コードを使用する場合、このビットは次のような意味を持ちます。

0 : アクティブライン

1 : ライン間同期

埋め込み同期の場合、このビットは DCMI_CR の CAPTURE ビットがセットされている場合にのみ意味を持ちます。

30.5.3 DCMI raw 割込みステータスレジスタ (DCMI_RIS)

DCMI_RIS は raw 割込みステータスを示し、読出し専用でアクセスできます。読出し時、このレジスタは DCMI_IER レジスタ値でマスクする前の関連する割込みのステータスを返します。

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
											r	r	r	r	r

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE_RIS** : ライン raw 割込みステータス

DCMI_HSYNC 信号がインアクティブ状態からアクティブ状態に変わると、このビットがセットされます。ラインが有効でなくてもハイになります。

埋め込み同期の場合、このビットは DCMI_CR の CAPTURE ビットがセットされている場合にのみセットされます。

DCMI_ICR レジスタの LINE_ISC ビットをセットすることで、クリアされます。

ビット 3 **VSYNC_RIS** : DCMI_VSYNC raw 割込みステータス

DCMI_VSYNC 信号がインアクティブ状態からアクティブ状態に変わると、このビットがセットされます。埋め込み同期の場合、このビットは DCMI_CR の CAPTURE ビットがセットされている場合にのみセットされます。

DCMI_ICR レジスタの VSYNC_ISC ビットをセットすることで、クリアされます。

ビット 2 **ERR_RIS** : 同期エラー raw 割込みステータス

0 : 同期エラーは検出されていません。

1 : 埋め込み同期文字が正しい順番で受信されません。

このビットは、埋め込み同期モードでのみ有効です。DCMI_ICR レジスタの ERR_ISC ビットをセットすることで、クリアされます。

注 : このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR_RIS** : オーバーラン raw 割込みステータス

0 : データバッファオーバーランは発生していません。

1 : データバッファオーバーランが発生し、データ FIFO が不正です。

DCMI_ICR レジスタの OVR_ISC ビットをセットすることで、このビットはクリアされます。

ビット 0 **FRAME_RIS** : キャプチャ完了 raw 割込みステータス

0 : 新規キャプチャはありません。

1 : フレームキャプチャが完了しています。

このビットは、フレームやウィンドウがキャプチャされるとセットされます。

クロップされたウィンドウの場合、このビットは、クロップの最後のラインの終了時にセットされます。キャプチャされたフレームが空でもセットされます (例 : フレーム外でクロップされたウィンドウなど)。

DCMI_ICR レジスタの FRAME_ISC ビットをセットすることで、このビットはクリアされます。

30.5.4 DCMI 割込み有効レジスタ (DCMI_IER)

DCMI_IER レジスタは、割込みを有効にするために使用します。DCMI_IER ビットの 1 つがセットされると、該当する割込みが有効になります。このレジスタは、読出しと書き込みの両方でアクセスできます。

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE _IE	VSYN C_IE	ERR _IE	OVR _IE	FRAME _IE
											rw	rw	rw	rw	rw

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE_IE** : ライン割込みイネーブル

0 : ライン受信時に割込みは生成されません。

1 : ラインすべてを受信したときに割込みが生成されます。

ビット 3 **VSYN_C_IE** : DCMI_VSYNC 割込みイネーブル

0 : 割込みは生成されません。

1 : DCMI_VSYNC がインアクティブからアクティブ状態に遷移するたびに割込みが生成されます。

DCMI_VSYNC 信号のアクティブ状態は、VSPOL ビットによって定義されます。

ビット 2 **ERR_IE** : 同期エラー割込みイネーブル

0 : 割込みは生成されません。

1 : 埋め込み同期コードが正しい順番で受信されない場合に割込みが生成されます。

注 : このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR_IE** : オーバーラン割込みイネーブル

0 : 割込みは生成されません。

1 : DMA が新しいデータ (32 ビット) を受信する前に最後のデータを転送できなかった場合に割込みが生成されます。

ビット 0 **FRAME_IE** : キャプチャ完了割込みイネーブル

0 : 割込みは生成されません。

1 : 受信したフレーム/クロップウィンドウ(クロップモードの場合)の終了時に割込みが生成されます。

30.5.5 DCMI マスク済み割込みステータスレジスタ (DCMI_MIS)

DCMI_MIS レジスタは、読み出し専用レジスタです。読み出し時、該当する割込みの現在のマスクステータス値 (DCMI_IER の値によって変わります) を返します。DCMI_IER 内の該当するイネーブルビットがセットされ、DCMI_RIS 内の該当ビットがセットされると、このレジスタ内のビットがセットされます。

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
											r	r	r	r	r

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE_MIS** : マスクされたライン割込みステータス

このビットは、マスクされたライン割込みのステータスを示します。

0 : ライン受信時に割込みは生成されません。

1 : ラインが完全に受信され、DCMI_IER の LINE_IE ビットがセットされると割込みが生成されます。

ビット 3 **VSYNC_MIS** : マスクされた VSYNC 割込みステータス

このビットは、マスクされた VSYNC 割込みのステータスを示します。

0 : DCMI_VSYNC の遷移で割込みは生成されません。

1 : DCMI_VSYNC がインアクティブからアクティブ状態に遷移し、DCMI_IER の VSYNC_IE ビットがセットされるたびに割込みが生成されます。

DCMI_VSYNC 信号のアクティブ状態は、VSPOL ビットによって定義されます。

ビット 2 **ERR_MIS** : マスクされた同期エラー割込みステータス

このビットは、マスクされた同期エラー割込みのステータスを示します。

0 : 同期エラー時、割込みは生成されません。

1 : 埋め込み同期コードが正しい順番で受信されず、DCMI_IER の ERR_IE ビットがセットされると割込みが生成されます。

注 : このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR_MIS** : マスクされたオーバーラン割込みステータス

このビットは、マスクされたオーバーフロー割込みのステータスを示します。

0 : オーバーラン時、割込みは生成されません。

1 : DMA が新しいデータ (32 ビット) を受信する前に最後のデータを転送できず、DCMI_IER の OVR_IE ビットがセットされると割込みが生成されます。

ビット 0 **FRAME_MIS** : マスクされたキャプチャ完了割込みステータス

このビットは、マスクされたキャプチャ完了割込みのステータスを示します。

0 : キャプチャ完了で割込みは生成されません。

1 : 受信したフレーム／クロップウィンドウ（クロップモードの場合）の終了時に DCMI_IER の FRAME_IE ビットがセットされると割込みが生成されます。

30.5.6 DCMI 割込みクリアレジスタ (DCMI_ICR)

DCMI_ICR レジスタは、書込み専用です。このレジスタのビットをセットすることによって DCMI_RIS および DCMI_MIS レジスタの対応するフラグをクリアします。0 を書き込んでも、ビットの値は変化しません。

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE _ISC	VSYNC_ ISC	ERR_ _ISC	OVR_ _ISC	FRAME_ ISC
											w	w	w	w	w

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE_ISC** : ライン割込みステータスクリア

このビットをセットすると DCMI_RIS レジスタの LINE_RIS フラグがクリアされます。

ビット 3 **VSYNC_ISC** : 垂直同期割込みステータスクリア

このビットをセットすると DCMI_RIS レジスタの VSYNC_RIS フラグがクリアされます。

ビット 2 **ERR_ISC** : 同期エラー割込みステータスクリア

このビットをセットすると DCMI_RIS レジスタの ERR_RIS フラグがクリアされます。

注 : このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR_ISC** : オーバーラン割込みステータスクリア

このビットをセットすると DCMI_RIS レジスタの OVR_RIS フラグがクリアされます。

ビット 0 **FRAME_ISC** : キャプチャ完了割込みステータスクリア

このビットをセットすると DCMI_RIS レジスタの FRAME_RIS フラグがクリアされます。

30.5.7 DCMI 埋め込み同期コードレジスタ (DCMI_ESCR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEC[7:0]								LEC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSC[7:0]								FSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **FEC[7:0]** : フレーム終了デリミタコード

このバイトによって、フレーム終了デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、FEC の形式で 4 バイトで構成されています。

FEC が 0xFF にプログラムされている場合、使用されていないすべてのコード (0xFF0000XY) はフレーム終了デリミタとして解釈されます。

ビット 23:16 **LEC[7:0]** : ライン終了デリミタコード

このバイトによって、ライン終了デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、LEC の形式で 4 バイトで構成されています。

ビット 15:8 **LSC[7:0]** : ライン開始デリミタコード

このバイトによって、ライン開始デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、LSC の形式で 4 バイトで構成されています。

ビット 7:0 **FSC[7:0]** : フレーム開始デリミタコード

このバイトによって、フレーム開始デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、FSC の形式で 4 バイトで構成されています。

FSC が 0xFF にプログラムされている場合、フレーム開始デリミタは検出されません。FEC コードの後に現れる最初の LSC は、フレームの開始デリミタとして解釈されます。

30.5.8 DCMI 埋め込み同期マスク解除レジスタ (DCMI_ESUR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEU[7:0]								LEU[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSU[7:0]								FSU[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **FEU[7:0]** : フレーム終了デリミタマスク解除

このバイトによって、フレーム終了デリミタのコードに適用するマスクを指定します。

0 : 受信したデータとフレーム終了デリミタを比較すると同時に、DCMI_ESCR の FEC バイトの該当ビットがマスクされます。

1 : 受信したデータとフレーム終了デリミタを比較すると同時に、DCMI_ESCR の FEC バイトの該当ビットが比較されます。

ビット 23:16 **LEU[7:0]** : ライン終了デリミタマスク解除

このバイトによって、ライン終了デリミタのコードに適用するマスクを指定します。

0 : 受信したデータとライン終了デリミタを比較すると同時に、DCMI_ESCR の LEC バイトの該当ビットがマスクされます。

1 : 受信したデータとライン終了デリミタを比較すると同時に、DCMI_ESCR の LEC バイトの該当ビットが比較されます。

ビット 15:8 **LSU[7:0]** : ライン開始デリミタマスク解除

このバイトによって、ライン開始デリミタのコードに適用するマスクを指定します。

0 : 受信したデータとライン開始デリミタを比較すると同時に、DCMI_ESCR の LSC バイトの該当ビットがマスクされます。

1 : 受信したデータとライン開始デリミタを比較すると同時に、DCMI_ESCR の LSC バイトの該当ビットが比較されます。

ビット 7:0 **FSU[7:0]** : フレーム開始デリミタマスク解除

このバイトによって、フレーム開始デリミタのコードに適用するマスクを指定します。

0 : 受信したデータとフレーム開始デリミタを比較すると同時に、DCMI_ESCR の FSC バイトの該当ビットがマスクされます。

1 : 受信したデータとフレーム開始デリミタを比較すると同時に、DCMI_ESCR の FSC バイトの該当ビットが比較されます。

30.5.9 DCMI クロップウィンドウ開始 (DCMI_CWSTRT)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	VST[12:0]												
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	HOFFCNT[13:0]													
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:16 **VST[12:0]** : 垂直方向開始ラインカウント

このライン番号で画像キャプチャを開始します。これより前のラインデータは無視されます。

0x0000 : ライン 1

0x0001 : ライン 2

0x0002 : ライン 3

...

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **HOFFCNT[13:0]** : 水平方向オフセットカウント

キャプチャ開始前にカウントするピクセルクロック数を示します。

30.5.10 DCMI クロップウィンドウサイズ (DCMI_CWSIZE)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	VLIN[13:0]													
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CAPCNT[13:0]													
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:16 **VLIN[13:0]** : 垂直方向ラインカウント

開始点からのキャプチャライン数を示します。

0x0000 : 1 本のライン

0x0001 : 2 本のライン

0x0002 : 3 本のライン

...

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **CAPCNT[13:0]** : キャプチャカウント

同一ライン上でキャプチャする開始点からのピクセルクロック数を示します。この値は、パラレルインタフェースのさまざまなデータ幅に対してワードアライメントが施されたデータに対応する必要があります。

0x0000 : 1 ピクセル
0x0001 : 2 ピクセル
0x0002 : 3 ピクセル
...

30.5.11 DCMI データレジスタ (DCMI_DR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

このデジタルカメラインタフェースは、DMA 転送を要求する前に受信したデータをすべて 32 ビット形式にまとめます。DMA 転送に十分な時間を与え、DMA オーバーランを回避するため、深さ 8 ワードの FIFO が使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BYTE3 [7:0]								BYTE2 [7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE1 [7:0]								BYTE0 [7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **BYTE3[7:0]** : データバイト 3

ビット 23:16 **BYTE2[7:0]** : データバイト 2

ビット 15:8 **BYTE1[7:0]** : データバイト 1

ビット 7:0 **BYTE0[7:0]** : データバイト 0

30.5.12 DCMI レジスタマップ

表 288. DCMI レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	DCMI_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBs	BSM[1:0]	Res.	Res.	ENABLE	Res.	Res.	Res.	EDM[1:0]	Res.	Res.	FCRC[1:0]	Res.	VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	リセット値												0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	
0x04	DCMI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNc	HSYNc		
	リセット値																														0	0	0		
0x08	DCMI_RIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNc_RIS	ERR_RIS	OVR_RIS	FRAME_RIS		
	リセット値																											0	0	0	0	0	0	0	
0x0C	DCMI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNc_IE	ERR_IE	OVR_IE	FRAME_IE			
	リセット値																											0	0	0	0	0	0	0	
0x10	DCMI_MIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNc_MIS	ERR_MIS	OVR_MIS	FRAME_MIS			
	リセット値																											0	0	0	0	0	0	0	
0x14	DCMI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNc_ISC	ERR_ISC	OVR_ISC	FRAME_ISC			
	リセット値																											0	0	0	0	0	0	0	
0x18	DCMI_ESCR	FEC[7:0]								LEC[7:0]								LSC[7:0]								FSC[7:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	DCMI_ESUR	FEU[7:0]								LEU[7:0]								LSU[7:0]								FSU[7:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	DCMI_CWSTRT	Res.	Res.	Res.	VST[12:0]												Res.	Res.	HOFFCNT[13:0]																
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x24	DCMI_CWSIZE	Res.	Res.	VLINe[13:0]												Res.	Res.	CAPCNT[13:0]																	
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	DCMI_DR	BYTE3 [7:0]								BYTE2 [7:0]								BYTE1 [7:0]								BYTE0 [7:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

31 パラレル同期スレーブインタフェース (PSSI)

PSSI ペリフェラルと DCMI (デジタルカメラインタフェース) は同じ回路を使用します。その結果、これらの 2つのペリフェラルを同時に使用することはできません。PSSI を使用する場合、DCMI レジスタにはアクセスできません。その逆も同様です。

さらに、PSSI と DCMI は同じオルタネート機能と割込みベクタを共有します ([セクション 31.3.2 : PSSI ビンおよび内部信号](#)を参照)。

31.1 概要

PSSI は、汎用の同期型 8/16ビットパラレルデータ入出力スレーブインタフェースです。これにより、データが有効になった時を示すデータ有効信号をトランスミッタが送信し、データをサンプリングする準備ができた時を示すフロー制御信号をレシーバが出力することができます。

31.2 PSSI の主な機能

PSSI ペリフェラルの主な機能は以下のとおりです。

- スレーブモード動作
- 8ビットまたは 16ビットのパラレルデータ入力または出力
- 8ワード (32バイト) FIFO
- データイネーブル (PSSI_DE) オルタネート機能入力およびレディ (PSSI_RDY) オルタネート機能出力

これらを選択すると、これらの信号によって、トランスミッタはデータが有効であること、レシーバはデータをサンプリングする準備ができたことを示すことができます。あるいはそのどちらも可能です。

31.3 PSSI の機能説明

PSSI は、高速のデータフローを送受信できる同期型パラレルスレーブインタフェースです。最大 16本のデータライン (PSSI_D[15:0]) と 1本のクロックライン (PSSI_PDCK) で構成されています。クロックの極性は、データがクロックの立上がりエッジまたは立下がりエッジのどちらでもキャプチャまたは送信されるように設定できます。

通常、汎用 DMA チャンネルは、データレジスタ (PSSI_DR) を介して 32ビットのパックデータを渡すために使用されます。

データフローは、連続、またはオプションの PSSI_DE (データイネーブル) および PSSI_RDY (レディ) 信号を使用してハードウェアによって同期されます。

[図 281](#) に PSSI ブロック図を示します。

31.3.1 PSSI ブロック図

図 281. PSSI のブロック図

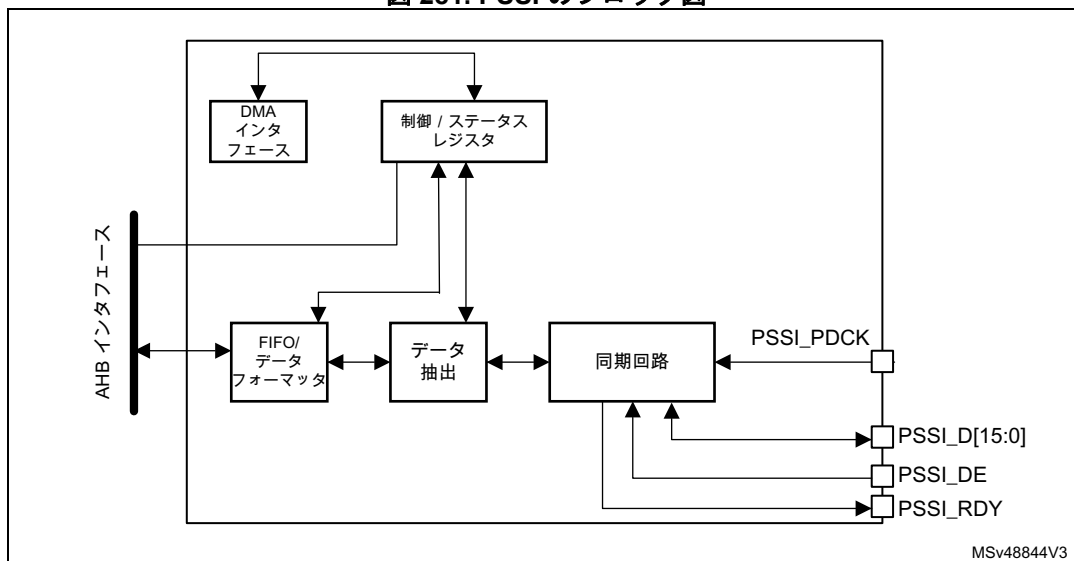
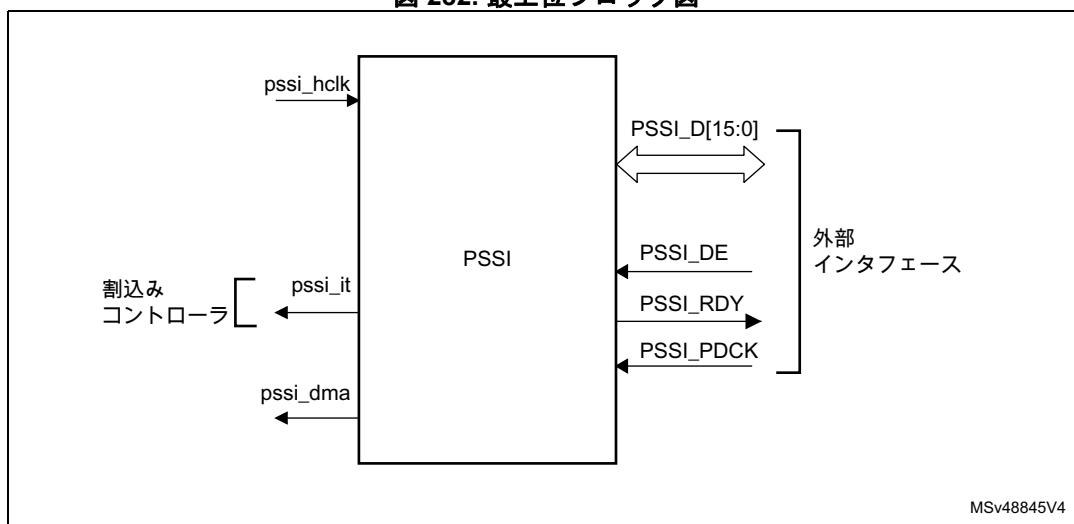


図 282. 最上位ブロック図



31.3.2 PSSI ピンおよび内部信号

PSSI インタフェースは 19本のピンで構成されていますが、パラレルデータを転送するには 9本の信号で十分です。表 289 に PSSI ピンを示します。

PSSI ENABLE ビット (PSSI_CR のビット 14) が 1 にセットされていると、オルタネート機能と割り込みベクタは PSSI に関連付けられます。そうでない場合は DCMI に関連付けられます。DCMI ENABLE ビット (DCMI_CR のビット 15) と PSSI ENABLE ビット (PSSI_CR のビット 14) を同時に 1 にセットしないでください。たとえば、オルタネート機能 PSSI_PDCK/DCMI_PIXCK を使用するように GPIO が設定されている場合、PSSI_CR/ENABLE が 1 にセットされていれば、アクティブになるのは PSSI_PDCK 機能です。

表 289. PSSI の入出力ピン

PSSI 信号名	共有される DCMI 信号	信号タイプ	説明
PSSI_PDCK	DCMI_PIXCK	入力	パラレルデータのクロック入力
PSSI_D[15:0]	DCMI_D[13:0]	入力／出力	送信時はデータ出力、受信時はデータ入力
PSSI_DE	DCMI_HSYNC	入力	データイネーブル信号：受信時のデータ有効信号または送信時のフロー制御信号
PSSI_RDY	DCMI_VSYNC	出力	レディ信号：受信時のフロー制御信号または送信時のデータ有効信号

表 290 に、PSSI の内部入力／出力信号を示します。

表 290. PSSI 内部入力／出力信号

内部信号名	信号タイプ	説明
psii_it	出力	割込み
psii_dma	出力	DMA リクエストあり
psii_hclk	入力	AHB クロック

31.3.3 PSSI クロック

AHB クロック周波数は PSSI_PDCK 周波数よりも少なくとも 2.5倍以上高くなければなりません。周波数比が 2.5未満の場合、データが転送中に破損または失われる可能性があります。

データ転送は PSSI_PDCK と同期しています。PSSI_PDCK の極性は、CKPOL ビット (PSSI_CR のビット 5) を介して、次のように設定できます。

- CKPOL = 0 の場合
 - 入力ピンは PSSI_PDCK の立下がりエッジでサンプリングされます。
 - 出力ピンは PSSI_PDCK の立上がりエッジで駆動されます。
- CKPOL = 1 の場合
 - 入力ピンは PSSI_PDCK の立上がりエッジでサンプリングされます。
 - 出力ピンは PSSI_PDCK の立下がりエッジで駆動されます。

31.3.4 PSSI データ管理

データの方向

データ転送の方向は、OUTEN 制御ビット (PSSI_CR のビット 31) を介して次のように設定されます。

- OUTEN を 0 にクリアすると (デフォルト設定)、PSSI は受信モードで動作し、データピンにデータが入力されます。
- OUTEN を 1 にセットすると、ペリフェラル は送信モードで動作し、データピンにデータが出力されます。

OUTEN は、ENABLE ビットが 0 にクリアされている場合にのみ変更できます。

データレジスタおよび DMA

データは、PSSI_DR データレジスタを使用して FIFO との間で以下のように転送されます。

- 受信モードでは、PSSI_DR を読み取ることによって FIFO からデータを読み出す必要があります。
- 送信モードでは、PSSI_DR に書き込むことによって FIFO にデータを書き込む必要があります。

PSSI_DR へのワード (32ビット) アクセスおよび PSSI_DR[15:0] へのハーフワード (16ビット) アクセスは、すべてのモードで許可されます。PSSI_DR[7:0] へのバイト (8ビット) アクセスは、PSSI が一度に 8ビットを転送するように設定されている場合 (PSSI_CR レジスタの EDM = 00) にのみ許可されます。

CPU の負荷を軽減するには、DMA を使用して PSSI FIFO との間でデータを転送することを推奨します。これを使用する場合、PSSI_DR レジスタを介してデータを転送するように DMA を設定する必要があります。32ビット転送を使用すると、帯域幅が最適化され、バス負荷が軽減されます。ただし、8ビットおよび 16ビットの転送も許可されています。

DMA を使用するには、PSSI DMA イネーブルビット (PSSI_CR の DMAEN) を 1 (デフォルト設定) にセットします。DMAEN が 1 にセットされている場合、FIFO で 32ビット転送の準備 (受信モードで 4つの有効バイト、または送信モードで 4つの空バイト) ができると、DMA 転送が開始されます。したがって、受信モードでは、DMA が 8ビット転送を実行するように設定されている場合でも、FIFO に 3バイト以下しかない場合、DMA 転送は開始されません。

RTT4B および RTT1B ステータスビット (PSSI_SR) は、CPU が FIFO との間で直接転送を実行するときに役立ちます。RTT4B が 1 にセットされると、FIFO が 4バイトを転送する準備ができていることになります。つまり、受信モードの FIFO で少なくとも 4つの有効なバイト、または送信モードで少なくとも 4つの空きバイトがある状態になります。RTT1B が 1 にセットされると、FIFO が 1バイトを転送する準備ができていることになります。つまり、受信モードの FIFO で少なくとも 1つの有効なバイト、または送信モードで少なくとも 1つの空きバイトがある状態になります。

8 ビットデータ

PSSI パラレルインタフェースは、EDM[1:0] 制御ビット (PSSI_CR のビット 11:10) に応じて、8ビットのデータ (D[7:0] を使用) または 16ビットのデータ (D[15:0] を使用) を転送できます。8ビットの設定を選択 (EDM[1:0] を 00 にセット) した場合、使っていない D[15:0] ピンを GPIO または他の機能に使用できます。

PSSI_CR の EDM[1:0] が 00 にプログラムされている場合、インタフェースは D[7:0] ピンを使用して 8ビットを転送します。この場合、D[15:8] は使用されず、32ビットワードを転送するには 4サイクルの PSSI_PDCK が必要です。

最下位バイト (ビット 7:0) は転送された最初のバイトに対応し、最上位バイト (ビット 31:24) は転送された 4番目のバイトに対応します。表 291 に、2つの 32ビットワード内のデータバイトの配置を示します。

表 291. 32 ビットワード (8 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:24	23:16	15:8	7:0
0	D _{n+3} [7:0]	D _{n+2} [7:0]	D _{n+1} [7:0]	D _n [7:0]
4	D _{n+7} [7:0]	D _{n+6} [7:0]	D _{n+5} [7:0]	D _{n+4} [7:0]

16 ビットデータ

PSSI_CR の EDM[1:0] が 11 にプログラムされている場合、インタフェースは D[15:0] ピンを使用して 16 ビットを転送します。この場合、32 ビットワードを転送するには 2 サイクルの PSSI_PDCK がが必要です。

最下位のハーフワード（ビット 15:0）は転送された最初のハーフワードに対応し、最上位のハーフワード（ビット 31:16）は転送された 2 番目のハーフワードに対応します。表 292 に、2 つの 32 ビットワード内のデータの配置を示します。

表 292. 32 ビットワード（16 ビット幅）でキャプチャされたデータバイトの配置

バイトアドレス	31:16	15:0
0	D _{n+1} [15:0]	D _n [15:0]
4	D _{n+3} [15:0]	D _{n+2} [15:0]

FIFO データバッファとエラー状態

8 ワードの FIFO は、パフォーマンスの向上に役立ち、オーバーランとアンダーランを回避します。

受信モードでレディ信号（PSSI_RDY）が無効になっている場合、FIFO がフルのときにクロックのアクティブエッジが発生すると、オーバーランエラーが生成されます。この場合、入力データは失われます。

送信モードでデータイネーブル信号（PSSI_DE）が無効になっている場合、FIFO が空のときにクロックのアクティブエッジが発生すると、アンダーランエラーが生成されます。この場合、予測不能なデータが出力されます。

OVR_RIS ステータスビットは、オーバーランまたはアンダーランが発生したことを示します。これらのイベントが発生した場合、割り込みを生成することができます。

31.3.5 PSSI オプション制御信号

データイネーブル（PSSI_DE）オルタネート機能入力

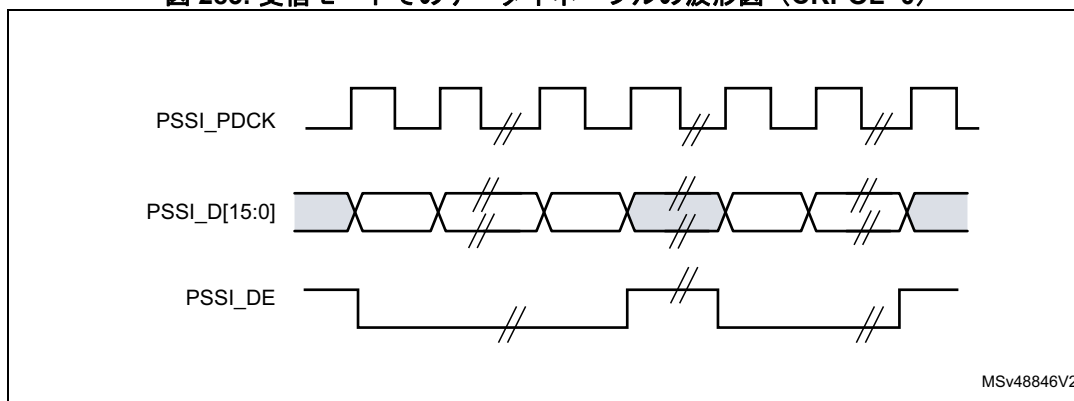
データイネーブル信号 PSSI_DE はオプション信号です。それは、現在のサイクル中にデータの転送が有効になっていることを示すために、データ転送元／トランスミッタによって駆動されます。PSSI_DE が非アクティブの場合、受信側は次のクロックエッジでデータをサンプリングしてはならないことを意味します。

このオルタネート機能信号は、DERDYCFG（PSSI_CR のビット 20:18）制御ビットを使用して有効にできます。PSSI_DE の極性は、DEPOL 制御ビット（PSSI_CR のビット 6）を介して設定されます。DEPOL を 0 にクリアすると PSSI_DE はアクティブローになり、DEPOL を 1 にセットするとアクティブハイになります。

PSSI_DE 信号の方向は、OUTEN の値によって定義されます。それはデータの方向と同じです。

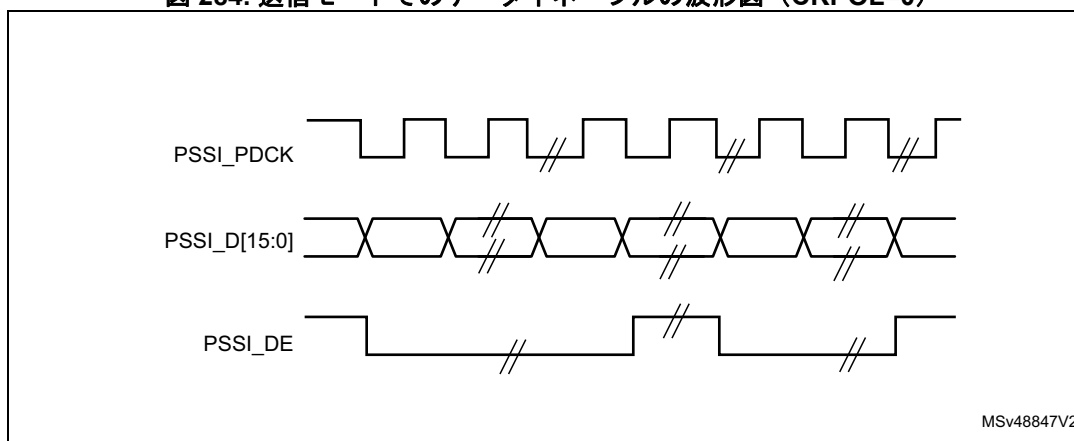
PSSI_DE オルタネート機能入力が（DERDYCFG を介して）受信モード（OUTEN が 0 にクリアされている状態）で有効になっている場合、PSSI はデータ（D[15:0]）のサンプリングに使用されるものと同じ PSSI_PDCK エッジで PSSI_DE をサンプリングします。PSSI_DE がアクティブな場合、サンプリングされたデータは FIFO に保存されます。そうでない場合、サンプリングされたデータは無効と見なされ、破棄されます。送信側デバイスは、PSSI_DE をデータ有効信号として使用し、現在のサイクルのデータが無効な場合に PSSI_DE を非アクティブにできます。このフロー制御機能により、アンダーランエラーを回避できます。

図 283. 受信モードでのデータイネーブルの波形図 (CKPOL=0)



PSSI_DE オルタネート出力機能が (DERDYCFG を介して) 送信モード (OUTEN=1) で有効になっている場合、PSSI はデータ (D[15:0]) の駆動に使用されるものと同じ PSSI_PDCK エッジで PSSI_DE を駆動します。新しい 8 または 16 ビットデータ (PSSI_CR の EDM[1:0] 制御ビットでプログラムされたもの) が内部 FIFO での送信に利用できる場合、このデータはデータ出力 (D[15:0]) に出力され、そして、PSSI_DE 出力は現在の PSSI_PDCK エッジでアクティブになります。そうでない場合 (TX FIFO が空の場合)、D[15:0] 出力は次のクロックエッジで変更されないまま、PSSI_DE 出力は非アクティブになります。

図 284. 送信モードでのデータイネーブルの波形図 (CKPOL=0)



レディ (PSSI_RDY) オルタネート機能出力

レディ信号 PSSI_RDY はオプション信号です。それは、受信デバイスによって駆動され、現在のサイクル中にデータを受け入れられるかどうかを示します。PSSI_RDY が非アクティブの場合、受信側は次のクロックエッジでデータをサンプリングしてはならないことを意味します。

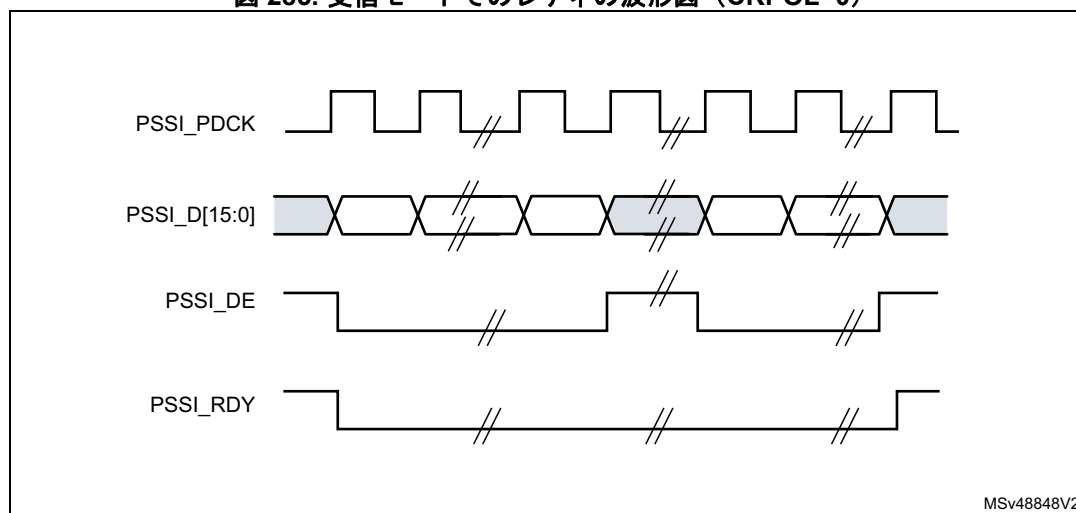
このオルタネート機能信号は、DERDYCFG 制御ビット (PSSI_CR のビット 20:18) を使用して有効にできます。PSSI_RDY の極性は、RDYPOL 制御ビット (PSSI_CR のビット 6) を介して設定されます。RDYPOL を 0 にクリアすると PSSI_RDY はアクティブラーになり、RDYPOL を 1 にセットするとアクティブハイになります。

PSSI_RDY 信号の方向は、OUTEN (PSSI_CR のビット 31) によって定義されます。PSSI_DE およびデータ信号とは逆の方向に設定されます。

PSSI_RDY オルタネート出力機能が (DERDYCFG を介して) 受信モード (OUTEN=0) で有効になっている場合、PSSI はデータ (D[15:0]) をサンプリングした後、PSSI_PDCK の半サイクル後に PSSI_RDY を駆動します。FIFO にさらにデータを受信するのに十分な空きスペースがある場合、

PSSI は PSSI_RDY 信号をアクティブにします。そうではなく、FIFO がフルでさらにデータを受信することができない場合、PSSI は PSSI_RDY 信号を非アクティブにします。送信側デバイスは、PSSI_RDY が非アクティブであることを検出すると、次のサイクルで現在のデータを繰り返す必要があります。このフロー制御機能により、システムが (DMA を介して) データフローに対応できない場合に、PSSI はオーバーランエラーを回避できます。

図 285. 受信モードでのレディの波形図 (CKPOL=0)



PSSI_RDY オルタネート入力機能が (DERDYCFG を介して) 送信モード (OUTEN=1) で有効になっている場合、PSSI はデータ (D[15:0]) が駆動されるのと逆の PSSI_PDCK のエッジで PSSI_RDY 信号をサンプリングします。PSSI_RDY 信号が非アクティブの場合、PSSI は同じデータ (D[15:0]) を保持し、次の PSSI_PDCK クロックサイクル中に有効なデータが利用可能であることが PSSI_DE によって通知されます。そうではなく、PSSI_RDY 信号がアクティブとしてサンプリングされると、TX FIFO (利用可能な場合) からの次のデータがデータ出力 (D[15:0]) に出力されます。TX FIFO に新しいデータがない場合、PSSI はデータ出力値を保持し、PSSI_DE 信号を (有効な場合) 非アクティブとして出力します。

受信デバイスは PSSI_RDY を使用してデータフローを制御し、システムが (DMA を介して) データフローに対応できない場合に、オーバーランエラーを回避できます。

双方向 PSSI_DE/PSSI_RDY 信号

PSSI_CR レジスタで DEPOL と RDYPOL の両方が 1 にセットされ、DERDYCFG が 111 または 100 にセットされている場合、1つのピンをデータイネーブル (PSSI_DE) 機能とレディ (PSSI_RDY) 機能の両方に使用できます。この場合、選択したオルタネート機能 (DERDYCFG = 111 の場合は PSSI_DE、または DERDYCFG = 100 の場合は PSSI_RDY) に対応する GPIO をオープンドレインとして設定する必要があります。他方のデバイスも、信号線をオープンドレインとして駆動するように設定する必要があり、弱いプルアップを信号線に適用する必要があります。

こうして、信号は双方向になります。送信側が信号線をローに駆動する (データが無効であることを示す) か、受信側が信号線をローに駆動する (現在のデータをサンプリングしていないことを示す) 場合、両方のデバイスは現在のサイクルでデータが転送されていないことを知ります。

図 286. 双方向 PSSI_DE/PSSI_RDY の波形

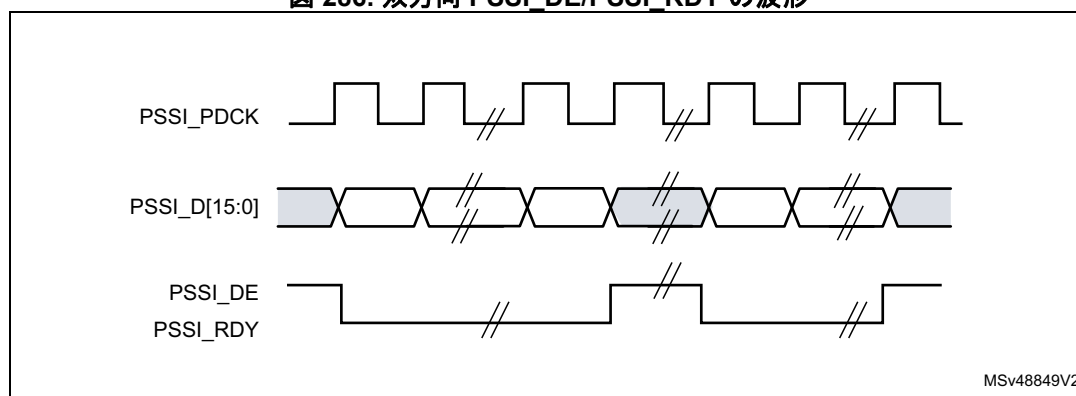
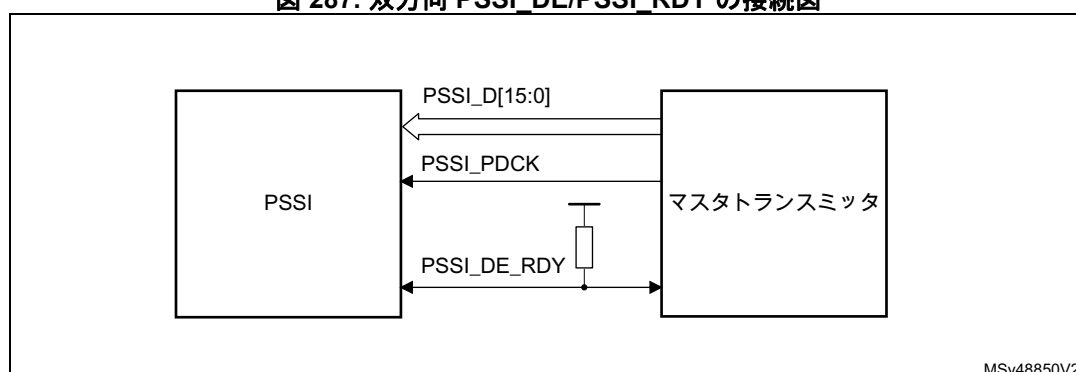


図 287. 双方向 PSSI_DE/PSSI_RDY の接続図



31.4 PSSI 割込み

PSSI は割込みを 1 つだけ (IT_OVR) 生成します。したがって、グローバル割込み (pssi_it) と同等です。割込みのリストについては、表 293 を参照してください。

PSSI と DCMI は同じ割込みベクタを共有します。PSSI ENABLE ビット (PSSI_CR のビット 14) が 1 にセットされていると、PSSI によってこれらの割込みがトリガされます。そうでない場合、それらは DCMI によって制御されます。

DCMI ENABLE ビット (DCMI_CR のビット 14) と PSSI ENABLE ビットを同時に 1 にセットしないでください。

表 293. PSSI 割込みリクエスト

項目 (割込みの 略称)	DCMI と 共用	割込みイベント	イベント フラグ	有効制御ビット	割込みのクリア 方法	低消費電力 モードの終了
IT_OVR	IT_OVR	受信モードでの オーバーラン または送信 モードでの アンダーランを 示す	OVR_RIS	OVR_IE	OVR_ISC	該当なし

31.5 PSSI レジスタ

PSSI_DR 以外の PSSI レジスタへ 8 ビットの書き込みまたは 16 ビットの書き込み操作を行うと、バスエラーになります。32ビットの読出しおよび書き込み操作が許可されています。

31.5.1 PSSI 制御レジスタ (PSSI_CR)

アドレスオフセット : 0x00

リセット値 : 0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTEN	DMAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DERDYCFG[2:0]			Res.	Res.
rw	rw										rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ENABLE	Res.	Res.	EDM[1:0]		Res.	RDYPOL	Res.	DEPOL	CKPOL	Res.	Res.	Res.	Res.	Res.
	rw			rw	rw		rw		rw	rw					

ビット 31 **OUTEN** : データ方向選択ビット

- 0 : 受信モード : データは PSSI_PDCK と同期して入力されます。
- 1 : 送信モード : データは PSSI_PDCK と同期して出力されます。

ビット 30 **DMAEN** : DMA 有効

- 0 : DMA 転送は無効になります。DMA 転送が無効になっているとき、ユーザアプリケーションは PSSI_DR レジスタに直接アクセスできます。
- 1 : DMA 転送は有効です (デフォルトの設定)。汎用 DMA コントローラの DMA チャンネルは、PSSI_DR との間の転送を実行するように設定する必要があります。

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:18 **DERDYCFG[2:0]** : データイネーブルおよびレディの設定

- 000 : PSSI_DE および PSSI_RDY の両方無効
 - 001 : PSSI_RDY のみ有効
 - 010 : PSSI_DE のみ有効
 - 011 : PSSI_RDY および PSSI_DE オルタネート機能の両方有効
 - 100 : PSSI_RDY および PSSI_DE 機能の両方有効 - PSSI_RDY ピンで双方向 ([1208ページの双方向 PSSI_DE/PSSI_RDY 信号](#) を参照)
 - 101 : PSSI_RDY 機能のみ有効ですが、PSSI_DE ピンに配置されます。
 - 110 : PSSI_DE 機能のみ有効ですが、PSSI_RDY ピンに配置されます。
 - 111 : PSSI_RDY および PSSI_DE 機能の両方有効 - PSSI_DE ピンで双方向 ([1208ページの双方向 PSSI_DE/PSSI_RDY 信号](#) を参照)
- PSSI_RDY 機能が PSSI_DE ピンに配置されている場合でも (設定 101 または 111)、極性を決定するのは RDYPOL ビットです。同様に、PSSI_DE 機能が PSSI_RDY ピンに配置されている場合でも (設定 110 または 111)、極性を決定するのは DEPOL ビットです。

ビット 17:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **ENABLE** : PSSI 有効化

0 : PSSI は無効です。

1 : PSSI は有効です。

ENABLE が 0 にクリアされると、FIFO の内容は一括されます。

注： **ENABLE = 1 のとき、ENABLE ビット自体を除いて、PSSI_CR の内容を変更しないでください。ENABLE が 0 から 1 に変わるとすぐに、すべての設定ビットが変化する可能性があります。DMA コントローラとすべての PSSI 設定レジスタを正しくプログラムしてから、ENABLE ビットを 1 にセットする必要があります。**
ENABLE ビットと DCMI ENABLE ビット (DCMI_CR のビット 15) を同時に 1 にセットしないでください。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **EDM[1:0]** : 拡張データモード

00 : インタフェースはパラレルデータクロックごとに 8 ビットデータをキャプチャします。

01 : 予約済み。選択できません。

10 : 予約済み。選択できません。

11 : インタフェースはパラレルデータクロックごとに 16 ビットデータをキャプチャします。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **RDYPOL** : レディ (PSSI_RDY) の極性

このビットは、データがパラレルインタフェース上で有効でないときの PSSI_RDY ピン上でのレベルを示します。

0 : PSSI_RDY はアクティブローです (0 は受信側が受信準備完了していることを示します)。

1 : PSSI_RDY はアクティブハイです (1 は受信側が受信準備完了していることを示します)。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DEPOL** : データイネーブル (PSSI_DE) の極性

このビットは、データがパラレルインタフェース上で有効でないときの PSSI_DE ピン上でのレベルを示します。

0 : PSSI_DE はアクティブローです (0 はデータが有効であることを示します)。

1 : PSSI_DE はアクティブハイです (1 はデータが有効であることを示します)。

ビット 5 **CKPOL** : パラレルデータのクロック極性

このビットは、OUTEN に応じて、パラレルクロックのキャプチャエッジまたは出力の駆動に使用されるエッジを設定します。

0 : 入力に対して立下がりエッジでアクティブ、または出力に対して立上がりエッジでアクティブ

1 : 入力に対して立上がりエッジでアクティブ、または出力に対して立下がりエッジでアクティブ

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

31.5.2 PSSI ステータスレジスタ (PSSI_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTT1B	RTT4B	Res.	Res.
												r	r		

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **RTT1B** : FIFO は 1バイトを転送する準備ができています

1 : FIFO は 1バイト (32ビット) の転送準備ができています。受信モードでは、少なくとも 1バイトの有効データが FIFO にあることを意味します。送信モードでは、少なくとも 1バイトの空きが FIFO にあることを意味します。

0 : FIFO は 1バイトの転送準備ができていません。

ビット 2 **RTT4B** : FIFO は 4バイトを転送する準備ができています

1 : FIFO は 4バイト (32ビット) の転送準備ができています。受信モードでは、少なくとも 4バイトの有効データが FIFO にあることを意味します。送信モードでは、少なくとも 4バイトの空きが FIFO にあることを意味します。

0 : FIFO は 4バイトの転送準備ができていません。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

31.5.3 PSSI raw 割込みステータスレジスタ (PSSI_RIS)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

PSSI_RIS は raw 割込みステータスを提供します。このレジスタは読出し専用です。読出し時、これは PSSI_IER レジスタ値でマスクする前の関連する割込みのステータスを返します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_RIS	Res.
														r	

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OVR_RIS** : データパッファのオーバーラン／アンダーラン raw 割込みステータス

0 : オーバーラン／アンダーランは発生しませんでした。

1 : オーバーラン／アンダーランが発生しました : 受信モードでのオーバーラン、送信モードでのアンダーラン。

このビットは、PSSI_ICR の OVR_ISC ビットに 1 を書き込むことによってクリアされます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

31.5.4 PSSI 割込み有効レジスタ (PSSI_IER)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

PSSI_IER レジスタは、割込みを有効にするために使用します。PSSI_IER ビットの 1つがセットされると、該当する割込みが有効になります。このレジスタは、読出しと書込みの両方のモードでアクセスできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_I E	Res.
														rw	

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OVR_IE** : データバッファのオーバーラン／アンダーラン割込みイネーブル

0 : 割込みは生成されません。

1 : オーバーランエラーまたはアンダーランエラーのどちらかが発生した場合、割込みが生成されます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

31.5.5 PSSI マスク済み割込みステータスレジスタ (PSSI_MIS)

この PSSI_MIS レジスタは読出し専用です。読出し時、これは、該当する割込みの現在のマスクされたステータス値 (PSSI_IER の値によって変わります) を返します。PSSI_IER 内の該当するイネーブルビットがセットされ、PSSI_RIS 内の該当ビットがセットされている場合、このレジスタ内のビットがセットされます。

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_ MIS	Res.
														r	

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OVR_MIS** : データバッファのオーバーラン／アンダーランのマスクされた割込みステータス

このビットは、PSSI_IER/OVR_IE と PSSI_RIS/OVR_RIS の両方とも 1 にセットされている場合のみ、1 にセットされます。

0 : オーバーラン／アンダーランエラーが発生したとき、割込みは生成されません。

1 : オーバーランまたはアンダーランエラーがあり、PSSI_IER で OVR_IE ビットがセットされている場合、割込みが生成されます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

31.5.6 PSSI 割込みクリアレジスタ (PSSI_ICR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

PSSI_ICR レジスタは、書込み専用です。このレジスタのビットに 1 を書き込むことによって PSSI_RIS および PSSI_MIS レジスタの対応するビットをクリアします。0 を書き込んでも、ビットの値は変化しません。このレジスタを読み出すと常に 0 が返されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_I SC	Res.
														w	

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OVR_ISC** : データバッファのオーバーラン／アンダーラン割込みステータスクリア

このビットに 1 を書き込むと、PSSI_RIS の OVR_RIS ビットがクリアされます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

31.5.7 PSSI データレジスタ (PSSI_DR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

受信モード (OUTEN = 0) では、DMA コントローラはこのレジスタから受信データを読み取る必要があります。PSSI_DR へ書込み操作を行うと、エラー応答が発生します。FIFO で有効なバイト数より多くのバイトが読み出されると、無効なバイトはゼロを返します。

送信モード (OUTEN = 1) では、DMA コントローラはこのレジスタに送信データを書き込む必要があります。PSSI_DR への読出し操作を行うと、エラー応答が発生します。

PSSI_DR に対して、32ビット、16ビット、および 8ビットのアクセスはすべてサポートされています。たとえば、16ビットの読出し／書込み操作では、2バイトが FIFO から削除／FIFO へ追加されます。ただし、8ビットアクセスは、PSSI が一度に 8ビットのデータを転送するように設定されている場合 (PSSI_CR の EDM=00) にのみ許可されます。EDM が 0 にセットされていない場合に PSSI_DR へ 8ビットアクセスを行うと、エラー応答が発生します。

すべてのアクセスはバイト 0 を含む必要があります。すなわち、8ビットアクセスはビット 7~0 に対して行い、16ビットアクセスはビット 15~0 に対して行う必要があります。バイト 0 を含まないアクセスを行うと、エラー応答が発生します。

PSSI_CR の ENABLE ビットが 0 にセットされているときに PSSI_DR にアクセスすると、エラー応答が発生します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BYTE3 [7:0]								BYTE2 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE1 [7:0]								BYTE0 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **BYTE3[7:0]** : データバイト 3

ビット 23:16 **BYTE2[7:0]** : データバイト 2

ビット 15:8 **BYTE1[7:0]** : データバイト 1

ビット 7:0 **BYTE0[7:0]** : データバイト 0

31.5.8 PSSI レジスタマップ

表 294. PSSI レジスタのマップとリセット値

オフ セット	レジスタ名とリ セット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	PSSI_CR	OUTEN	DMAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DERDYCFG		Res.	Res.	Res.	ENABLE[2:0]	Res.	Res.	Res.	EDM		Res.	RDYPOL	Res.	DEPOL	CKPOL	Res.	Res.	Res.	Res.	Res.
	リセット値	0	1										0	0	0				0			0	0		0	0	0	0					
0x04	PSSI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTT1B	RTT4B	Res.	Res.
	リセット値																													0	0		
0x08	PSSI_RIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_RIS	Res.
	リセット値																														0		
0x0C	PSSI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_IE	Res.
	リセット値																														0		
0x10	PSSI_MIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_MIS	Res.
	リセット値																															0	
0x14	PSSI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR_ISC	Res.
	リセット値																															0	
0x18- 0x24	予約済み	Res.																															
0x28	PSSI_DR	BYTE3 [7:0]								BYTE2 [7:0]								BYTE1 [7:0]								BYTE0 [7:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

32 真性乱数発生器 (RNG)

32.1 概要

RNG は、完全エントロピー出力を 32 ビットサンプルとしてアプリケーションに提供する真性乱数発生器です。ライブエントロピーソース（アナログ）と内部conditioning component(条件付けコンポーネント)から構成されています。

RNG は NIST SP 800-90B 準拠のエントロピーソースで、非決定論的乱数発生器 (NDRBG) を構築するために使用できます。

RNG 真性乱数発生器は、NIST SP800-90B に従って事前認定されています。また、ドイツ BSI 文書の統計的検定 AIS-31 (T0~T8) 規格に従って試験を行いました。

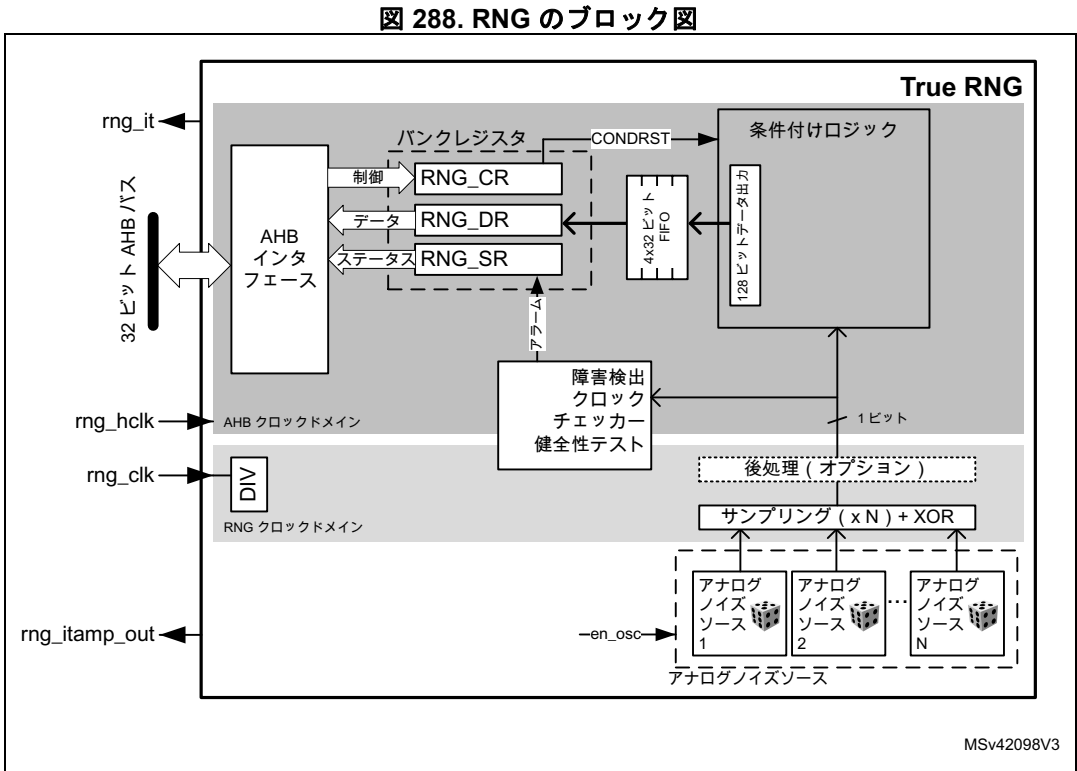
32.2 RNG の主な機能

- RNG は、NIST SP800-90B 承認条件付けステージで条件付けされたアナログエントロピーソースによって作成された 32 ビットの真の乱数を供給します。
- 非決定論的乱数発生器 (NDRBG) を構築するためのエントロピーソースとして使用できます。
- NIST 構成では、 $f_{\text{AHB}} < f_{\text{threshold}}$ の場合、412 AHB クロックサイクル（それ以外の場合は 256 RNG クロックサイクル）ごとに 4 つの 32 ビットのランダムサンプルを生成します。
- RNG には、起動時テストおよび NIST SP800-90B 承認の継続的健全性テスト（繰り返し回数テストおよび適応比率テスト）が内蔵されており、特定のエラー管理と関連付けられています。
- 本機能を無効にして消費電力を低減できます。また、自動低消費電力モードで有効にできます（デフォルト設定）。
- 32 ビットワードのシングルアクセスでのみアクセス可能な AMBA® AHB スレーブペリフェラルが搭載されています（それ以外には AHB バスエラーが発生し、書込みアクセスは無視されます）。

32.3 RNG の機能説明

32.3.1 RNG ブロック図

図 288 に RNG ブロック図を示します。



32.3.2 RNG 内部信号

表 295 では、RNG をより良く理解するために内部の信号のリストを記載しております。STM32 (パッド上) の入出力の信号ではありません。

表 295. RNG 内部入力／出力信号

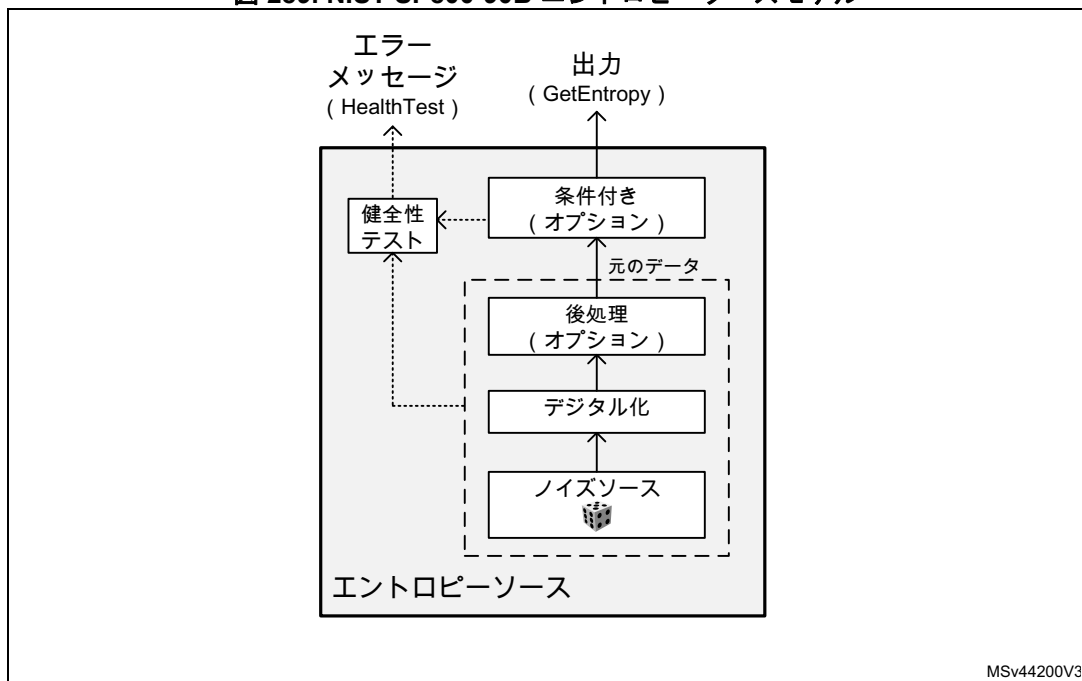
信号名	信号タイプ	説明
rng_it	デジタル出力	RNG グローバル割込みリクエスト
rng_hclk	デジタル入力	AHB クロック
rng_clk	デジタル入力	RNG 専用クロック、rng_hclk と非同期
rng_itamp_out	デジタル出力	予期しないハードウェアフォールトが発生したとき、TAMP への RNG 内部タンパイメント信号 (XOR されます)。この信号がトリガされると、RNG はランダムサンプルの配信を停止し、再び使用可能になるにはリセットと新しい初期化が必要です。

32.3.3 乱数の生成

真性乱数発生器 (RNG) は、決定論的な間隔で AHB インタフェースを介して真にランダムなデータを伝送します。

その境界内で、RNG には [図 289](#) に示されている必要なすべての NIST コンポーネントが統合されています。これらのコンポーネントは、アナログノイズソース、デジタル化ステージ、条件付けアルゴリズム、健全性監視ブロック、およびエントロピーソースとの相互作用に使用される 2 つのインタフェース、すなわち GetEntropy と HealthTest です。

図 289. NIST SP800-90B エントロピーソースモデル



上に図示されたコンポーネントの詳細を以下に示します。

ノイズソース

ノイズソースは、非決定論的なエントロピーを生成する構成要素のノイズソースで、これは出力の不確実性に関連したものです(デジタル化されたノイズソースやソースデータ)このノイズソースは 1 ビットのサンプルを提供します。以下の要素で構成されます。

- それぞれ排他的論理和がとられたフリーランニングのリングオシレータの出力 3 つに基づく複数のアナログノイズソース (x6)。セクション 32.3.8 : RNG の低消費電力時の取り扱いに記載しているとおり、電力低減のためにそれらのアナログオシレータは無効化できます。
- 単一のアナログ出力に向けた、すべてのノイズソースの排他的論理和。
- 専用クロック入力 (rng_clk (内蔵分周回路あり)) によってクロック供給されたこの出力のサンプリングステージ。1 ビットの元データ出力を伝送します。

このノイズソースサンプリングは、AHB インタフェースクロック周波数 (rng_hclk) から独立しています。ソフトウェアは統合された分周回路を使用してサンプリング周波数を減少させることができます。

注： [セクション 32.6 : RNG エントロピーソース検証](#)では、推奨される RNG クロック周波数および関連する分周値が示されています。

後処理

NIST 構成では、サンプリングされたノイズソースに後処理は適用されません。非 NIST 構成 B ([セクション 32.6.2](#) で定義されている) では、正規化バイアス解除が適用されます。すなわち、ビットの半分はサンプルノイズソースから取得し、あとの半分は反転したサンプルノイズソースから取得します。

条件付け

RNG の条件付けコンポーネントは、結果として得られる固定長ビット列出力 (128 ビット) のエントロピー率を増加させる決定論的機能です。NIST SP800-90B の目標は、出力での完全なエントロピーです (128 ビット)。

2 つの乱数生成の間に必要な時間、RNG を初期化してから最初のサンプルを利用できるまでに必要な時間については、[セクション 32.5 : RNG 処理時間](#)に記載しています。

出力バッファ

データ出力バッファは、条件付けコンポーネントから出力された 32 ビットワードを 4 つまで格納できます。RNG_DR レジスタを通じて出力 FIFO から 4 ワード読み出されると、128 ビット条件付け出力レジスタの内容は出力 FIFO にプッシュされ、新たな条件付けが自動的に始まります。[セクション 32.5 : RNG 処理時間](#)に指定された数のクロックサイクルの後に、新たな 4 ワードが条件付け出力レジスタに追加されます。

乱数が RNG_DR レジスタを介して使用できるときは必ず、DRDY フラグが 0 から 1 に変わります。このフラグは、RNG_DR レジスタから 4 ワード読み出した後、出力バッファが空になるまでハイのままになります。

注： 割込みが有効の場合、このデータレディフラグが 0 から 1 に遷移するときに割込みが生成されます。その後、上記で説明したように割込みは RNG によって自動的にクリアされます。

健全性チェック

このコンポーネントはエントロピーソース全体（そのノイズソースとともに）を開始し、その後期待通りに実行し、高い確立と信頼性ですばやく不具合の取り込みを保証することを確実にします。

RNG は、NIST SP800-90B に準拠して、以下のような健全性チェック機能を実行します。記述されている閾値は、レジスタ RNG_HTCR の推奨値に対応します（[セクション 32.6.2](#) の設定 A を参照）。

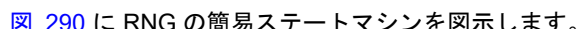
- リセットしてから RNG を初めてエントロピーソースとして使用するまでに実行される起動時健全性テスト：
 - 繰り返し回数テストで、ノイズソースが 42 ビット以上連続して一定値（0 または 1）を提供した場合にエラーフラグを立てます。
 - 連続する 1024 ビットのウィンドウで実行される適応比率テスト：RNG は、ノイズソース出力の最初のビットが 711 回以上繰り返されていないことを検証します。
 - 既知解テストで、条件付けステージを検証します。
- ノイズソースの出力で無制限に実行する継続的健全性テスト：
 - 起動時テストで実行されるものと同様の繰り返し回数テスト。
 - 起動時テストで実行されるものと同様の適応比率テスト。
- ベンダ固有の継続テスト
 - 遷移回数テスト。ノイズソースが 2 ビットパターン（01 または 10）を 32 回以上連続して提供した場合にエラーフラグを立てます。
 - 「遅すぎる」サンプリングクロックのリアルタイム検出。1 RNG クロックサイクル（分周前）が 32 で分周された AHB クロックサイクルより小さい場合にエラーフラグを立てます。
- デジタル化ノイズソース（元のデータ）のオン・デマンド・テスト
 - エントロピーソースを再起動し、起動時テストを再実行することでサポートされます（[セクション 32.3.4 : RNG 初期化](#)のソフトウェアリセットシーケンスを参照）。その他のオン・デマンド・テスト（ソフトウェアベース）はサポートされていません。

エラーコンディションが検出された時に、RNG_SR レジスタの CECS および SECS ステータスビットに表示されます。詳細は[セクション 32.3.7 : エラー管理](#)を参照してください。

注：エラーが検出された場合、割込みを生成することができます。

上記の健全性テストの閾値は、RNG_HTCR レジスタの値を変更することで変更されます。詳細については、[セクション 32.6 : RNG エントロピーソース検証](#)を参照してください。

32.3.4 RNG 初期化

 [図 290](#) に RNG の簡易ステートマシンを図示します。

RNG を有効化した後（RNG_CR の RNGEN = 1）、以下の一連のイベントが発生します。

- アナログノイズソースが有効になると、デフォルトでは RNG はアナログ出力のサンプリングを開始して 128 ビットの条件付けシフトレジスタに入力するまで 16 RNG クロックサイクル待ちます。
- 条件付けハードウェアは起動すると、元のデータサンプルで起動時動作テストと既知解テストを自動的にトリガします。
- 起動時健全性テストが完了するとき、この間に、3 つの 128 ビットのノイズソースサンプルが使用されます。
- 条件付けステージの内部入力データバッファに 128 ビットが再入力され、RNG 設定（NIST または非 NIST）によって定義された多数の条件付けが実行されます。その後、出力バッファには後処理の結果が入力されます。
- 出力バッファは RNG の使用方法に応じて自動的に再入力されます。

関連する初期化時間については、[セクション 32.5 : RNG 処理時間](#)を参照してください。

図 290. RNG 初期化の概要

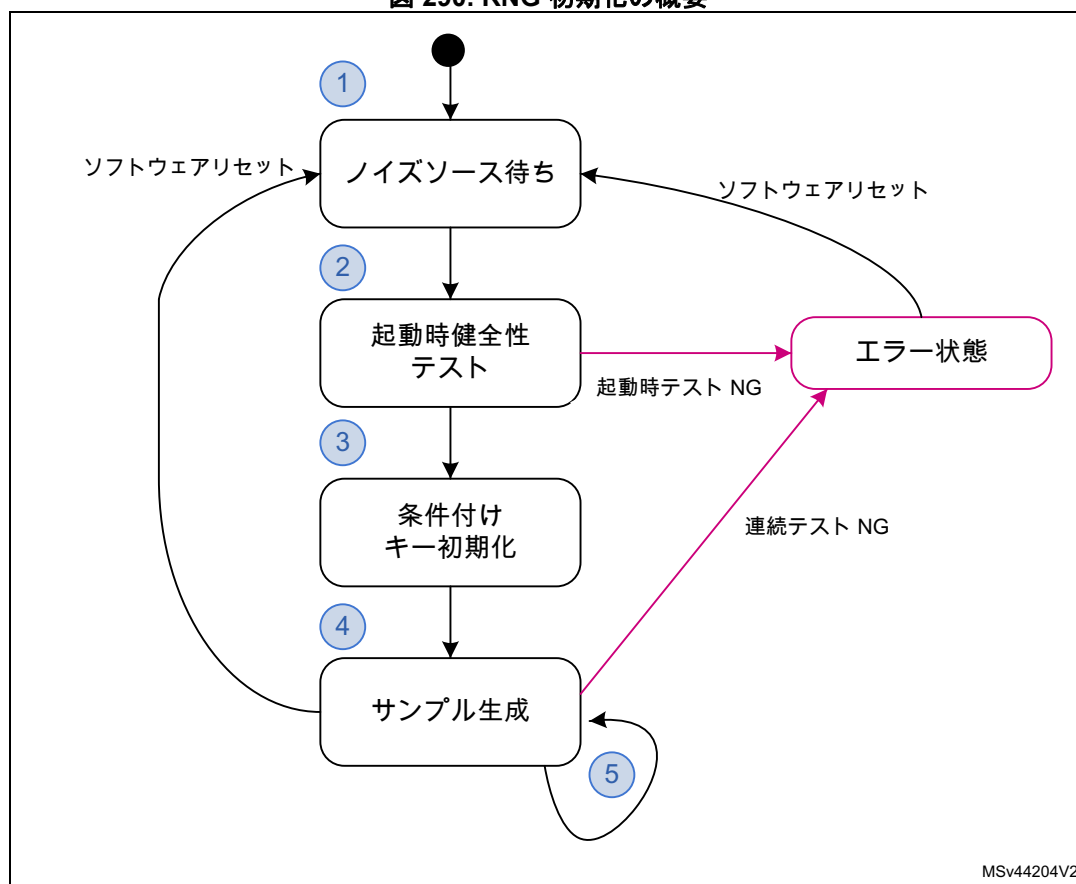


図 290 では、以下によって実施される考えられるソフトウェアリセットのシーケンスも説明しています。

1. 同じ RNG 設定および新しい CLKDIV で、RNG_CR レジスタにビット RNGEN = 0 および CONDRST = 1 を書き込みます（必要に応じて）。
2. 次に、RNG_CR レジスタに RNGEN = 1 および CONDRST = 0 を書き込みます。
3. 初期化が完了した後、乱数が準備できるまで待ちます。

注： RNG ペリフェラルが RCC（ハードウェアリセット）を介してリセットされると、RNG レジスタで最適なランダム性のための RNG 設定が失われます。CONFIGLOCK をセットしたソフトウェアリセットでは、RNG 設定が保持されます。

32.3.5 RNG 操作

通常動作

割込みを使用して RNG を作動させるには、次の手順を推奨します。

1. [セクション 32.6: RNG エントロピーソース検証](#)を参照して、特定の RNG 設定がアプリケーションに必要なかどうかを確認します。
 - 必要であれば、RNG_CR レジスタに正しい RNG 設定とともにビット CONDRST = 1 を書き込みます。次に、RNG_CR レジスタにビット CONDRST = 0、割込み有効化ビット IE = 1、RNG 有効化ビット RNGEN = 1 を、さらに書き込みます。
 - 必要でない場合、RNG_CR レジスタに割込み有効化ビット IE = 1、RNG 有効化ビット RNGEN1 = 1 を書き込みます。
2. 乱数の準備ができたとき、またはエラーが発生したとき、割込みが生成されるようになります。したがって、割込みがあるたびに次のことをチェックします。
 - 発生したエラーはありません。RNG_SR レジスタで SEIS および CEIS ビットを 0 にセットする必要があります。
 - 乱数が準備できています。RNG_SR レジスタで DRDY ビットを 1 にセットする必要があります。
 - 上記 2 つの条件に当てはまる場合、最大で連続 4 回まで RNG_DR レジスタの内容が読み出せるようになります。条件付け出力バッファに有効なデータがある場合には、アプリケーションによってさらに 4 ワード読み出すことができます（この場合には DRDY ビットはハイのままです）。上記の条件のどちらかまたは両方に当てはまらない場合、RNG_DR レジスタを読み出さないでください。エラーが発生した場合には、[セクション 32.3.7](#)に記載されたリカバリシーケンスを使用する必要があります。

ポーリングモードで RNG を作動させるには、次の手順を推奨します。

1. [セクション 32.6: RNG エントロピーソース検証](#)を参照して、特定の RNG 設定がアプリケーションに必要なかどうかを確認します。
 - 必要であれば、RNG_CR レジスタに正しい RNG 設定とともにビット CONDRST = 1 を書き込みます。次に、RNG_CR レジスタにビット CONDRST = 0、RNG 有効化ビット RNGEN = 1 を、さらに書き込みます。
 - 必要でない場合、RNG_CR レジスタの RNGEN ビットを 1 にセットして、RNG を有効にするだけです。
2. RNG_SR レジスタを読み出し、次を確認します。
 - 発生したエラーはありません (SEIS および CEIS ビットは 0 にセットされていなければなりません)。
 - 乱数が準備できています (DRDY ビットは 1 にセットされていなければなりません)。
3. 上記の条件に当てはまる場合、最大で連続 4 回まで RNG_DR レジスタの内容を読み出します。条件付け出力バッファに有効なデータがある場合にはアプリケーションによってさらに 4 ワード読み出すことができます（この場合には DRDY ビットはハイのままです）。上記の条件のどちらかまたは両方に当てはまらない場合、RNG_DR レジスタを読み出さないでください。エラーが発生した場合には、[セクション 32.3.7](#)に記載されたリカバリシーケンスを使用する必要があります。

注： データが準備できていない場合 (DRDY = 0)、RNG_DR は 0 を返します。
RNG_DR がゼロではないことを常に確認することをお勧めします。ゼロの場合、RNG_SR ポーリングと RND_DR 出力読出しの間にシードエラーが発生した（まれなイベント）ことを意味するからです。

乱数生成周期がアプリケーションに対して懸念となる場合、および NIST の準拠が必須ではない場合、[セクション 32.6 : RNG エントロピーソース検証](#)に記述されている RNG の設定“B”を使用して、より

高速な RNG 設定を選択できます。乱数生成速度の増加に関する概要は、[セクション 32.5 : RNG 処理時間](#)に示しています。

低消費電力動作

消費電力がアプリケーションに対して懸念となる場合、[セクション 32.3.8 : RNG の低消費電力時の取り扱い](#)に示すように低消費電力にする取り扱いを使用できます。

ソフトウェア後処理

AIS-31 または NIST SP800-90B の承認を満たすために期待される特定のソフトウェア後処理／条件付けはありません。

内蔵されている健全性チェック機能の説明は[セクション 32.3.3 : 乱数の生成](#)を参照してください。

32.3.6 RNG クロック供給

RNG は、AHB バスクロックと専用 RNG クロックの 2 種類の異なるクロックで動作します。

AHB クロックは、AHB バンクレジスタおよび条件付けコンポーネントにクロック供給するために使用されます。RNG クロックは、プログラム可能な分周回路 (RNG_CR レジスタの CLKDIV ビットフィールドを参照) と組み合わせてノイズソースサンプリングに使用されます。推奨されるクロック設定は、[セクション 32.6 : RNG エントロピーソース検証](#)で詳しく説明されています。

注 : RNG_CR レジスタの CED ビットが 0 にセットされている場合、内部分周器の前の RNG クロック周波数は 32 で分周された AHB クロック周波数より高くする必要があります。そうしないと、クロックチェッカーが必ずクロックエラーのフラグ (RNG_SR レジスタの CECS = 1) を立てます。

詳細 (AHB および RNG クロックドメイン) については、[セクション 32.3.1 : RNG ブロック図](#)を参照してください。

32.3.7 エラー管理

このセクションに詳細を示しているとおり、乱数生成と並行して、健全性チェックブロックにより、正しいノイズソース動作と RNG ソースクロックの周波数が検証されます。関連するエラー状態も記載されています。

クロックエラー検出

クロックエラー検出が有効なとき (CED = 0)、RNG クロック周波数が低すぎる場合、RNG は CEIS と CECS の両方のビットを 1 にセットして、クロックエラーが発生したことを示します。この場合、アプリケーションによって RNG クロックが正しく設定されていること ([セクション 32.3.6 : RNG クロック供給](#)を参照) をチェックして、CEIS ビットの割込みフラグをクリアする必要があります。CECS ビットは、クロック条件が正常なときに自動的にクリアされます。

注 : クロックエラーは、生成された乱数には影響しないため、アプリケーションは RNG_DR レジスタを読み出すことができます。

CEIS は、RNG によって CECS が 1 にセットされているときのみセットされます。

ノイズソースエラー検出

ノイズソース (またはシード) エラーが発生すると、RNG は乱数生成を停止し、SEIS および SECS ビットを両方 1 にセットして、シードエラーが発生したことを示します。RNG_DR レジスタにおいて値が利用可能な場合であっても、エントロピーが十分でない可能性があるので使用することはできません。

シードエラーからの完全なリカバリを行うには、次のシーケンスを使用する必要があります。

1. CONDRST に 1、そして 0 を書き込んでソフトウェアリセットします（詳細はビットフィールドの説明を参照）。SECS がセットされている場合のみ、このステップが必要です。実際、SEIS がセットされ、SECS がクリアされていると、RNG が自動的にリセット（自動リセット）を実行したことを意味します。この場合、アプリケーションは SEIS ビット割り込みフラグをクリアする必要があります。
2. 手順 1 で SECS がセットされていた（自動リセットではない）場合、RNG_CR レジスタで CONDRST がクリアされるまで待ってから、RNG_SR レジスタで SEIS がクリアされていることを確認します。そうでない場合、RNG_SR レジスタの SEIS ビットをクリアするだけです。
3. 手順 1 で SECS がセットされていた（自動リセットではない）場合、RNG によって SECS がクリアされるまで待ちます。乱数生成が正常に戻ります。

注： シードエラーの後、SECS がクリアされると、RNG は乱数の生成を再開します。

アプリケーションが RNG_CR レジスタの ARDIS ビットを設定すると、自動リセットは無効になります。手順 1 で CONDRST を使用する必要があります。

RNG タンパエラー

RNG によって予期しないエラーが検出されると、TAMP ペリフェラルで内部タンパイイベントがトリガされ、RNG はランダムデータの提供を停止します。

このイベントが発生すると、セキュアアプリケーションが集中リセット管理またはグローバル SoC リセットのいずれかを使用して RNG ペリフェラルをリセットする必要があります。その後、再び RNG を適切に初期化する必要があります。

32.3.8 RNG の低消費電力時の取り扱い

消費電力が懸念される場合、RNG_CR レジスタの RNGEN ビットを 0 にセットして DRDY ビットを 1 にセットするとすぐに、RNG を無効化できます。RNGEN = 0 であっても後処理ロジックと出力バッファは動作を続けていますので、ソフトウェアは次の機能を使用できます。

- 出力バッファに有効なワードが存在する場合には、RNG_DR レジスタから乱数をもう 4 個読み出すことができます。
- 条件付け出力内部レジスタに有効なビットが存在する場合には、RNG_DR レジスタから乱数をもう 4 個読み出すことができます。そうではない場合には、期待される新しいノイズソースビット閾値（NIST モードでは 128 ビット）に達して、条件付けが完了するまでアプリケーションで RNG を再度有効にする必要があります。
期待される条件付けの回数に達した場合のみ（NISTC = 0 の場合は 2 回）、新たな 4 つの乱数ワードを使用できるようになります。全体的な時間については、[セクション 32.5 : 1226 ページの RNG 処理時間](#)を参照してください。

RNG を無効化する場合、すべてのアナログシード発生回路を無効にします。その消費電力は、データシートの電气的特性セクションに示されています。また、RNG クロックによってクロックの供給を受けるロジックはすべてクロック停止にします。この方法では、RNG 初期化時間のために、RNG_DR レジスタでランダムサンプルが利用可能となるまでのレイテンシが増加することに注意してください。

初期化中に（すなわち DRDY ビットが初めて立ち上がるよりも十分前に）RNG ブロックが無効化された場合、RNGEN ビットが 1 にセットされると、初期化シーケンスは停止したところから再開されます（アプリケーションが RNG_CR レジスタの CONDRST ビットを使用して条件付けロジックをリセットしていない限り）。

アプリケーションが RNG クロックを停止する必要がある場合は、RNGEN ビットをクリアしてから RCC を使用して RNG カーネルクロックを停止するまでの間に 2 RNG カーネルクロックサイクル待つことをお勧めします。

また、RNG が非アクティブになる電力モードにアプリケーションが入る必要がある場合は、RNGEN ビットをクリアしてから PWR を使用して低消費電力モードに入るまでに 2 RNG カーネルクロック サイクル待つことをお勧めします。

上記の 2 つのケースでは、RNG アナログ発振器がアクティブなままのときの予期せぬ電力消費を避けるために、アプリケーションは RNG_CR レジスタのビット 13 をセットすることができます。このビットをセットすると、RNGEN ビットがセットされている (RNG がアクティブになっている) 間、わずかな電力消費しか追加されません。

注： RNG が非アクティブ化される (つまり、保留されるか、または使用できない) 電力モードは、PWR セクションを参照してください。

32.4 RNG 割込み

RNG では、割込みは次のイベントによって生成できます。

- データレディフラグ
- シードエラー、を参照 [セクション 32.3.7 : エラー管理](#)
- クロックエラー、を参照 [セクション 32.3.7 : エラー管理](#)

表 296 に示すように、専用割込み有効化制御ビットが使用できます。

表 296. RNG 割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ	有効化制御ビット	割込みのクリア方法
RNG	データレディフラグ	DRDY	IE	なし (自動)
	シードエラーフラグ	SEIS	IE	SEIS に 0 を書き込むか、CONDRST に 1 を書き込んでから 0 を書き込みます。
	クロックエラーフラグ	CEIS	IE	CEIS に 0 を書き込みます。

RNG_CR レジスタのマスクビットまたは全体的な割込み制御ビット IE を変更することにより、上記の割込みソースを個別に有効にしたり無効にしたりすることができます。個別の割込みソースのステータスは RNG_SR レジスタから読み出すことができます。

注： 割込みは、RNG が有効化されている場合にのみ生成されます。

32.5 RNG 処理時間

表 297 で説明されている推奨設定 A では、4 つの 32 ビットデータの 2 セットの間の時間は次のいずれかです。

- $f_{\text{AHB}} < f_{\text{threshold}}$ の場合、 $206 \times N$ AHB サイクル (条件付けステージが制限要因)、または
- $f_{\text{AHB}} \geq f_{\text{threshold}}$ の場合、 $128 \times N$ RNG サイクル (ノイズソースステージが制限要因)

ここで、 $f_{\text{threshold}} = 1.6 \times f_{\text{RNG}}$ 、たとえば、 $f_{\text{RNG}} = 48 \text{ MHz}$ の場合、 77 MHz 。N の値は 2 です。

注： CLKDIV が 0 ではない場合、 f_{RNG} は内部分周比を考慮する必要があります。

設定 B が選択された場合、性能数値は次のようになります。

- $f_{\text{AHB}} < f_{\text{threshold}}$ の場合、206 AHB サイクル、または
- $f_{\text{AHB}} \geq f_{\text{threshold}}$ の場合、32 RNG サイクル

ここで、 $f_{\text{threshold}} = 6.5 \times f_{\text{RNG}}$ 。

32.6 RNG エントロピーソース検証

32.6.1 概要

RNG から使用できるエントロピーの量を評価するために、STMicroelectronics はドイツの BSI AIS-31 統計テスト (T0~T8) および NIST SP800-90B テストスイートを使用してペリフェラルをテストしました。結果は要求に応じて提供でき、顧客側でテストを再現できます。

32.6.2 検証条件

STMicroelectronics は、以下の条件下で RNG 真性乱数発生器をテストしました。

- RNG クロック rng_clk= 48 MHz
- RNG の設定については、[表 297 : RNG の設定](#)で説明しています。設定 A のみ NIST SP800-90B を認証できます。

表 297. RNG の設定

RNG 設定	RNG_CR ビット						ループ数 (N)	RNG_ HTCR レジスタ	RNG_ NCSR register
	NISTC ビット	RNG_ CONFIG1 [5:0]	CLKDIV [3:0]	RNG_ CONFIG2 [2:0]	RNG_ CONFIG3 [3:0]	CED ビット			
A	0	0x0F	0x0 ⁽¹⁾	0x0 ⁽²⁾	0xD	0	2	0x0000AAC7 ⁽³⁾	0x0000 0492
B	1	0x18	0x0	0x0	0x0	0	1	0x0000AAC7	0x0003 FFFF

1. NIST 認証の場合、ノイズソースのサンプリングは 48 MHz 以下でなければなりません。したがって、RNG クロックが 48 MHz と異なる場合は、CLKDIV のこの値を調整する必要があります。詳細については、[セクション 32.7.1](#) の CLKDIV ビットフィールドの説明を参照してください。
2. RNG の電力消費が重要な場合は、0x1 値を推奨します。詳細については、[セクション 32.3.8 : RNG の低消費電力時の取り扱い](#) の最後を参照してください。
3. 繰り返しテストでは 42 に、適応テストでは 711 に対応します。詳細については、[1221ページの健全性チェック](#)を参照してください。

32.6.3 データ収集

統計テストを行うためには、サンプルを元データレベルでのエントロピーソースからだけでなく、エントロピーソースの出力からも収集する必要があります。データ収集と統計テストスイートの実行の詳細については、www.st.com から入手できるアプリケーションノート「STM32 microcontrollers random number generation validation using NIST statistical test suite」(AN4230)を参照してください。

製品で上記サンプルを取得する必要がある場合は、STMicroelectronics までご連絡ください。

32.7 RNG レジスタ

RNG は、制御レジスタ、データレジスタ、ステータスレジスタと連動します。

32.7.1 RNG 制御レジスタ (RNG_CR)

アドレスオフセット : 0x000

リセット値 : 0x0080 0D00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CONFIGLOCK	CONDRST	Res.	Res.	Res.	Res.	RNG_CONFIG1 [5:0]						CLKDIV[3:0]			
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RNG_CONFIG2 [2:0]			NISTC	RNG_CONFIG3 [3:0]				ARDIS	Res.	CED	Res.	IE	RNGEN	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw		

ビット 31 CONFIGLOCK : RNG 設定ロック

0 : RNG_HTCR および RNG_CR 設定ビット [29:4] への書き込みが許可されます。

1 : RNG_HTCR および RNG_CR 設定ビット [29:4] への書き込みは、次の RNG リセットまで無視されます。このビットフィールドは 1 回だけセットできます。このビットがセットされると、RNG がリセットされた場合のみ 0 にリセットできます。

ビット 30 CONDRST : 条件付きソフトリセット

1 を書き込んでから 0 を書き込んで条件付けロジックをリセットし、すべての FIFO をクリアして、RNG_SR がクリアされた状態で新しい RNG 初期化処理を開始します。RNG_CR および RNG_HTCR レジスタは、CONDRST によっては変更されません。

設定ビット [29:4] をセットするのと同じアクセスで、このビットを 1 にセットする必要があります。言い換えると、CONDRST ビットが 1 にセットされるときには、ビット [29:4] にも正しい設定を書き込む必要があります。

CONDRST がソフトウェアによって 0 にセットされると、その値はリセット処理が完了したときに 0 になります。これには約 2 AHB クロックサイクルと 2 RNG クロックサイクルを必要とします。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:20 RNG_CONFIG1[5:0] : RNG 設定 1

RNG 設定に予約済みです (ビットフィールド 1)。セクション 32.6 : RNG エントロピーソース検証に記載された推奨される値を使用して初期化する必要があります。

RNG_CONFIG1 のあらゆるビットへの書き込みは、CONFIGLOCK が 0 になっている間に、同じアクセスで CONDRST ビットが 1 にセットされる場合のみ考慮されます。CONFIGLOCK = 1 の場合、このビットへの書き込みは無視されます。

ビット 19:16 CLKDIV[3:0] : クロック分周回路ファクタ

この値は、受信 RNG クロックで動作するプログラム可能な内部分周回路 (1~16) の設定に使用されます。これらのビットに書き込めるのは、コアが無効 (RNGEN = 0) のときだけです。

0x0 : 分周回路以降の内部 RNG クロックは、受信 RNG クロックと類似しています。

0x1 : 内部 RNG クロックごとに 2 RNG クロックサイクルです。

0x2 : 内部 RNG クロックあたり 2^2 (= 4) RNG サイクルです。

.....

0xF : 内部クロックあたり 2^{15} RNG クロックサイクルです (たとえば、受信した 48MHz RNG クロックは 1.5 kHz 内部 RNG クロックになります)

これらのビットへの書き込みは、CONFIGLOCK が 0 になっている間に、同じアクセスで CONDRST ビットが 1 にセットされる場合のみ考慮されます。CONFIGLOCK = 1 の場合、このビットへの書き込みは無視されます。

ビット 15:13 RNG_CONFIG2[2:0] : RNG 設定 2

RNG 設定に予約済みです (ビットフィールド 2)。RNG の電力消費が重要な場合は、ビット 13 をセットすることができます。[セクション 32.3.8 : RNG の低消費電力時の取り扱い](#)を参照してください。詳細は、RNG_CONFIG1 ビットフィールドの説明を参照してください。

ビット 12 NISTC : NIST カスタム

0 : NIST 準拠 RNG のハードウェアデフォルト値この設定では、128 ビット出力ごとの 2 回の条件付けループが実行され、256 ビットのノイズソースが使用されます。

1 : NIST 準拠 RNG のカスタム値提案される設定については、[セクション 32.6 : RNG エントロピーソース検証](#)を参照してください。

このビットへの書込みは、CONFIGLOCK が 0 のままで同じアクセスで CONDRST ビットが 1 にセットされる場合のみ考慮されます。CONFIGLOCK = 1 の場合、このビットへの書込みは無視されます。

ビット 11:8 RNG_CONFIG3[3:0] : RNG 設定 3

RNG 設定に予約済みです (ビットフィールド 3)。詳細は、RNG_CONFIG1 ビットフィールドの説明を参照してください。

このレジスタの NISTC ビットがクリアされている場合、RNG_CONFIG3 のビットフィールドの値は RNG によって無視されます。

ビット 7 ARDIS : オートリセットは無効です。

0 : ノイズソースエラーが発生すると、RNG はオートリセットを実行して、SECS ビットをクリアします。

1 : ノイズソースエラーが発生したとき、乱数生成を再開するには、アプリケーションは CONDRST に 1 を書き込んでから 0 を書き込むことによって、RNG をリセットする必要があります。

オートリセットが有効なとき、アプリケーションは、やはり、ノイズソースエラー後に SEIS ビットをクリアする必要があります。

このビットへの書込みは、CONFIGLOCK が 0 のままで同じアクセスで CONDRST ビットが 1 にセットされる場合のみ考慮されます。CONFIGLOCK = 1 の場合、このビットへの書込みは無視されます。

ビット 6 予約済みであり、リセット値に保持する必要があります。**ビット 5 CED : クロックエラー検出**

0 : クロックエラー検出は有効です。

1 : クロックエラー検出は無効です。

RNG が有効になっている場合、動作中にクロックエラー検出を有効化/無効化できません。CED を有効化/無効化するには、RNG を無効化する必要があります。

このビットへの書込みは、CONFIGLOCK が 0 になっている間に、同じアクセスで CONDRST ビットが 1 にセットされる場合のみ考慮されます。CONFIGLOCK = 1 の場合、このビットへの書込みは無視されます。

ビット 4 予約済みであり、リセット値に保持する必要があります。**ビット 3 IE : 割込みイネーブル**

0 : RNG 割込みは無効です。

1 : RNG 割込みは有効です。RNG_SR レジスタで DRDY = 1、SEIS = 1 または CEIS = 1 になると、割込みは直ちにペンディングとなります。

ビット 2 RNGEN : 真性乱数発生器イネーブル

0 : 真性乱数発生器は無効です。アナログノイズソースの電力がオフになり、RNG クロックによって供給されるロジックがゲートされます。

1 : 真性乱数発生器は有効です。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

32.7.2 RNG ステータスレジスタ (RNG_SR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SEIS** : シードエラー割込みステータス

このビットは、SECS として同じ時間でセットされます。0 を書き込むことによってクリアされます (CONDRST が使用されていない場合)。1 を書き込んでも、ビットの値は変化しません。

0 : 障害シーケンスは検出されませんでした。

1 : 1 つ以上の障害シーケンスが検出されました。詳細については、**SECS** ビットの説明を参照してください。

RNG_CR レジスタで IE = 1 である場合、割込みが保留となります。

ビット 5 **CEIS** : クロックエラー割込みステータス

このビットは、CECS として同じ時間でセットされます。0 を書き込むことによってクリアされます。1 を書き込んでも、ビットの値は変化しません。

0 : RNG クロックは正常です ($f_{\text{RNGCLK}} > f_{\text{HCLK}}/32$)。

1 : 内部分周回路より前の RNG クロックが遅すぎると検出されました ($f_{\text{RNGCLK}} < f_{\text{HCLK}}/32$)。

RNG_CR レジスタで IE = 1 である場合、割込みが保留となります。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **SECS** : シードエラーの現在のステータス

0 : 障害のあるシーケンスは現在のところ検出されていません。SEIS ビットがセットされている場合、これは障害のあるシーケンスが検出されたが、回復していることを意味します。

1 : 次の障害シーケンスのうち 1 つ以上が検出されました。

- ランタイム時の繰り返し回数テストに失敗しました (ノイズソースが 24 ビット以上連続して一定値 0 または 1 を提供したか、32 回以上連続して 2 ビットパターン 01 または 10 を提供しました)。
- ノイズソースで起動時または継続の適応比率テストに失敗しました。
- 起動の後処理/条件付け正常性チェックに失敗しました。

ビット 1 **CECS** : クロックエラーの現在のステータス

0 : RNG クロックは正常です ($f_{\text{RNGCLK}} > f_{\text{HCLK}}/32$)。CEIS ビットがセットされている場合、これは遅いクロックが検出されたが、回復していることを意味します。

1 : RNG クロックは遅すぎます ($f_{\text{RNGCLK}} < f_{\text{HCLK}}/32$)。

注 : **CECS** ビットは、RNG_CR レジスタの CED ビットが 0 にセットされたときにのみ有効です。

ビット 0 **DRDY** : データレディ

0 : RNG_DR レジスタがまだ有効でなく、乱数データは利用できません。

1 : RNG_DR レジスタに有効な乱数データが入っています。

出力バッファが空になると (RNG_DR レジスタの読出し後)、新しい乱数値が生成されるまで、このビットは 0 に戻ります。

注 : ペリフェラルが無効化されている場合 (RNG_CR レジスタの RNGEN = 0) に、DRDY ビットが立ち上がることがあります。

RNG_CR レジスタで IE = 1 の場合、DRDY = 1 のときに割込みが生成されます。

32.7.3 RNG データレジスタ (RNG_DR)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

RNG_DR レジスタは読み出し専用レジスタであり、読み出されると 32 ビットの乱数値を返します。このレジスタの内容は、RNGEN = 0 の場合でも、DRDY = 1 で、値が 0x0 でないときに有効です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RNDATA[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RNDATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RNDATA[31:0]** : 乱数データ

DRDY = 1 のときに有効な 32 ビット乱数データです。DRDY = 0 のとき、RNDATA の値は 0 です。

DRDY がセットされているとき、RNG_DR がゼロではないことを常に確認することをお勧めします。ゼロの場合、RNG_SR ポーリングと RNG_DR 出力読み出しの間にシードエラーが発生した（まれなイベント）ことを意味するからです。

32.7.4 RNG 健全性テスト制御レジスタ (RNG_HTCR)

アドレスオフセット : 0x010

リセット値 : 0x0000 : 72AC

RNG_HTCR への書き込みは、RNG_CR で CONDRST ビットがセットされ、CONFIGLOCK ビットがクリアされている場合のみ考慮されます。CONFIGLOCK=1 の場合、このレジスタへの書き込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HTCFG[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTCFG[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **HTCFG[31:0]** : 健全性テストの設定

この設定は健全性テストを設定するために RNG によって使用されます。推奨値については [セクション 32.6 : RNG エントロピーソース検証](#) を参照してください。

注： 推奨値と異なる値が書き込まれた場合、このレジスタの読み出しを含む RNG の動作は保証されません。

32.7.5 RNG レジスタマップ

表 298. RNG レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	RNG_CR	CONFIGLOCK	CONDRST	Res.	Res.	Res.	Res.	RNG_CONFIG1 [5:0]						CLKDIV [3:0]				RNG_CONFIG2 [2:0]			NISTC	RNG_CONFIG3 [3:0]			ARDIS	Res.	CED	Res.	IE	RNGEN	Res.	Res.	
	リセット値	0	0					0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0		0		0		
0x004	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY	
	リセット値																									0	0			0	0	0	0
0x008	RNG_DR	RNDATA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	RNG_HTCR	HTCFG[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

33 AES ハードウェアアクセラレータ (AES)

33.1 概要

AES ハードウェアアクセラレータ (AES) は、NIST によって規定されている高度暗号化標準 (AES : Advanced Encryption Standard) に準拠してデータの暗号化や復号を行います。

AES は、キー長が 128 または 256 ビットの ECB、CBC、CTR、GCM、GMAC、CCM 連鎖 (チェーン) モードをサポートしています。AES は、SAES の制御の下で SAES ペリフェラルに格納されたキーをハードウェアによってロードできる機能を備えています。

このペリフェラルは、入力および出力データの DMA シングル転送をサポートしています (2 つの DMA チャンネルが必要です)。

33.2 AES の主な特徴

- NIST FIPS 公報 197『高度暗号化標準 (Advanced encryption standard (AES))』(2001 年 11 月) に準拠
- 次の複数の連鎖モードで暗号化と復号 :
 - 電子コードブック (ECB) モード
 - 暗号ブロック連鎖 (CBC) モード
 - カウンタ (CTR) モード
 - ガロアカウンタモード (GCM)
 - ガロアメッセージ認証コード (GMAC) モード
 - CBC-MAC 付きカウンタ (CCM) モード
- 128 ビットおよび 256 ビットの暗号キー長に対応した、128 ビットデータブロックの処理
 - 128 ビットまたは 256 ビットキーを使用して 1 個の 128 ビットブロックを処理する場合は ECB モードでそれぞれ 51 または 75 クロックサイクルの遅延
- 専用のキーバスを使用し、SAES によって制御されるサイドチャンネル耐性のある SAES ペリフェラルとのオプションのキー共有 (共有キーモード)
- ECB/CBC 復号の最終ラウンドキーを計算するための統合キースケジューラ
- 暗号化キーを格納するための 256 ビットの書込み専用レジスタ (32 ビットレジスタ 8 個)
- 初期化ベクタ (32 ビットレジスタ 4 個) を格納するための 128 ビットのレジスタ
- データ入力および出力用の 32 ビットバッファ
- 2 つのチャンネル (1 チャンネルは入力データ用、1 チャンネルは処理済みデータ用) を使用したダイレクトメモリアクセス (DMA) のサポートによる自動データフロー制御。シングル転送のみサポートされます。
- 1、8、16、32 ビットデータをサポートするデータスワッピングロジック
- 32 ビットワードのシングルアクセスでのみアクセス可能な AMBA AHB スレーブペリフェラル。他のアクセスタイプは AHB エラーとなります。32 ビット書込み以外では、レジスタの内容が破損する可能性があります。
- AES がより優先順位の高い別メッセージを処理する必要がある場合は、ソフトウェアでメッセージをサスペンドしてから元のメッセージのレジュームが可能

33.3 AES の実装

デバイスには、次の表に従って実装された 1 つの AES ペリフェラルがあります。そこでは、SAES ペリフェラルによって生成されたキーが使用できます。比較のため、SAES ペリフェラルも表に含めています。

表 299. AES および SAES の機能

モードまたは機能 ⁽¹⁾	AES	SAES
ECB、CBC 連鎖（チェーン動作）	X	X
CTR、CCM、GCM 連鎖	X	X
AES 128 ビット ECB 暗号化（サイクル単位）	51	480
DHUK および BHK のキーの選択	-	X
サイドチャネル攻撃への耐性	-	X
SAES と AES 間での共有キー	X	
キー長（ビット単位）	128, 256	128, 256

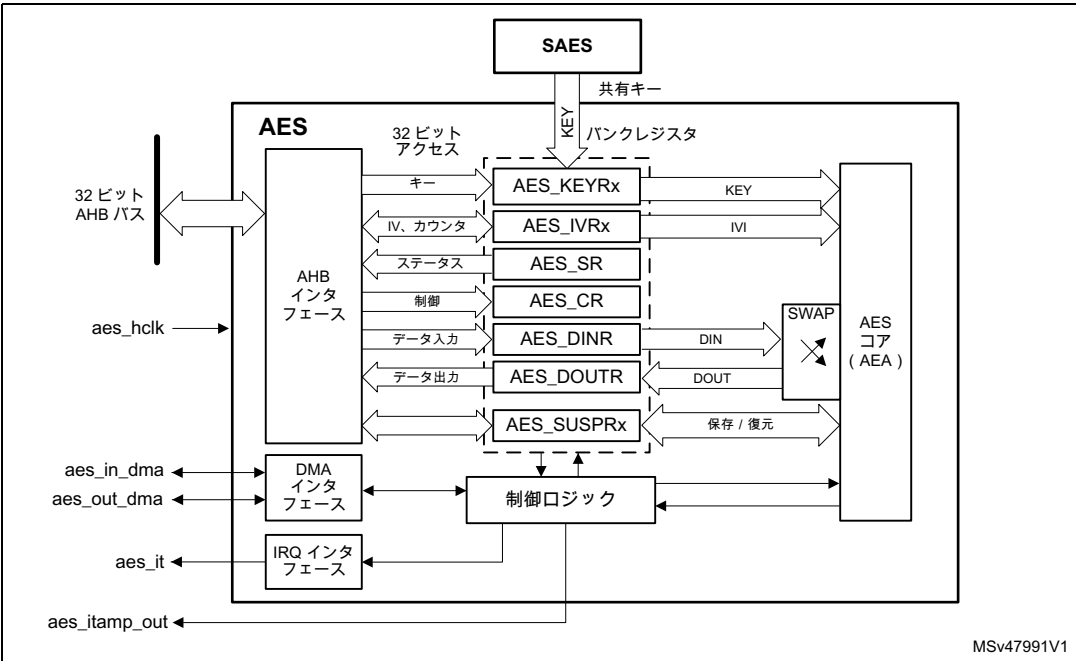
1. X：サポートされています。

33.4 AES機能詳細

33.4.1 AES ブロック図

図 291 に AES のブロック図を示します。

図 291. AES ブロック図



33.4.2 AES 内部信号

表 300 には、AES ペリフェラルにインタフェース接続するユーザ関連内部信号を示しています。

表 300. AES 内部入力／出力信号

信号名	信号タイプ	説明
aes_hclk	入力	AHB バスクロック
aes_it	出力	AES 割込みリクエスト
aes_in_dma	入力／出力	AES 入力データ DMA シングルリクエスト／確認応答
aes_out_dma	入力／出力	AES 処理済みデータ DMA シングルリクエスト／確認応答
aes_itamp_out	出力	予期しないハードウェアフォールトが発生したときにトリガされる、TAMP へのタンパイイベント信号 (XOR されます)。この信号がトリガされると、AES はキーレジスタを自動的にクリアします。AES を再び使用可能にするには、リセットが必要です。

33.4.3 AES のリセットおよびクロック

AES ペリフェラルは、AHB バスクロックによってクロック供給されます。

AES には RCC 内に専用のリセットビットがあります。

33.4.4 AES 対称暗号の実装

AES ハードウェアアクセラレータ (AES) は、高度暗号化標準 (AES) を使用して 16 バイトのデータブロックを暗号化または復号する 32 ビット AHB ペリフェラルです。また、表 301 に要約されている承認済みの AES 対称キーセキュリティ機能のセットも実装されています。これらの機能は NIST PUB 140-3 の認定を受けることができます。

表 301. AES の承認済みの対称キー機能

動作	アルゴリズム	仕様	キービット長	連鎖モード
暗号化、復号	AES	FIPS PUB 197 NIST SP800-38A	128, 256	ECB、CBC、CTR
認証済みの暗号化または復号		NIST SP800-38C NIST SP800-38D		GCM、CCM
暗号ベースのメッセージ認証コード		NIST SP800-38D		GMAC

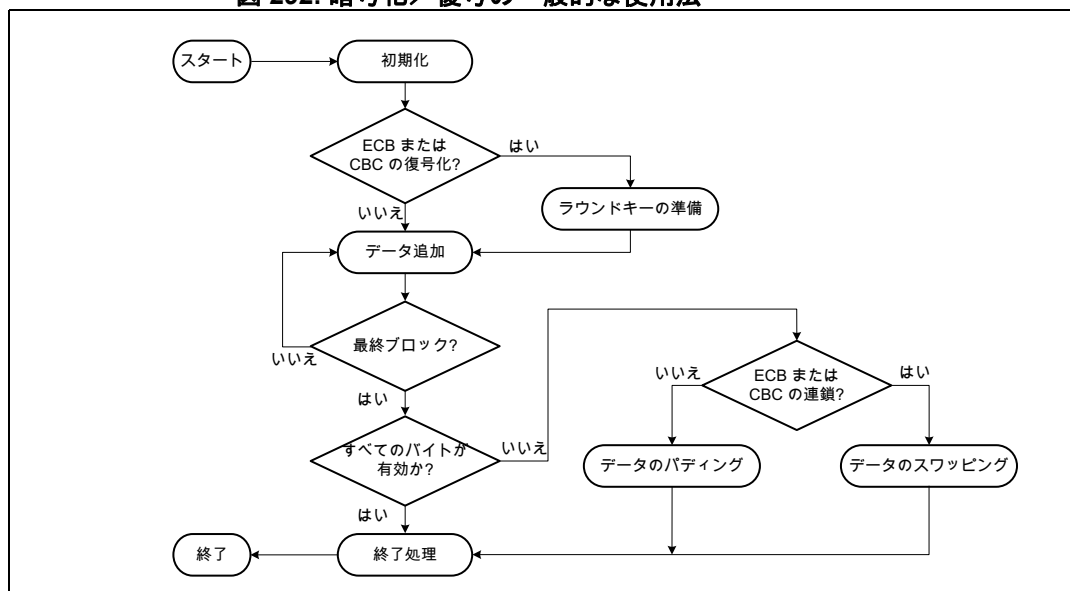
AES は、CPU によって直接使用することも、2 つの DMA チャンネル (1 つは平文用、もう 1 つは暗号文用) を使用して間接的に使用することもできます。

セクション 33.4.8 で説明しているシーケンスに従って、任意の AES 処理を一時停止し、その後、再開することができます。

33.4.5 AES の暗号化または復号の一般的な使用法

暗号化または復号の一般的な動作を以下の図に示します。

図 292. 暗号化／復号の一般的な使用法



初期化

AES ペリフェラルを連鎖モードに従って初期化します。詳細については、[セクション 33.4.9 : AES 基本連鎖モード \(ECB、CBC\)](#) および [セクション 33.4.10 : AES カウンタ \(CTR\) モード](#) を参照してください。

データ追加

本セクションでは処理用のデータを追加するさまざまな方法を説明します。ECB または CBC 連鎖モードでは、処理するデータのサイズが 16 バイトの倍数ではない場合、[セクション 33.4.7 : AES の暗号文借用およびデータパディング](#)を参照してください。これらの場合の最終ブロックの管理は、このセクションで説明しているものより複雑になります。

CPU をポーリングモードで使したデータの追加

この方法では、フラグポーリングを使用して次のシーケンスでデータ追加を制御します。

1. KEYVALID がセットされているとき、AES_CR レジスタの EN ビットをセットして、AES ペリフェラルを有効にします（まだ有効化されていない場合）。
2. ペイロードが完全に処理されるまで次のサブシーケンスを繰り返します。
 - a) AES_DINR レジスタに 4 つの入力データワードを書き込みます。
 - b) ステータスフラグ CCF が AES_ISR レジスタでセットされるまで待つ、AES_DOUTR レジスタから 4 つのデータワードを読み出します。
 - c) AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
 - d) 次の処理ブロックが最後のブロックである場合は、（該当する場合）データを 0 でパディングして完全なブロックを取得し、GCM ペイロード暗号化または CCM ペイロード復号の場合は、無効なバイト数を (NPBLB[3:0] を使用して) 指定します（そうしないと、誤ったタグ計算になります）。

3. 処理したばかりのデータブロックはメッセージの最終ブロックであるため、必要に応じてメッセージ/ペイロードの一部ではないデータを破棄してから、EN をクリアして AES ペリフェラルを無効にします。

注： キーを AES プロセッサに送信できるようにするため、AES_DINR レジスタへの連続する 2 つの書き込みの間に最大 3 つのウェイトサイクルが自動的に挿入されます。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

CPU を割り込みモードで使したデータの追加

この方法は、AES ペリフェラルからの割り込みを使用して次のシーケンスでデータ追加を制御します。

1. AES_IER レジスタの CCFIE ビットをセットして、AES からの割り込みを有効にします。
2. KEYVALID がセットされているとき、EN をセットして、AES ペリフェラルを有効にします（まだ有効化されていない場合）。
3. まず、AES_DINR レジスタに 4 つの入力データワードを書き込みます。
4. AES 割り込みサービスルーチンでデータを処理します。割り込みごとに次の手順に従います。
 - a) AES_DOUTR レジスタから 4 つの出力データワードを読み出します。
 - b) AES_ICR レジスタの CCF ビットをセットして、CCF フラグ、保留中の割り込みをクリアします。
 - c) 次の処理ブロックがメッセージの最後のブロックである場合は、（該当する場合）データを 0 でパディングして完全なブロックを取得し、GCM ペイロード暗号化または CCM ペイロード復号の場合は、無効なバイト数を（NPBLB[3:0] を通して）指定します（そうしないと、誤ったタグ計算になります）。次に、4e) に進みます。
 - d) 処理したばかりのデータブロックがメッセージの最終ブロックだった場合、必要に応じてメッセージ/ペイロードの一部ではないデータを破棄してから、EN をクリアして AES ペリフェラルを無効にし、割り込みサービスルーチンを中止します。
 - e) AES_DINR レジスタに次の 4 つの入力データワードを書き込み、割り込みサービスルーチンを中止します。

注： AES は連続した読み出し操作または書き込み操作の間の遅延を許容するため、2 つの AES 計算の間に別のペリフェラルからの割り込みがある場合などに受け入れることができます。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

DMA を使したデータの追加

この方法では、すべての転送と処理が DMA および AES によって管理されます。次のように実行します：

1. 処理するメッセージの最後のブロックが 16 バイトより短い場合は、ブロックの残りの部分に 0 をパディングして、最後の 4 ワードのデータブロックを準備します。
2. 処理するデータをメモリから AES ペリフェラル入力へ、処理したデータを AES ペリフェラル出力からメモリに転送するよう DMA コントローラを設定します（[セクション 33.6 : AES DMA リクエスト](#) に記載）。転送完了時に割り込みを生成するよう DMA コントローラを設定します。GCM のペイロード暗号化または CCM のペイロード復号では、最後の 4 ワードブロックを 0 でパディングしている場合は、DMA 転送にそのブロックを含めないでください。その代わりに、[CPU をポーリングモードで使したデータの追加](#) に示したシーケンスをこの最終ブロックに使用する必要があります。これは AES が正しいタグを計算できるように、NPBLB[3:0] ビットフィールドをセットアップしてからブロックを処理する必要があるためです。
3. KEYVALID がセットされているとき、EN をセットして、AES ペリフェラルを有効にします（まだ有効化されていない場合）。
4. DMAINEN と DMAOUTEN をセットして、DMA リクエストを有効にします。

5. 転送完了を示す DMA 割込みが発生したら、メモリから AES 処理済みデータを取得します。

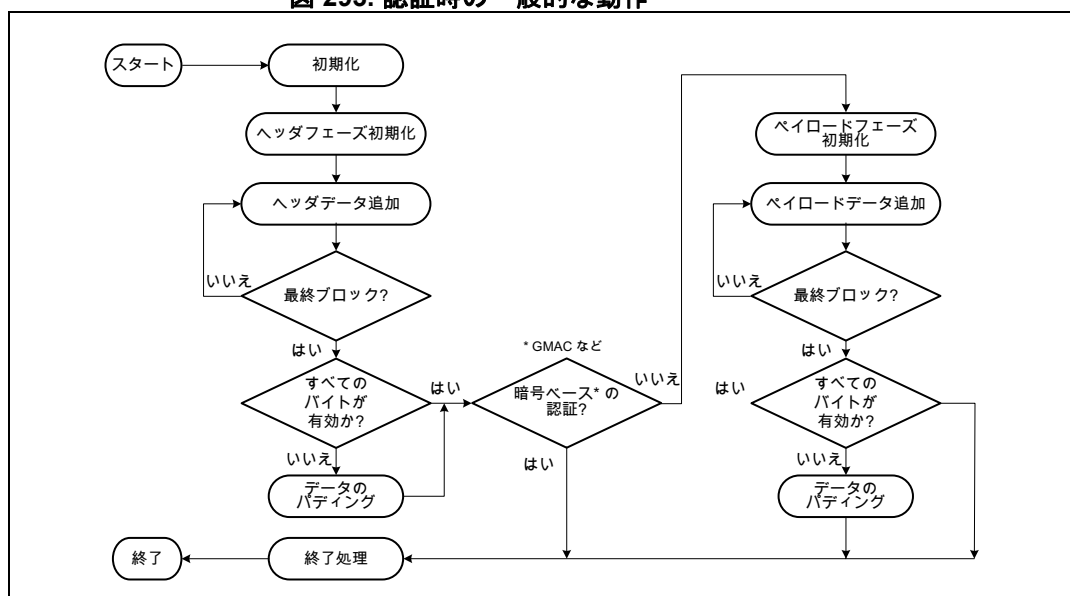
注： AES_DOUTR レジスタの読出しは計算フェーズの最後にソフトウェア操作なしで自動的に DMA によって管理されるため、この方法では CCF フラグを使用しません。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

33.4.6 AES の認証済み暗号化、復号、および暗号ベースのメッセージ認証

認証済み暗号化または復号、および暗号ベースのメッセージ認証の一般的な動作を以下の図に示します。

図 293. 認証時の一般的な動作



セクション 33.4.11 : AES ガロア/カウンタモード (GCM) および セクション 33.4.13 : AESCBC-MAC 付きカウンタ (CCM) で、AES がサポートしているシーケンスの詳細を説明しています。

図に示したように、暗号ベースのメッセージ認証フローではペイロードフェーズが省略されます。AES でサポートされているシーケンスの詳細は、セクション 33.4.12 : AES ガロアメッセージ認証コード (GMAC : Galois message authentication code) を参照してください。

33.4.7 AES の暗号文借用およびデータパディング

サイズがブロックサイズ (16 バイト) の倍数ではないメッセージを管理するために ECB または CBC モードで AES を使用するときには、アプリケーションでは NIST 特別公報 800-38A 『ブロック暗号の推奨動作モード : CBC モード用暗号文借用の 3 つの変種』 (Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation) に記載されているもののような、暗号文借用技術を使用します。AES はこのような技術を実装していないため、アプリケーションは最後から 2 番目のブロックのデータを使用して、入力データの最終ブロックを完成させる必要があります。

注： 暗号文借用技術は、この参考マニュアルには記載されていません。

同様に、ECB または CBC 以外のモードでは、不完全な入力データブロック (16 バイトより短い入力データを持つブロック) を、暗号化の前に 0 でパディングする必要があります。つまり、データストリングの後端に余分なビットを付加する必要があります。復号後は、余分なビットを破棄する必要があります。AES では、最終ブロックに対する自動データパディング操作が行われないため、アプリ

ケーションが本書に示された推奨に従って、サイズが 16 ビットの倍数ではないメッセージを管理する必要があります。

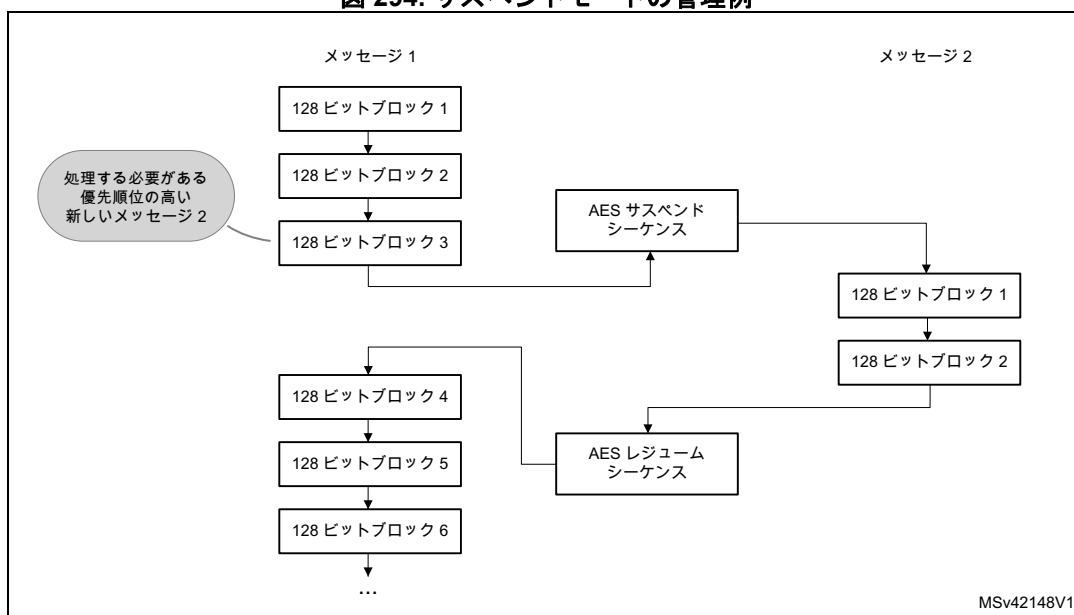
33.4.8 AES のサスペンドおよびレジューム操作

優先順位の高い別のメッセージを処理するには、メッセージをサスペンドすることが可能です。優先順位の高いメッセージを送信した後、サスペンドしたメッセージを再開できます。これは、暗号化と復号モードの両方に適用されます。

サスペンドおよびレジューム操作を行っても連鎖操作は途切れません。AES が再び有効になるとすぐにメッセージ処理を再開して、次のデータブロックを受信できます。

図 294 に、サスペンドおよびレジューム操作の例を示します。メッセージ 1 は、より短く優先順位の高いメッセージ 2 を送信するためにサスペンドされています。

図 294. サスペンドモードの管理例



サスペンドおよびレジューム操作の詳細は、各連鎖モード専用のセクションに説明があります。

33.4.9 AES 基本連鎖モード (ECB、CBC)

ECB は最も単純な動作モードです。連鎖操作も、特別な初期化ステージ也没有ありません。メッセージはブロックに分割され、各ブロックが個別に暗号化または復号されます。ECB で復号するときは、最初のブロックを処理する前に特別なキースケジューリングが必要となります。

図 295 と 図 296 に、電子コードブック (ECB : electronic codebook) の暗号化と復号のそれぞれの連鎖の実行手順を示します。ECB 連鎖モードを選択するには、CHMOD[2:0] に 0x0 を書き込みます。

図 295. ECB 暗号化

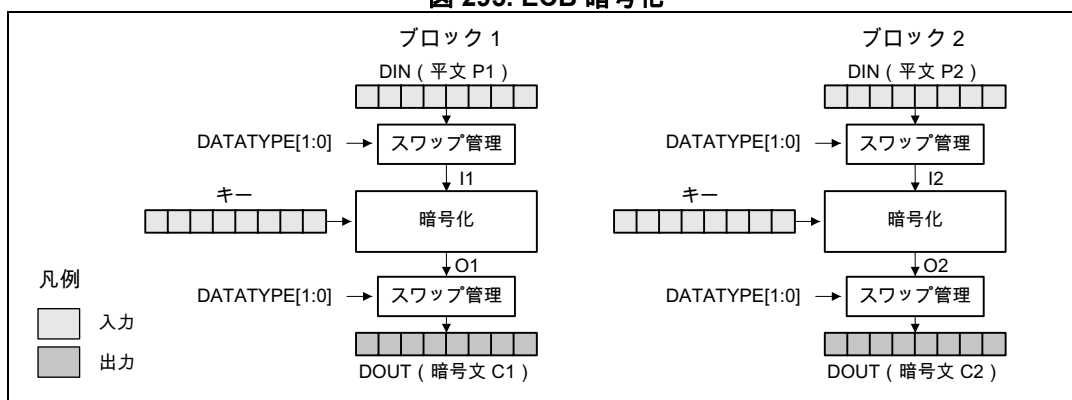
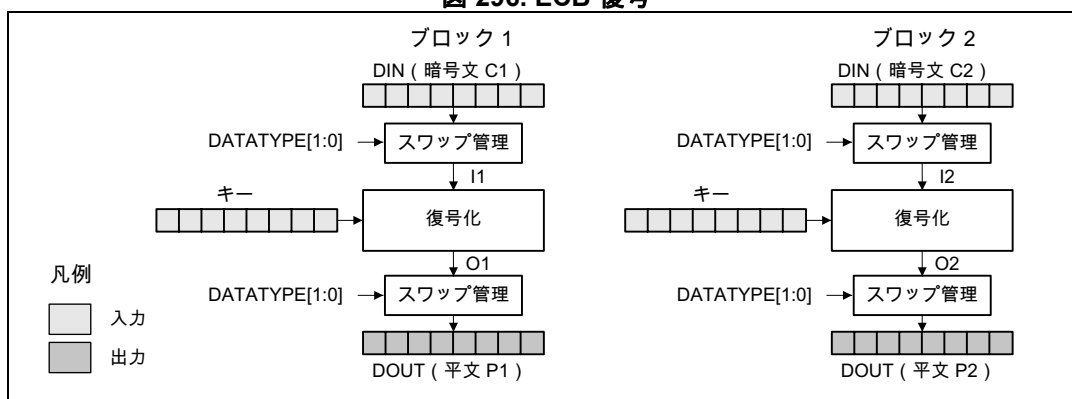


図 296. ECB 復号



CBC 暗号化モードでは、各ブロックの出力が次のブロックの入力で連鎖されます。各メッセージを一意にするために、最初のブロック処理時に初期化ベクタが使用されます。CBC で復号するときには、最初のブロックを処理する前に特別なキースケジューリングが必要となります。

図 297 と 図 298 に、暗号ブロック連鎖 (CBC) の暗号化と復号のそれぞれの連鎖の実行手順を示します。この連鎖モードを選択するには、CHMOD[2:0] に 0x1 を書き込みます。

図 297. CBC 暗号化

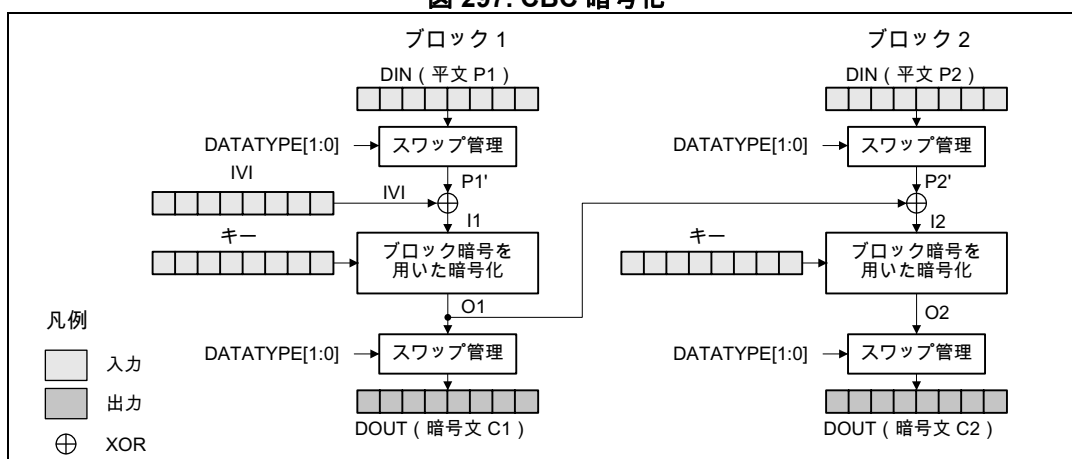
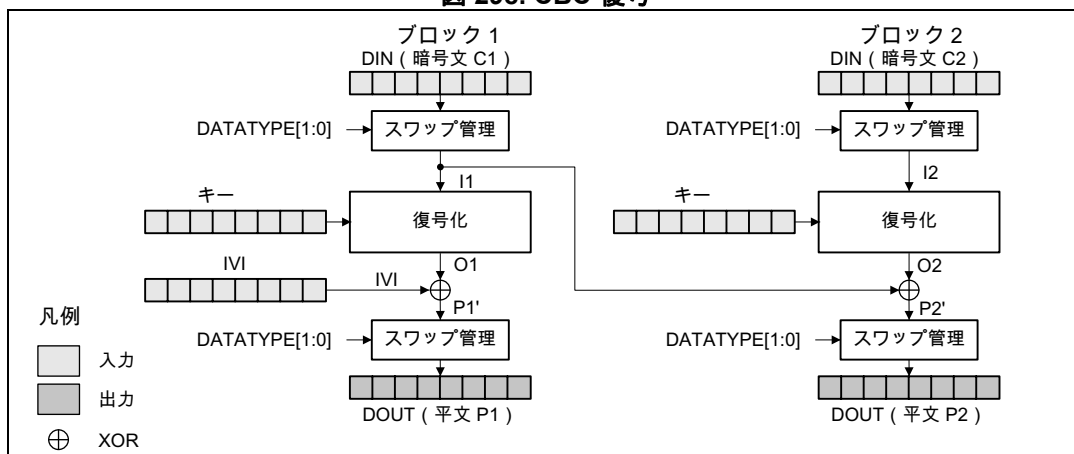


図 298. CBC 復号



詳細については、NIST 特別公報 800-38A『ブロック暗号の推奨動作モード』（（Special Publication 800-38C, NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation））を参照してください。

ECB および CBC 暗号化プロセス

このプロセスは [セクション 33.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして AES ペリフェラルを無効にします。
2. 次のように AES_CR レジスタを初期化します。
 - 暗号化モード (MODE[1:0] に 0x0 を書き込む) で、ECB または CBC 連鎖モードを選択します (CHMOD[2:0] に 0x0 または 0x1 を書き込む)。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。
 - KMOD[1:0] を使ってキーモードを選択します。SAES ペリフェラルからキーを取得する場合は、KMOD[1:0] に 0x2 を書き込み、そうでない場合は 0x0 のままにします。
3. 前の手順で CBC モードが選択された場合は、AES_IVRx レジスタに初期化ベクタを書き込みます。
4. KMOD[1:0] が 0x0 の場合は、AES_KEYRx レジスタにキーを書き込みます。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます ([セクション 33.4.14](#) を参照)。
5. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
6. EN を設定して AES ペリフェラルを有効にします。
7. 平文データを追加します。
 - a) それが最後から 2 番目または最後のブロックであり、メッセージの平文テキストのサイズが 16 バイトの倍数ではない場合は、[セクション 33.4.7](#) のガイダンスに従ってください。
 - b) [セクション 33.4.5](#) の説明に従って平文ブロックを AES に追加し、AES_DOUTR レジスタを 4 回読み取って暗号文ブロックを保存します。
 - c) 最後から 3 番目の平文ブロックが暗号化されるまで手順 b) を繰り返します。最後の 2 つのブロックについて、手順 a) と b) を実施します。
8. シーケンスを終了します: EN をクリアして AES ペリフェラルを無効にします。

ECB/CBC 復号プロセス

このプロセスは [セクション 33.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして AES ペリフェラルを無効にします。
2. 次のように AES_CR レジスタを初期化します。
 - キー導出モードを選択します (MODE[1:0] に 0x1 を書き込みます)。この操作中は、CHMOD[2:0] ビットフィールドは意味がありません。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。
 - KMOD[1:0] を使ってキーモードを選択します。SAES ペリフェラルからキーを取得する場合は、KMOD[1:0] に 0x2 を書き込み、そうでない場合は 0x0 のままにします。
3. KMOD[1:0] が 0x0 の場合は、AES_KEYRx レジスタにキーを書き込みます。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます ([セクション 33.4.14](#) を参照)。
4. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
5. EN を設定して AES ペリフェラルを有効にします。ペリフェラルは、ただちにキー準備用の AES ラウンドを開始します。
6. AES_ISR レジスタの CCF フラグがセットされるまで待ちます。
7. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。復号キーは AES コアで使用可能となり、AES は自動的に無効化されます。
8. 復号モード (MODE[1:0] に 0x2 を書き込む) で、ECB または CBC 連鎖モードを選択します (CHMOD[2:0] に 0x0 または 0x1 を書き込む)。その他のパラメータは変更しないでください。
9. 前の手順で CBC モードが選択された場合は、AES_IVRx レジスタに初期化ベクタを書き込みます。
10. EN を設定して AES ペリフェラルを有効にします。
11. 暗号化データを追加します。
 - a) それが最後から 2 番目または最後のブロックであり、メッセージの暗号文のサイズが 16 バイトの倍数ではない場合は、[セクション 33.4.7](#) のガイダンスに従ってください。
 - b) [セクション 33.4.5](#) の説明に従って暗号文ブロックを AES に追加し、AES_DOUTR レジスタを 4 回読み取って平文ブロックを保存します (MSB が先)。
 - c) 最後から 3 番目の暗号文ブロックが復号されるまで手順 [b](#)) を繰り返します。最後の 2 つのブロックについて、手順 [a](#)) と [b](#)) を実施します。
12. シーケンスを終了します: EN をクリアして AES ペリフェラルを無効にします。

ECB/CBC モードでのサスペンド/レジューム動作

メッセージの処理をサスペンドするには、次の手順に従います。

1. DMA が使用されている場合には、AES_CR レジスタの DMAINEN ビットをクリアして、入力 FIFO への AES DMA 転送を停止します。
2. DMA が使用されていない場合は、AES_DOUTR レジスタを 4 回読み出して、最後に処理されたブロックを保存します。DMA が使用されている場合には、AES_ISR レジスタで CCF フラグがセットされるまで待ち、AES_CR レジスタの DMAOUTEN ビットをクリアして、出力 FIFO からの DMA 転送を停止します。
3. DMA が使用されていない場合は、AES_ISR レジスタの CCF フラグがセットされる (計算が完了する) まで待ちます。
4. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
5. 初期化ベクタレジスタを保存します (データ処理中に AES_IVRx レジスタが変更される CBC モードでのみ必要です)。

6. EN をクリアして AES ペリフェラルを無効にします。
7. AES_CR レジスタを保存して、キーレジスタが優先順位の高いメッセージを処理するのに必要でなければ、それをクリアします。
8. DMA が使用されている場合には、DMA コントローラのステータス (AES の入力および出力データ転送のポインタ、残りバイト数など) を保存します。

メッセージの処理をレジュームするには、次の手順に従います。

1. DMA が使用されている場合には、DMA コントローラを設定して残りの入力 FIFO と出力 FIFO の転送を完了させます。
2. EN をクリアして AES ペリフェラルを無効にします。
3. AES_CR レジスタを復元してから (正しい KEYSIZE で)、AES_KEYRx レジスタを復元します。KMOD[1:0] が 0x2 の場合は、キーを再び SAES ペリフェラルから転送する必要があります ([セクション 33.4.14](#) を参照)。
4. [ECB/CBC 復号プロセス](#) に示したように復号キーを準備します (ECB または CBC 復号のみ必要)。
5. 保存した設定を使用して AES_IVRx レジスタを復元します (CBC モードのみ必要)。
6. EN を設定して AES ペリフェラルを有効にします。
7. DMA が使用されている場合には、DMAINEN と DMAOUTEN をセットして、AES DMA 転送を有効にします。

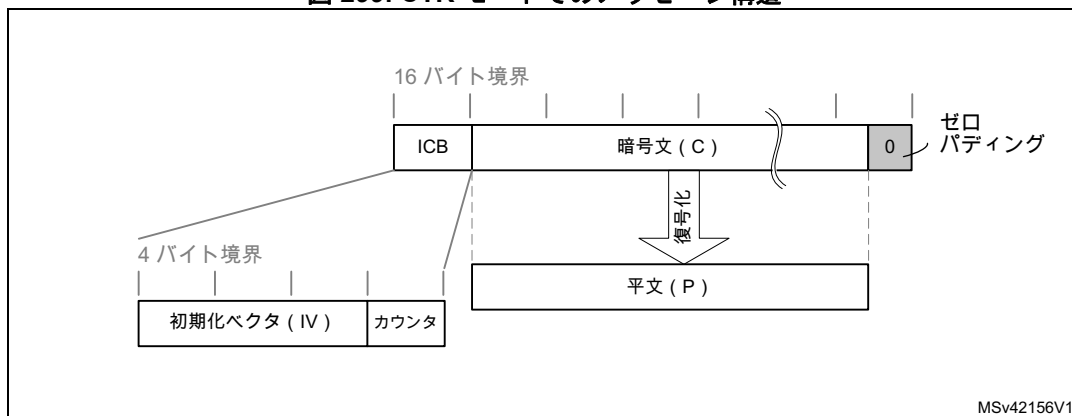
注： アプリケーションには元のキーがわかっているため、キーレジスタを保存する必要はありません。

33.4.10 AES カウンタ (CTR) モード

CTR モードでは、AES コアを使用してキーストリームを生成します。キーは、その後平文との排他的論理和をとって暗号化されます。AES コアは暗号化モードで必ず使用されるため、ECB モードや CBC モードとは異なり、CTR 復号にはキースケジューリングは必要ありません。

CTR モードでの典型的なメッセージ構造を [図 299](#) に示します。

図 299. CTR モードでのメッセージ構造



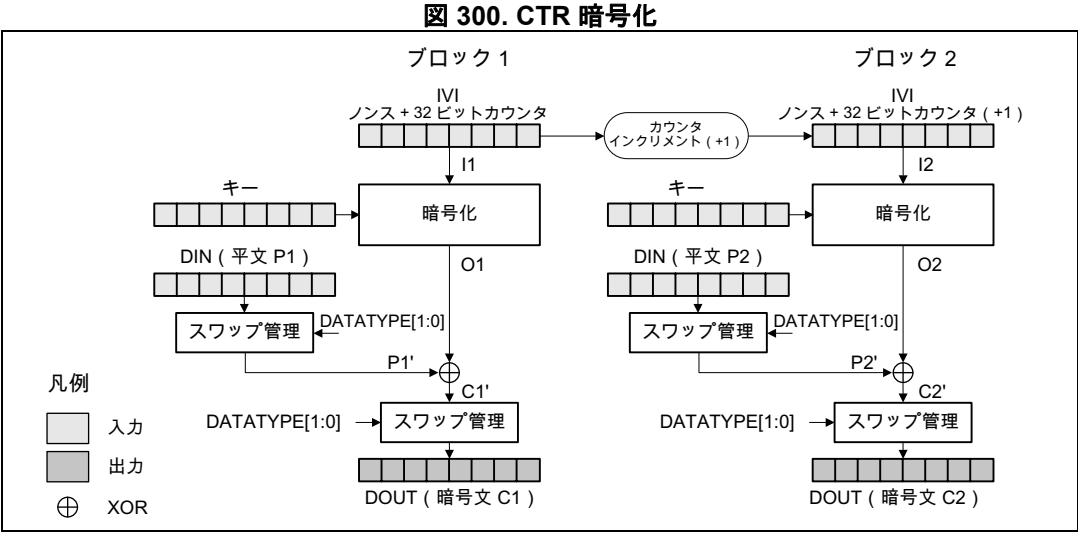
このメッセージの構造は次のとおりです。

- 16 バイトの初期カウンタブロック (ICB) は、次の 2 つの異なるフィールドで構成されています。
 - 初期化ベクタ (IV)** : ある特定のキーでの暗号化サイクルごとに一意でなければならない 96 ビットの値です。
 - カウンタ** : ブロック処理が完了するたびにインクリメントされる 32 ビットのビッグエンディアンの整数です。カウンタの初期値は 1 に設定する必要があります。
- 平文 P は、既知の長さの暗号文 C として暗号化されます。この長さは 16 バイトの倍数でなくとも構いませんが、その場合には平文のパディングが必要です。

詳細については、NIST 特別公報 800-38A 『ブロック暗号の推奨動作モード』 ((Special Publication 800-38C, NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation)) を参照してください。

CTR の暗号化と復号

図 300 には、AES ペリフェラルでのカウンタ (CTR) の連鎖の実行手順 (暗号化) を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x2 を書き込みます。



AES の初期化ベクタは、表 302 に示すように初期化する必要があります。

表 302. カウンタモードの初期化ベクタの定義

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0] 32 ビットカウンタ = 0x0001

CTR の暗号化と復号のプロセス

このプロセスは [セクション 33.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして AES ペリフェラルを無効にします。
2. AES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード (MODE[1:0] に 0x0 または 0x2 を書き込む) で、CTR 連鎖モードを選択します (CHMOD[2:0] に 0x2 を書き込む)。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。
 - KMOD[1:0] を使ってキーモードを選択します。SAES ペリフェラルからキーを取得する場合は、KMOD[1:0] に 0x2 を書き込み、そうでない場合は 0x0 のままにします。
3. [表 302](#) に従って、AES_IVRx レジスタに初期化ベクタを書き込みます。
4. KMOD[1:0] が 0x0 の場合は、AES_KEYRx レジスタにキーを書き込みます。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます ([セクション 33.4.14](#) を参照)。
5. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
6. EN を設定して AES ペリフェラルを有効にします。
7. データを追加します。
 - a) これが最終ブロックであり、ブロックの平文 (暗号化) または暗号文 (復号) のサイズが 16 バイトを下回る場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 33.4.5](#) の説明に従ってデータブロックを AES に追加し、AES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します (MSB が先)。
 - c) 最後から 2 番目のブロックが処理されるまで手順 [b](#)) を繰り返します。平文の最終ブロック (暗号化のみ) に対しては、[a](#)) および [b](#)) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、メッセージの一部ではないビットを破棄します。
8. シーケンスを終了します : EN をクリアして AES ペリフェラルを無効にします。

CTR モードでのサスペンド／レジューム動作

CBC モードと同様に、メッセージを中断して、優先順位の高いメッセージを送信してから、中断したメッセージを再開することができます。CBC のサスペンドおよびレジュームのシーケンスの詳細については、[セクション 33.4.9 : AES 基本連鎖モード \(ECB、CBC\)](#) を参照してください。

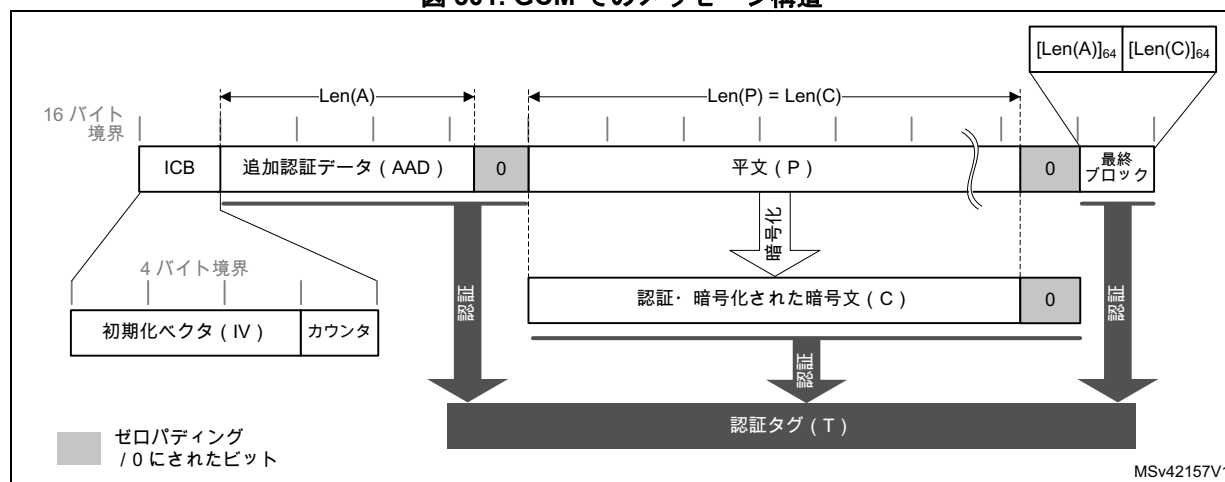
注 : CBC モードと同様に、レジューム動作中に IV レジスタを再ロードする必要があります。

33.4.11 AES ガロア／カウンタモード (GCM)

AES ガロア／カウンタモード (GCM) では、平文メッセージを対応する暗号文およびタグ（メッセージ認証コードとも言います）に暗号化して認証できます。

GCM モードは、機密性については AES カウンタモードに基づいています。固定の有限体に乗算器を使用して、メッセージ認証コードを計算します。GCM モードでの一般的なメッセージ構造を以下の図に示します。

図 301. GCM でのメッセージ構造



メッセージには、次の構造があります。

- 16 バイトの初期カウンタブロック (ICB)** は、次の 2 つの異なるフィールドで構成されています。
 - 初期化ベクタ (IV)** : ある特定のキーでの暗号化サイクルごとに一意でなければならない 96 ビットの値です。GCM 規格では 96 ビット未満の IV に対応していますが、この場合には規則が厳格に適用されます。
 - カウンタ** : ブロック処理が完了するたびにインクリメントされる 32 ビットのビッグエンディアンの整数です。NIST 仕様によると、ペイロードの最初のブロックを処理するときのカウンタ値は 0x2 です。
- 認証済みヘッダ AAD** (追加認証データとも言います) は、既知の長さである Len(A) を持ちます。この長さは 16 バイトの倍数でなくても構いませんが、 2^{64} ビットを超えてはなりません。メッセージのこの部分には認証だけ行われて、暗号化はされません。
- 平文メッセージ P** は暗号文 C として認証と暗号化の両方が行われ、既知の長さである Len(P) を持ちます。この長さは 16 バイトの倍数でなくても構いませんが、16 バイトの $2^{32}-2$ ブロック分を超えることはできません。
- 最終ブロック** には、表 304 に示すとおり、AAD ヘッダ長 (ビット [32:63]) とペイロード長 (ビット [96:127]) の情報が含まれます。

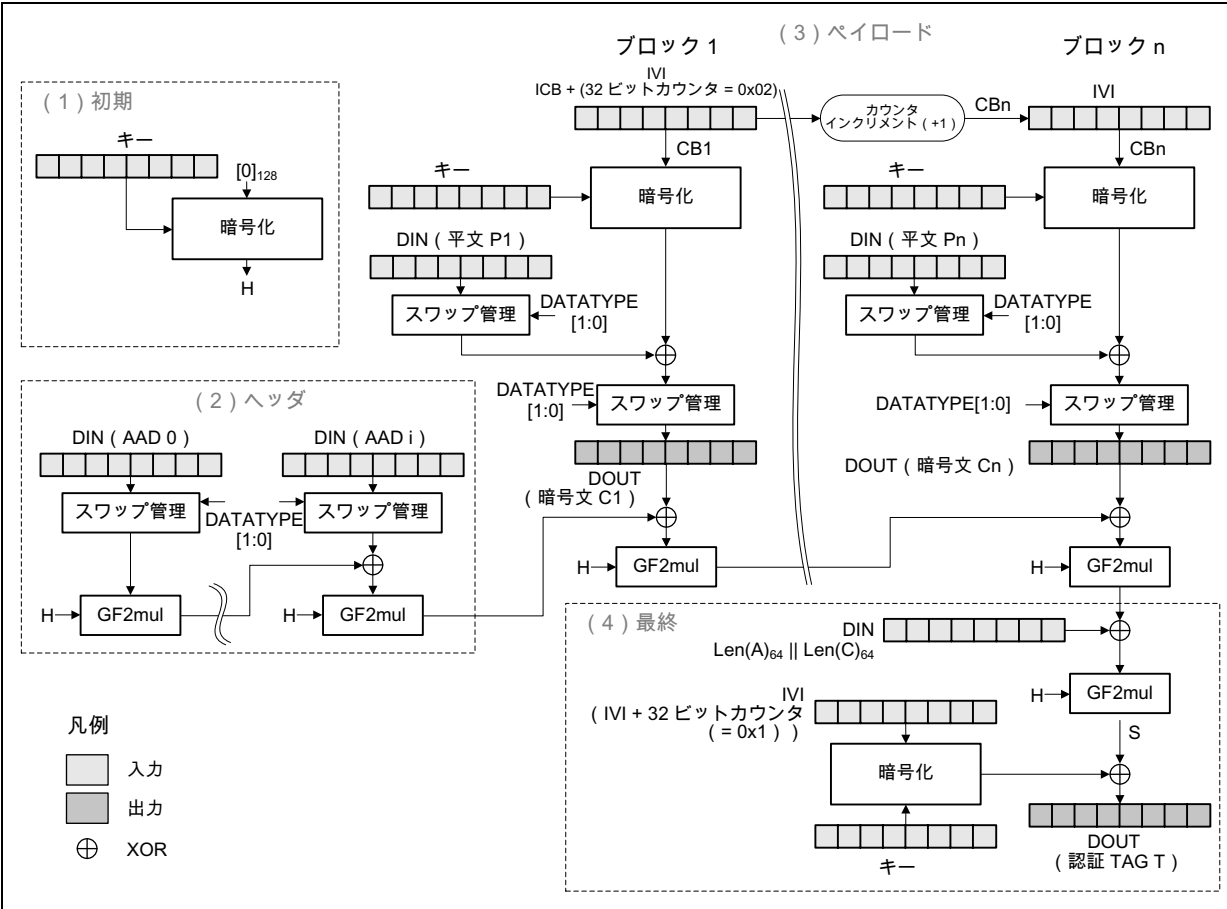
GCM 規格では、暗号文 C は平文 P と同じビット長であることと規定されています。

メッセージ (AAD または P) の長さが 16 バイトの倍数ではない場合には、特別なパディング処理が必要となります。

詳細は、NIST 特別公報 800-38D『ブロック暗号の推奨動作モード - ガロア／カウンタモード (GCM) および GMAC』(NIST Special Publication 800-38D, Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation - Galois/Counter Mode (GCM) and GMAC) を参照してください。

図 302 には、AES ペリフェラルでの GCM の連鎖の実行手順（暗号化）を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x3 を書き込みます。

図 302. GCM 認証暗号化



最初のカウンタブロック（CB1）は、アプリケーションソフトウェアによって初期カウンタブロック ICB から（表 303 に定義されているとおり）導出されます。

表 303. GCM モードでの IV レジスタの初期化

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
ICB[127:96]	ICB[95:64]	ICB[63:32]	ICB[31:0] 32 ビットカウンタ = 0x0002

GCM メッセージの最後のブロックには、表 304 に示すように、AAD ヘッダーの長さとペイロードの長さの情報が含まれています。

表 304. GCM 最終ブロックの定義

AES_DINR への ワード順	最初のワード	2 番目のワード	3 番目のワード	4 番目のワード
入力データ	AAD 長 [63:32]	AAD 長 [31:0]	ペイロード長 [63:32]	ペイロード長 [31:0]

GCM の暗号化と復号のプロセス

このプロセスは [セクション 33.4.6](#) で説明されており、以下の一連のイベントが含まれます。

GCM の初期化

1. EN をクリアして AES ペリフェラルを無効にします。
2. AES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード (MODE[1:0] に 0x0 または 0x2 を書き込む) で、GCM 連鎖モードを選択します (CHMOD[2:0] に 0x3 を書き込む)。MODE[1:0] に 0x1 は書き込まないでください。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。
 - KMOD[1:0] を使ってキーモードを選択します。SAES ペリフェラルからキーを取得する場合は、KMOD[1:0] に 0x2 を書き込み、そうでない場合は 0x0 のままにします。
 - GCMPH[1:0] に 0x0 を書き込んで、GCM 初期化フェーズを選択します。
3. [表 303](#) に従って、AES_IVRx レジスタに初期化ベクタを書き込みます。
4. KMOD[1:0] が 0x0 の場合は、AES_KEYRx レジスタにキーを書き込みます。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます ([セクション 33.4.14](#) を参照)。
5. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
6. EN をセットしてハッシュキーの計算を開始します。計算が完了すると、EN は自動的にクリアされます。
7. CCF フラグが AES_ISR レジスタでセットされ、GCM ハッシュサブキー (H) の計算完了が示されるまで待ちます。
8. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。

GCM ヘッドフェーズ

9. ヘッドフェーズを初期化します。
 - a) GCMPH[1:0] に 0x1 を書き込んで、GCM ヘッドフェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) EN を設定して AES ペリフェラルを有効にします。
10. ヘッドデータを追加します。
 - a) これが最終ブロックであり、ブロックの AAD が 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 33.4.5](#) で説明したように、AES にデータブロックを追加します。
 - c) 最後から 2 番目の AAD データブロックが処理されるまで、手順 [b\)](#) を繰り返します。最後のブロックに対して、手順 [a\)](#) と [b\)](#) を実施します。

注： このフェーズは、AAD がなければ ($\text{Len}(A) = 0$ であれば) スキップできます。

ヘッドフェーズではデータは読み込まれません。

GCM ペイロードフェーズ

11. ペイロードフェーズを初期化します。
 - a) GCMPH[1:0] に 0x2 を書き込んで、GCM ペイロードフェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) ヘッドフェーズをスキップした場合、EN をセットして、AES ペリフェラルを有効にします。

12. ペイロードデータを追加します。
 - a) これが最終ブロックであり、ブロックのメッセージが 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 33.4.5](#) の説明に従ってデータブロックを AES に追加し、AES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します。
 - c) 最後から 2 番目の平文ブロックが暗号化されるか、暗号文の最終ブロックが復号されるまで手順 b) を繰り返します。平文の最終ブロック（暗号化のみ）に対しては、a) および b) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、ペイロードの一部ではないビットを破棄します。

注： このフェーズは、ペイロードがなければ (Len(C) = 0) スキップできます (GMAC モードを参照)。

GCM の終了処理

13. GCM_PH[1:0] に 0x3 を書き込んで、GCM 最終フェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
14. AES_DINR レジスタに最終 GCM ブロックを書き込みます。これは、[表 304](#) に示すとおり、AAD ビットとペイロードビットの長さを連結したものです。
15. AES_ISR レジスタの CCF フラグがセットされるまで待ちます。
16. AES_DOUTR レジスタを 4 回読み出して GCM 認証タグを取得します。
17. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
18. EN をクリアして AES ペリフェラルを無効にします。認証された復号である場合には、生成されたタグをメッセージとともに渡された予測タグと比較します。

注： 最終フェーズでは、データは通常どおり (スワップなしに) AES_DINR に書き込まれますが、AES_DOUTR から読み出されるタグデータにはスワップが適用されます。

ヘッダフェーズまたはペイロードフェーズから最終フェーズに遷移する際に、AES を無効にしないでください。そうしないと、誤った結果になります。

GCM モードでのサスペンド／レジューム動作

メッセージの処理をサスペンドするには、次の手順に従います。

1. DMA が使用されている場合には、DMAINEN をクリアして、入力 FIFO への AES DMA 転送を停止します。DMA が使用されていない場合は、AES_ISR レジスタの CCF フラグがセットされる (計算が完了する) まで待ちます。
2. ペイロードフェーズで、DMA が使用されていない場合は、AES_DOUTR レジスタを 4 回読み出して、最後に処理されたブロックを保存します。DMA が使用されている場合には、AES_ISR レジスタで CCF フラグがセットされるまで待ってから、DMAOUTEN をクリアして、出力 FIFO からの DMA 転送を停止します。
3. AES_ICR レジスタの CCF ビットをセットして、AES_ISR レジスタの CCF フラグをクリアします。
4. メモリに AES_SUSPRx レジスタを保存します。
5. ペイロードフェーズでは、AES_IVRx レジスタがデータ処理中に初期値から変化するため、それらを保存します。ヘッダフェーズでは、この手順は不要です。
6. EN をクリアして AES ペリフェラルを無効にします。
7. メモリに現在の AES_CR 設定を保存します。アプリケーションが元のキー値を知っていますので、キーレジスタを保存する必要はありません。
8. DMA が使用されている場合には、DMA コントローラのステータス (AES 入力データ転送のポインタ、残りバイト数など) を保存します。ペイロードフェーズでは、AES 出力データ転送のポインタも保存します。

メッセージの処理をレジュームするには、次の手順に従います。

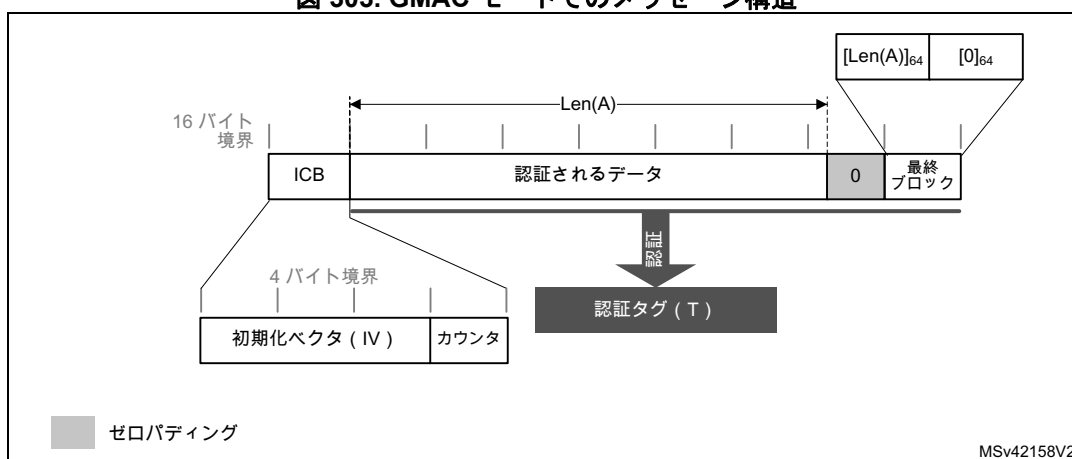
1. DMA が使用されている場合には、DMA コントローラを設定して残りの入力 FIFO の転送を完了させます。ペイロードフェーズでは、残りの出力 FIFO の転送のためにも DMA コントローラを設定します。
2. EN をクリアして AES ペリフェラルを無効にします。
3. メモリに事前に保存したサスペンドレジスタの値を、対応する AES_SUSPRx レジスタに書き戻します。
4. ペイロードフェーズでは、メモリに事前に保存した初期化ベクタレジスタの値を、対応する AES_IVRx レジスタに書き戻します。ヘッダフェーズでは、初期設定値を AES_IVRx レジスタに書き戻します。
5. KMOD[1:0] が 0x0 の場合は、AES_CR と AES_KEYRx のレジスタに初期設定値を復元します。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます(セクション 33.4.14 を参照)。
6. EN を設定して AES ペリフェラルを有効にします。
7. DMA が使用されている場合には、DMAINEN (とペイロードフェーズの場合は DMAOUTEN も) をセットして、AES DMA リクエストを有効にします。

33.4.12 AES ガロアメッセージ認証コード (GMAC : Galois message authentication code)

ガロアメッセージ認証コード (GMAC) では、平文を認証して、対応するタグ情報 (メッセージ認証コードとも言います) を生成することができます。

平文の認証済みデータのみで構成されたメッセージ (すなわち、ヘッダのみでペイロードなし) に適用されることを除けば、GMAC は GCM と似ています。GMAC での一般的なメッセージ構造を以下の図に示します。

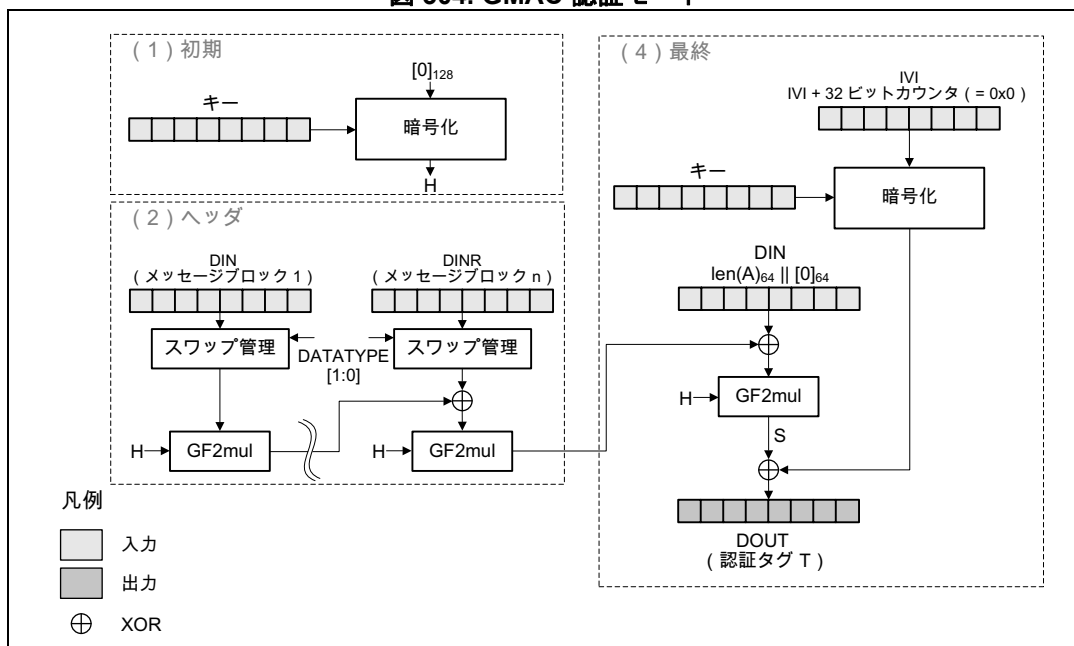
図 303. GMAC モードでのメッセージ構造



詳細は、NIST 特別公報 800-38D『ブロック暗号の推奨動作モード - ガロア/カウンタモード (GCM) および GMAC』(NIST Special Publication 800-38D, Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation - Galois/Counter Mode (GCM) and GMAC) を参照してください。

図 304 には、AES ペリフェラルでの GMAC 連鎖の実行手順を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x3 を書き込みます。

図 304. GMAC 認証モード



GMAC アルゴリズムはヘッダのみ含まれているメッセージに適用される GCM アルゴリズムに相当します。その結果、ペイロードフェーズが省略されること以外の手順と設定がすべて GCM と同じです。

GMAC でのサスペンド／レジューム動作

GMAC モードでは、ヘッダフェーズのみが割込み可能であることを除いて、GCM で示されたシーケンスが適用されます。

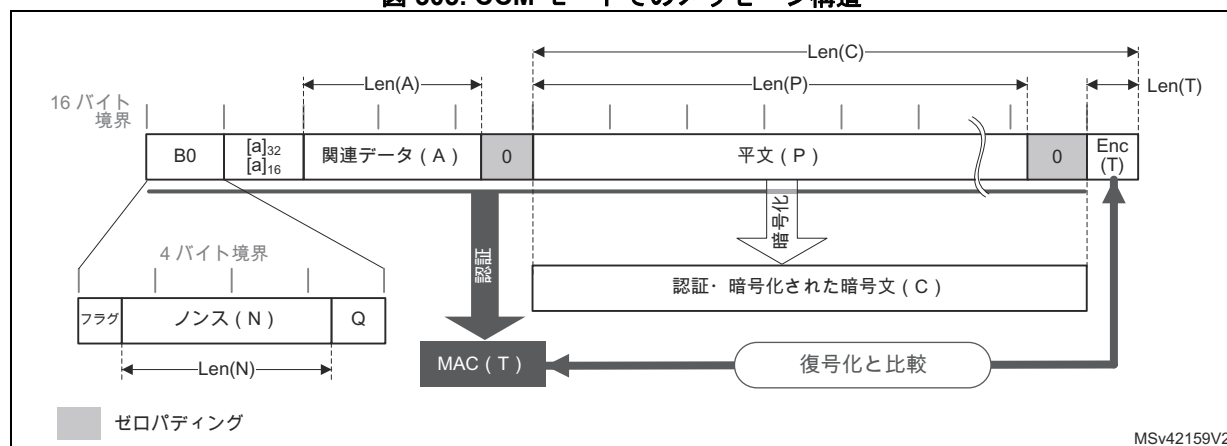
33.4.13 AESCBC-MAC 付きカウンタ (CCM)

AES 暗号ブロック連鎖-メッセージ認証コード付きカウンタ (CCM) アルゴリズムでは、平文を暗号化して認証し、対応する暗号文およびタグ (メッセージ認証コードとも言います) を生成することができます。機密性の確保のため、CCM アルゴリズムは AES カウンタモード処理に基づいています。暗号ブロック連鎖技術を使ってメッセージ認証コードを生成します。これは、一般的に CBC-MAC と呼ばれます。

注: NIST は、CCM 仕様の関係以外では CBC-MAC を認証モードとして承認していません。

CCM での一般的なメッセージ構造を以下の図に示します。

図 305. CCM モードでのメッセージ構造



メッセージの構造は次のとおりです。

- 16 バイトの初期認証ブロック (B0)** は、次の 3 つの異なるフィールドで構成されています。
 - Q** : P のオクテット長 (Len(P)) を表すビットストリングです。
 - ノンズ (N : Nonce)** : Len(P) サイズの使い捨て値です (新しい通信のたびに新しいノンズを割り当てる必要があります)。Len(N) + Len(P) の合計は、15 バイトにする必要があります。
 - フラグ** : 規格に規定されている、制御情報のための 4 つのフラグを格納する最上位オクテットです。値 **t** (バイト単位での MAC の長さ) と **Q** (Len(P) < 2^{8Q} バイトとなる平文の長さ) をエンコードするための 3 ビットのストリングが 2 つ格納されています。**Q** に関連付けられるカウンタブロックの範囲は 2^{8Q-4} となり、**Q** の最大値が 8 である場合、暗号で使用するカウンタブロックは 60 ビットでなければなりません。
- 16 バイトブロック (B)** は関連データ (A) に関連付けられます。メッセージのこの部分には認証だけ行われて、暗号化はされません。この部分は既知の長さである Len(A) を持っており、この長さは 16 バイトの倍数でなくとも構いません (図 305 参照)。規格には、最初のメッセージブロック (B1) の上位側ビットにおいて、バイト数で表現された関連データ長 (a) は次の定義のようにエンコードされなければならないとも示されています。
 - $0 < a < 2^{16} - 2^8$ である場合には、[a]₁₆ のようにエンコードされ、2 バイトとなります。
 - $2^{16} - 2^8 < a < 2^{32}$ である場合には、0xff || 0xfe || [a]₃₂ のようにエンコードされ、6 バイトとなります。
 - $2^{32} < a < 2^{64}$ である場合には、0xff || 0xff || [a]₆₄ のようにエンコードされ、10 バイトとなります。
- 16 バイトブロック (B)** は既知の長さ Len(P) を持ち、暗号文 C として認証と暗号化の両方が行われている平文メッセージ P に関連付けられます。この長さは 16 バイトの倍数でなくとも構いません (図 305 参照)。
- 長さ Len(T) の暗号化された **MAC (T)** が、全長が Len(C) である暗号文 C に追加されます。

メッセージ (A または P) の長さが 16 バイトの倍数ではない場合には、特別なパディング処理が必要となります。

注 : CCM 連鎖モードは、関連データのみ (ペイロードなし) にも使用できます。

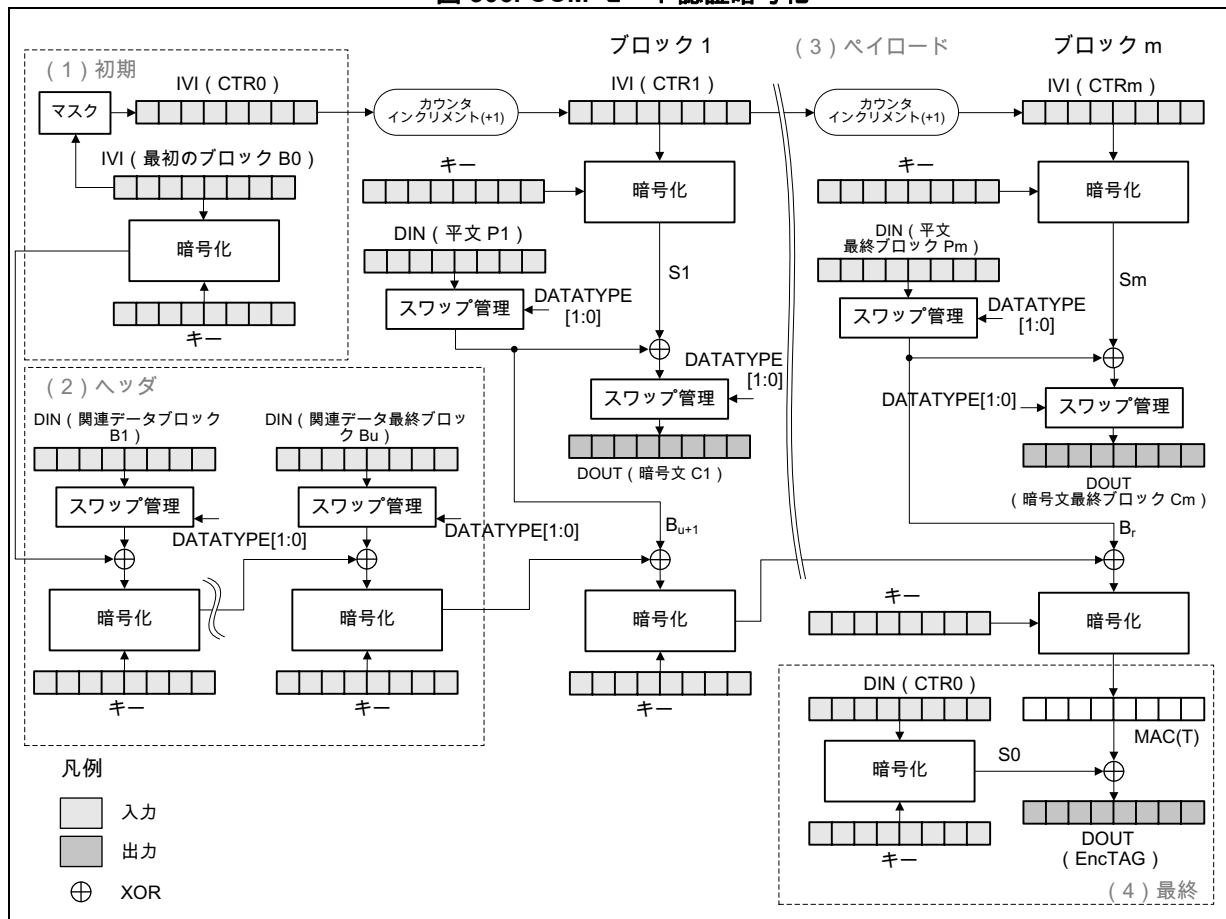
例として、NIST 特別公報 800-38C のセクション C.1 には次の値（16 進数）が示されています。

N : 10111213 141516 (Len(N) = 56 ビットまたは 7 バイト)
 A : 00010203 04050607 (Len(A) = 64 ビットまたは 8 バイト)
 P : 20212223 (Len(P) = 32 ビットまたは 4 バイト)
 T : 6084341B (Len(T) = 32 ビットまたは t = 4)
 B0 : 4F101112 13141516 00000000 00000004
 B1 : 00080001 02030405 06070000 00000000
 B2 : 20212223 00000000 00000000 00000000
 CTR0 : 0710111213 141516 00000000 00000000
 CTR1 : 0710111213 141516 00000000 00000001

詳細については、NIST 特別公報 800-38C 『ブロック暗号の推奨動作モード - 認証および機密性のための CCM モード』を参照してください。(NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation - The CCM Mode for Authentication and Confidentiality)

図 306 には、AES ペリフェラルでの CCM の連鎖の実行手順（暗号化）を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x4 を書き込みます。

図 306. CCM モード認証暗号化



CCM メッセージの最初のブロック (B0) は、表 305 で定義されているようにアプリケーションによって準備される必要があります。

表 305. CCM モードでの IV レジスタの初期化

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
B0 [127:96] ⁽¹⁾	B0 [95:64]	B0 [63:32]	B0 [31:0] ⁽²⁾

1. 上位側 5 ビットはクリアされます (フラグビット)。
2. セットされるビット 0 を除く、Q の長さビットはクリアされます。

AES は、NIST が指定する最大 64 ビットのカウンタをサポートしています。

CCM の暗号化と復号のプロセス

このプロセスは [セクション 33.4.6](#) で説明されており、以下の一連のイベントが含まれます。

CCM の初期化

1. EN をクリアして AES ペリフェラルを無効にします。
2. AES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード (MODE[1:0] に 0x0 または 0x2 を書き込む) で、CCM 連鎖モードを選択します (CHMOD[2:0] に 0x4 を書き込む)。MODE[1:0] に 0x1 は書き込まないでください。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。
 - KMOD[1:0] を使ってキーモードを選択します。SAES ペリフェラルからキーを取得する場合は、KMOD[1:0] に 0x2 を書き込み、そうでない場合は 0x0 のままにします。
 - GCMPH[1:0] に 0x0 を書き込んで、CCM 初期化フェーズを選択します。
3. [表 305](#) に従って、AES_IVRx レジスタに B0 データを書き込みます。
4. KMOD[1:0] が 0x0 の場合は、AES_KEYRx レジスタにキーを書き込みます。KMOD[1:0] が 0x2 の場合は、キーは SAES ペリフェラルから転送されます ([セクション 33.4.14](#) を参照)。
5. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
6. EN をセットして最初のマスク計算を開始します。計算が完了すると、EN ビットは自動的にクリアされます。
7. AES_ISR レジスタの CCF フラグがセットされるまで待ちます。
8. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。

CCM ヘッダフェーズ

9. ヘッダフェーズを初期化します。
 - a) CCM 連鎖ルールに従って、メッセージに関連付けられた (B1) データの最初のブロックを準備します。
 - b) GCMPH[1:0] に 0x1 を書き込んで、CCM ヘッダフェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
 - c) EN を設定して AES ペリフェラルを有効にします。
10. ヘッダデータを追加します。
 - a) これが最終ブロックであり、ブロックの関連データが 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 33.4.5](#) で説明したように、AES にデータブロックを追加します。
 - c) 最後から 2 番目の関連データブロックが処理されるまで、手順 [b\)](#) を繰り返します。最後のブロックに対して、手順 [a\)](#) と [b\)](#) を実施します。

注： このフェーズは、関連データがなければ ($\text{Len}(A) = 0$ であれば) スキップできます。
ヘッダフェーズではデータは読み込まれません。

CCM ペイロードフェーズ

11. ペイロードフェーズを初期化します。
 - a) GCMPH[1:0] に 0x2 を書き込んで、CCM ペイロードフェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) ヘッダフェーズをスキップした場合、EN をセットして、AES ペリフェラルを有効にします。
12. ペイロードデータを追加します。
 - a) 暗号化においてのみ、これが最終ブロックであり、ブロックの平文が 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) セクション 33.4.5 の説明に従ってデータブロックを AES に追加し、AES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します。
 - c) 最後から 2 番目の平文ブロックが暗号化されるか、暗号文の最終ブロックが復号されるまで手順 b) を繰り返します。平文の最終ブロック（暗号化のみ）に対しては、a) および b) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、ペイロードの一部ではないビットを破棄します。

注： このフェーズは、ペイロードがなければ ($\text{Len}(P) = 0$ または $\text{Len}(C) = \text{Len}(T)$ であれば) スキップできます。

暗号文 C を復号するときには、暗号化されたタグ情報である $\text{LSB}_{\text{Len}(T)}(C)$ を削除します。

CCM の終了処理

13. GCMPH[1:0] に 0x3 を書き込んで、CCM 最終フェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
14. AES_ISR レジスタの CCF フラグがセットされるまで待ちます。
15. AES_DOUTR レジスタを 4 回読み出して CCM 認証タグを取得します。
16. AES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
17. EN をクリアして AES ペリフェラルを無効にします。認証された復号である場合には、生成されたタグをメッセージとともに渡された予測タグと比較します。有効なタグを取得するために、タグの長さで認証タグ出力をマスクします。

注： 最終フェーズでは、AES_DOUTR レジスタから読み出されるタグデータにはスワップが適用されます。
ヘッダフェーズまたはペイロードフェーズから最終フェーズに遷移する際に、AES を無効にしないでください。そうしないと、誤った結果になります。

CCM モードでのサスペンドおよびレジューム動作

ヘッダフェーズまたはペイロードフェーズでメッセージの処理をサスペンドするには、次の手順に従います。

1. DMA が使用されている場合には、DMAINEN をクリアして、入力 FIFO への AES DMA 転送を停止します。DMA が使用されていない場合は、AES_ISR レジスタの CCF フラグがセットされる（計算が完了する）まで待ちます。
2. ペイロードフェーズで、DMA が使用されていない場合は、AES_DOUTR レジスタを 4 回読み出して、最後に処理されたブロックを保存します。DMA が使用されている場合には、AES_ISR レジスタで CCF フラグがセットされるまで待ってから、DMAOUTEN をクリアして、出力 FIFO からの DMA 転送を停止します。
3. AES_ICR レジスタの CCF ビットをセットして、AES_ISR レジスタの CCF フラグをクリアします。
4. メモリに AES_SUSPRx レジスタを保存します。

5. データ処理中に変更された IV レジスタを保存します。
6. EN をクリアして AES ペリフェラルを無効にします。
7. メモリに現在の AES_CR 設定を保存します。アプリケーションが元のキー値を知っているため、キーレジスタを保存する必要はありません。
8. DMA が使用されている場合には、DMA コントローラのスレータス (AES 入力データ転送のポイント、残りバイト数など) を保存します。ペイロードフェーズでは、AES 出力データ転送のポイントも保存します。

メッセージの処理をレジュームするには、次の手順に従います。

1. DMA が使用されている場合には、DMA コントローラを設定して残りの入力 FIFO の転送を完了させます。ペイロードフェーズでは、残りの出力 FIFO の転送のためにも DMA コントローラを設定します。
2. EN をクリアして AES ペリフェラルを無効にします。
3. メモリに事前に保存したサスペンドレジスタの値を、対応する AES_SUSPRx レジスタに書き戻します。
4. 保存されている設定を用いて AES_IVRx レジスタを復元します。
5. KMOD[1:0] が 0x0 の場合は、AES_CR と AES_KEYRx のレジスタに初期設定値を復元します。KMOD[1:0] が 0x2 の場合は、キーを再び SAES ペリフェラルから転送する必要があります (セクション 33.4.14 を参照)。
6. EN を設定して AES ペリフェラルを有効にします。
7. DMA が使用されている場合には、DMAINEN (とペイロードフェーズの場合は DMAOUTEN も) をセットして、AES DMA リクエストを有効にします。

33.4.14 AES セキュア・コプロセッサとの AES キー共有

AES ペリフェラルは SAES ペリフェラルをセキュリティ・コプロセッサとして使用できます。セキュア・アプリケーションが、堅牢な SAES ペリフェラルでキーを準備し、準備が整うと、AES アプリケーションは専用のハードウェアキーバスを介してこの準備されたキーをロードできます。

推奨されるシーケンスについては、これ以降と、この文書の [SAES](#) セクションの共有キーを使用した SAES 操作のセクションで説明します。

1. SAES ペリフェラルでは、アプリケーションは共有キーモードで (KMOD[1:0] を 0x2 にして) 共有するキーを暗号化 (ラップ) します。
2. AES ペリフェラルで共有キーが必要になるたびに、アプリケーションは共有キーモードで (KMOD[1:0] を 0x2 にして) 共有キーを SAES ペリフェラルで復号します。
3. 共有キーが復号 (アンラップ) され、SAES_KEYRx レジスタにロードされると、AES と共有できるようになります。共有キーを AES にロードするには、アプリケーションは KEYSIZE を適切に設定し、KMOD[1:0] に 0x2 を書き込みます。KEYVALID がクリアされると、キーはハードウェアによって AES_KEYRx レジスタに自動的に転送され、AES_SR レジスタの BUSY フラグがセットされます。
4. キー転送が完了すると、BUSY フラグがクリアされ、AES_SR レジスタの KEYVALID フラグがセットされます。BUSY ビットがクリアされたときに KEYVALID がセットされない場合、または AES_ISR レジスタに KEIF フラグがセットされる場合は、KEYSIZE 値が正しくないか、転送中に予期しないイベント (DPA エラー、改ざんイベント、または転送終了前に KEYVALID がクリアされたなど) が発生したかのいずれかです。このようなエラーが発生した場合は、両方のペリフェラルの IPRST ビットを使用して両ペリフェラルをリセットし、キー共有プロセス全体を再起動します。

キー共有シーケンスが完了すると、AES は有効な共有キーで初期化されます。その後、アプリケーションは KMOD[1:0] に 0x0 を書き込むことで、通常のキーモードでデータを処理できるようになります。

注： SAES ペリフェラルが未使用であり、正しくキー共有状態にある限り、AES ペリフェラル内のこのシーケンスは複数回（たとえば、サスペンド/レジューム状況を管理するため）実行できます。

33.4.15 AES データレジスタおよびデータスワッピング

データの入出力

16 バイトのデータブロックは、AES_DINR レジスタ（ビットフィールド DIN[31:0]）に 4 つの 32 ビットワードを最上位ワード（ビット [127:96]）から最下位ワード（ビット [31:0]）まで連続で書き込むことで、AES ペリフェラルに入力されます。

16 バイトのデータブロックは、AES_DOUTR レジスタ（ビットフィールド DOUT[31:0]）から 4 つの 32 ビットワードを最上位ワード（ビット [127:96]）から最下位ワード（ビット [31:0]）まで連続で読み出すことで、AES ペリフェラルから取得されます。

16 バイトのデータブロックの 4 つの 32 ビットワードは、ビッグエンディアン順に（最上位ワードを一番小さいアドレスに）連続してメモリに格納する必要があります。詳細については、表 306 の「スワッピングなし」オプションを参照してください。

データスワッピング

AES ペリフェラルは、AES 処理コアに読み込まれる前の AES_DINR レジスタの入力データワード、および AES_DOUTR レジスタに送られる前の AES 処理コアから出力されたデータについて、ビット、バイト、ハーフワードのスワッピングまたはスワッピングなしに設定できます。データのタイプによって選択します。たとえば、ASCII テキストストリームの場合、バイトスワッピングを使用します。

データスワッピングのタイプは、DATATYPE[1:0] で選択します。この選択は AES の入力と出力の両方に適用されます。

注： AES キーレジスタ（AES_KEYRx）と初期化ベクタレジスタ（AES_IVRx）のデータは、選択されたスワップモードの影響を受けません。

AES のデータスワッピング機能の要約を、表 306 と図 307 に示します。

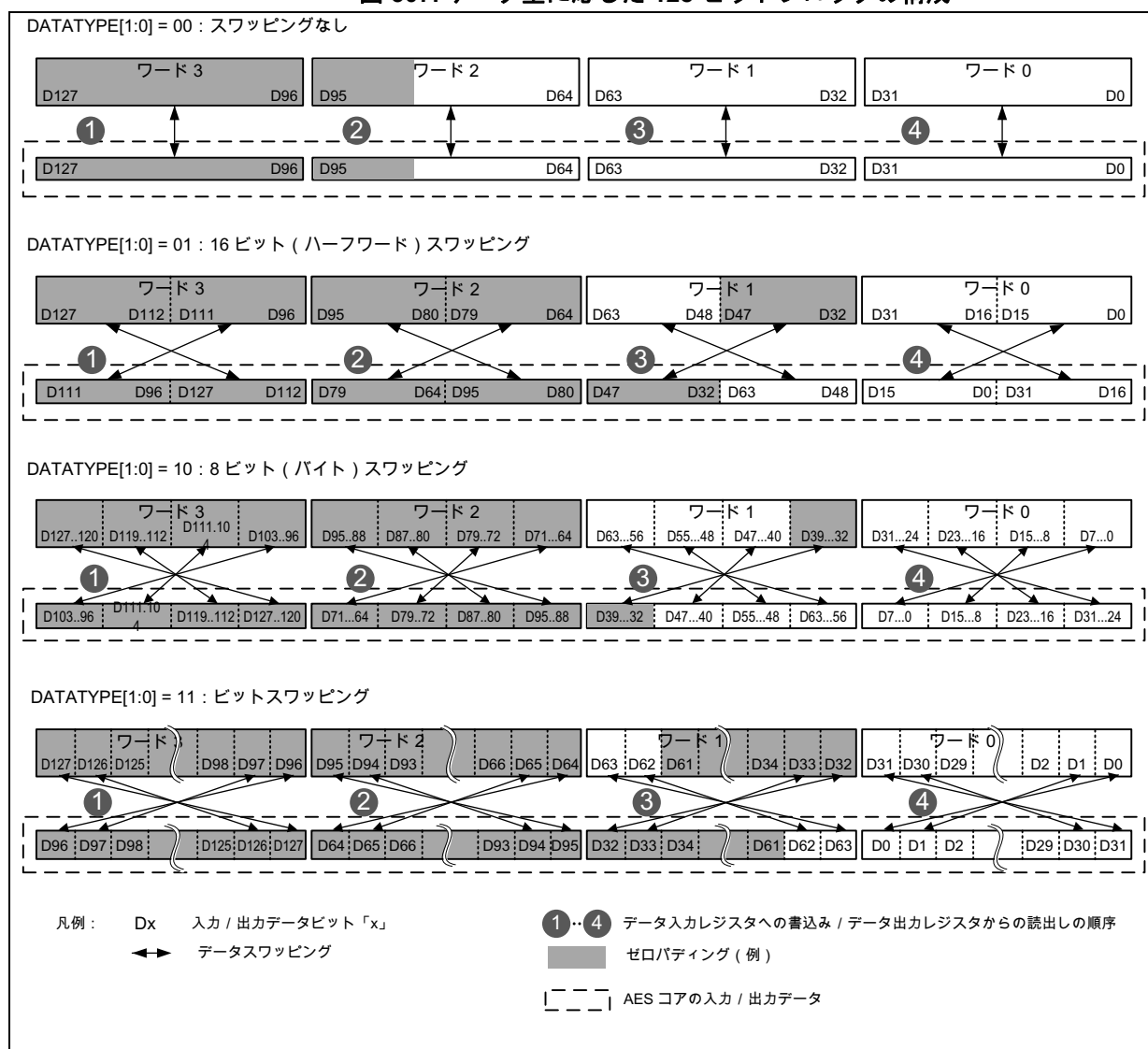
表 306. AES データスワッピングの例

DATATYPE[1:0]	実施される スワッピング処理	データブロック
		システムメモリデータ（ビッグエンディアン）
0x0	スワッピングなし	Block[127..64] : 0x04EEF672 : 2E04CE96
		Block[63..0] : 0x4E6F7720 69732074
		Address @、word[127..96] : 0x04EEF672
		Address @ + 0x4、word[95..64] : 0x2E04CE96 Address @ + 0x8、word[63..32] : 0x4E6F7720 Address @ + 0xC、word[31..0] : 0x69732074
0x1	ハーフワード (16 ビット) スワッピング	Block[63..0] : 0x4E6F 7720 6973 2074
		Address @、word[63..32] : 0x7720 4E6F
		Address @ + 0x4、word[31..0] : 0x2074 6973
0x2	バイト (8 ビット) スワッピング	Block[63..0] : 0x4E 6F 77 20 69 73 20 74
		Address @、word[63..32] : 0x20 77 6F 4E
		Address @ + 0x4、word[31..0] : 0x74 20 73 69

表 306. AES データスワッピングの例 (続き)

DATATYPE[1:0]	実施される スワッピング処理	データブロック
		システムメモリデータ (ビッグエンディアン)
0x3	ビットスワッピング	Block[63..32] : 0x4E6F7720 0100 1110 0110 1111 0111 0111 0010 0000 Block[31..0] : 0x69732074 0110 1001 0111 0011 0010 0000 0111 0100
		Address @、word[63..32] : 0x04EE : F672 0000 0100 1110 1110 1111 0110 0111 0010 Address @ + 0x4、word[31..0] : 0x2E04 : CE96 0010 1110 0000 0100 1100 1110 1001 0110

図 307. データ型に応じた 128 ビットブロックの構成



データパディング

図 307 では、データスワッピング後の 0 にされたビットによって AES コア入力バッファの最上位ビット側に隣接したゾーンが形成される場合など、0 でパディングされるメモリデータブロックの例も示しています。この例では、次を含む入力データブロックのパディングが示されています。

- 84 メッセージビット (DATATYPE[1:0] = 0x0)
- 48 メッセージビット (DATATYPE[1:0] = 0x1)
- 56 メッセージビット (DATATYPE[1:0] = 0x2)
- 34 メッセージビット (DATATYPE[1:0] = 0x3)

33.4.16 AES キーレジスタ

8 つの AES_KEYRx 書込み専用レジスタには、表 307 に示すように、暗号化キーまたは復号キーの情報が保存されます。読出しはセキュリティ上の理由で許されません。

注： メモリおよび AES キーレジスタでは、キーはリトルエンディアン形式で格納され、最上位バイトが最上位アドレスに配置されます。

表 307. AES_KEYRx レジスタでのキーエンディアン (128/256 ビットのキー)

AES_KEYR7 [31:0]	AES_KEYR6 [31:0]	AES_KEYR5 [31:0]	AES_KEYR4 [31:0]	AES_KEYR3 [31:0]	AES_KEYR2 [31:0]	AES_KEYR1 [31:0]	AES_KEYR0 [31:0]
-	-	-	-	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
KEY[255:224]	KEY[223:192]	KEY[191:160]	KEY[159:128]	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]

キーレジスタは、DATATYPE[1:0] ビットフィールドで制御されるデータスワッピング機能による影響を受けません。

AES ペリフェラルが有効になっている (EN ビットがセットされている) 場合、AES_KEYRx レジスタへの書込み操作は無視されます。アプリケーションでは、キーレジスタを修正する前にこれを確認する必要があります。

キー全体は AES の計算を開始する前に書き込む必要があります。通常のキーモード (KMOD[1:0] が 0x0 の場合) では、キーレジスタは常に昇順または降順で書き込まれる必要があります。書込みシーケンスは次のようになります。

- KEYSIZE がクリアされている場合、AES_KEYRx (x = 0 から 3 または x = 3 から 0)
- KEYSIZE がセットされている場合、AES_KEYRx (x = 0 から 7 または x = 7 から 0)

注： KEYSIZE は、キーを書き込む前に書き込む必要があります。

最初のキーレジスタが書き込まれると直ちに KEYVALID フラグがクリアされます。キーレジスタの書込みシーケンスが完了すると、KEYVALID がセットされ、EN が書き込めるようになります。エラーが発生すると、KEYVALID はクリアされ、KEIF がセットされます (セクション 33.4.18 を参照)。

33.4.17 AES 初期化ベクタレジスタ

4 つの AES_IVRx レジスタには、表 308 に示されているように初期化ベクタ (IV) 情報が格納されます。これらは、AES ペリフェラルが無効 (EN クリア) の場合にのみ書込みできます。

注： メモリおよび AES IV レジスタでは、初期化ベクタはリトルエンディアン形式で格納され、最上位バイトが最上位アドレスに配置されます。

表 308. AES_IVRx レジスタ上に配置される IVI ビットフィールド

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0]

初期化ベクタ情報は、選択した連鎖モードによって異なります。AES_IVRx レジスタは、使用されると、AES 計算サイクルごとに更新されます（サスペンドモードの管理に役立ちます）。

初期化ベクタレジスタは、DATATYPE[1:0] で制御されるデータスワッピング機能による影響を受けません。

33.4.18 AES エラー管理

このセクションに記載するエラーは、AES ペリフェラルによって管理されます。

読出しエラーフラグ (RDERRF)

AES_DOUTR レジスタの予期しない読出し試行があると 0 が返され、RDERRF フラグと RWEIF フラグがセットされます。RDERRF は、計算フェーズまたは入力フェーズ中にトリガされます。

注： RDERRF が立っても AES は無効にならず、処理を継続します。

RWEIE ビットがセットされていた場合、割込みが生成されます。詳細については、[セクション 33.5: AES 割込み](#) を参照してください。

AES_ICR レジスタの RWEIF ビットをセットすることで、RDERRF と RWEIF フラグはクリアされます。

書込みエラーフラグ (WDERR)

AES_DINR レジスタの予期しない書込み試行は無視され、WRERRF フラグと RWEIF フラグがセットされます。WRERRF は、計算フェーズまたは出力フェーズ中にトリガされます。

注： WRERRF が立っても AES は無効にならず、処理を継続します。

RWEIE ビットがセットされていた場合、割込みが生成されます。詳細については、[セクション 33.5: AES 割込み](#) を参照してください。

AES_ICR レジスタの RWEIF ビットをセットすることで、WRERRF と RWEIF フラグはクリアされます。

キーエラー割込みフラグ (KEIF)

AES_ISR レジスタの KEIF フラグをセットし、AES_SR レジスタの KEYVALID ビットをクリアするエラーの原因は複数あります。

- **キー書込みシーケンスエラー：**キーレジスタの書込みシーケンスの間違いが検出されると、トリガされます。詳細については、[セクション 33.4.16: AES キーレジスタ](#) を参照してください。
- **キー共有サイズの不一致エラー：**KMOD[1:0] が 0x2 で、AES ペリフェラルの KEYSIZE が SAES ペリフェラルの KEYSIZE と一致しない場合にトリガされます。
- **キー共有エラー：**SAES 共有キーの AES ペリフェラルへの転送に失敗すると、トリガされます。詳細については、[セクション 33.4.14: AES セキュア・コプロセッサとの AES キー共有](#) を参照してください。

KEIF フラグは、AES_ICR レジスタの対応するビットでクリアします。AES_IER レジスタの KEIE ビットがセットされている場合、割込みが生成されます。詳細については、[セクション 33.5 : AES 割込み](#) を参照してください。

キー共有エラーが発生した場合は、対応する制御レジスタの IPRST ビットによって AES と SAES のペリフェラルの両方をリセットしてから、キー共有シーケンスを再開します。

注： キーエラーがあった場合は、AES を無効にして再設定する前に KEIF フラグをクリアしてください。

33.5 AES 割込み

AES ペリフェラルによって生成され個別にマスク可能な割込みソースは複数あり、次のイベントが通知されます。

- 計算完了 (CCF)
- 読出しエラー (RDERRF)
- 書込みエラー (WRERRF)
- キーエラー (KEIF)

AES エラーの詳細については、[セクション 33.4.18 : AES エラー管理](#)を参照してください。

これらのソースは、Cortex® CPU 割込みコントローラに接続する AES ペリフェラルからの共通の割込み信号に結合されます。アプリケーションは、AES_IER レジスタの対応するイネーブルビットをセット/クリアすることにより、AES 割込みソースを個別に有効または無効にすることができます。

個別のマスク可能な割込みソースのステータスは AES_ISR レジスタから読み出すことができます。それらは、AES_ICR レジスタの対応ビットをセットすることでクリアされます。

[表 309](#) に、使用可能な機能の概要を示します。

表 309. AES 割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ	イネーブルビット	割込みのクリア方法
AES	計算完了フラグ	CCF	CCFIE	CCF のセット ⁽¹⁾
	読出しエラーフラグ	RDERRF ⁽²⁾	RWEIE	RWEIF のセット ⁽¹⁾
	書込みエラーフラグ	WRERRF ⁽²⁾		
	キーエラーフラグ	KEIF	KEIE	KEIF のセット ⁽¹⁾

1. AES_ICR レジスタのビット。

2. AES_SR レジスタのフラグであり、AES_ISR レジスタのフラグ RWEIF によって反映されます。

33.6 AES DMA リクエスト

AES ペリフェラルには、DMA (ダイレクトメモリアクセス) コントローラに接続するインタフェースが搭載されています。DMA の動作は、AES_CR レジスタの DMAINEN ビットと DMAOUTEN ビットを通じて制御されます。キー導出が選択されている場合 (MODE[1:0] が 0x1 の場合)、これらのビットを設定しても効果はありません。

AES ではシングル DMA リクエストのみがサポートされています。

AES による DMA の使用方法の詳細については、[セクション 33.4.5: AES の暗号化または復号の一般的な使用法](#) の DMA を使用したデータの追加サブセクションを参照してください。

DMA を用いたデータ入力

DMAINEN をセットすると、AES への DMA 書込みが有効になります。すると、AES は入力フェーズ中に、16 バイトのデータブロックごとに AES_DINR レジスタに書き込む一連のシングル DMA リクエスト（4 個の 32 ビットワード、MSB が先）を開始します。

注： 選択されたアルゴリズムとモードに応じて、特別なパディングや暗号文借用が必要となることがあります（[セクション 33.4.7](#) を参照）。

DMA を用いたデータ出力

DMAOUTEN をセットすると、AES からの DMA 読出しが有効になります。すると、AES は出力フェーズ中に、16 バイトのデータブロックごとに AES_DOUTR レジスタから読み出す一連のシングル DMA リクエスト（4 個の 32 ビットワード、MSB が先）を開始します。

16 バイトデータブロックの処理終了時の出力フェーズの後に、AES によって次のデータブロック（ある場合）の新しい入力フェーズに自動的に切り替えられます。

DMA モードでは、AES_DOUTR レジスタの読取りは計算フェーズの終了時に DMA によって自動的に管理されるため、CCF フラグは使用する必要がありません。CCF フラグは、ソフトウェアによるデータ転送管理に再び遷移したときのみクリアする必要があります。

注： メッセージサイズによっては、最後のブロックでアプリケーションによって余分なバイトを破棄する必要があります。

DMA 転送の停止

AES が無効になる（EN がクリアされる）か、DMA イネーブルビット（入力データの場合は DMAINEN、出力データの場合は DMAOUTEN）がクリアされると、すべての DMA リクエスト信号がデアサートされます。

33.7 AES の処理遅延時間

次の表は、動作モードごとの 16 バイトのデータブロックの処理遅延時間を示しています。

表 310. ECB、CBC および CTR での処理遅延時間

キー長	動作モード	連鎖アルゴリズム	クロックサイクル数
128 ビット	暗号化または復号 ⁽¹⁾	ECB、CBC、CTR	51
	キーの準備	-	59
256 ビット	暗号化または復号 ⁽¹⁾	ECB、CBC、CTR	75
	キーの準備	-	82

1. キーの準備時間を除く（ECB および CBC のみ）。

表 311. GCM および CCM での処理遅延時間（クロックサイクル数）

キー長	動作モード	連鎖アルゴリズム	初期化フェーズ	ヘッダフェーズ ⁽¹⁾	ペイロードフェーズ ⁽¹⁾	最終フェーズ ⁽¹⁾
128 ビット	暗号化／復号	GCM	64	35	51	59
		CCM	63	55	114	58
256 ビット	暗号化／復号	GCM	88	35	75	75
		CCM	87	79	162	82

1. データ挿入では、AHB バスで AES により強制されるウェイトステートが含まれる可能性があります（最大 3 サイクル、通常は 1 サイクル）。

33.8 AES レジスタ

レジスタは、32 ビットワードのシングルアクセスでのみアクセス可能。他のアクセスタイプは AHB エラーとなります。32 ビット書き込み以外では、レジスタの内容が破損する可能性があります。

33.8.1 AES 制御レジスタ (AES_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPRST	Res.	Res.	Res.	Res.	Res.	KMOD[1:0]		NPBLB[3:0]				Res.	KEYSIZE	Res.	CHMOD[2]
rw						rw	rw	rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GCMIPH[1:0]		DMAOUTEN	DMAINEN	Res.	Res.	Res.	Res.	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw

ビット 31 **IPRST** : AES ペリフェラルのソフトウェアリセット

このビットをセットすると AES ペリフェラルがリセットされ、IPRST ビット自体を除くすべてのレジスタがそれぞれのデフォルト値に設定されます。したがって、すべてのキー関連データは失われます。このため、安全性の低いアプリケーションに AES を引き渡す前にこのビットをセットすることをお勧めします。

各設定レジスタへの書き込み中は、このビットをローに保つ必要があります。

ビット 30:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **KMOD[1:0]** : キーモード選択

このビットフィールドでは、アプリケーションが AES キーをどのように使用できるかを定義します。KMOD[1:0] を 0 以外にセットする場合は、KEYSIZE を正しく初期化する必要があります。

0x0 : 通常キーモード。AES レジスタは自由に使用できます。

0x2 : 共有キーモード。SAES ペリフェラルで共有キーモードが適切に初期化されている場合、SAES キーレジスタに保存されているデータが AES ペリフェラルによってそのキーレジスタに自動的にロードされます。BUSY ビットがクリアされ、AES_SR レジスタに KEYVALID がセットされている場合に、キーの値は AES キーレジスタで入手できます。それ以外の場合は、AES_ISR レジスタにキーエラーフラグ KEIF がセットされます。

その他 : 予約済み

BUSY がセットされている場合、および書き込みアクセスの前に EN がセットされていて、その書き込みアクセスによってクリアされない場合、このビットフィールドへの書き込みの試みは無視されます。

ビット 23:20 **NPBLB[3:0]** : 最終ブロックにおけるパディングバイト数

このパディング情報は、GCM ペイロード暗号化または CCM ペイロード復号の最終ブロックを処理する前に、ソフトウェアによって入力される必要があります。そうしないと、認証タグの計算が不正になります。

0x0 : 全バイトが有効 (パディングなし)

0x1 : 最後の LSB バイトに対するパディング

.....

0xF : 最終ブロックの LSB バイト 15 個に対するパディング。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **KEYSIZE** : キー長選択

このビットフィールドでは、AES が使用するキーのビット単位の長さを定義します。

0 : 128 ビット

1 : 256 ビット

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットへの書込みの試みは無視されます。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:13 **GCMPH[1:0]** : GCM または CCM フェーズの選択

このビットフィールドで、GCM、GMAC、または CCM 連鎖モードでのみ適用できるフェーズを選択します。

0x0 : 初期化フェーズ

0x1 : ヘッダフェーズ

0x2 : ペイロードフェーズ

0x3 : 最終フェーズ

ビット 12 **DMAOUTEN** : DMA 出力有効

このビットにより、データフェーズ中に、DMA による AES からの送信データ転送のための DMA リクエストの自動生成が有効になります。

0 : 無効化

1 : イネーブル

MODE[1:0] が 0x1 (キー導出) のとき、このビットの設定は無視されます。

ビット 11 **DMAINEN** : DMA 入力イネーブル

このビットにより、データフェーズ中に、DMA による AES への着信データ転送のための DMA リクエストの自動生成が有効になります。

0 : 無効化

1 : イネーブル

MODE[1:0] が 0x1 (キー導出) のとき、このビットの設定は無視されます。

ビット 10:7 予約済みであり、リセット値に保持する必要があります。

ビット 16、**CHMOD[2:0]** : 連鎖モード

6:5 このビットフィールドで、AES 連鎖モードを選択します。

0x0 : 電子コードブック (ECB)

0x1 : 暗号ブロック連鎖 (CBC)

0x2 : カウンタモード (CTR)

0x3 : ガロアカウンタモード (GCM) およびガロアメッセージ認証コード (GMAC)

0x4 : CBC-MAC 付きカウンタ (CCM)

上記以外 : 予約済み

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 4:3 **MODE[1:0]** : 動作モード

このビットフィールドで、AES 動作モードを選択します。

0x0 : 暗号化

0x1 : キー導出 (またはキー準備) (ECB/CBC 復号の場合のみ)

0x2 : 復号

0x3 : 予約済み

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 2:1 **DATATYPE[1:0]** : データ型

このビットフィールドは、データスワッピングのモードを選択して、AES_DINR レジスタに書き込まれるデータまたは AES_DOUTR レジスタから読み出されるデータのフォーマットを定義します。このスワッピングは [セクション 33.4.15 : AES データレジスタおよびデータスワッピング](#) に定義されています。

0x0 : スワッピングなし (32 ビットデータ)

0x1 : ハーフワードスワッピング (16 ビットデータ)

0x2 : バイトスワッピング (8 ビットデータ)

0x3 : ビットレベルスワッピング

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 0 **EN** : イネーブル

このビットは、AES ペリフェラルを有効または無効にします。

0 : 無効化

1 : イネーブル

このビットをクリアしてからセットすれば、いつでも AES ペリフェラルを再初期化できます。

このビットは、キーの準備 (MODE[1:0] が 0x1) の完了時、および GCM/GMAC/CCM 初期化フェーズの完了時に、ハードウェアによって自動的にクリアされます。

– このビットは、KEYVALID がクリアされている限り、セットできません。

33.8.2 AES ステータスレジスタ (AES_SR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	Res.	Res.	Res.	BUSY	WRERRF	RDERRF	Res.
								r				r	r	r	

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **KEYVALID** : キー有効フラグ

このビットは、KEYSIZE で定義されたサイズのキーが AES_KEYRx キーレジスタにロードされると、ハードウェアによってセットされます。

0 : キーは無効

1 : キーは有効

EN ビットは、KEYVALID がセットされている場合にのみセットできます。

KMOD[1:0] が 0 のときの通常モードでは、キーを正しい順序でキーレジスタに書き込む必要があります。そうしないと、KEIF フラグがセットされ、KEYVALID はクリアされたままになります。

KMOD[1:0] が 0 と異なる場合、AES によって BUSY フラグが自動的にセットされます。キーが正常にロードされると、BUSY がクリアされ、KEYVALID がセットされます。エラーが発生すると、通常、KEIF がセットされ、BUSY はクリアされ、KEYVALID はクリアされたままになります。

KEIF がセットされている場合、AES_ICR レジスタを通じてクリアする必要があります。クリアしないと、KEYVALID がセットされません。詳細については、KEIF フラグの説明を参照してください。

キーのローディングの詳細については、[セクション 33.4.16 : AES キーレジスタ](#) を参照してください。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **BUSY** : ビジー

このフラグは、AES がアイドルかビジーかを示します。

0 : アイドル

1 : ビジー

AES が無効になっているとき (EN がローのとき)、または最後の処理が完了したときに、アイドルとして AES にフラグが立てられます。

AESがブロックデータを処理しているとき、キーを準備しているとき (ECB または CBC の復号のみ)、あるいは SAES ペリフェラルから共有キーを転送しているとき、AES にビジーとしてフラグが立てられます。

GCM 暗号化が選択されている場合、優先度の高いメッセージを管理するために現在のプロセスを一時停止する前に、このフラグを 0 にする必要があります。

ビット 2 **WRERRF** : 書き込みエラーフラグ

このビットは、AES_DINR レジスタへの予期しない書き込みが発生したときに、セットされます。WRERRF ビットがセットされても、AES の動作に影響はありません。

0 : エラーはありません。

1 : 計算中またはデータ出力フェーズ中に、AES_DINR レジスタへの予期しない書き込みが発生しました。

AES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、AES_ICR レジスタの RWEIF ビットをセットしてクリアします。

ビット 1 **RDERRF** : 読出しエラーフラグ

このビットは、AES_DOUTR レジスタに対する予期しない読出しが発生したときに、セットされます。RDERRF ビットがセットされても、AES の動作に影響はありません。

0 : エラーはありません。

1 : 計算中またはデータ入力フェーズ中に、AES_DOUTR レジスタに対する予期しない読出しが発生しました。

AES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、AES_ICR レジスタの RWEIF ビットをセットしてクリアします。

ビット 0 予約済みであり、リセット値に保持する必要があります。

33.8.3 AES データ入力レジスタ (AES_DINR)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **DIN[31:0]** : データ入力

入力フェーズ中にこのビットフィールドに 4 連続で書き込むことによって、AES ペリフェラルに 16 バイトブロックの入力データすべてを書き込みます。1 回目から 4 回目の書き込みまで、対応するデータの重みは [127:96]、[95:64]、[63:32]、[31:0] となります。書き込みごとに、32 ビット入力バッファのデータが DATATYPE[1:0] ビットフィールドに従ってデータスワッピングブロックによって処理され、AES コアの 16 バイト入力バッファに書き込まれます。

これを読み出すと 0 が返されます。

33.8.4 AES データ出力レジスタ (AES_DOUTR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOUT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **DOUT[31:0]** : データ出力

読出し専用ビットフィールドは、32 ビット出力バッファをフェッチします。計算完了 (CCF フラグセット) 時にこのビットフィールドを 4 連続で読み出すことによって、ペリフェラルから 16 バイトブロックの出力データすべてを実質的に読み出せます。

出力バッファに達する前に、AES コアによって生成されたデータは、

DATATYPE[1:0] ビットフィールドに従ってデータスワッピングブロックによって処理されます。

1 回目から 4 回目までの読出し操作でのデータの重みは、[127:96]、[95:64]、[63:32]、[31:0] となります。

33.8.5 AES キーレジスタ 0 (AES_KEYR0)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[31:0]** : 暗号化キー、ビット [31:0]

これらは、AES_CR レジスタの MODE[1:0] ビットフィールドに応じて、書込み専用ビットフィールドである KEY[255:0] の AES 暗号化または復号キーのうちのビット [31:0] です。

AES が有効になっている (EN ビットがセットされている) 場合、AES_KEYRx レジスタへの書込みは無視されます。キーが SAES ペリフェラルから取得される場合 (KMOD[1:0] が 0x2 の場合) も、キーレジスタへの書込みは無視され、AES_ISR レジスタの KEIF ビットがセットされます。

KMOD[1:0] が 0x0 の場合、特別の書込みシーケンスが必要です。このシーケンスでは、KEYVALID フラグがセットされるシーケンス完了書込みを除き、AES_KEYRx レジスタへ有効な書込みをすると、KEYVALID フラグがクリアされます。AES_SR レジスタの KEYVALID フラグの説明も参照してください。

33.8.6 AES キーレジスタ 1 (AES_KEYR1)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[63:48]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[63:32]** : 暗号化キー、ビット [63:32]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.7 AES キーレジスタ 2 (AES_KEYR2)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[95:80]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[95:64]** : 暗号化キー、ビット [95:64]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.8 AES キーレジスタ 3 (AES_KEYR3)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[127:112]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[127:96]** : 暗号化キー、ビット [127:96]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.9 AES 初期化ベクタレジスタ 0 (AES_IVR0)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[31:0]** : 初期化ベクタ入力、ビット [31:0]

AES_IVRx レジスタには、選択された連鎖モードに応じて、128 ビットの初期化ベクタまたはノンスが格納されます。この値は、各計算ラウンド（該当する場合）後にハードウェアによって更新されます。

このレジスタへの書込みは、EN ビットが AES_SR レジスタでセットされている場合は無視されます。

33.8.10 AES 初期化ベクタレジスタ 1 (AES_IVR1)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[63:48]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[63:32]** : 初期化ベクタ入力、ビット [63:32]

IVI[128:0] ビットフィールドの説明については、AES_IVR0 レジスタを参照してください。

33.8.11 AES 初期化ベクタレジスタ 2 (AES_IVR2)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[95:80]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[95:64]** : 初期化ベクタ入力、ビット [95:64]

IVI[128:0] ビットフィールドの説明については、AES_IVR0 レジスタを参照してください。

33.8.12 AES 初期化ベクタレジスタ 3 (AES_IVR3)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[127:112]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[127:96]** : 初期化ベクタ入力、ビット [127:96]

IVI[128:0] ビットフィールドの説明については、AES_IVR0 レジスタを参照してください。

33.8.13 AES キーレジスタ 4 (AES_KEYR4)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[159:144]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[143:128]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[159:128]** : 暗号化キー、ビット [159:128]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.14 AES キーレジスタ 5 (AES_KEYR5)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[191:176]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[175:160]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[191:160]** : 暗号化キー、ビット [191:160]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.15 AES キーレジスタ 6 (AES_KEYR6)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[223:208]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[207:192]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[223:192]** : 暗号化キー、ビット [223:192]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.16 AES キーレジスタ 7 (AES_KEYR7)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[255:240]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[239:224]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[255:224]** : 暗号化キー、ビット [255:224]

KEY[255:0] ビットフィールドの説明、および AES_KEYRx レジスタへの書込みに関する情報については、AES_KEYR0 レジスタを参照してください。

33.8.17 AES サスペンドレジスタ (AES_SUSPRx)

アドレスオフセット : $0x040 + x * 0x4$ 、($x = 0 \sim 7$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUSP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUSP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GCM、GMAC、または CCM の現在のタスクの処理が、優先順位の高いタスクを処理するためにサスペンドされる場合、AES の内部レジスタのステータスはすべて AES_SUSPRx レジスタ内に保存されます。詳細については、[セクション 33.4.8 : AES のサスペンドおよびレジューム操作](#)を参照してください。

EN ビットが AES_SR レジスタでクリアされている場合は、このレジスタに対する読出しには 0 が返されます。AES_SUSPRx レジスタは、GCM、GMAC、または CCM 以外の連鎖モードでは使用されません。

33.8.18 AES 割込み有効レジスタ (AES_IER)

アドレスオフセット : 0x300

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIE	RWEIE	CCFIE
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEIE** : キーエラー割込みイネーブル

このビットは、KEIF (キーエラーフラグ) がセットされた場合の AES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

ビット 1 **RWEIE** : 読出しまたは書き込みエラー割込み有効

このビットは、RWEIF (読出しおよび/または書き込みエラーフラグ) がセットされた場合の AES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

ビット 0 **CCFIE** : 計算完了フラグ割込み有効

このビットは、CCF (計算完了フラグ) がセットされた場合の AES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

33.8.19 AES割込みステータスレジスタ (AES_ISR)

アドレスオフセット : 0x304

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIF	RWEIF	CCF
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEIF** : キーエラー割込みフラグ

この読み出し専用ビットは、キー情報がキーレジスタにロードできなかったときに、ハードウェアによってセットされます。

0 : キーエラーは検出されていません。

1 : キー情報がキーレジスタにロードできませんでした。

AES_IER レジスタの KEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、AES_ICR レジスタの対応ビットをセットしてクリアします。

KEIF は、次のいずれかのイベント時に立てられます。

–AES_KEYRx レジスタへの書き込みが、次の正しい順序で行われなかった。(KEYSIZE がクリアされている場合は、AES_KEYR0、AES_KEYR1、AES_KEYR2、AES_KEYR3 レジスタの順か、またはその逆。KEYSIZE がセットされている場合は、AES_KEYR0、AES_KEYR1、AES_KEYR2、AES_KEYR3、AES_KEYR4、AES_KEYR5、AES_KEYR6、AES_KEYR7 の順か、またはその逆)。

–AES が、SAES パリフェラルによって共有されるキーをロードできなかった (KMOD[1:0] = 0x2 の場合)。

KEIF は、アプリケーションソフトウェアでクリアする必要があります。クリアしないと、KEYVALID がセットされません。

ビット 1 **RWEIF** : 読み出しまたは書き込みエラー割込みフラグ

この読み出し専用ビットは、AES_SR レジスタの RDERRF または WRERRF エラーフラグがセットされたときに、ハードウェアによってセットされます。

0 : 読み出しまたは書き込みエラーは検出されていません。

1 : 読み出しまたは書き込みエラーが検出されました。

AES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、AES_ICR レジスタの対応ビットをセットしてクリアします。

キー導出モードが選択されている場合、このフラグは意味を持ちません。

詳細については、AES_SR レジスタを参照してください。

ビット 0 **CCF** : 計算完了フラグ

このフラグは計算が完了したかどうかを示します。これは DMAOUTEN ビットがクリアされている場合にのみ重要であり、DMAOUTEN がセットされている場合はハイのままになる可能性があります。

0 : 未完了

1 : 完了

AES_IER レジスタの CCFIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、AES_ICR レジスタの対応ビットをセットしてクリアします。

33.8.20 AES 割込みクリアレジスタ (AES_ICR)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIF	RWEIF	CCF
													w	w	w

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEIF** : キーエラー割込みフラグクリア

このビットをセットすると AES_ISR レジスタの KEIF ステータスビットがクリアされます。

ビット 1 **RWEIF** : 読みまたは書き込みエラー割込みフラグクリア

このビットをセットすると、AES_ISR レジスタの RWEIF ステータスビットがクリアされ、AES_SR レジスタの RDERRF フラグと WRERRF フラグの両方がクリアされます。

ビット 0 **CCF** : 計算完了フラグクリア

このビットをセットすると AES_ISR レジスタの CCF ステータスビットがクリアされます。

33.8.21 AES レジスタマップ

表 312. AESレジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	AES_CR	IPRST	Res.	Res.	Res.	Res.	Res.	KMOD[1]	KMOD[0]	NPBLB[3:0]				Res.	KEYSIZE	Res.	CHMOD[2]	Res.	GCMIPH[1:0]	DMAOUTEN	DMAINEN	Res.	Res.	Res.	Res.	Res.	CHMOD[1:0]	MODE[1:0]	DATATYPE[1:0]		EN		
	リセット値	0						0	0	0	0	0	0		0		0		0	0	0	0					0	0	0	0	0	0	0
0x004	AES_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	Res.	Res.	Res.	BUSY	WRERRF	RDERRF	Res.
	リセット値																								0				0	0	0	0	0
0x008	AES_DINR	DIN[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	AES_DOUTR	DOUT[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	AES_KEYR0	KEY[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	AES_KEYR1	KEY[63:32]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	AES_KEYR2	KEY[95:64]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 312. AESレジスタマップとリセット値 (続き)

[illegible]

表 312. AESレジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x304	AES_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																														0	0	0
0x308	AES_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																														0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

34 セキュアな AES コプロセッサ (SAES)

34.1 概要

セキュアな AES コプロセッサ (SAES) は、NIST によって規定されている高度暗号化標準 (AES : Advanced Encryption Standard) に準拠してデータの暗号化や復号を行います。これには、差動電力解析 (DPA) を含むサイドチャネル攻撃 (SCA) に対する保護、認定済みの SESIP、および PSA セキュリティ保証レベル 3 などが組み込まれています。

SAES では、キー長が 128 ビットまたは 256 ビットの ECB、CBC、CTR、GCM、GMAC、および CCM の連鎖モードに加え、ハードウェア秘密キー暗号化／復号 (ラップキーモード) や高速 AES ペリフェラルとのキー共有 (共有キーモード) などの特別なモードをサポートしています。

SAES には、ハードウェアによって STM32 のハードウェア秘密マスターキー (ブートハードウェアキー BHK および導出ハードウェア固有キー DHUK) をロードできる機能があり、これらのキーはアプリケーションで使用できますが、読み取ることはできません。

このペリフェラルは、入力および出力データの DMA シングル転送をサポートしています (2 つの DMA チャンネルが必要です)。また、真性乱数発生器 (TRNG) および AES ペリフェラルとハードウェア的に連動しています。

34.2 SAES の主な特徴

- NIST FIPS 公報 197 『高度暗号化標準 (Advanced encryption standard (AES))』(2001 年 11 月) に準拠
- 次の複数の連鎖モードで暗号化と復号 :
 - 電子コードブック (ECB) モード
 - 暗号ブロック連鎖 (CBC) モード
 - カウンタ (CTR) モード
 - ガロアカウンタモード (GCM)
 - ガロアメッセージ認証コード (GMAC) モード
 - CBC-MAC 付きカウンタ (CCM) モード
- 差動電力解析 (DPA) を含むサイドチャネル攻撃 (SCA) に対する保護、認定済みの SESIP、および PSA セキュリティ保証レベル 3 など
- 128 ビットおよび 256 ビットの暗号キー長に対応した、128 ビットデータブロックの処理
 - 128 ビットまたは 256 ビットキーを使用して 1 個の 128 ビットブロックを処理する場合は ECB モードでそれぞれ 480 または 680 クロックサイクルの遅延
- ハードウェア秘密キー暗号化/復号 (ラップキーモード)
- 専用のキーバスを使用し、SAES によって制御される高速 AES ペリフェラルとのオプションのキー共有 (共有キーモード)
- ECB/CBC 復号の最終ラウンドキーを計算するための統合キースケジューラ
- 暗号化キーを格納するための 256 ビットの書込み専用レジスタ (32 ビットレジスタ 8 個)
 - XOR 結合できる 2 つのハードウェア秘密キー (BHK、DHUK) を 128 ビットまたは 256 ビットでハードウェアローディングするオプション
- キーのセキュリティコンテキスト強制
- 初期化ベクタ (32 ビットレジスタ 4 個) を格納するための 128 ビットのレジスタ
- データ入力および出力用の 32 ビットバッファ
- 2 つのチャネル (1 チャネルは入力データ用、1 チャネルは処理済みデータ用) を使用したダイレクトメモリアクセス (DMA) のサポートによる自動データフロー制御。シングル転送のみサポートされます。
- 1、8、16、32 ビットデータをサポートするデータスワッピングロジック
- 32 ビットワードのシングルアクセスでのみアクセス可能な AMBA AHB スレーブペリフェラル。他のアクセスタイプは AHB エラーとなります。32 ビット書込み以外では、レジスタの内容が破損する可能性があります。
- SAES がより優先順位の高い別メッセージを処理する必要がある場合は、ソフトウェアでメッセージをサスペンドしてから元のメッセージのレジュームが可能

34.3 SAES の実装

デバイスには、次の表に従って実装された 1 つの SAES ペリフェラルがあります。これは、AES ペリフェラルとキーを共有します。比較のため、AES ペリフェラルも表に含めています。

表 313. AES および SAES の機能

モードまたは機能 ⁽¹⁾	AES	SAES
ECB、CBC 連鎖 (チェーン動作)	X	X
CTR、CCM、GCM 連鎖	X	X
AES 128 ビット ECB 暗号化 (サイクル単位)	51	480
DHUK および BHK のキーの選択		X
サイドチャネル攻撃への耐性	-	X
SAES と AES 間での共有キー	X	
キー長 (ビット単位)	128, 256	128, 256

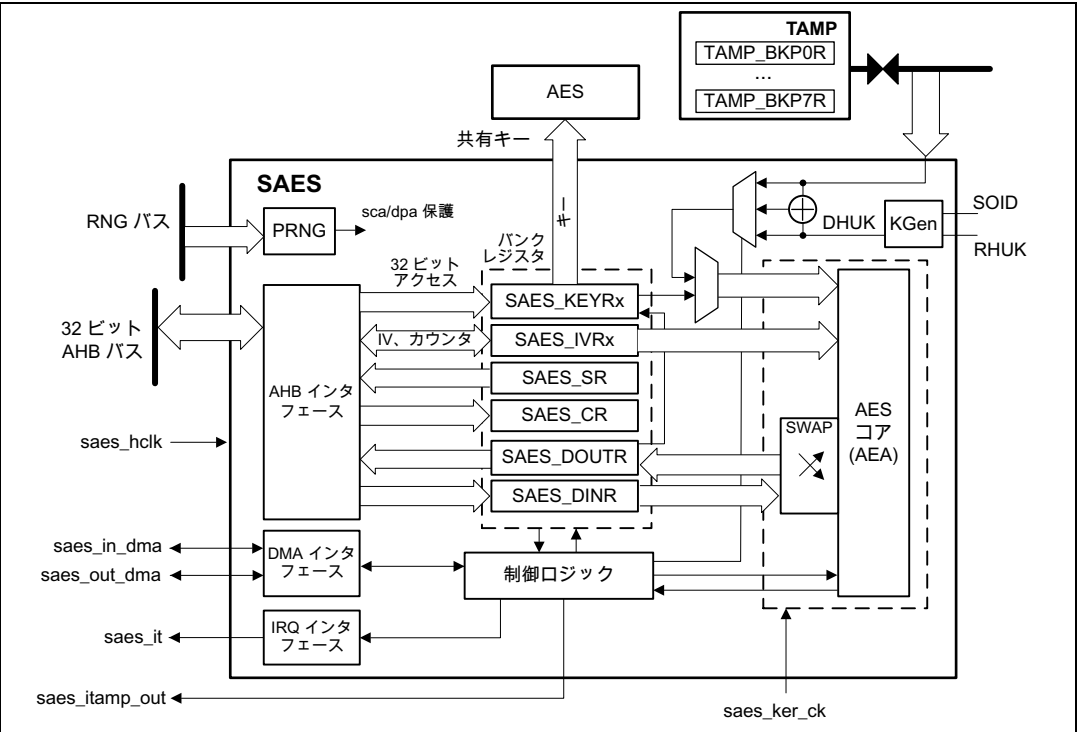
1. X: サポートされています。

34.4 SAES機能詳細

34.4.1 SAES ブロック図

図 308 に SAES のブロック図を示します。

図 308. SAES ブロック図



注: AES は AES ペリフェラルを表わします。

34.4.2 SAES 内部信号

表 314 には、SAES ペリフェラルにインタフェース接続するユーザ関連内部信号を示しています。

表 314. SAES 内部入力／出力信号

信号名	信号タイプ	説明
saes_hclk	入力	AHB バスクロック
saes_ker_ck	入力	SAES カーネルクロック
saes_it	出力	SAES 割込みリクエスト
saes_in_dma	入力／出力	SAES 入力データ DMA シングルリクエスト／確認応答
saes_out_dma	入力／出力	SAES 処理済みデータ DMA シングルリクエスト／確認応答
saes_itamp_out	出力	予期しないハードウェアフォールトが発生したときにトリガされる、TAMP へのタンパイイベント信号 (XOR されます)。この信号がトリガされると、SAES はキーレジスタを自動的にクリアします。SAES を再び使用可能にするには、リセットが必要です。
RHUK	入力	256 ビットのルートハードウェア固有キー（不揮発性、デバイスごとに固有、およびソフトウェアに対して秘密）。導出ハードウェア固有キー（DHUK）を内部で計算するために使用されます。
BHK ⁽¹⁾	入力	改ざん防止のセキュアバックアップレジスタに格納されている 256 ビットのブートハードウェアキー（BHK）で、ブート中にセキュアコードによって書き込まれます。このキーは、一度書き込まれると、製品をリセットするまではアプリケーションから読み書きができなくなります。
SOID	入力	各 DHUK の計算中に使用される静的なオペレーティング ID のハードウェア入力。これを変更すると、キーレジスタが消去され、KEIF フラグがセットされます。

1. TAMP ペリフェラル内の一連のバックアップレジスタに接続されており、アプリケーションソフトウェアによって書き込まれ、その後、読取り／書き込みがロックされます（詳細については [セクション 34.4.17](#) を参照）。

34.4.3 SAES のリセットおよびクロック

SAES ペリフェラルは、AHB バスクロックによってクロック供給されます。SAES インスタンスにも RCC でプログラムされた専用カーネルクロックが備わっています。

SAES には RCC 内に専用のリセットビットがあります。

34.4.4 SAES 対称暗号の実装

セキュア AES コプロセッサ (SAES) は、高度暗号化標準 (AES) を使用して 16 バイトのデータブロックを暗号化または復号する 32 ビット AHB ペリフェラルです。また、[表 315](#) に要約されている承認済みの AES 対称キーセキュリティ機能のセットも実装されています。これらの機能は NIST PUB 140-3 の認定を受けることができます。

表 315. SAES の承認済みの対称キー機能

動作	アルゴリズム	仕様	キービット長	連鎖モード
暗号化、復号	AES	FIPS PUB 197 NIST SP800-38A	128, 256	ECB、CBC、CTR
認証済みの暗号化または復号		NIST SP800-38C NIST SP800-38D		GCM、CCM
暗号ベースのメッセージ認証コード		NIST SP800-38D		GMAC

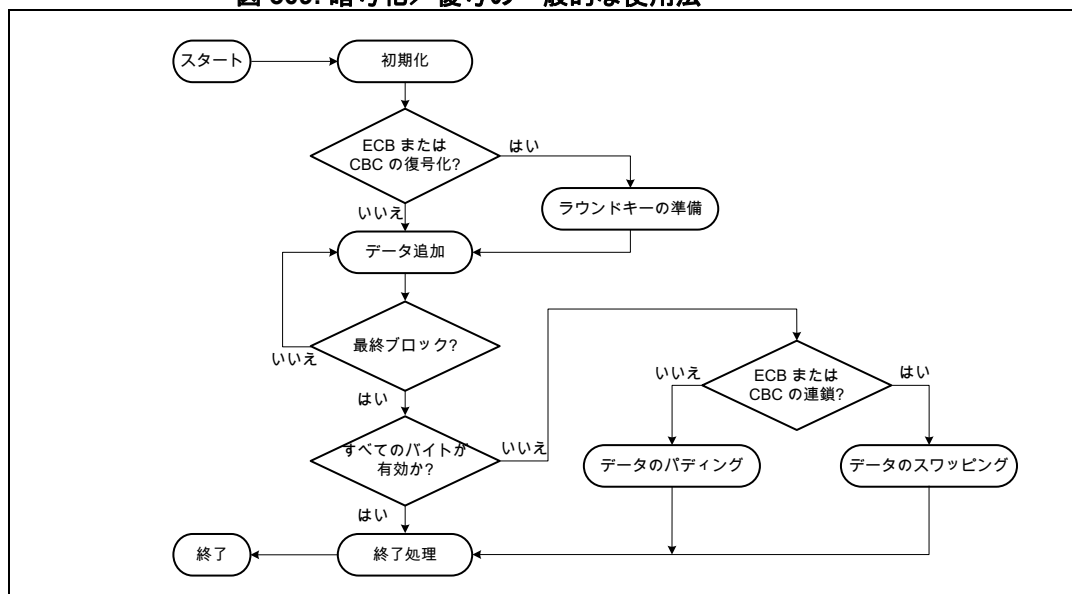
SAES は、CPU によって直接使用することも、2 つの DMA チャンネル（1 つは平文用、もう 1 つは暗号文用）を使用して間接的に使用することもできます。

[セクション 34.4.8](#) で説明しているシーケンスに従って、任意の SAES 処理を一時停止し、その後、再開することができます。

34.4.5 SAES の暗号化または復号の一般的な使用法

暗号化または復号の一般的な動作を以下の図に示します。

図 309. 暗号化／復号の一般的な使用法



初期化

SAES ペリフェラルを連鎖モードに従って初期化します。詳細については、[セクション 34.4.9 : SAES 基本連鎖モード \(ECB、CBC\)](#) および [セクション 34.4.10 : SAES カウンタ \(CTR\) モード](#) を参照してください。

データ追加

本セクションでは処理用のデータを追加するさまざまな方法を説明します。ECB または CBC 連鎖モードでは、処理するデータのサイズが 16 バイトの倍数ではない場合、[セクション 34.4.7 : SAES の暗号文借用およびデータパディング](#)を参照してください。これらの場合の最終ブロックの管理は、このセクションで説明しているものより複雑になります。

CPU をポーリングモードで使したデータの追加

この方法では、フラグポーリングを使用して次のシーケンスでデータ追加を制御します。

1. KEYVALID がセットされているとき、SAES_CR レジスタの EN ビットをセットして、SAES ペリフェラルを有効にします（まだ有効化されていない場合）。
2. ペイロードが完全に処理されるまで次のサブシーケンスを繰り返します。
 - a) SAES_DINR レジスタに 4 つの入力データワードを書き込みます。
 - b) ステータスフラグ CCF が SAES_ISR レジスタでセットされるまで待つて、SAES_DOUTR レジスタから 4 つのデータワードを読み出します。
 - c) SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
 - d) 次の処理ブロックが最後のブロックである場合は、（該当する場合）データを 0 でパディングして完全なブロックを取得し、GCM ペイロード暗号化または CCM ペイロード復号の場合は、無効なバイト数を（NPBLB[3:0] を使用して）指定します（そうしないと、誤ったタグ計算になります）。
3. 処理したばかりのデータブロックはメッセージの最終ブロックであるため、必要に応じてメッセージ/ペイロードの一部ではないデータを破棄してから、EN をクリアして SAES ペリフェラルを無効にします。

注： キーを AES コプロセッサに送信できるようにするため、SAES_DINR レジスタへの連続する 2 つの書き込みの間に最大 3 つのウェイトサイクルが自動的に挿入されます。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

CPU を割り込みモードで使したデータの追加

この方法は、SAES ペリフェラルからの割り込みを使用して次のシーケンスでデータ追加を制御します。

1. SAES_IER レジスタの CCFIE ビットをセットして、SAES からの割り込みを有効にします。
2. KEYVALID がセットされているとき、EN をセットして、SAES ペリフェラルを有効にします（まだ有効化されていない場合）。
3. まず、SAES_DINR レジスタに 4 つの入力データワードを書き込みます。
4. SAES 割り込みサービスルーチンでデータを処理します。割り込みごとに次の手順に従います。
 - a) SAES_DOUTR レジスタから 4 つの出力データワードを読み出します。
 - b) SAES_ICR レジスタの CCF ビットをセットして、CCF フラグ、保留中の割り込みをクリアします。
 - c) 次の処理ブロックがメッセージの最後のブロックである場合は、（該当する場合）データを 0 でパディングして完全なブロックを取得し、GCM ペイロード暗号化または CCM ペイロード復号の場合は、無効なバイト数を（NPBLB[3:0] を通して）指定します（そうしないと、誤ったタグ計算になります）。次に、4e) に進みます。
 - d) 処理したばかりのデータブロックがメッセージの最終ブロックだった場合、必要に応じてメッセージ/ペイロードの一部ではないデータを破棄してから、EN をクリアして SAES ペリフェラルを無効にし、割り込みサービスルーチンを中止します。
 - e) SAES_DINR レジスタに次の 4 つの入力データワードを書き込み、割り込みサービスルーチンを中止します。

注： SAES は連続した読出し操作または書き込み操作の間の遅延を許容するため、2 つの SAES 計算の間に別のペリフェラルからの割り込みがある場合などに受け入れることができます。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

DMA を使用したデータの追加

この方法では、すべての転送と処理が DMA および SAES によって管理されます。次のように実行します：

1. 処理するメッセージの最後のブロックが 16 バイトより短い場合は、ブロックの残りの部分に 0 をパディングして、最後の 4 ワードのデータブロックを準備します。
2. 処理するデータをメモリから SAES ペリフェラル入力へ、処理したデータを SAES ペリフェラル出力からメモリに転送するよう DMA コントローラを設定します（[セクション 34.6 : SAES DMA リクエスト](#)に記載）。転送完了時に割込みを生成するよう DMA コントローラを設定します。GCM のペイロード暗号化または CCM のペイロード復号では、最後の 4 ワードブロックを 0 でパディングしている場合は、DMA 転送にそのブロックを含めないでください。その代わりに、[CPU をポーリングモードで使用したデータの追加](#)に示したシーケンスをこの最終ブロックに使用する必要があります。これは SAES が正しいタグを計算できるように、NPBLB[3:0] ビットフィールドをセットアップしてからブロックを処理する必要があります。
3. KEYVALID がセットされているとき、EN をセットして、SAES ペリフェラルを有効にします（まだ有効化されていない場合）。
4. DMAINEN と DMAOUTEN をセットして、DMA リクエストを有効にします。
5. 転送完了を示す DMA 割込みが発生したら、メモリから SAES 処理済みデータを取得します。

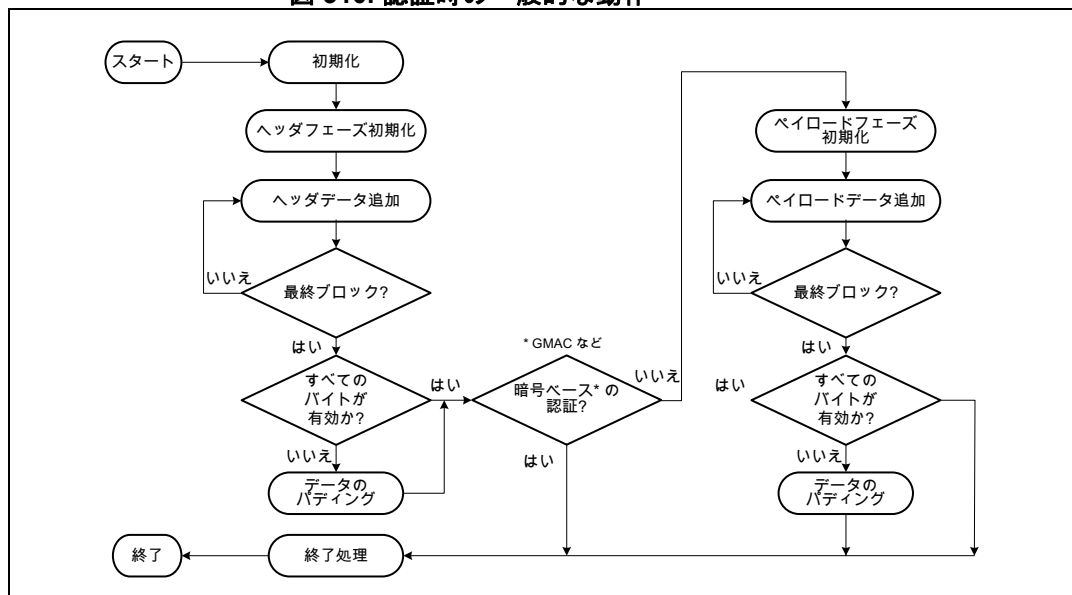
注： SAES_DOUTR レジスタの読出しは計算フェーズの最後にソフトウェア操作なしで自動的に DMA によって管理されるため、この方法では CCF フラグを使用しません。

NPBLB[3:0] ビットフィールドは、GCM、GMAC、CCM の連鎖モードのヘッダフェーズでは使用しません。

34.4.6 SAES の認証済み暗号化、復号、および暗号ベースのメッセージ認証

認証済み暗号化または復号、および暗号ベースのメッセージ認証の一般的な動作を以下の図に示します。

図 310. 認証時の一般的な動作



[セクション 34.4.11 : SAES ガロア／カウンタモード \(GCM\)](#) および [セクション 34.4.13 : SAES の CBC-MAC 付きカウンタ \(CCM\)](#) で、SAES がサポートしているシーケンスの詳細を説明しています。

図に示したように、暗号ベースのメッセージ認証フローではペイロードフェーズが省略されます。SAES でサポートされているシーケンスの詳細は、[セクション 34.4.12 : SAES ガロアメッセージ認証コード \(GMAC : Galois message authentication code\)](#) を参照してください。

34.4.7 SAES の暗号文借用およびデータパディング

サイズがブロックサイズ (16 バイト) の倍数ではないメッセージを管理するために ECB または CBC モードで SAES を使用するときには、アプリケーションでは NIST 特別公報 800-38A『ブロック暗号の推奨動作モード : CBC モード用暗号文借用の 3 つの変種』(Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation) に記載されているもののような、暗号文借用技術を使用します。SAES はこのような技術を実装していないため、アプリケーションは最後から 2 番目のブロックのデータを使用して、入力データの最終ブロックを完成させる必要があります。

注 : 暗号文借用技術は、この参考マニュアルには記載されていません。

同様に、ECB または CBC 以外のモードでは、不完全な入力データブロック (16 バイトより短い入力データを持つブロック) を、暗号化の前に 0 でパディングする必要があります。つまり、データストリングの後端に余分なビットを付加する必要があります。復号後は、余分なビットを破棄する必要があります。SAES では、最終ブロックに対する自動データパディング操作が行われないため、アプリケーションが本書に示された推奨に従って、サイズが 16 ビットの倍数ではないメッセージを管理する必要があります。

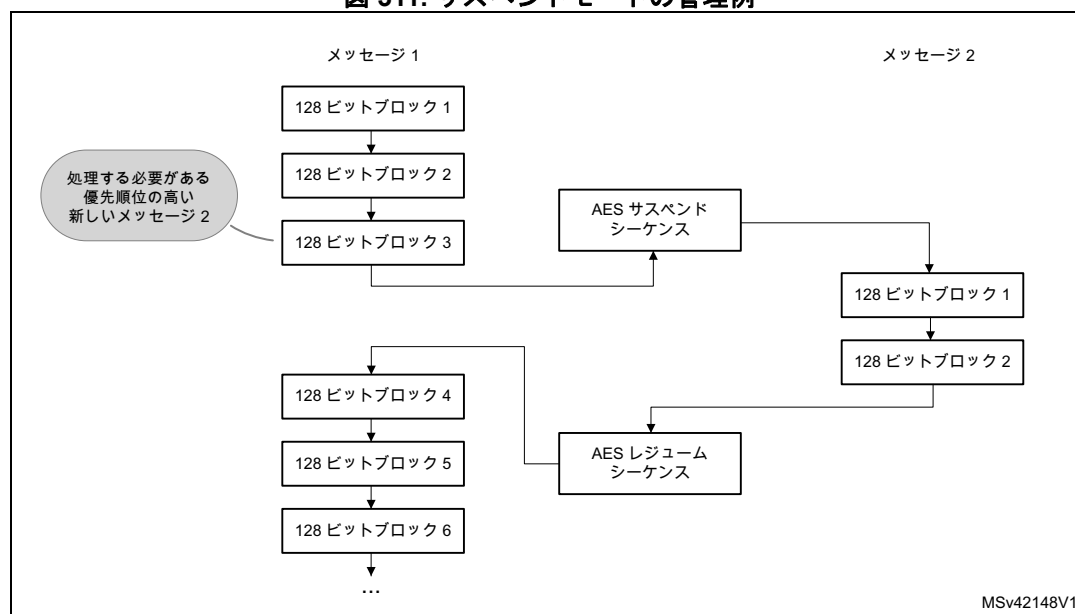
34.4.8 SAES のサスペンドおよびレジューム操作

優先順位の高い別のメッセージを処理するには、メッセージをサスペンドすることが可能です。優先順位の高いメッセージを送信した後、サスペンドしたメッセージを再開できます。これは、暗号化と復号モードの両方に適用されます。

サスペンドおよびレジューム操作を行っても連鎖操作は途切れません。SAES が再び有効になるとすぐにメッセージ処理を再開して、次のデータブロックを受信できます。

[図 311](#) に、サスペンドおよびレジューム操作の例を示します。メッセージ 1 は、より短く優先順位の高いメッセージ 2 を送信するためにサスペンドされています。

図 311. サスペンドモードの管理例



サスペンドおよびレジューム操作の詳細は、各連鎖モード専用のセクションに説明があります。

34.4.9 SAES 基本連鎖モード (ECB、CBC)

ECB は最も単純な動作モードです。連鎖操作も、特別な初期化ステージ也没有。メッセージはブロックに分割され、各ブロックが個別に暗号化または復号されます。ECB で復号するときは、最初のブロックを処理する前に特別なキースケジューリングが必要となります。

図 312 と 図 313 に、電子コードブック (ECB : electronic codebook) の暗号化と復号のそれぞれの連鎖の実行手順を示します。ECB 連鎖モードを選択するには、CHMOD[2:0] に 0x0 を書き込みます。

図 312. ECB 暗号化

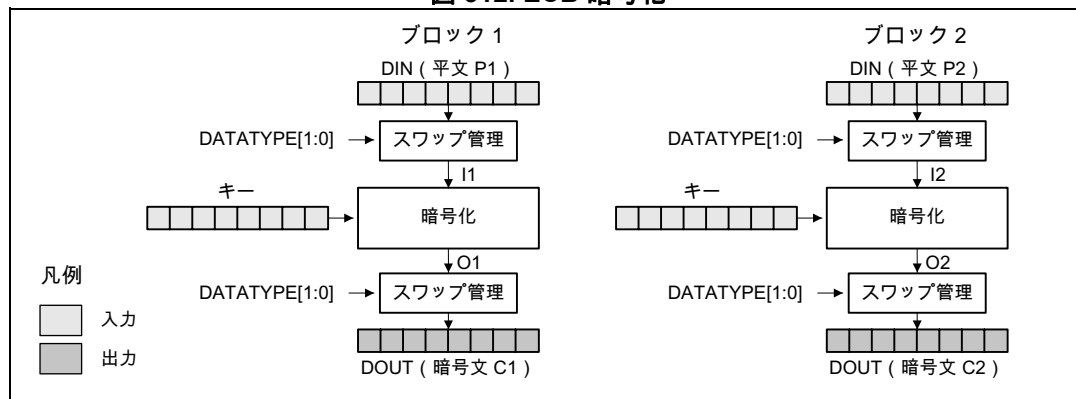
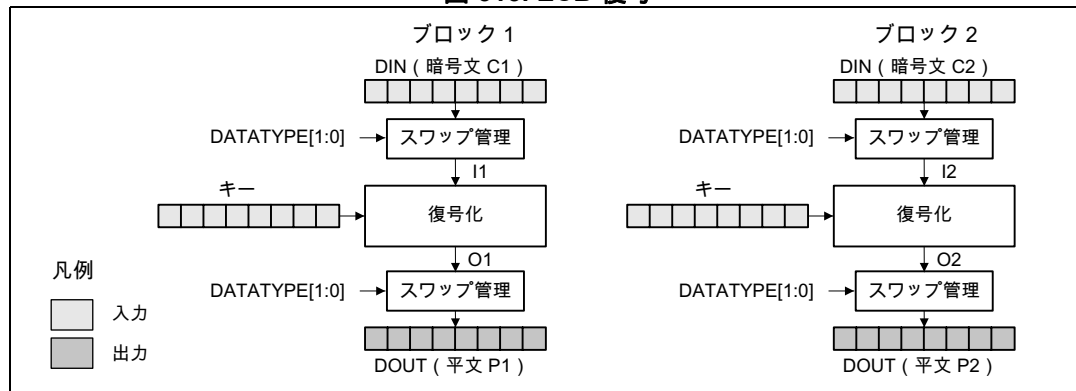


図 313. ECB 復号



CBC 暗号化モードでは、各ブロックの出力が次のブロックの入力で連鎖されます。各メッセージを一意にするために、最初のブロック処理時に初期化ベクタが使用されます。CBC で復号するときは、最初のブロックを処理する前に特別なキースケジューリングが必要となります。

図 314 と 図 315 に、暗号ブロック連鎖 (CBC) の暗号化と復号のそれぞれの連鎖の実行手順を示します。この連鎖モードを選択するには、CHMOD[2:0] に 0x1 を書き込みます。

図 314. CBC 暗号化

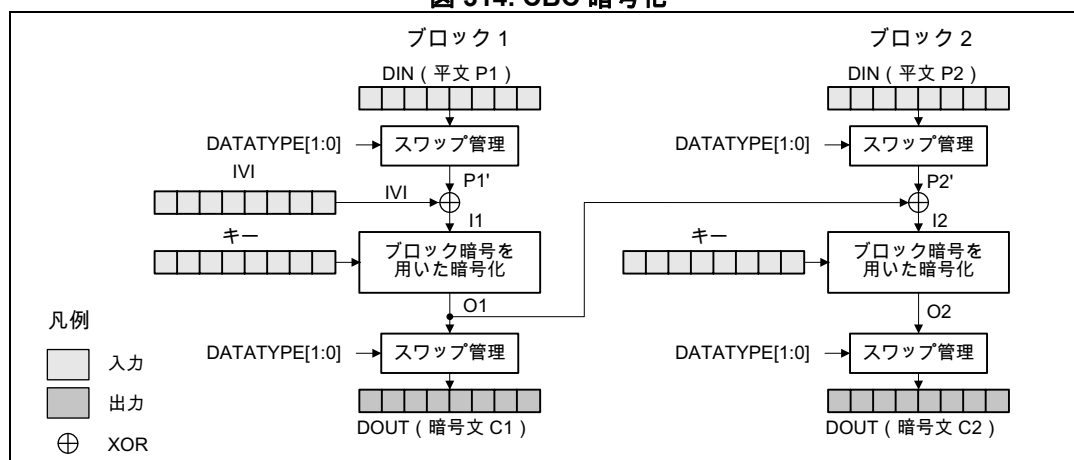
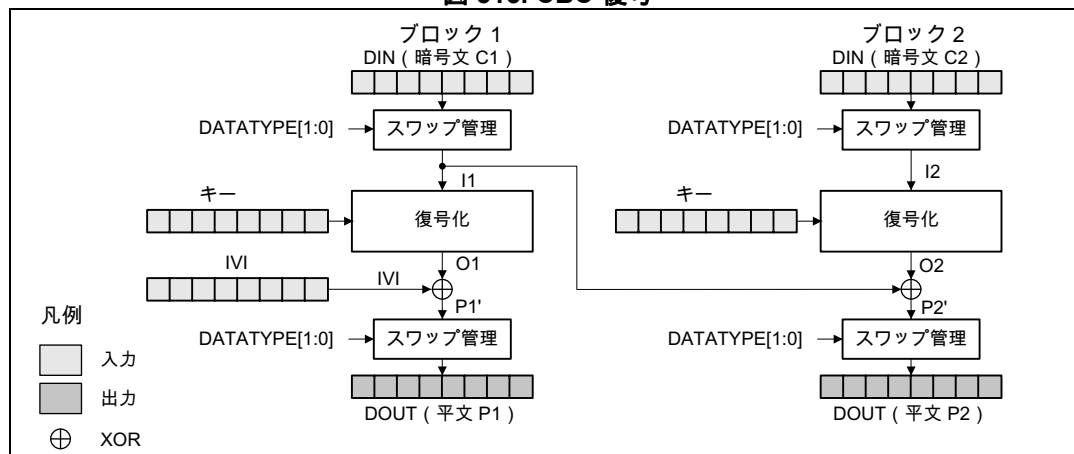


図 315. CBC 復号



詳細については、NIST 特別公報 800-38A『ブロック暗号の推奨動作モード』を参照してください。

ECB および CBC 暗号化プロセス

このプロセスは [セクション 34.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. 次のように SAES_CR レジスタを初期化します。
 - 暗号化モード (MODE[1:0] に 0x0 を書き込む) で、ECB または CBC 連鎖モードを選択します (CHMOD[2:0] に 0x0 または 0x1 を書き込む)。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。キーを別のセキュリティコンテキスト (別のセキュア属性) と共有してはならない場合は、KEYPROT ビットもセットする必要があります。
 - KMOD[1:0] に 0x0 を書き込んで通常のキーモードを選択します。その他の KMOD[1:0] 値については、[セクション 34.4.14](#) (ラップキー) および [セクション 34.4.15](#) (共有キー) を参照してください。

4. 前の手順で CBC モードが選択された場合は、SAES_IVRx レジスタに初期化ベクタを書き込みます。
5. SAES_KEYRx レジスタにキーを書き込みます。あるいは、0x0 とは異なる値を KEYSEL[2:0] に書き込むことによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
6. KEYVALID がセットされる（キーのロードが完了する）まで待ちます。
7. EN を設定して SAES ペリフェラルを有効にします。
8. 平文データを追加します。
 - a) それが最後から 2 番目または最後のブロックであり、メッセージの平文テキストのサイズが 16 バイトの倍数ではない場合は、[セクション 34.4.7](#) のガイダンスに従ってください。
 - b) [セクション 34.4.5](#) の説明に従って平文ブロックを SAES に追加し、SAES_DOUTR レジスタを 4 回読み取って暗号文ブロックを保存します。
 - c) 最後から 3 番目の平文ブロックが暗号化されるまで手順 b) を繰り返します。最後の 2 つのブロックについて、手順 a) と b) を実施します。
9. シーケンスを終了します：EN をクリアして SAES ペリフェラルを無効にします。

ECB/CBC 復号プロセス

このプロセスは [セクション 34.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる（RNG 乱数のフェッチが進行中でなくなる）まで待ちます。
3. 次のように SAES_CR レジスタを初期化します。
 - キー導出モードを選択します（MODE[1:0] に 0x1 を書き込みます）。この操作中は、CHMOD[2:0] ビットフィールドは意味がありません。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。キーを別のセキュリティコンテキスト（別のセキュア属性）と共有してはならない場合は、KEYPROT ビットもセットする必要があります。
 - KMOD[1:0] に 0x0 を書き込んで通常のキーモードを選択します。その他の KMOD[1:0] 値については、[セクション 34.4.14](#)（ラップキー）および [セクション 34.4.15](#)（共有キー）を参照してください。
4. SAES_KEYRx レジスタにキーを書き込みます。あるいは、0x0 とは異なる値を KEYSEL[2:0] に書き込むことによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
5. KEYVALID がセットされる（キーのロードが完了する）まで待ちます。
6. EN を設定して SAES ペリフェラルを有効にします。ペリフェラルは、ただちにキー準備用の AES ラウンドを開始します。
7. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
8. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。復号キーは AES コアで使用可能となり、SAES は自動的に無効化されます。
9. 復号モード（MODE[1:0] に 0x2 を書き込む）で、ECB または CBC 連鎖モードを選択します（CHMOD[2:0] に 0x0 または 0x1 を書き込む）。その他のパラメータは変更しないでください。
10. 前の手順で CBC モードが選択された場合は、SAES_IVRx レジスタに初期化ベクタを書き込みます。
11. EN を設定して SAES ペリフェラルを有効にします。

12. 暗号化データを追加します。
 - a) それが最後から 2 番目または最後のブロックであり、メッセージの暗号文のサイズが 16 バイトの倍数ではない場合は、[セクション 34.4.7](#) のガイダンスに従ってください。
 - b) [セクション 34.4.5](#) の説明に従って暗号文ブロックを SAES に追加し、SAES_DOUTR レジスタを 4 回読み取って平文ブロックを保存します (MSB が先)。
 - c) 最後から 3 番目の暗号文ブロックが復号されるまで手順 [b](#)) を繰り返します。最後の 2 つのブロックについて、手順 [a](#)) と [b](#)) を実施します。
13. シーケンスを終了します: EN をクリアして SAES ペリフェラルを無効にします。

ECB/CBC モードでのサスペンド/レジューム動作

以下のシーケンスは通常のキーモード (KMOD[1:0] が 0x0) で有効です。

メッセージの処理をサスペンドするには、次の手順に従います。

1. DMA が使用されている場合には、SAES_CR レジスタの DMAINEN ビットをクリアして、入力 FIFO への SAES DMA 転送を停止します。
2. DMA が使用されていない場合は、SAES_DOUTR レジスタを 4 回読み出して、最後に処理されたブロックを保存します。DMA が使用されている場合には、SAES_ISR レジスタで CCF フラグがセットされるまで待ち、SAES_CR レジスタの DMAOUTEN ビットをクリアして、出力 FIFO からの DMA 転送を停止します。
3. DMA が使用されていない場合は、SAES_ISR レジスタの CCF フラグがセットされる (計算が完了する) まで待ちます。
4. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
5. 初期化ベクタレジスタを保存します (データ処理中に SAES_IVRx レジスタが変更される CBC モードでのみ必要です)。
6. EN をクリアして SAES ペリフェラルを無効にします。
7. SAES_CR レジスタを保存して、キーレジスタが優先順位の高いメッセージを処理するのに必要でなければ、それをクリアします。
8. DMA が使用されている場合には、DMA コントローラのステータス (SAES の入力および出力データ転送のポインタ、残りバイト数など) を保存します。

メッセージの処理をレジュームするには、次の手順に従います。

1. DMA が使用されている場合には、DMA コントローラを設定して残りの入力 FIFO と出力 FIFO の転送を完了させます。
2. EN をクリアして SAES ペリフェラルを無効にします。
3. SAES_CR レジスタを復元してから (正しい KEYSIZE で)、SAES_KEYRx レジスタを復元します。あるいは、KEYSEL[2:0] を使用して、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17: SAES キーレジスタ](#) を参照してください。
4. [ECB/CBC 復号プロセス](#) に示したように復号キーを準備します (ECB または CBC 復号のみ必要)。
5. 保存した設定を使用して SAES_IVRx レジスタを復元します (CBC モードのみ必要)。
6. EN を設定して SAES ペリフェラルを有効にします。
7. DMA が使用されている場合には、DMAINEN と DMAOUTEN をセットして、SAES DMA 転送を有効にします。

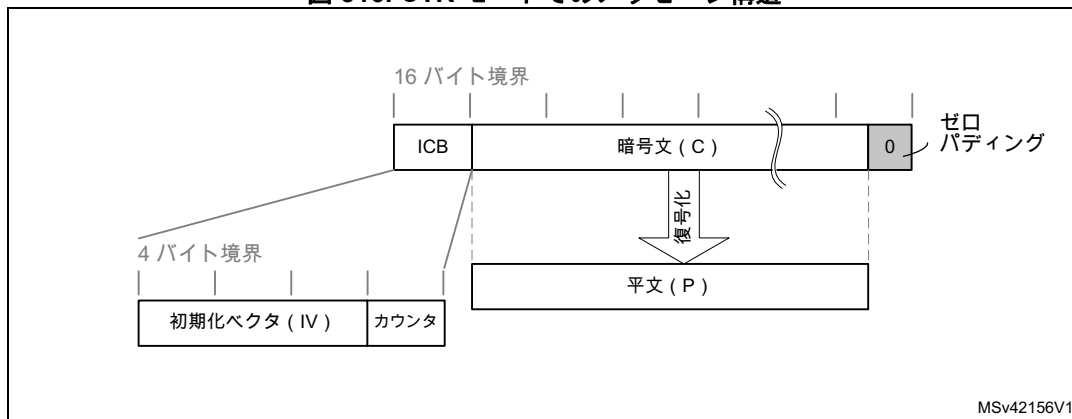
注: アプリケーションには元のキーがわかっているため、キーレジスタを保存する必要はありません。

34.4.10 SAES カウンタ (CTR) モード

CTR モードでは、AES コアを使用してキーストリームを生成します。キーは、その後平文との排他的論理和をとって暗号化されます。AES コアは暗号化モードで必ず使用されるため、ECB モードや CBC モードとは異なり、CTR 復号にはキースケジューリングは必要ありません。

CTR モードでの典型的なメッセージ構造を図 316 に示します。

図 316. CTR モードでのメッセージ構造



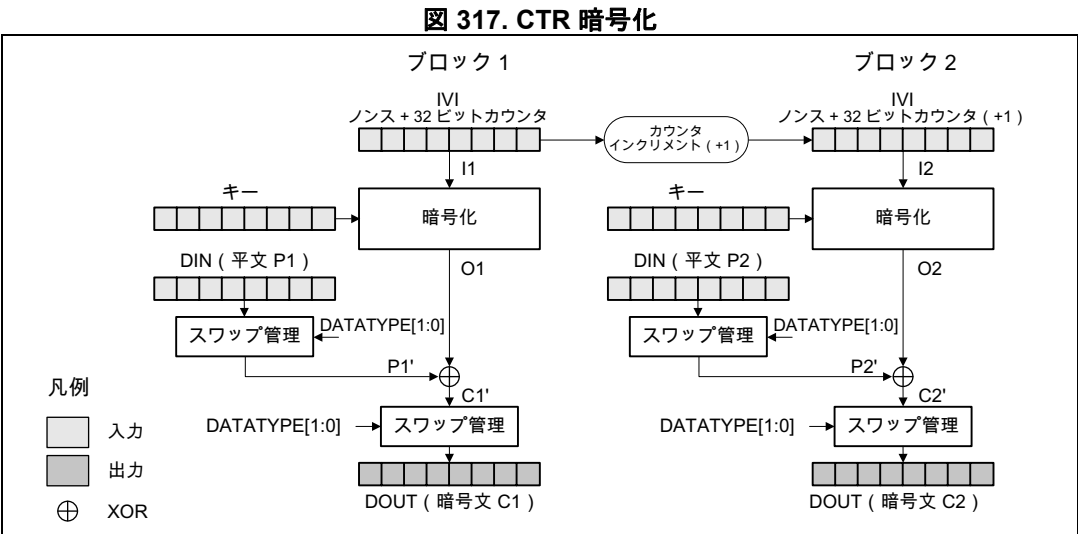
このメッセージの構造は次のとおりです。

- 16 バイトの初期カウンタブロック (ICB) は、次の 2 つの異なるフィールドで構成されています。
 - 初期化ベクタ (IV)** : ある特定のキーでの暗号化サイクルごとに一意でなければならない 96 ビットの値です。
 - カウンタ** : ブロック処理が完了するたびにインクリメントされる 32 ビットのビッグエンディアンの整数です。カウンタの初期値は 1 に設定する必要があります。
- 平文 P は、既知の長さの暗号文 C として暗号化されます。この長さは 16 バイトの倍数でなくとも構いませんが、その場合には平文のパディングが必要です。

詳細については、NIST 特別公報 800-38A 『ブロック暗号の推奨動作モード』を参照してください。

CTR の暗号化と復号

図 317 には、SAES ペリフェラルでのカウンタ (CTR) の連鎖の実行手順 (暗号化) を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x2 を書き込みます。



SAES の初期化ベクタは、表 316 に示すように初期化する必要があります。

表 316. カウンタモードの初期化ベクタの定義

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0] 32 ビットカウンタ = 0x0001

CTR の暗号化と復号のプロセス

このプロセスは [セクション 34.4.5](#) で説明されており、以下の一連のイベントが含まれます。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる（RNG 乱数のフェッチが進行中でなくなる）まで待ちます。
3. SAES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード（MODE[1:0] に 0x0 または 0x2 を書き込む）で、CTR 連鎖モードを選択します（CHMOD[2:0] に 0x2 を書き込む）。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。キーを別のセキュリティコンテキスト（別のセキュア属性）と共有してはならない場合は、KEYPROT ビットもセットする必要があります。
 - KMOD[1:0] に 0x0 を書き込んで、通常のキーモードを選択します。その他の KMOD[1:0] 値については、[セクション 34.4.14](#)（ラップキー）および [セクション 34.4.15](#)（共有キー）を参照してください。
4. [表 316](#) に従って、SAES_IVRx レジスタに初期化ベクタを書き込みます。
5. SAES_KEYRx レジスタにキーを書き込みます。あるいは、0x0 とは異なる値を KEYSEL[2:0] に書き込むことによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
6. KEYVALID がセットされる（キーのロードが完了する）まで待ちます。
7. EN を設定して SAES ペリフェラルを有効にします。

8. データを追加します。
 - a) これが最終ブロックであり、ブロックの平文（暗号化）または暗号文（復号）のサイズが 16 バイトを下回る場合、ブロックの残り部分を 0 でパディングします。
 - b) セクション 34.4.5 の説明に従ってデータブロックを SAES に追加し、SAES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します（MSB が先）。
 - c) 最後から 2 番目のブロックが処理されるまで手順 b) を繰り返します。平文の最終ブロック（暗号化のみ）に対しては、a) および b) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、メッセージの一部ではないビットを破棄します。
9. シーケンスを終了します：EN をクリアして SAES ペリフェラルを無効にします。

CTR モードでのサスペンド／レジューム動作

CBC モードと同様に、メッセージを中断して、優先順位の高いメッセージを送信してから、中断したメッセージを再開することができます。CBC のサスペンドおよびレジュームのシーケンスの詳細については、セクション 34.4.9：SAES 基本連鎖モード（ECB、CBC）を参照してください。

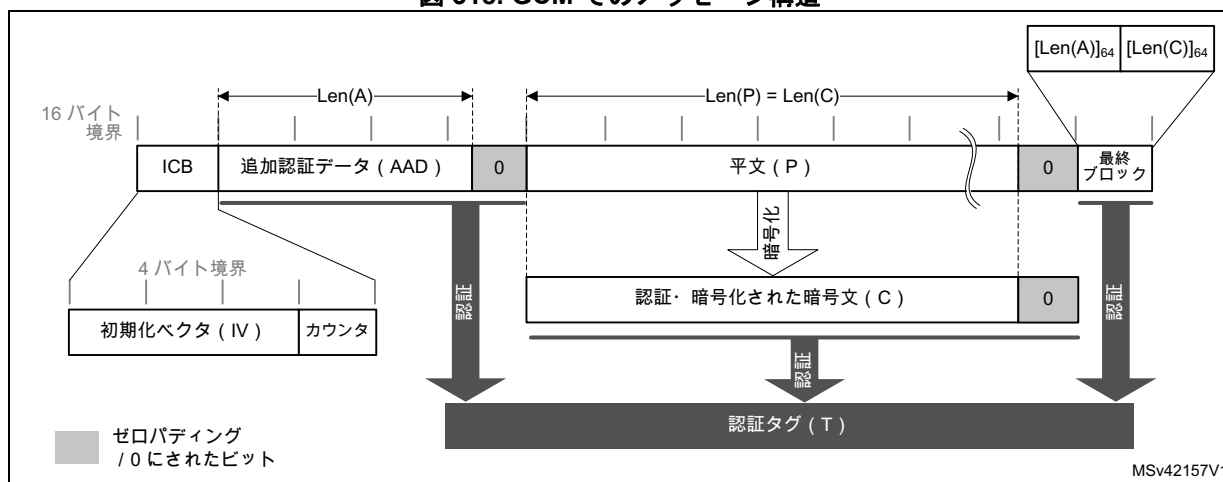
注： CBC モードと同様に、レジューム動作中に IV レジスタを再ロードする必要があります。

34.4.11 SAES ガロア／カウンタモード（GCM）

AES ガロア／カウンタモード（GCM）では、平文メッセージに対応する暗号文およびタグ（メッセージ認証コードとも言います）に暗号化して認証できます。

GCM モードは、機密性については AES カウンタモードに基づいています。固定の有限体に乗算器を使用して、メッセージ認証コードを計算します。GCM モードでの一般的なメッセージ構造を以下の図に示します。

図 318. GCM でのメッセージ構造



メッセージには、次の構造があります。

- **16 バイトの初期カウンタブロック (ICB)** は、次の 2 つの異なるフィールドで構成されています。
 - **初期化ベクタ (IV)**：ある特定のキーでの暗号化サイクルごとに一意でなければならない 96 ビットの値です。GCM 規格では 96 ビット未満の IV に対応していますが、この場合には規則が厳格に適用されます。
 - **カウンタ**：ブロック処理が完了するたびにインクリメントされる 32 ビットのビッグエンディアンの整数です。NIST 仕様によると、ペイロードの最初のブロックを処理するときのカウンタ値は 0x2 です。

- **認証済みヘッダ AAD** (追加認証データともいいます) は、既知の長さである Len(A) を持ちます。この長さは 16 バイトの倍数でなくても構いませんが、 2^{64} ビットを超えてはなりません。メッセージのこの部分には認証だけ行われて、暗号化はされません。
- **平文メッセージ P** は暗号文 C として認証と暗号化の両方が行われ、既知の長さである Len(P) を持ちます。この長さは 16 バイトの倍数でなくても構いませんが、16 バイトの $2^{32}-2$ ブロック分を超えることはできません。
- **最終ブロック** には、表 318 に示すとおり、AAD ヘッダ長 (ビット [32:63]) とペイロード長 (ビット [96:127]) の情報が含まれます。

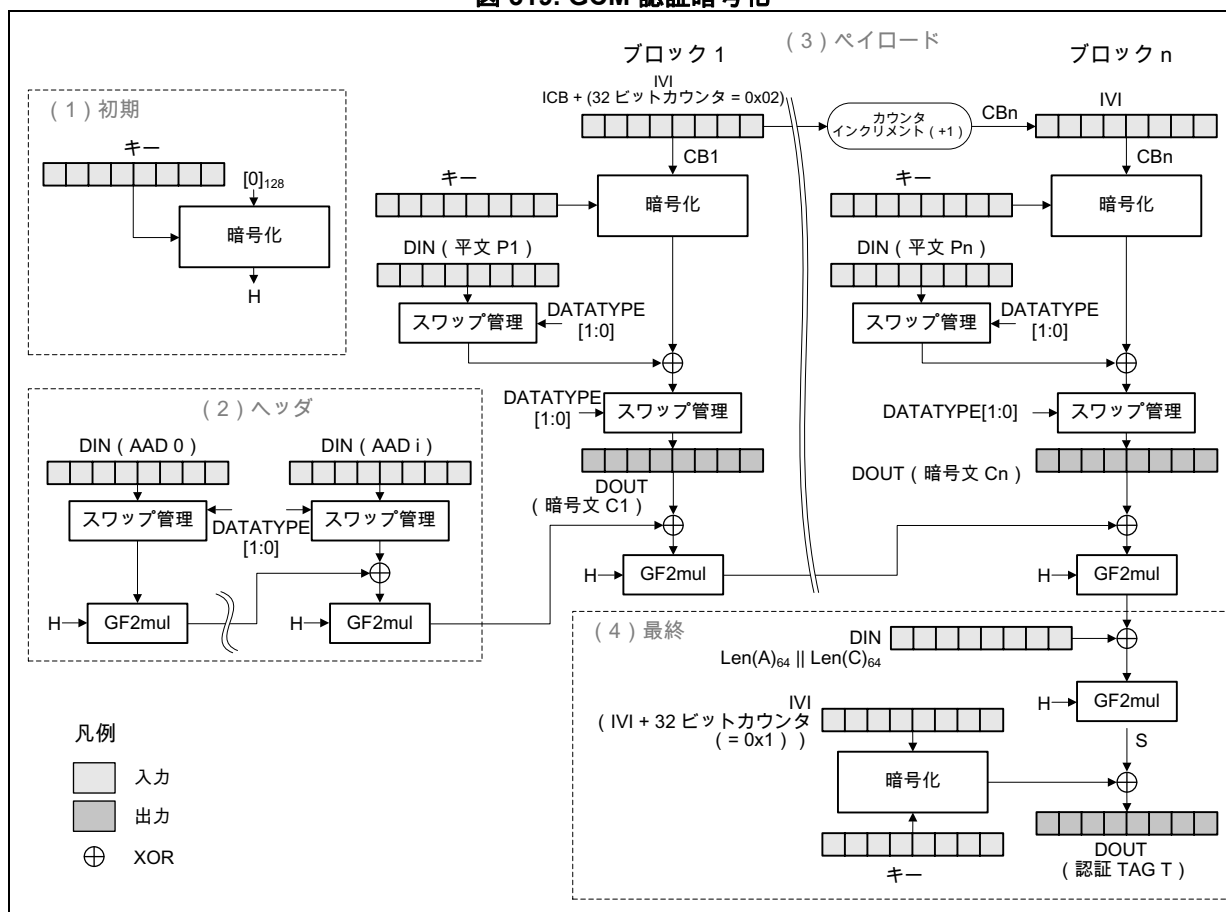
GCM 規格では、暗号文 C は平文 P と同じビット長であることと規定されています。

メッセージ (AAD または P) の長さが 16 バイトの倍数ではない場合には、特別なパディング処理が必要となります。

詳細は、NIST 特別公報 800-38D『ブロック暗号の推奨動作モード - ガロア/カウンタモード (GCM) および GMAC』を参照してください。

図 319 には、SAES ペリフェラルでの GCM の連鎖の実行手順 (暗号化) を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x3 を書き込みます。

図 319. GCM 認証暗号化



最初のカウンタブロック (CB1) は、アプリケーションソフトウェアによって初期カウンタブロック ICB から (表 317 に定義されているとおり) 導出されます。

表 317. GCM モードでの IV レジスタの初期化

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
ICB[127:96]	ICB[95:64]	ICB[63:32]	ICB[31:0] 32 ビットカウンタ = 0x0002

GCM メッセージの最後のブロックには、表 318 に示すように、AAD ヘッダーの長さとペイロードの長さの情報が含まれています。

表 318. GCM 最終ブロックの定義

SAES_DINR への ワード順	最初のワード	2 番目のワード	3 番目のワード	4 番目のワード
入力データ	AAD 長 [63:32]	AAD 長 [31:0]	ペイロード長 [63:32]	ペイロード長 [31:0]

GCM の暗号化と復号のプロセス

このプロセスは [セクション 34.4.6](#) で説明されており、以下の一連のイベントが含まれます。

GCM の初期化

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. SAES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード (MODE[1:0] に 0x0 または 0x2 を書き込む) で、GCM 連鎖モードを選択します (CHMOD[2:0] に 0x3 を書き込む)。MODE[1:0] に 0x1 は書き込まないでください。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。キーを別のセキュリティコンテキスト (セキュア、非セキュア、特定 CPU など) と共有してはならない場合は、KEYPROT もセットします。
 - KMOD[1:0] に 0x0 を書き込んで、通常のキーモードを選択します。その他の KMOD[1:0] 値については、[セクション 34.4.14](#) (ラップキー) および [セクション 34.4.15](#) (共有キー) を参照してください。
 - GCMPPH[1:0] に 0x0 を書き込んで、GCM 初期化フェーズを選択します。
4. [表 317](#) に従って、SAES_IVRx レジスタに初期化ベクタを書き込みます。
5. SAES_KEYRx レジスタにキーを書き込みます。あるいは、0x0 とは異なる値を KEYSEL[2:0] に書き込むことによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
6. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
7. EN をセットしてハッシュキーの計算を開始します。計算が完了すると、EN は自動的にクリアされます。
8. CCF フラグが SAES_ISR レジスタでセットされ、GCM ハッシュサブキー (H) の計算完了が示されるまで待ちます。
9. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。

GCM ヘッダフェーズ

10. ヘッダフェーズを初期化します。
 - a) GCMPH[1:0] に 0x1 を書き込んで、GCM ヘッダフェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) EN を設定して SAES ペリフェラルを有効にします。
11. ヘッダデータを追加します。
 - a) これが最終ブロックであり、ブロックの AAD が 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 34.4.5](#) で説明したように、SAES にデータブロックを追加します。
 - c) 最後から 2 番目の AAD データブロックが処理されるまで、手順 [b\)](#) を繰り返します。最後のブロックに対して、手順 [a\)](#) と [b\)](#) を実施します。

注： このフェーズは、AAD がなければ ($\text{Len(A)} = 0$ であれば) スキップできます。
ヘッダフェーズではデータは読み込まれません。

GCM ペイロードフェーズ

12. ペイロードフェーズを初期化します。
 - a) GCMPH[1:0] に 0x2 を書き込んで、GCM ペイロードフェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) ヘッダフェーズをスキップした場合、EN をセットして、SAES ペリフェラルを有効にします。
13. ペイロードデータを追加します。
 - a) これが最終ブロックであり、ブロックのメッセージが 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 34.4.5](#) の説明に従ってデータブロックを SAES に追加し、SAES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します。
 - c) 最後から 2 番目の平文ブロックが暗号化されるか、暗号文の最終ブロックが復号されるまで手順 [b\)](#) を繰り返します。平文の最終ブロック（暗号化のみ）に対しては、[a\)](#) および [b\)](#) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、ペイロードの一部ではないビットを破棄します。

注： このフェーズは、ペイロードがなければ ($\text{Len(C)} = 0$) スキップできます (GMAC モードを参照)。

GCM の終了処理

14. 暗号化の場合のみ：SAES_SR レジスタの BUSY フラグがクリアされるまで待ちます。
15. GCMPH[1:0] に 0x3 を書き込んで、GCM 最終フェーズを選択します。GCM の初期化中に書き込まれた他の設定は変更しないでください。
16. SAES_DINR レジスタに最終 GCM ブロックを書き込みます。これは、[表 318](#) に示すとおり、AAD ビットとペイロードビットの長さを連結したものです。
17. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
18. SAES_DOUTR レジスタを 4 回読み出して GCM 認証タグを取得します。
19. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
20. EN をクリアして SAES ペリフェラルを無効にします。認証された復号である場合には、生成されたタグをメッセージとともに渡された予測タグと比較します。

注： 最終フェーズでは、データは通常どおり (スワップなしに) SAES_DINR に書き込まれますが、SAES_DOUTR から読み出されるタグデータにはスワップが適用されます。
ヘッダフェーズまたはペイロードフェーズから最終フェーズに遷移する際に、SAES を無効にしないでください。そうしないと、誤った結果になります。

GCM モードでのサスペンド／レジューム動作

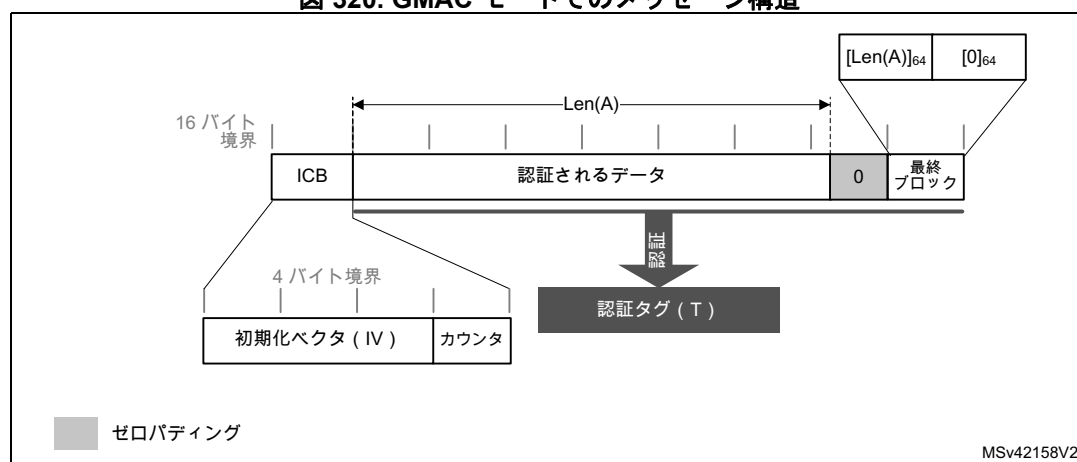
GCM モードではサスペンド／レジューム動作はサポートされていません。

34.4.12 SAES ガロアメッセージ認証コード (GMAC : Galois message authentication code)

ガロアメッセージ認証コード (GMAC) では、平文を認証して、対応するタグ情報 (メッセージ認証コードとも言います) を生成することができます。

平文の認証済みデータのみで構成されたメッセージ (すなわち、ヘッダのみでペイロードなし) に適用されることを除けば、GMAC は GCM と似ています。GMAC での一般的なメッセージ構造を以下の図に示します。

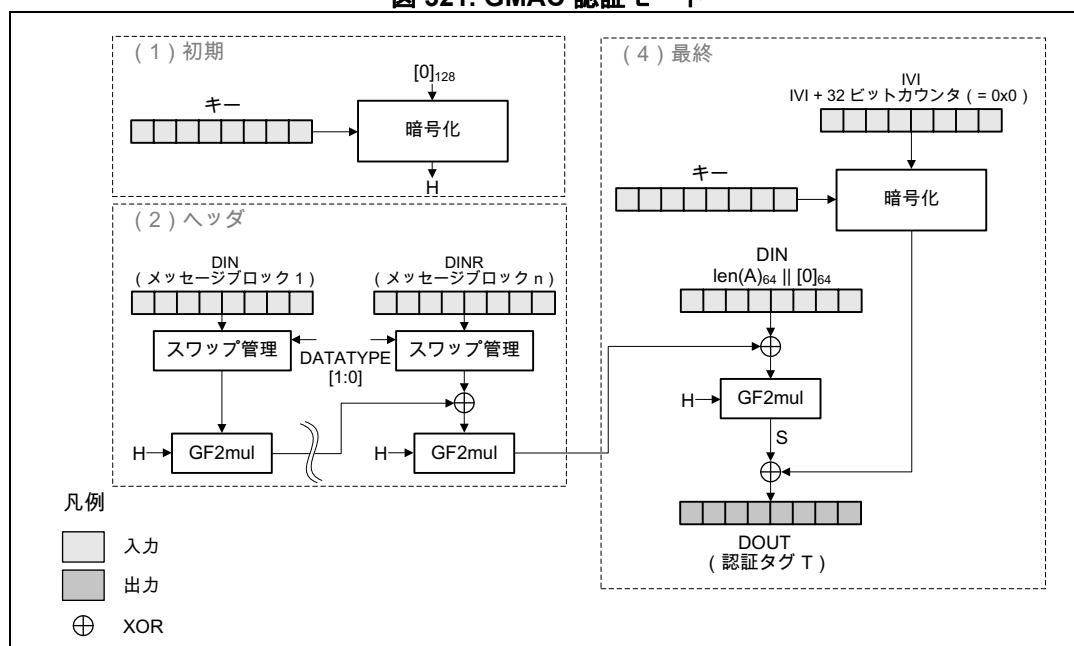
図 320. GMAC モードでのメッセージ構造



詳細は、NIST 特別公報 800-38D『ブロック暗号の推奨動作モード - ガロア／カウンタモード (GCM) および GMAC』を参照してください。

図 321 には、SAES ペリフェラルでの GMAC 連鎖の実行手順を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x3 を書き込みます。

図 321. GMAC 認証モード



GMAC アルゴリズムはヘッダのみ含まれているメッセージに適用される GCM アルゴリズムに相当します。その結果、ペイロードフェーズが省略されること以外の手順と設定がすべて GCM と同じです。

GMAC でのサスペンド／レジューム動作

GMAC モードではサスペンド／レジューム動作はサポートされていません。

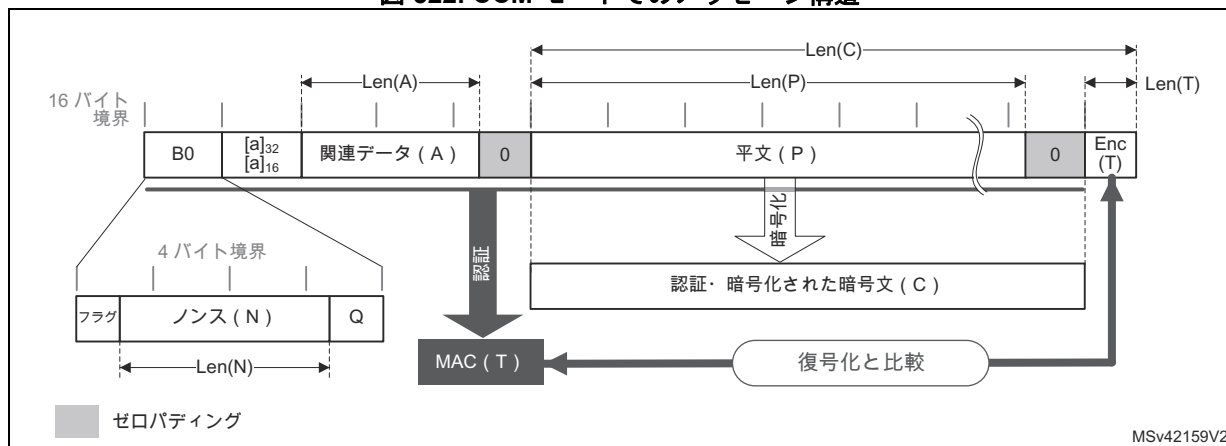
34.4.13 SAES の CBC-MAC 付きカウンタ (CCM)

AES 暗号ブロック連鎖-メッセージ認証コード付きカウンタ (CCM) アルゴリズムでは、平文を暗号化して認証し、対応する暗号文およびタグ (メッセージ認証コードとも言います) を生成することができます。機密性の確保のため、CCM アルゴリズムは AES カウンタモード処理に基づいています。暗号ブロック連鎖技術を使ってメッセージ認証コードを生成します。これは、一般的に CBC-MAC と呼ばれます。

注： NIST は、CCM 仕様に関係以外では CBC-MAC を認証モードとして承認していません。

CCM での一般的なメッセージ構造を以下の図に示します。

図 322. CCM モードでのメッセージ構造



メッセージの構造は次のとおりです。

- 16 バイトの初期認証ブロック (B0)** は、次の 3 つの異なるフィールドで構成されています。
 - Q** : P のオクテット長 (Len(P)) を表すビットストリングです。
 - ノンス (N : Nonce)** : Len(P) サイズの使い捨て値です (新しい通信のたびに新しいノンスを割り当てる必要があります)。Len(N) + Len(P) の合計は、15 バイトにする必要があります。
 - フラグ** : 規格に規定されている、制御情報のための 4 つのフラグを格納する最上位オクテットです。値 **t** (バイト単位での MAC の長さ) と **Q** (Len(P) < 2^{8Q} バイトとなる平文の長さ) をエンコードするための 3 ビットのストリングが 2 つ格納されています。**Q** に関連付けられるカウンタブロックの範囲は 2^{8Q-4} となり、**Q** の最大値が 8 である場合、暗号で使用するカウンタブロックは 60 ビットでなければなりません。
- 16 バイトブロック (B)** は関連データ (A) に関連付けられます。
 メッセージのこの部分には認証だけ行われて、暗号化はされません。この部分は既知の長さである Len(A) を持っており、この長さは 16 バイトの倍数でなくとも構いません (図 322 参照)。規格には、最初のメッセージブロック (B1) の上位側ビットにおいて、バイト数で表現された関連データ長 (a) は次の定義のようにエンコードされなければならないとも示されています。
 - $0 < a < 2^{16} - 2^8$ である場合には、[a]₁₆ のようにエンコードされ、2 バイトとなります。
 - $2^{16} - 2^8 < a < 2^{32}$ である場合には、0xff || 0xfe || [a]₃₂ のようにエンコードされ、6 バイトとなります。
 - $2^{32} < a < 2^{64}$ である場合には、0xff || 0xff || [a]₆₄ のようにエンコードされ、10 バイトとなります。
- 16 バイトブロック (B)** は既知の長さ Len(P) を持ち、暗号文 C として認証と暗号化の両方が行われている平文メッセージ P に関連付けられます。この長さは 16 バイトの倍数でなくとも構いません (図 322 参照)。
- 長さ Len(T) の暗号化された **MAC (T)** が、全長が Len(C) である暗号文 C に追加されます。

メッセージ (A または P) の長さが 16 バイトの倍数ではない場合には、特別なパディング処理が必要となります。

注 : CCM 連鎖モードは、関連データのみ (ペイロードなし) にも使用できます。

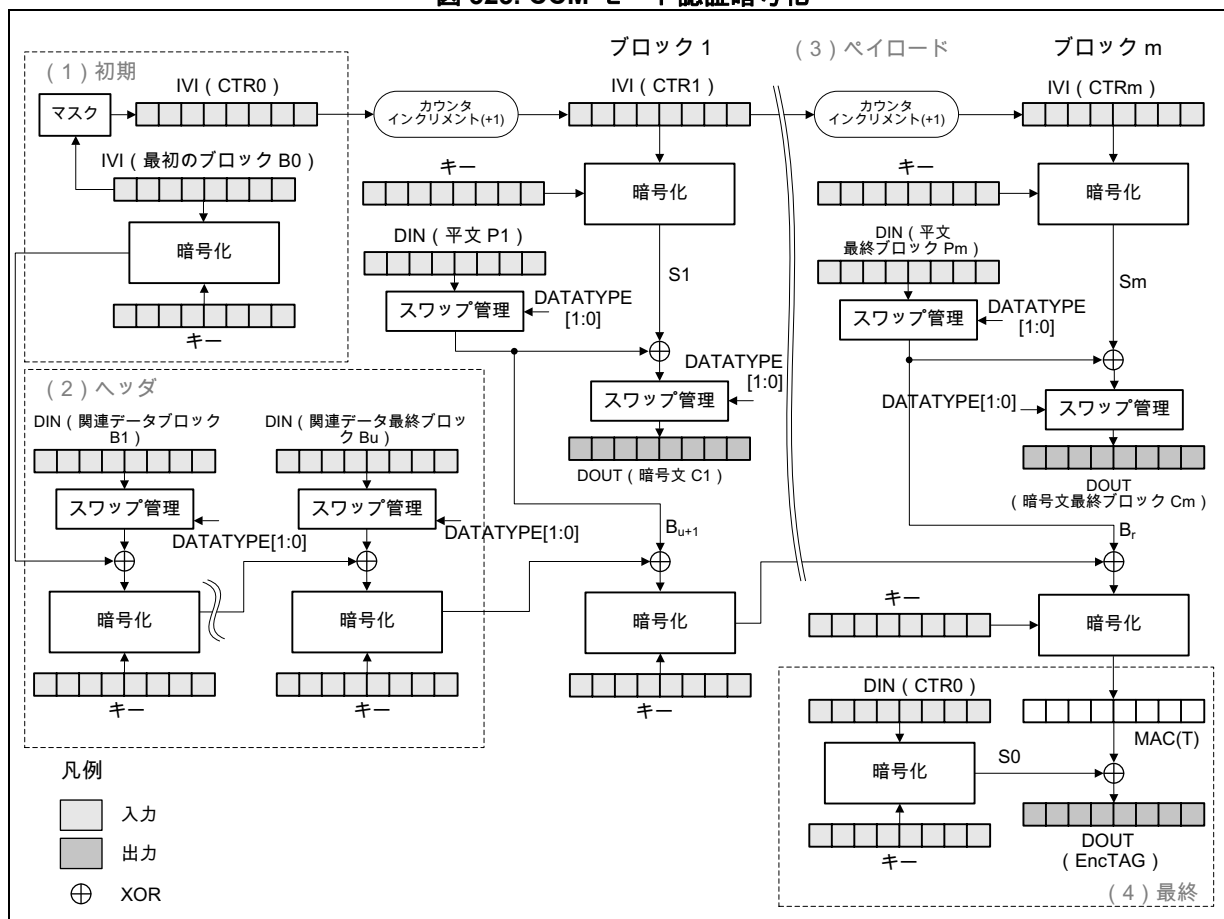
例として、NIST 特別公報 800-38C のセクション C.1 には次の値 (16 進数) が示されています。

N : 10111213 141516 (Len(N) = 56 ビットまたは 7 バイト)
 A : 00010203 04050607 (Len(A) = 64 ビットまたは 8 バイト)
 P : 20212223 (Len(P) = 32 ビットまたは 4 バイト)
 T : 6084341B (Len(T) = 32 ビットまたは t = 4)
 B0 : 4F101112 13141516 00000000 00000004
 B1 : 00080001 02030405 06070000 00000000
 B2 : 20212223 00000000 00000000 00000000
 CTR0 : 0710111213 141516 00000000 00000000
 CTR1 : 0710111213 141516 00000000 00000001

詳細については、NIST 特別公報 800-38C 『ブロック暗号の推奨動作モード - 認証および機密性のための CCM モード』を参照してください。(NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation - The CCM Mode for Authentication and Confidentiality)

図 323 には、SAES ペリフェラルでの CCM の連鎖の実行手順 (暗号化) を示しています。この連鎖モードを選択するには、CHMOD[2:0] に 0x4 を書き込みます。

図 323. CCM モード認証暗号化



CCM メッセージの最初のブロック (B0) は、表 319 で定義されているようにアプリケーションによって準備される必要があります。

表 319. CCM モードでの IV レジスタの初期化

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
B0 [127:96] ⁽¹⁾	B0 [95:64]	B0 [63:32]	B0 [31:0] ⁽²⁾

1. 上位側 5 ビットはクリアされます (フラグビット)。
2. セットされるビット 0 を除く、Q の長さビットはクリアされます。

SAES は、NIST が指定する最大 64 ビットのカウンタをサポートしています。

CCM の暗号化と復号のプロセス

このプロセスは [セクション 34.4.6](#) で説明されており、以下の一連のイベントが含まれます。

CCM の初期化

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. SAES_CR レジスタを初期化します。
 - 暗号化モードまたは復号モード (MODE[1:0] に 0x0 または 0x2 を書き込む) で、CCM 連鎖モードを選択します (CHMOD[2:0] に 0x4 を書き込む)。MODE[1:0] に 0x1 は書き込まないでください。
 - DATATYPE[1:0] でデータタイプを設定します。
 - KEYSIZE でキー長を設定します。キーを別のセキュリティコンテキストと共有してはならない場合は、KEYPROT ビットもセットします。
 - KMOD[1:0] に 0x0 を書き込んで、通常のキーモードを選択します。その他の KMOD[1:0] 値については、[セクション 34.4.14](#) (ラップキー) および [セクション 34.4.15](#) (共有キー) を参照してください。
 - GCMPH[1:0] に 0x0 を書き込んで、CCM 初期化フェーズを選択します。
4. [表 319](#) に従って、SAES_IVRx レジスタに B0 データを書き込みます。
5. SAES_KEYRx レジスタにキーを書き込みます。あるいは、0x0 とは異なる値を KEYSEL[2:0] に書き込むことによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
6. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
7. EN をセットして最初のマスク計算を開始します。計算が完了すると、EN ビットは自動的にクリアされます。
8. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
9. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。

CCM ヘッダフェーズ

10. ヘッダフェーズを初期化します。
 - a) CCM 連鎖ルールに従って、メッセージに関連付けられた (B1) データの最初のブロックを準備します。
 - b) GCMPH[1:0] に 0x1 を書き込んで、CCM ヘッダフェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
 - c) EN を設定して SAES ペリフェラルを有効にします。

11. ヘッダデータを追加します。
 - a) これが最終ブロックであり、ブロックの関連データが 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 34.4.5](#) で説明したように、SAES にデータブロックを追加します。
 - c) 最後から 2 番目の関連データブロックが処理されるまで、手順 [b\)](#) を繰り返します。最後のブロックに対して、手順 [a\)](#) と [b\)](#) を実施します。

注： このフェーズは、関連データがなければ ($\text{Len}(A) = 0$ であれば) スキップできます。
ヘッダフェーズではデータは読み込まれません。

CCM ペイロードフェーズ

12. ペイロードフェーズを初期化します。
 - a) GCMPH[1:0] に 0x2 を書き込んで、CCM ペイロードフェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
 - b) ヘッダフェーズをスキップした場合、EN をセットして、SAES ペリフェラルを有効にします。
13. ペイロードデータを追加します。
 - a) 暗号化においてのみ、これが最終ブロックであり、ブロックの平文が 16 バイトより小さい場合、ブロックの残り部分を 0 でパディングします。
 - b) [セクション 34.4.5](#) の説明に従ってデータブロックを SAES に追加し、SAES_DOUTR レジスタを 4 回読み取って結果ブロックを保存します。
 - c) 最後から 2 番目の平文ブロックが暗号化されるか、暗号文の最終ブロックが復号されるまで手順 [b\)](#) を繰り返します。平文の最終ブロック（暗号化のみ）に対しては、[a\)](#) および [b\)](#) の手順に従います。最終ブロックに対しては、最終ブロックが 16 バイトより小さい場合、ペイロードの一部ではないビットを破棄します。

注： このフェーズは、ペイロードがなければ ($\text{Len}(P) = 0$ または $\text{Len}(C) = \text{Len}(T)$ であれば) スキップできます。

暗号文 C を復号するときには、暗号化されたタグ情報である $\text{LSB}_{\text{Len}(T)}(C)$ を削除します。

CCM の終了処理

14. GCMPH[1:0] に 0x3 を書き込んで、CCM 最終フェーズを選択します。CCM の初期化中に書き込まれた他の設定は変更しないでください。
15. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
16. SAES_DOUTR レジスタを 4 回読み出して CCM 認証タグを取得します。
17. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
18. EN をクリアして SAES ペリフェラルを無効にします。認証された復号である場合には、生成されたタグをメッセージとともに渡された予測タグと比較します。有効なタグを取得するために、タグの長さで認証タグ出力をマスクします。

注： 最終フェーズでは、SAES_DOUTR レジスタから読み出されるタグデータにはスワップが適用されます。
ヘッダフェーズまたはペイロードフェーズから最終フェーズに遷移する際に、SAES を無効にしないでください。そうしないと、誤った結果になります。

CCM モードでのサスペンドおよびレジューム動作

ヘッダフェーズまたはペイロードフェーズでメッセージの処理をサスペンドするには、次の手順に従います。

1. DMA が使用されている場合には、DMAINEN をクリアして、入力 FIFO への SAES DMA 転送を停止します。DMA が使用されていない場合は、SAES_ISR レジスタの CCF フラグがセットされる（計算が完了する）まで待ちます。
2. ペイロードフェーズで、DMA が使用されていない場合は、SAES_DOUTR レジスタを 4 回読み出して、最後に処理されたブロックを保存します。DMA が使用されている場合には、SAES_ISR レジスタで CCF フラグがセットされるまで待ってから、DMAOUTEN をクリアして、出力 FIFO からの DMA 転送を停止します。
3. SAES_ICR レジスタの CCF ビットをセットして、SAES_ISR レジスタの CCF フラグをクリアします。
4. メモリに SAES_SUSPRx レジスタを保存します。
5. データ処理中に変更された IV レジスタを保存します。
6. EN をクリアして SAES ペリフェラルを無効にします。
7. メモリに現在の SAES_CR 設定を保存します。アプリケーションが元のキー値を知っていますので、キーレジスタを保存する必要はありません。
8. DMA が使用されている場合には、DMA コントローラのステータス（SAES 入力データ転送のポインタ、残りバイト数など）を保存します。ペイロードフェーズでは、SAES 出力データ転送のポインタも保存します。

メッセージの処理をレジュームするには、次の手順に従います。

1. DMA が使用されている場合には、DMA コントローラを設定して残りの入力 FIFO の転送を完了させます。ペイロードフェーズでは、残りの出力 FIFO の転送のためにも DMA コントローラを設定します。
2. EN をクリアして SAES ペリフェラルを無効にします。
3. メモリに事前に保存したサスペンドレジスタの値を、対応する SAES_SUSPRx レジスタに書き戻します。
4. 保存されている設定を用いて SAES_IVRx レジスタを復元します。
5. SAES_CR と SAES_KEYRx のレジスタに初期設定値を復元します。あるいは、0x0 とは異なる値を KEYSEL[2:0] にセットすることによって、キーレジスタとは異なるキーソースを選択します。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
6. EN を設定して SAES ペリフェラルを有効にします。
7. DMA が使用されている場合には、DMAINEN（とペイロードフェーズの場合は DMAOUTEN も）をセットして、SAES DMA リクエストを有効にします。

34.4.14 ラップキーを使った SAES の動作

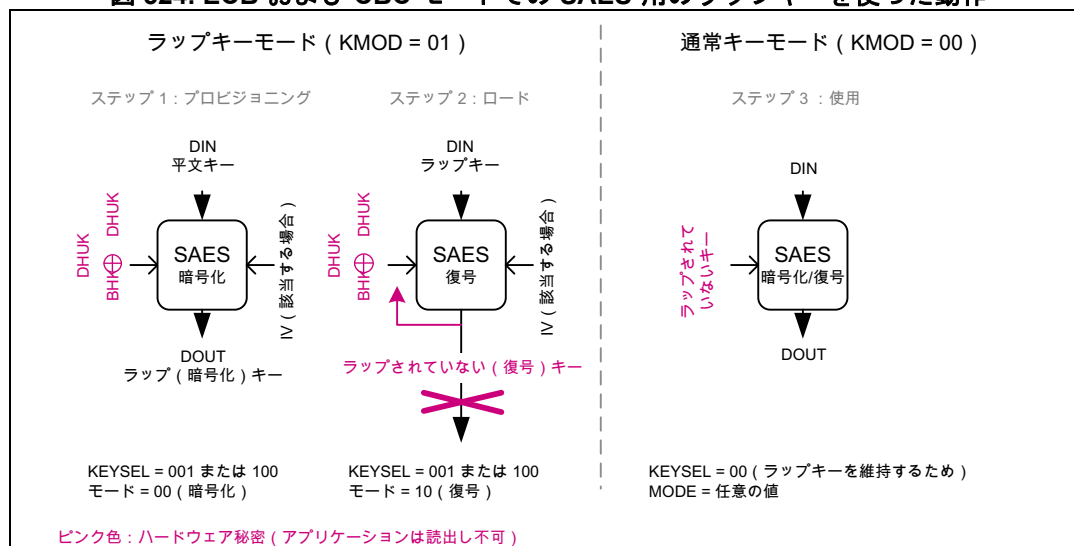
SAES ペリフェラルは、ハードウェア秘密キー DHUK を使用して、アプリケーションキー BHK と XOR 演算して、またはしないで、アプリケーションキーをラップ（暗号化）およびアンラップ（復号）することができます。この機能を使用すると、AES キーを平文（暗号化されていない文）で公開することなく、アプリケーションソフトウェアで使用可能にすることができます。

ラップキーシーケンスは小さすぎるため、サスペンド／レジュームすることができません。SAES では、ラップされていないキーを使ってキーをアンラップすることはできません。

ECB および CBC モードでの SAES 用のラップキーを使った動作

ECB および CBC モードにおいて SAES 用のキーをラップまたはアンラップする方法の概要を図 324 に示します。ラップキーを保護するためには、KEYSEL[2:0] に 0x1 または 0x4 を書き込んで DHUK を選択します。あるいは、対応するレジスタが TAMP ペリフェラルで読取り／書込みロックされている場合は、KEYSEL[2:0] に 0x2 を書き込んで BHK を選択します。

図 324. ECB および CBC モードでの SAES 用のラップキーを使った動作



注： DHUK 値は、特権、KMOD[1:0]、KEYSEL[2:0]、CHMOD[2:0]、および SAES ペリフェラルがセキュアか非セキュアかによって異なります。

SAES のためのキーラッピング

キーをラップする（つまり暗号化する）ための推奨シーケンスは次のとおりです。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. 次のように SAES_CR レジスタを初期化します。
 - 暗号化モード (MODE[1:0] に 0x0 をセット) で、ECB または CBC 連鎖モードを選択します (CHMOD[2:0] に 0x0 または 0x1 を書き込む)。
 - 32 ビットデータタイプを選択します (DATATYPE[1:0] を 0x0 にセット)。
 - KEYSIZE でキー長を設定します。この情報は、暗号化キーと暗号化されるキーの両方に使用されます。
 - KMOD[1:0] に 0x1 を書き込んでラップキーモードを選択します。
4. 前の手順で CBC モードが選択された場合は、SAES_IVRx レジスタに初期化ベクタを書き込みます。
5. KEYSEL[2:0] に 0x1 または 0x4 を書き込んで、DHUK キーソースを選択します。KEYSEL[2:0] が 0x4 の場合の使用についての詳細は、[セクション 34.4.17](#) を参照してください。
6. KEYVALID がセットされる (DHUK のロードが完了する) まで待ちます。
7. EN を設定して SAES ペリフェラルを有効にします。
8. SAES_DINR レジスタに 4 回書き込んで、暗号化するキーを入力します (MSB が先。[1309 ページの表 321](#) を参照)。
9. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。

10. SAES_DOUTR レジスタを 4 回読み出して暗号化されたキー (MSB が先) を取得します。その後、SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。
11. KEYSIZE がセットされている場合は、8. から 10. の手順を繰り返します。
12. EN をクリアして SAES ペリフェラルを無効にします。

注： ラップキーモードでの暗号化は、ECB または CBC が選択されている場合のみサポートされています。

SAES のためのキーアンラッピング

ECB/CBC を使用して、ラップされた (暗号化された) キーをアンラップ (または復号) する推奨シーケンスは次のとおりです。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. 次のように SAES_CR レジスタを初期化します。
 - キー導出モード (MODE[1:0] を 0x1 にセット) で、ラッピングプロセス中に使用された連鎖モードを選択します (CHMOD[2:0] を 0x0 または 0x1 にセット)。
 - 32 ビットデータタイプを選択します (DATATYPE[1:0] を 0x0 にセット)。
 - ラッピングプロセス中に使用されたキー長を KEYSIZE で設定します。この情報は、復号キーと復号されるキーの両方に使用されます。
 - KMOD[1:0] に 0x1 を書き込んでラップキーモードを選択します。
4. KEYSEL[2:0] に 0x1 または 0x4 を書き込んで、DHUK キーソースを選択します。KEYSEL[2:0] が 0x4 の場合の使用についての詳細は、[セクション 34.4.17](#) を参照してください。
5. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
6. SAES_CR で EN ビットをセットしてペリフェラルを有効にします。
7. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
8. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。復号キーは AES コアで使用可能となり、SAES は自動的に無効化されます。
9. 復号モードを選択します (MODE[1:0] を 0x2 にセット)。その他のパラメータは変更しません。
10. 前の手順で CBC モードが選択された場合は、SAES_IVRx レジスタに初期化ベクタを書き込みます。
11. EN を設定して SAES ペリフェラルを有効にします。
12. SAES_DINR レジスタに 4 回書き込んで、復号するキーを入力します (MSB が先。[1309 ページの表 321](#) を参照)。
13. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。その後、SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。SAES_DOUTR を読み出すと 0 が返り、読出しエラー (RDERR) がトリガされます。
14. KEYSIZE がセットされている場合は、12 と 13 の手順を繰り返します。
15. EN をクリアして SAES ペリフェラルを無効にします。

このシーケンスの最後に、復号されたラップキーは、アプリケーションであらゆる AES 操作 (通常のキーモード) にすぐに使用できるようになります。手順 3 で KEYPROT ビットをクリアした場合、復号されたラップキーは、異なるセキュリティコンテキスト (異なるセキュリティまたはコンパートメント ID 属性) で実行されているアプリケーションと共有できます。

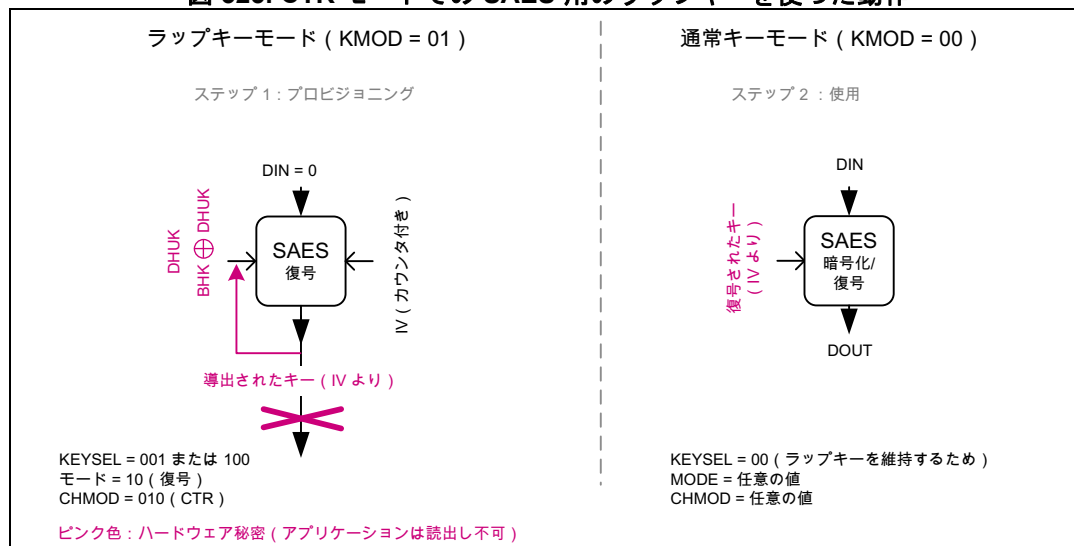
注： KMOD[1:0] = 0x1 (ラップキー) および MODE[1:0] = 0x2 (復号) の場合、SAES_DOUTR レジスタへ読出しアクセスを行うと、読出しエラー (RDERR) がトリガされます。

KEYSEL[2:0] が 0x1 (DHUK) または 0x4 (DHUK XOR BHK) の場合、アプリケーションソフトウェアは暗号化および復号に同じ権限、セキュリティ、KMOD[1:0]、CHMOD[2:0]、および KEYSIZE コンテキストを使用する必要があります。そうしないと、誤った結果になります。

CTR モードでの SAES 用のラップキーを使った動作

CTR モードにおいて SAES 用のキーをアンラップする方法の概要を図 325 に示します。導出キーを保護するためには、KEYSEL[2:0] に 0x1 または 0x4 を書き込んで DHUK を選択します。あるいは、対応するレジスタが TAMP ペリフェラルで読取り／書き込みロックされている場合は、KEYSEL[2:0] に 0x2 を書き込んで BHK を選択します。

図 325. CTR モードでの SAES 用のラップキーを使った動作



注: DHUK 値は、特権、KMOD[1:0]、KEYSEL[2:0]、CHMOD[2:0]、および SAES ペリフェラルがセキュアか非セキュアかによって異なります。

CTR を使用した SAES のラップキーモードの推奨シーケンスは次のとおりです。

1. EN をクリアして SAES ペリフェラルを無効にします。
2. BUSY がクリアされる (RNG 乱数のフェッチが進行中でなくなる) まで待ちます。
3. 次のように SAES_CR レジスタを初期化します。
 - 復号モード (MODE[1:0] を 0x2 にセット) で、CTR 連鎖モードを選択します (CHMOD[2:0] を 0x2 にセット)。他の MODE[1:0] 値はサポートされていません。
 - 32 ビットデータタイプを選択します (DATATYPE[1:0] を 0x0 にセット)。
 - KEYSIZE でキー長を設定します。これは、暗号化キーと共有するキーに使用されます。
 - KMOD[1:0] に 0x1 を書き込んでラップキーモードを選択します。
4. SAES_IVR0 の最下位 2 ビットを 0 のままにして、初期化ベクタを SAES_IVRx レジスタに書き込みます。
5. KEYSEL[2:0] に 0x1 または 0x4 を書き込んで、DHUK キーソースを選択します。KEYSEL[2:0] が 0x4 の場合の使用についての詳細は、[セクション 34.4.17](#) を参照してください。
6. KEYVALID がセットされる (キーのロードが完了する) まで待ちます。
7. EN を設定して SAES ペリフェラルを有効にします。
8. SAES_ISR レジスタの CCF フラグがセットされるまで待ちます。
9. SAES_ICR レジスタの CCF ビットをセットして CCF フラグをクリアします。導出されたハードウェア秘密キーが SAES_KEYRx レジスタで使用可能になります。
10. KEYSIZE がセットされている場合は、[8](#) と [9](#) の手順を繰り返します。
11. EN をクリアして SAES ペリフェラルを無効にします。

このシーケンスの最後に、SAES_IVRx レジスタ内の公開データから導出されたハードウェア秘密キーは、アプリケーションであらゆる AES 操作（通常のキーモード）にすぐに使用できるようになります。

注： KMOD[1:0] を 0x1（ラップキー）に、CHMOD[2:0] を 0x2（CTR 連鎖）に、MODE を 0x0（暗号化）に設定すると、SAES_CR レジスタの EN ビットが自動的にクリアされてペリフェラルが無効になります。

34.4.15 共有キーを使った SAES の動作

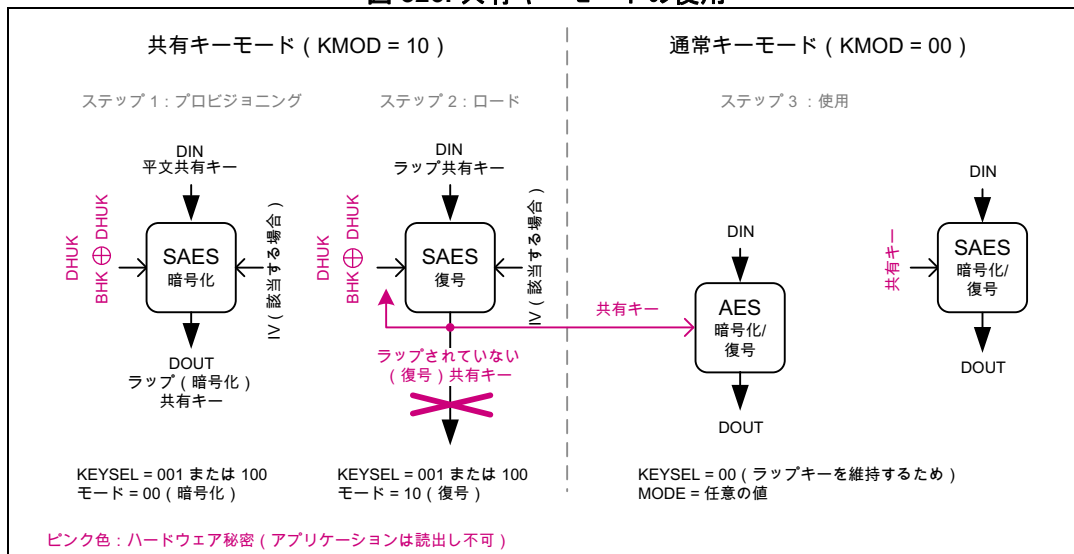
SAES ペリフェラルは、アプリケーションキー BHK と XOR 演算された、またはされない、ハードウェア秘密キー DHUK を使用してラップされたアプリケーションキーを共有することができます。この機能を使用すると、アプリケーションソフトウェアは AES キーを平文（暗号化されていない文）で公開することなく、AES ペリフェラルで 使用できるようにすることができます。

共有キーシーケンスは小さすぎるため、サスペンド/レジュームすることができません。SAES では、ラップされていないキーを使って共有キーをアンラップすることはできません。

注： SAES に保存されているキーが AES と共有されると、KEYPROT ビットによる保護が失われます。保護については、[セクション 34.4.17 : SAES キーレジスタ](#)で詳細に説明されています。

AES ペリフェラルと共有するためにキーをラップまたはアンラップする方法の概要を図 326 に示します。共有キーを保護するためには、KEYSEL[2:0] に 0x1 または 0x4 を書き込んで DHUK を選択する必要があります。あるいは、対応するレジスタが TAMP ペリフェラルで読取り/書込みロックされている場合は、KEYSEL[2:0] に 0x2 を書き込んで BHK を選択します。

図 326. 共有キーモードの使用



注： DHUK 値は、特権、KMOD[1:0]、KSHAREID、KEYSEL[2:0]、CHMOD[2:0]、および SAES ペリフェラルがセキュアか非セキュアかによって異なります。

手順 3 で、AES は AES ペリフェラルを表わします。

AES ペリフェラルのためのキーラッピング

SAES が AES ペリフェラルとキーを共有できるようにするには、キーを一度暗号化（ラップ）する必要があります。共有キーの暗号化シーケンスはラップキーの場合と同じで、[図 326](#) の手順 3 で KMOD[1:0] を 0x2（共有キー）にセットし、KSHAREID[1:0] を 0x0 に維持します。詳細については、[SAES のためのキーラッピング](#)を参照してください。

注： 共有キーモードでの暗号化は、ECB または CBC が選択されている場合のみサポートされています。

AES ペリフェラルのためのキーのアンラッピング（共有キー）

SAES で、AES ペリフェラルとキーを共有する必要があるたびに、共有暗号化キーを SAES で復号（アンラップ）し、AES でロードしなければなりません。全体のシーケンスは、次の項に記載しています。

SAES ペリフェラルでのシーケンス

共有キーの復号シーケンスはラップキーの場合と同じで、[図 326](#) の手順 3 で KMOD[1:0] を 0x2（共有キー）にセットし、KSHAREID[1:0] を 0x0 に維持します。詳細については、[SAES のためのキーアンラッピング](#)を参照してください。

共有キーモードで、復号モードが選択された場合（MODE[1:0] が 0x2 の場合）、SAES_DOUTR レジスタへ読出しアクセスを行うと、読出しエラー（RDERR）がトリガされます。

注： KEYSEL[2:0] ビットフィールドは自動的にクリアされるため、SAES では、復号された共有キーを共有する代わりに直接使用できます。この場合、KMOD[1:0] には 0x0（通常のキーモード）を書き込む必要があります。

AES ペリフェラルでのシーケンス

共有キーが SAES キーレジスタで復号されると、SAES ペリフェラルがキー共有状態、つまり KMOD[1:0] が 0x2 で KEYVALID がセットされた状態のままであれば、復号されたキーは AES ペリフェラルと共有できます。キー共有対象ペリフェラル AESでのシーケンスは、本書の該当セクションの「セキュア AES コプロセッサとの AES のキー共有」で説明されています。SAES ペリフェラルが未使用であり、正しくキー共有状態にある限り、このシーケンスは複数回（たとえば、サスペンド／レジューム状況を管理するため）実行できます。

注： KMOD[1:0] が 0x2 で AES ペリフェラルに BUSY がセットされていて、AES と SAESの KEYSIZE 値が異なる場合、キー共有は失敗し、両方のペリフェラルで KEIF フラグが立てられます。

KEYSEL[2:0] が 0x1（DHUK）または 0x4（DHUK XOR BHK）の場合、アプリケーションソフトウェアは暗号化および復号に同じ権限、セキュリティ、KMOD[1:0] / KSHAREID[1:0]、CHMOD[2:0]、および KEYSIZE コンテキストを使用する必要があります。そうしないと、誤った結果になります。

34.4.16 SAES データレジスタおよびデータスワッピング

データの入出力

16 バイトのデータブロックは、SAES_DINR レジスタ（ビットフィールド DIN[31:0]）に 4 つの 32 ビットワードを最上位ワード（ビット [127:96]）から最下位ワード（ビット [31:0]）まで連続で書き込むことで、SAES ペリフェラルに入力されます。

16 バイトのデータブロックは、SAES_DOUTR レジスタ（ビットフィールド DOUT[31:0]）から 4 つの 32 ビットワードを最上位ワード（ビット [127:96]）から最下位ワード（ビット [31:0]）まで連続で読み出すことで、SAES ペリフェラルから取得されます。

16 バイトのデータブロックの 4 つの 32 ビットワードは、ビッグエンディアン順に（最上位ワードを一番小さいアドレスに）連続してメモリに格納する必要があります。詳細については、表 320 の「スワッピングなし」オプションを参照してください。

データスワッピング

SAES ペリフェラルは、AES 処理コアに読み込まれる前の SAES_DINR レジスタの入力データワード、および SAES_DOUTR レジスタに送られる前の AES 処理コアから出力されたデータについて、ビット、バイト、ハーフワードのスワッピングまたはスワッピングなしに設定できます。データのタイプによって選択します。たとえば、ASCII テキストストリームの場合、バイトスワッピングを使用します。

データスワッピングのタイプは、DATATYPE[1:0] で選択します。この選択は SAES の入力と出力の両方に適用されます。

注： SAES キーレジスタ (SAES_KEYRx) と初期化ベクタレジスタ (SAES_IVRx) のデータは、選択されたスワップモードの影響を受けません。

SAES のデータスワッピング機能の要約を、表 320 と 図 327 に示します。

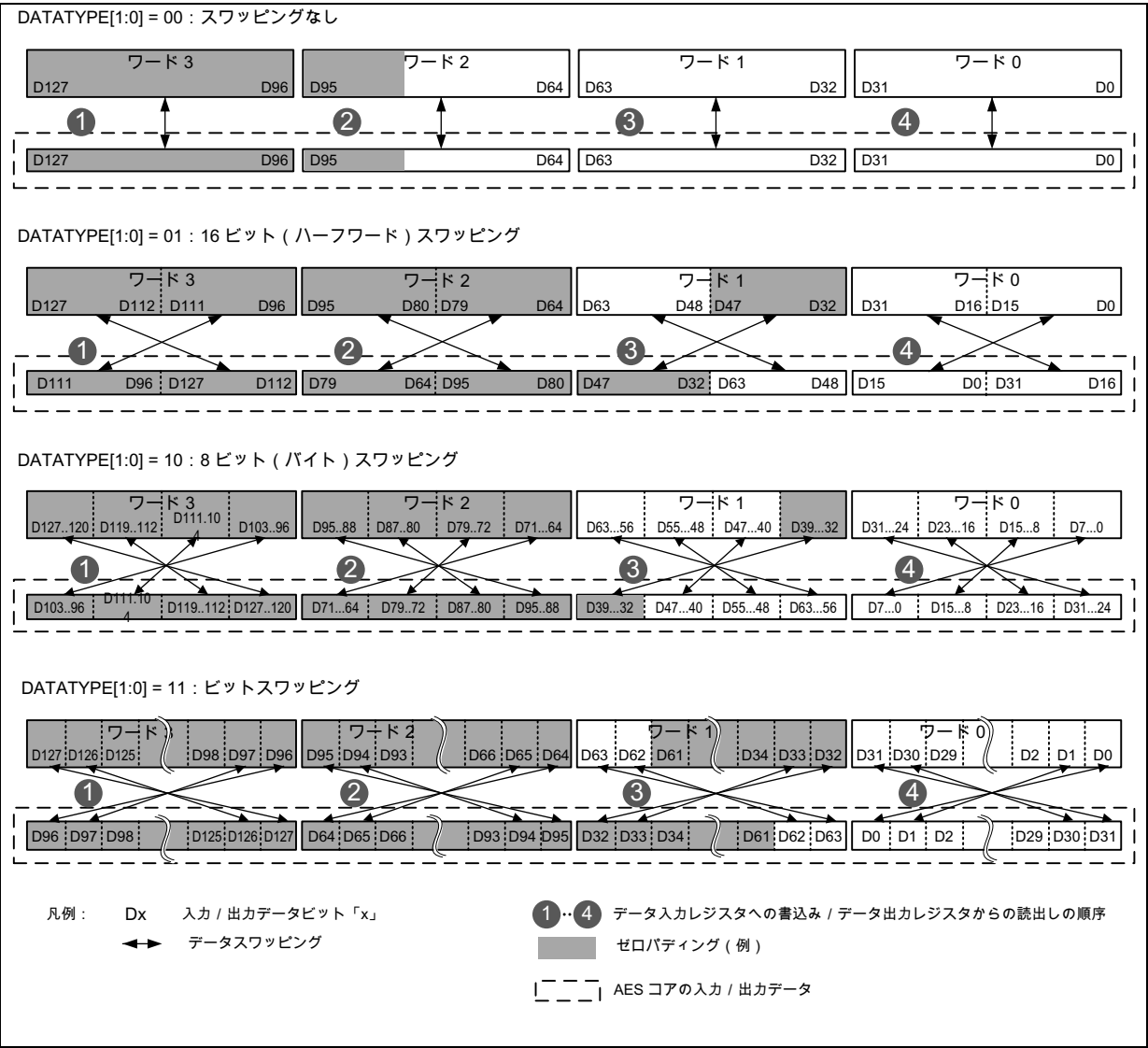
表 320. AES データスワッピングの例

DATATYPE[1:0]	実施される スワッピング処理	データブロック
		システムメモリデータ（ビッグエンディアン）
0x0	スワッピングなし	Block[127..64] : 0x04EEF672 : 2E04CE96
		Block[63..0] : 0x4E6F7720 69732074
		Address @、word[127..96] : 0x04EEF672
		Address @ + 0x4、word[95..64] : 0x2E04CE96
0x1	ハーフワード (16 ビット) スワッピング	Address @ + 0x8、word[63..32] : 0x4E6F7720
		Address @ + 0xC、word[31..0] : 0x69732074
		Block[63..0] : 0x4E6F 7720 6973 2074
		Address @、word[63..32] : 0x7720 4E6F
0x2	バイト (8 ビット) スワッピング	Address @ + 0x4、word[31..0] : 0x2074 6973
		Block[63..0] : 0x4E 6F 77 20 69 73 20 74
		Address @、word[63..32] : 0x2077 6F4E
		Address @ + 0x4、word[31..0] : 0x7420 7369

表 320. AES データスワッピングの例 (続き)

DATATYPE[1:0]	実施される スワッピング処理	データブロック
		システムメモリデータ (ビッグエンディアン)
0x3	ビット スワッピング	Block[63..32] : 0x4E6F7720 0100 1110 0110 1111 0111 0111 0010 0000
		Block[31..0] : 0x69732074 0110 1001 0111 0011 0010 0000 0111 0100
		Address @、word[63..32] : 0x04EE : F672 0000 0100 1110 1110 1111 0110 0111 0010
		Address @ + 0x4、word[31..0] : 0x2E04 : CE96 0010 1110 0000 0100 1100 1110 1001 0110

図 327. データ型に応じた 128 ビットブロックの構成



データパディング

図 327 では、データスワッピング後の 0 にされたビットによって AES コア入力バッファの最上位ビット側に隣接したゾーンが形成される場合など、0 でパディングされるメモリデータブロックの例も示しています。この例では、次を含む入力データブロックのパディングが示されています。

- 84 メッセージビット (DATATYPE[1:0] = 0x0)
- 48 メッセージビット (DATATYPE[1:0] = 0x1)
- 56 メッセージビット (DATATYPE[1:0] = 0x2)
- 34 メッセージビット (DATATYPE[1:0] = 0x3)

34.4.17 SAES キーレジスタ

8 つの SAES_KEYRx 書き込み専用レジスタには、表 321 に示すように、暗号化キーまたは復号キーの情報が保存されます。読出しはセキュリティ上の理由で許されません。

注： メモリおよび SAES キーレジスタでは、キーはリトルエンディアン形式で格納され、最上位バイトが最上位アドレスに配置されます。

表 321. SAES_KEYRx レジスタでのキーエンディアン (128/256 ビットのキー)

SAES_KEYR7 [31:0]	SAES_KEYR6 [31:0]	SAES_KEYR5 [31:0]	SAES_KEYR4 [31:0]	SAES_KEYR3 [31:0]	SAES_KEYR2 [31:0]	SAES_KEYR1 [31:0]	SAES_KEYR0 [31:0]
-	-	-	-	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
KEY[255:224]	KEY[223:192]	KEY[191:160]	KEY[159:128]	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
TAMP_BKP7R [31:0]	TAMP_BKP6R [31:0]	TAMP_BKP5R [31:0]	TAMP_BKP4R [31:0]	TAMP_BKP3R [31:0]	TAMP_BKP2R [31:0]	TAMP_BKP1R [31:0]	TAMP_BKP0R [31:0]

キーレジスタは、DATATYPE[1:0] ビットフィールドで制御されるデータスワッピング機能による影響を受けません。

SAES ペリフェラルが有効 (EN ビットセット) になっていて、KEYSEL[2:0] が 0 と異なる場合、SAES_KEYRx レジスタへの書き込み操作は無視されます。アプリケーションでは、キーレジスタを修正する前にこれを確認する必要があります。

キー全体は AES の計算を開始する前に書き込む必要があります。通常のキーモード (KMOD[1:0] が 0x0 の場合) で、KEYSEL[2:0] が 0x0 の場合、キーレジスタは常に昇順または降順で書き込まれる必要があります。書き込みシーケンスは次のようになります。

- KEYSIZE がクリアされている場合、SAES_KEYRx (x = 0 から 3 または x = 3 から 0)
- KEYSIZE がセットされている場合、SAES_KEYRx (x = 0 から 7 または x = 7 から 0)

注： KEYSIZE は、キーを書き込む前に書き込む必要があります。

最初のキーレジスタが書き込まれると直ちに KEYVALID フラグがクリアされます。キーレジスタの書き込みシーケンスが完了すると、KEYVALID がセットされ、EN が書き込めるようになります。エラーが発生すると、KEYVALID はクリアされ、KEIF がセットされます (セクション 34.4.19 を参照)。

キーの選択

KEYSEL[2:0] が 0x0 の場合、アプリケーションはキーを SAES_KEYRx レジスタに書き込む必要があります。

KEYSEL[2:0] が 0x1 の場合、不揮発性の秘密のルートハードウェア固有キーから SAES 内部で計算された導出ハードウェア固有キー (DHUK) が、KEYSIZE 情報に基づいてキーレジスタに直接ロード

されます。キー導出機能のおかげで、セキュアな SAES はセキュアな DHUK を使用し、非セキュアな SAES は非セキュアな DHUK を使用します。

KEYSEL[2:0] が 0x2 の場合、耐タンパー性のあるセキュアバックアップレジスタに格納されているブートハードウェアキー (BHK) は、セキュアなアプリケーションがすべての TAMP_BKPxR レジスタ (KEYSIZE がクリアされている場合は $x = 0 \sim 3$ 、KEYSIZE がセットされている場合は $x = 0 \sim 7$) を昇順または降順で 1 回読み取ることによって、キーレジスタに完全に転送されます。表 321 を参照してください。

KEYSEL[2:0] が 0x4 の場合、DHUK と BHK の XOR 結合は、セキュアなアプリケーションがすべての TAMP_BKPxR レジスタ (KEYSIZE がクリアされている場合は $x = 0 \sim 3$ 、KEYSIZE がセットされている場合は $x = 0 \sim 7$) を昇順または降順で 1 回読み取ることによって、キーレジスタに完全に転送されます。表 321 を参照してください。

KEYVALID がセットされている場合、同じ 0 以外の値で KEYSEL[2:0] を繰り返し書き込むと、DHUK または BHK のロードのみがトリガされます。KEYVALID をクリアするための推奨方法は、IPRST をセットすることです。このような方法は、たとえば、同じ BHK を選択して (KEYSEL[2:0] を 0x2 に設定)、ECB 復号から ECB 暗号化に切り替える場合に必要です。すべての KEYSEL[2:0] 値について、キーのローディングシーケンスを開始すると、BUSY フラグが設定され、KEYVALID フラグがクリアされます。

KEYSIZE で定義されたビット量が SAES_KEYRx レジスタに転送されると、BUSY がクリアされ、KEYVALID がセットされ、EN ビットが書き込めるようになります。エラーが発生すると、BUSY および KEYVALID はクリアされ、KEIF がセットされます (セクション 34.4.19 を参照)。

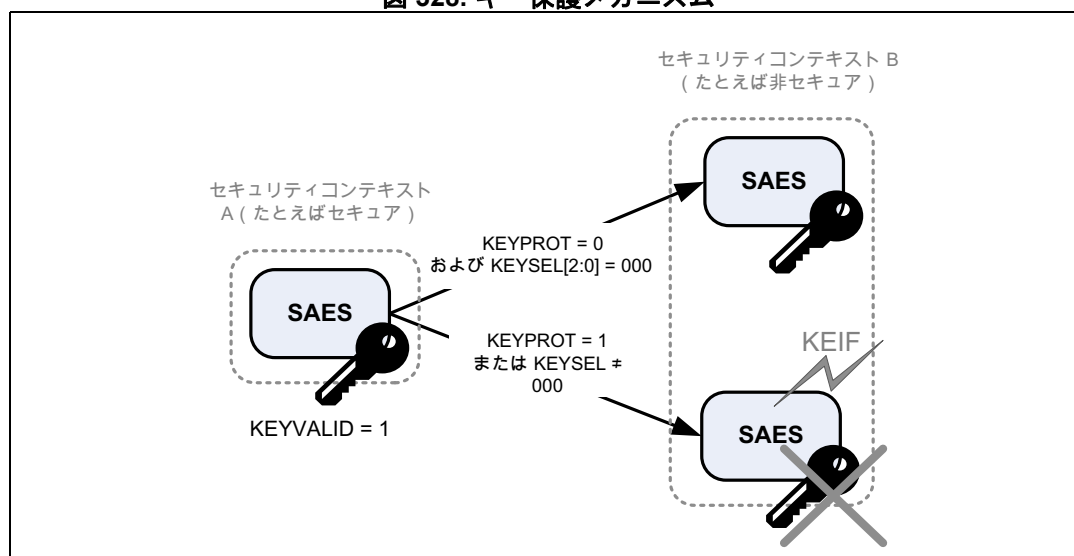
注: DHUK、BHK、およびそれらの XOR 結合は、どのソフトウェアでも (セキュアであっても) 読み取ることができません。

キーの保護

図 328 に示すように、アプリケーションが SAES_KEYRx にキーを書き込む前に KEYPROT ビットを設定すると、異なるセキュリティコンテキスト (つまり、異なるセキュリティ属性) で動作しているアプリケーションが、KEYVALID が設定されている SAES レジスタへアクセスすると、KEIF エラーフラグがトリガされます。

注: KEYSEL[2:0] 値が 0 (通常のキー) と異なる場合、キーレジスタは自動的に保護されます。

図 328. キー保護メカニズム



34.4.18 SAES 初期化ベクタレジスタ

4 つの SAES_IVRx レジスタには、表 322 に示されているように初期化ベクタ (IV) 情報が格納されます。これらは、SAES ペリフェラルが無効 (EN クリア) の場合にのみ書込みできます。

注： メモリおよび SAES IV レジスタでは、初期化ベクタはリトルエンディアン形式で格納され、最上位バイトが最上位アドレスに配置されます。

表 322. SAES_IVRx レジスタ上に配置される IVI ビットフィールド

SAES_IVR3[31:0]	SAES_IVR2[31:0]	SAES_IVR1[31:0]	SAES_IVR0[31:0]
IVI[127:96]	IVI[95:64]	IVI[63:32]	IVI[31:0]

初期化ベクタ情報は、選択した連鎖モードによって異なります。SAES_IVRx レジスタは、使用されると、AES 計算サイクルごとに更新されます (サスペンドモードの管理に役立ちます)。

初期化ベクタレジスタは、DATATYPE[1:0] で制御されるデータスワッピング機能による影響を受けません。

34.4.19 SAES エラー管理

このセクションに記載するエラーは、SAES ペリフェラルによって管理されます。

読出しエラーフラグ (RDERRF)

SAES_DOUTR レジスタの予期しない読出し試行があると 0 が返され、RDERRF フラグと RWEIF フラグがセットされます。RDERRF は、計算フェーズまたは入力フェーズ中にトリガされます。

注： 特に指定がない限り、RDERRF が立っても SAES は無効にならず、処理を続けます。

RWEIE ビットがセットされていた場合、割込みが生成されます。詳細については、[セクション 34.5: SAES 割込み](#) を参照してください。

SAES_ICR レジスタの RWEIF ビットをセットすることで、RDERRF と RWEIF フラグはクリアされます。

書込みエラーフラグ (WDERR)

SAES_DINR レジスタの予期しない書込み試行は無視され、WRERRF フラグと RWEIF フラグがセットされます。WRERRF は、計算フェーズまたは出力フェーズ中にトリガされます。

注： 特に指定がない限り、WRERRF が立っても SAES は無効にならず、処理を続けます。

RWEIE ビットがセットされていた場合、割込みが生成されます。詳細については、[セクション 34.5: SAES 割込み](#) を参照してください。

SAES_ICR レジスタの RWEIF ビットをセットすることで、WRERRF と RWEIF フラグはクリアされます。

キーエラー割込みフラグ (KEIF)

SAES_ISR レジスタの KEIF フラグをセットし、SAES_SR レジスタの KEYVALID ビットをクリアするエラーの原因は複数あります。

- **キー保護エラー**: KEYVALID がセットされているときに、KEYPROT がセットされているか、または KEYSEL[2:0] が 0 と異なる場合、キーのロードに使用したセキュリティコンテキストとは異なるセキュリティコンテキスト（つまり、異なるセキュリティ属性）で動作しているアプリケーションが SAES にアクセスしたときに、このエラーがトリガされます。
- **キー書き込みシーケンスエラー**: キーレジスタの書き込みシーケンスの間違いが検出されると、トリガされます。詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。
- **キー共有サイズの不一致エラー**: KMOD[1:0] が 0x2 で、AES ペリフェラルの KEYSIZE が SAES ペリフェラルの KEYSIZE と一致しない場合にトリガされます。
- **キー共有エラー**: SAES 共有キーの AES ペリフェラルへの転送に失敗すると、トリガされます。詳細については、[セクション 34.4.15 : 共有キーを使った SAES の動作](#)を参照してください。
- **ハードウェア秘密キーローディングエラー**: DHUK または BHK の SAES へのロードに失敗したときにトリガされます。KEYSEL[2:0] の 0x1 (DHUK) 設定、0x2 (BHK) 設定 または 0x4 (DHUK XOR BHK) 設定は無効になります。

KEIF フラグは、SAES_ICR レジスタの対応するビットでクリアします。SAES_IER レジスタの KEIE ビットがセットされている場合、割込みが生成されます。詳細については、[セクション 34.5 : SAES 割込み](#)を参照してください。

キー選択エラーが発生した場合、KEIF フラグをクリアすると、キー選択プロセスが自動的に再開されます。問題が解決しない場合（たとえば、RHUK ロード失敗など）、パワーオンリセットが必要になることがあります。

キー共有エラーが発生した場合は、対応する制御レジスタの IPRST ビットによって AES と SAES のペリフェラルの両方をリセットしてから、キー共有シーケンスを再開します。

注 : キーエラーがあった場合は、SAES を無効にして再設定する前に KEIF フラグをクリアしてください。

RNG エラー割込みフラグ (RNGEIF)

SAES は、RCC で IP リセットがトリガされた後、RNG ペリフェラルから乱数を自動的に取得します。RNGEIF がセットされると、SAES は使用できません。

RNG ペリフェラルから乱数を取得しているときにエラーが検出されると（たとえば、不正なエントロピーが原因で）、SAES_ISR レジスタの RNGEIF フラグがセットされます。このフラグは、SAES_ICR レジスタの対応ビットをセットしてクリアします。SAES_IER レジスタの RNGEIE ビットがセットされている場合、割込みが生成されます。詳細については、[セクション 34.5 : SAES 割込み](#)を参照してください。

RNG エラーが発生した場合 :

- RNG ペリフェラルの AHB クロックが有効で、かつ、このペリフェラルにノイズソース（またはシード）エラーが保留されていないことを確認してください。
- RNGEIF をクリアするか、IPRST をセットすることによってペリフェラルをリセットしてください。その後、BUSY フラグがクリアされれば、RNG からの乱数取得が完了したことになります。

注 : RNGEIF のエラーを避けるために、SAES AHB クロックが起動されるたびに RNG AHB クロックを起動することをお勧めします。

DPA エラーについて

予期しないエラーが発生すると、TAMP ペリフェラルの SAES 内部改ざんイベントがトリガされ、SAES コプロセッサの処理が停止します。

通常の動作を再開するには、RCC またはグローバルリセットを通じて SAES ペリフェラルをリセットします。

34.5 SAES 割込み

SAES ペリフェラルによって生成され個別にマスク可能な割込みソースは複数あり、次のイベントが通知されます。

- 計算完了 (CCF)
- 読出しエラー (RDERRF)
- 書込みエラー (WRERRF)
- キーエラー (KEIF)
- RNG エラー (RNGEIF)

SAES エラーの詳細については、[セクション 34.4.19 : SAES エラー管理](#)を参照してください。

これらのソースは、Cortex® CPU 割込みコントローラに接続する SAES ペリフェラルからの共通の割込み信号に結合されます。アプリケーションは、SAES_IER レジスタの対応するイネーブルビットをセット/クリアすることにより、SAES 割込みソースを個別に有効または無効にすることができます。

個別のマスク可能な割込みソースのステータスは SAES_ISR レジスタから読み出すことができます。それらは、SAES_ICR レジスタの対応ビットをセットすることでクリアされます。

[表 323](#) に、使用可能な機能の概要を示します。

表 323. SAES 割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ	イネーブルビット	割込みのクリア方法
SAES	計算完了フラグ	CCF	CCFIE	CCF のセット ⁽¹⁾
	読出しエラーフラグ	RDERRF ⁽²⁾	RWEIE	RWEIF のセット ⁽¹⁾
	書込みエラーフラグ	WRERRF ⁽²⁾		
	キーエラーフラグ	KEIF	KEIE	KEIF のセット ⁽¹⁾
	RNG エラーフラグ	RNGEIF	RNGEIE	RNGEIF のセット ⁽¹⁾

1. SAES_ICR レジスタのビット。

2. SAES_SR レジスタのフラグであり、SAES_ISR レジスタのフラグ RWEIF によって反映されます。

34.6 SAES DMA リクエスト

SAES ペリフェラルには、DMA (ダイレクトメモリアクセス) コントローラに接続するインターフェースが搭載されています。DMA の動作は、SAES_CR レジスタの DMAINEN ビットと DMAOUTEN ビットを通じて制御されます。キー導出が選択されている場合 (MODE[1:0] が 0x1 の場合)、これらのビットを設定しても効果はありません。

SAES ではシングル DMA リクエストのみがサポートされています。

SAES による DMA の使用方法の詳細については、[セクション 34.4.5 : SAES の暗号化または復号の一般的な使用法](#) の DMA を使用したデータの追加サブセクションを参照してください。

DMA を用いたデータ入力

DMAINEN をセットすると、SAES への DMA 書込みが有効になります。すると、SAES は入力フェーズ中に、16 バイトのデータブロックごとに SAES_DINR レジスタに書き込む一連のシングル DMA リクエスト (4 個の 32 ビットワード、MSB が先) を開始します。

注 : 選択されたアルゴリズムとモードに応じて、特別なパディングや暗号文借用が必要となることがあります ([セクション 34.4.7](#)を参照)。

DMA を用いたデータ出力

DMAOUTEN をセットすると、SAES からの DMA 読出しが有効になります。すると、SAES は出力フェーズ中に、16 バイトのデータブロックごとに SAES_DOUTR レジスタから読み出す一連のシングル DMA リクエスト (4 個の 32 ビットワード、MSB が先) を開始します。

16 バイトデータブロックの処理終了時の出力フェーズの後に、SAES によって次のデータブロック (ある場合) の新しい入力フェーズに自動的に切り替えられます。

DMA モードでは、SAES_DOUTR レジスタの読取りは計算フェーズの終了時に DMA によって自動的に管理されるため、CCF フラグは使用する必要がありません。CCF フラグは、ソフトウェアによるデータ転送管理に再び遷移したときのみクリアする必要があります。

注 : メッセージサイズによっては、最後のブロックでアプリケーションによって余分なバイトを破棄する必要があります。

DMA 転送の停止

SAES が無効になる (EN がクリアされる) か、DMA イネーブルビット (入力データの場合は DMAINEN、出力データの場合は DMAOUTEN) がクリアされると、すべての DMA リクエスト信号がデアサートされます。

34.7 SAES の処理遅延時間

次の表は、動作モードごとの 16 バイトのデータブロックの処理遅延時間を示しています。

表 324. ECB、CBC および CTR での処理遅延時間

キー長	動作モード	連鎖アルゴリズム	クロックサイクル数 ⁽¹⁾
128 ビット	暗号化または復号 ⁽²⁾	ECB、CBC、CTR	480
	キーの準備	-	145
256 ビット	暗号化または復号 ⁽²⁾	ECB、CBC、CTR	680
	キーの準備	-	230

1. SAES カーネルクロック

2. キーの準備時間を除く (ECB および CBC のみ)。

表 325. GCM および CCM での処理遅延時間 (SAES カーネルクロックサイクル数)

キー長	動作モード	連鎖アルゴリズム	初期化フェーズ	ヘッダフェーズ ⁽¹⁾	ペイロードフェーズ ⁽¹⁾	最終フェーズ ⁽¹⁾
128 ビット	モード 1 : 暗号化／ モード 3 : 復号	GCM	490	72 ⁽²⁾	480 ⁽³⁾	490
		CCM	490	490	800	490
256 ビット	モード 1 : 暗号化／ モード 3 : 復号	GCM	650	72 ⁽²⁾	690 ⁽³⁾	650
		CCM	650	680	1350	650

1. データ挿入では、AHB バスで SAES により強制されるウェイトステートが含まれる可能性があります (最大 3 サイクル、通常は 1 サイクル)。

2. カーネルクロックではなく、SAES AHB クロックサイクル数 (ガロアの乗算器のみ)。

3. 暗号化モードの最悪のケースとして、最後のブロック計算には 72 AHB クロックサイクルを追加します。

34.8 SAES レジスタ

レジスタは、32 ビットワードのシングルアクセスでのみアクセス可能。他のアクセスタイプは AHB エラーとなります。32 ビット書き込み以外では、レジスタの内容が破損する可能性があります。

34.8.1 SAES 制御レジスタ (SAES_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPRST	KEYSEL[2:0]			KSHAREID[1:0]		KMOD[1:0]		NPBLB[3:0]				KEYPROT	KEYSIZE	Res.	CHMOD[2]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GCMPTH[1:0]		DMAOUTEN	DMAINEN	Res.	Res.	Res.	Res.	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw

ビット 31 **IPRST** : SAES ペリフェラルのソフトウェアリセット

このビットをセットすると SAES ペリフェラルがリセットされ、IPRST ビット自体を除くすべてのレジスタがそれぞれのデフォルト値に設定されます。したがって、すべてのキー関連データは失われます。このため、安全性の低いアプリケーションに SAES を引き渡す前にこのビットをセットすることをお勧めします。

各設定レジスタへの書き込み中は、このビットをローに保つ必要があります。

ビット 30:28 **KEYSEL[2:0]** : キーの選択

このビットフィールドにより、AES 暗号コアで使用するキー情報のソースを規定します。

0x0 : ソフトウェアキー (キーレジスタ SAES_KEYx にロードされます)

0x1 : 導出ハードウェア固有キー (DHUK)

0x2 : ブートハードウェアキー (BHK)

0x4 : DHUK と BHK の XOR

その他 : 予約済み (使用した場合は、IPRST により SAES のフリーズを解除してください)

KEYSEL[2:0] が 0 と異なる場合、選択したキーの値は、BUSY ビットがクリアされ、SAES_SR レジスタに KEYVALID がセットされているときに、キーレジスタで入手できます。そうでないときは、キーエラーフラグ KEIF がセットされます。KEYVALID がクリアされている場合、同じ 0 以外の値で KEYSEL[2:0] を繰り返し書き込むと、DHUK または BHK のロードのみがトリガされます。

アプリケーションソフトウェアが KEYSEL[2:0] ビットフィールドに書き込むことによってキー選択を変更すると、キーレジスタは直ちに消去され、KEYVALID フラグがクリアされます。

復号プロセスの最後に、KMOD[1:0] が 0 以外の場合、KEYSEL[2:0] はクリアされます。

このビットフィールドに 0 以外の値がセットされ、KEYVALID がセットされている場合、アプリケーションは、ロードされたキーを持つ SAES の所有権を、別のセキュリティコンテキスト (セキュア、非セキュアなど) で実行されているアプリケーションに転送することはできません。より具体的には、レジスタへのアクセスのセキュリティが SAES によって記録された情報と一致しない場合、KEIF フラグがセットされます。

SAES_SR レジスタの BUSY フラグがセットされている場合、および書き込みアクセスの前に SAES_CR レジスタの EN ビットがセットされていて、その書き込みアクセスによってクリアされない場合、このビットフィールドへの書き込みの試みは無視されます。

ビット 27:26 **KSHAREID[1:0]** : キー共有 ID

このビットフィールドでは、KMOD[1:0] を 0x2 (共有キー) にした復号プロセスの終了時に、どのターゲットが専用のハードウェアバスを使用して SAES キーレジスタを読み取ることができるかを定義します。

0x0 : AES ペリフェラル

その他 : 予約済み

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 25:24 **KMOD[1:0]** : キーモード選択

このビットフィールドでは、アプリケーションが SAES キーをどのように使用できるかを定義します。KMOD[1:0] を 0 以外にセットする場合は、KEYSIZE を正しく初期化する必要があります。

0x0 : 通常キーモード。キーレジスタは自由に使用でき、SAES_DINR レジスタと SAES_DOUTR レジスタには特定の使用や保護は適用されません。

0x1 : SAES 用ラップキーモード。キーレジスタにロードされたキーは、AES キーを暗号化または復号するためにのみ使用できます。したがって、復号が選択されると、復号プロセスが成功した後、0 として読み出される SAES_DOUTR レジスタが SAES キーレジスタに自動的にロードされます。

0x2 : 共有キーモード。復号プロセス (アンラップ) が成功した後、SAES キーレジスタは KSHAREID[1:0] ビットフィールドに記述されているペリフェラルと共有されます。この共有は、KMOD[1:0] が 0x2 および KEYVALID = 1 である間のみ有効です。復号が選択された場合、復号プロセスが成功した後、0 として読み出される SAES_DOUTR レジスタが SAES キーレジスタに自動的にロードされます。

その他 : 予約済み

KMOD[1:0] が 0 以外の場合、異なるセキュリティドメイン (セキュアまたは非セキュアなど) に属するアプリケーションが使用できるように SAES ペリフェラルを設定しようとすると、キーが自動的に消去され、KEIF フラグがセットされます。

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 23:20 **NPBLB[3:0]** : 最終ブロックにおけるパディングバイト数

このパディング情報は、GCM ベイロード暗号化または CCM ベイロード復号の最終ブロックを処理する前に、ソフトウェアによって入力される必要があります。そうしないと、認証タグの計算が不正になります。

0x0 : 全バイトが有効 (パディングなし)

0x1 : 最後の LSB バイトに対するパディング

.....

0xF : 最終ブロックの LSB バイト 15 個に対するパディング。

ビット 19 **KEYPROT** : キーの保護

これをセットすると、ハードウェアベースのキー保護が有効になります。

0 : KEYVALID がセットされ、KEYSEL[2:0] = 0 の場合、アプリケーションは、ロードされたキーを持つ SAES の所有権を、別のセキュリティコンテキスト (非セキュア、セキュアなど) で実行されているアプリケーションに転送することができます。

1 : KEYVALID がセットされている場合、いずれかのレジスタに対して、キーをロードしたアプリケーションのコンテキストと一致しないセキュリティコンテキスト (たとえば、セキュア、非セキュアなど) を有するアクセスが検出されると、キーエラーフラグ (KEIF) がセットされます。

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットへの書込みの試みは無視されます。

ビット 18 **KEYSIZE** : キー長選択

このビットフィールドでは、SAES が使用するキーのビット単位の長さを定義します。

0 : 128 ビット

1 : 256 ビット

KMOD[1:0] が 0x1 または 0x2 の場合、KEYSIZE により暗号化または復号するキーの長さも定義されます。

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットへの書込みの試みは無視されます。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:13 **GCMPPH[1:0]** : GCM または CCM フェーズの選択

このビットフィールドで、GCM、GMAC、または CCM 連鎖モードでのみ適用できるフェーズを選択します。

0x0 : 初期化フェーズ

0x1 : ヘッダフェーズ

0x2 : ペイロードフェーズ

0x3 : 最終フェーズ

ビット 12 **DMAOUTEN** : DMA 出力有効

このビットにより、データフェーズ中に、DMA による SAES からの送信データ転送のための DMA リクエストの自動生成が有効になります。

0 : 無効化

1 : イネーブル

MODE[1:0] が 0x1 (キー導出) のとき、このビットの設定は無視されます。

ビット 11 **DMAINEN** : DMA 入力イネーブル

このビットにより、データフェーズ中に、DMA による SAES への着信データ転送のための DMA リクエストの自動生成が有効になります。

0 : 無効化

1 : イネーブル

MODE[1:0] が 0x1 (キー導出) のとき、このビットの設定は無視されます。

ビット 10:7 予約済みであり、リセット値に保持する必要があります。

ビット 16、**CHMOD[2:0]** : 連鎖モード

6:5 このビットフィールドで、AES 連鎖モードを選択します。

0x0 : 電子コードブック (ECB)

0x1 : 暗号ブロック連鎖 (CBC)

0x2 : カウンタモード (CTR)

0x3 : ガロアカウンタモード (GCM) およびガロアメッセージ認証コード (GMAC)

0x4 : CBC-MAC 付きカウンタ (CCM)

上記以外 : 予約済み

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 4:3 **MODE[1:0]** : 動作モード

このビットフィールドで、SAES 動作モードを選択します。

0x0 : 暗号化

0x1 : キー導出 (またはキー準備) (ECB/CBC 復号の場合のみ)

0x2 : 復号

0x3 : 予約済み

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 2:1 **DATATYPE[1:0]** : データ型

このビットフィールドは、データスワッピングのモードを選択して、SAES_DINR レジスタに書き込まれるデータまたは SAES_DOUTR レジスタから読み出されるデータのフォーマットを定義します。このスワッピングは [セクション 34.4.16 : SAES データレジスタおよびデータスワッピング](#) に定義されています。

0x0 : スワッピングなし (32 ビットデータ)

0x1 : ハーフワードスワッピング (16 ビットデータ)

0x2 : バイトスワッピング (8 ビットデータ)

0x3 : ビットレベルスワッピング

BUSY がセットされている場合、および書込みアクセスの前に EN がセットされていて、その書込みアクセスによってクリアされない場合、このビットフィールドへの書込みの試みは無視されます。

ビット 0 EN : イネーブル

このビットは、SAES ペリフェラルを有効または無効にします。

0 : 無効化

1 : イネーブル

このビットをクリアしてからセットすれば、いつでも SAES ペリフェラルを再初期化できます。KMOD[1:0] が 0x0 と異なる場合、代わりに IPRST ビットを使用することをお勧めします。

このビットは、キーの準備 (MODE[1:0] が 0x1) の完了時、および GCM/GMAC/CCM 初期化フェーズの完了時に、ハードウェアによって自動的にクリアされます。

このビットは、KEYVALID がクリアされているか、SAES が以下の設定のいずれかになっている限り、セットできません。

- KMOD[1:0] が 0x1 (ラップ)、CHMOD[2:0] が 0x3 (GCM)
- KMOD[1:0] が 0x1 (ラップ)、CHMOD[2:0] が 0x2 (CTR)、MODE[1:0] が 0x0 (暗号化)

34.8.2 SAES ステータスレジスタ (SAES_SR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEYVALID	Res.	Res.	Res.	BUSY	WRERRF	RDERRF	Res.
								r				r	r	r	

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 KEYVALID : キー有効フラグ

このビットは、KEYSIZE で定義されたサイズのキーが SAES_KEYRx キーレジスタにロードされると、ハードウェアによってセットされます。

0 : キーは無効

1 : キーは有効

EN ビットは、KEYVALID がセットされている場合にのみセットできます。

KEYSEL[2:0] が 0 のときの通常モードでは、キーを正しい順序でキーレジスタに書き込む必要があります。そうしないと、KEIF フラグがセットされ、KEYVALID はクリアされたままになります。

KEYSEL[2:0] が 0 と異なる場合、SAES によって BUSY フラグが自動的にセットされます。キーが正常にロードされると、BUSY がクリアされ、KEYVALID がセットされます。エラーが発生すると、KEIF がセットされ、BUSY はクリアされ、KEYVALID はクリアされたままになります。

KEIF がセットされている場合、SAES_ICR レジスタを通じてクリアする必要があります。クリアしないと、KEYVALID がセットされません。詳細については、KEIF フラグの説明を参照してください。

キーのローディングの詳細については、[セクション 34.4.17 : SAES キーレジスタ](#)を参照してください。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **BUSY** : ビジー

このフラグは、SAES がアイドルかビジーかを示します。

0 : アイドル

1 : ビジー

SAES が無効になっているとき (EN がローのとき)、または最後の処理が完了したときに、アイドルとして SAES にフラグが立てられます。

SAES がブロックデータを処理しているとき、キーを準備しているとき (ECB または CBC の復号のみ)、RNG から乱数を取得しているとき、あるいはターゲットペリフェラルへ共有キーを転送しているとき、SAES にビジーとしてフラグが立てられます。

GCM 暗号化が選択されている場合、優先度の高いメッセージを管理するために現在のプロセスを一時停止する前に、このフラグを 0 にする必要があります。BUSY は、GCM 最終フェーズを選択する前にもクリアする必要があります。

ビット 2 **WRERRF** : 書きみエラーフラグ

このビットは、SAES_DINR レジスタへの予期しない書き込みが発生したときに、セットされます。WRERRF ビットがセットされても、SAES の動作に影響はありません。

0 : エラーはありません。

1 : 計算中またはデータ出力フェーズ中に、SAES_DINR レジスタへの予期しない書き込みが発生しました。

SAES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの RWEIF ビットをセットしてクリアします。

ビット 1 **RDERRF** : 読出しエラーフラグ

このビットは、SAES_DOUTR レジスタに対する予期しない読出しが発生したときに、セットされます。RDERRF ビットがセットされても、SAES の動作に影響はありません。

0 : エラーはありません。

1 : 計算中またはデータ入力フェーズ中に、SAES_DOUTR レジスタに対する予期しない読出しが発生しました。

SAES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの RWEIF ビットをセットしてクリアします。

ビット 0 予約済みであり、リセット値に保持する必要があります。

34.8.3 SAES データ入力レジスタ (SAES_DINR)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **DIN[31:0]** : データ入力

入力フェーズ中にこのビットフィールドに 4 連続で書き込むことによって、SAES ペリフェラルに 16 バイトブロックの入力データすべてを書き込めます。1 回目から 4 回目の書き込みまで、対応するデータの重みは [127:96]、[95:64]、[63:32]、[31:0] となります。書き込みごとに、32 ビット入力バッファのデータが DATATYPE[1:0] ビットフィールドに従ってデータスワッピングブロックによって処理され、AES コアの 16 バイト入力バッファに書き込まれます。

これを読み出すと 0 が返されます。

34.8.4 SAES データ出力レジスタ (SAES_DOUTR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

MODE[1:0] が 0x2 で EN がセットされている場合に KMOD[1:0] が 0x1 または 0x2 であるときに読み出すと、読出しエラーがトリガされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DOUT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **DOUT[31:0]** : データ出力

読出し専用ビットフィールドは、32 ビット出力バッファをフェッチします。計算完了 (CCF フラグセット) 時にこのビットフィールドを 4 連続で読み出すことによって、ペリフェラルから 16 バイトブロックの出力データすべてを実質的に読み出せます。

出力バッファに達する前に、AES コアによって生成されたデータは、

DATATYPE[1:0] ビットフィールドに従ってデータスワッピングブロックによって処理されます。

1 回目から 4 回目までの読出し操作でのデータの重みは、[127:96]、[95:64]、[63:32]、[31:0] となります。

34.8.5 SAES キーレジスタ 0 (SAES_KEYR0)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[31:0]** : 暗号化キー、ビット [31:0]

これらは、SAES_CR レジスタの MODE[1:0] ビットフィールドに応じて、書き込み専用ビットフィールドである KEY[255:0] の AES 暗号化または復号キーのうちのビット [31:0] です。

SAES が有効になっている (EN ビットがセットされている) 場合、SAES_KEYRx レジスタへの書き込みは無視されます。KEYSEL[2:0] が 0 と異なり、KEYVALID が 0 の場合も、キーレジスタへの書き込みは無視され、SAES_ISR レジスタの KEIF ビットがセットされます。

KMOD[1:0] が 0x0 の場合、特別の書き込みシーケンスが必要です。このシーケンスでは、KEYVALID フラグがセットされるシーケンス完了書き込みを除き、AES_KEYRx レジスタへ有効な書き込みをすると、KEYVALID フラグがクリアされます。AES_SR レジスタの KEYVALID フラグの説明も参照してください。

34.8.6 SAES キーレジスタ 1 (SAES_KEYR1)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[63:48]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[63:32]** : 暗号化キー、ビット [63:32]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.7 SAES キーレジスタ 2 (SAES_KEYR2)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[95:80]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[95:64]** : 暗号化キー、ビット [95:64]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.8 SAES キーレジスタ 3 (SAES_KEYR3)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[127:112]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[127:96]** : 暗号化キー、ビット [127:96]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.9 SAES 初期化ベクタレジスタ 0 (SAES_IVR0)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[31:0]** : 初期化ベクタ入力、ビット [31:0]

SAES_IVRx レジスタには、選択された連鎖モードに応じて、128 ビットの初期化ベクタまたはノンスが格納されます。この値は、各計算ラウンド（該当する場合）後にハードウェアによって更新されます。

このレジスタへの書込みは、EN ビットが SAES_SR レジスタでセットされている場合は無視されます。

34.8.10 SAES 初期化ベクタレジスタ 1 (SAES_IVR1)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[63:48]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[63:32]** : 初期化ベクタ入力、ビット [63:32]

IVI[128:0] ビットフィールドの説明については、SAES_IVR0 レジスタを参照してください。

34.8.11 SAES 初期化ベクタレジスタ 2 (SAES_IVR2)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[95:80]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[95:64]** : 初期化ベクタ入力、ビット [95:64]

IVI[128:0] ビットフィールドの説明については、SAES_IVR0 レジスタを参照してください。

34.8.12 SAES 初期化ベクタレジスタ 3 (SAES_IVR3)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[127:112]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **IVI[127:96]** : 初期化ベクタ入力、ビット [127:96]

IVI[128:0] ビットフィールドの説明については、SAES_IVR0 レジスタを参照してください。

34.8.13 SAES キーレジスタ 4 (SAES_KEYR4)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[159:144]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[143:128]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[159:128]** : 暗号化キー、ビット [159:128]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.14 SAES キーレジスタ 5 (SAES_KEYR5)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[191:176]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[175:160]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[191:160]** : 暗号化キー、ビット [191:160]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.15 SAES キーレジスタ 6 (SAES_KEYR6)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[223:208]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[207:192]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[223:192]** : 暗号化キー、ビット [223:192]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.16 SAES キーレジスタ 7 (SAES_KEYR7)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[255:240]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[239:224]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **KEY[255:224]** : 暗号化キー、ビット [255:224]

KEY[255:0] ビットフィールドの説明、および SAES_KEYRx レジスタへの書込みに関する情報については、SAES_KEYR0 レジスタを参照してください。

34.8.17 SAES サスペンドレジスタ (SAES_SUSPRx)

アドレスオフセット : $0x040 + x * 0x4$ 、($x = 0 \sim 7$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUSP[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUSP[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **SUSP[31:0]** : サスペンドデータ

CCM の現在のタスクの処理が、優先順位の高いタスクを処理するためにサスペンドされる場合、SAES の内部レジスタのステータスはすべて SAES_SUSPRx レジスタ内に保存されます。詳細については、[セクション 34.4.8 : SAES のサスペンドおよびレジューム操作](#)を参照してください。

EN ビットが SAES_SR レジスタでクリアされている場合は、このレジスタに対する読出しには 0 が返されます。

SAES_SUSPRx レジスタは、CCM 以外の連鎖モードでは使用されません。

34.8.18 SAES 割込み有効レジスタ (SAES_IER)

アドレスオフセット : 0x300

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGIE	KEIE	RWEIE	CCIE
												rW	rW	rW	rW

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **RNGIE** : RNG エラー割込みイネーブル

このビットは、RNGEIF (RNG エラーフラグ) がセットされた場合の SAES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

ビット 2 **KEIE** : キーエラー割込みイネーブル

このビットは、KEIF (キーエラーフラグ) がセットされた場合の SAES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

ビット 1 **RWEIE** : 読出しまたは書き込みエラー割込み有効

このビットは、RWEIF (読出しおよび/または書き込みエラーフラグ) がセットされた場合の SAES 割込み生成を有効化または無効化 (マスク) します。

0 : 無効化 (マスク)

1 : 有効化 (マスクされない)

ビット 0 **CCFIE** : 計算完了フラグ割込み有効

このビットは、CCF（計算完了フラグ）がセットされた場合の SAES 割込み生成を有効化または無効化（マスク）します。

0 : 無効化（マスク）

1 : 有効化（マスクされない）

34.8.19 SAES割込みステータスレジスタ (SAES_ISR)

アドレスオフセット : 0x304

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **RNGEIF** : RNG エラー割込みフラグ

この読み出し専用ビットは、RNG バスインタフェースでエラー（たとえば、不正なエントロピー）が検出されたときに、ハードウェアによってセットされます。

0 : RNG バスは機能しています。

1 : RNG バスインタフェースでエラー（ランダム・シード取得エラー）が検出されました。

SAES_IER レジスタの RNGEIF ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの対応ビットをセットしてクリアします。クリア操作によって、RNG ペリフェラルからの新しい乱数のリロードがトリガされます。

ビット 2 KEIF : キーエラー割込みフラグ

この読み出し専用ビットは、キー情報がキーレジスタにロードできなかったとき、またはキーレジスタの使用が禁止されているときに、ハードウェアによってセットされます。

0 : キーエラーは検出されていません。

1 : キー情報がキーレジスタにロードできなかったか、またはキーレジスタの使用が禁止されています。

SAES_IER レジスタの KEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの対応ビットをセットしてクリアします。

KEIF は、次のいずれかのイベント時に立てられます。

–SAES が、DHUK をロードできない (KEYSEL[2:0] = 0x1 または 0x4 の場合)。

–SAES が、BHK をロードできない (KEYSEL[2:0] = 0x2 または 0x4 の場合)。

–AES AES が、SAES パリフェラルによって共有されるキーをロードできない (KMOD[1:0] = 0x2 の場合)。

–KEYVALID がセットされており、KEYPROT がセットされているか、KEYSEL[2:0] が 0x0 以外のどちらかである。キーをロードするアプリケーションのセキュリティコンテキスト (セキュアまたは非セキュア) が、SAES_CR または SAES_DOUT へのアクセスのセキュリティ属性と一致しない。この場合、KEYVALID および EN ビットはクリアされます。

–SAES_KEYRx レジスタへの書き込みが、次の正しい順序で行われていない。(KEYSIZE がクリアされている場合は、SAES_KEYR0、SAES_KEYR1、SAES_KEYR2、SAES_KEYR3 レジスタの順か、またはその逆。KEYSIZE がセットされている場合は、SAES_KEYR0、SAES_KEYR1、SAES_KEYR2、SAES_KEYR3、SAES_KEYR4、SAES_KEYR5、SAES_KEYR6、SAES_KEYR7 の順か、またはその逆)。

KEIF は、アプリケーションソフトウェアでクリアする必要があります。クリアしないと、KEYVALID がセットされません。

ビット 1 RWEIF : 読み出しまたは書き込みエラー割込みフラグ

この読み出し専用ビットは、SAES_SR レジスタの RDERRF または WRERRF エラーフラグがセットされたときに、ハードウェアによってセットされます。

0 : 読み出しまたは書き込みエラーは検出されていません。

1 : 読み出しまたは書き込みエラーが検出されました。

SAES_IER レジスタの RWEIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの対応ビットをセットしてクリアします。

キー導出モードが選択されている場合、このフラグは意味を持ちません。

詳細については、SAES_SR レジスタを参照してください。

ビット 0 CCF : 計算完了フラグ

このフラグは計算が完了したかどうかを示します。これは DMAOUTEN ビットがクリアされている場合にのみ重要であり、DMAOUTEN がセットされている場合はハイのままになる可能性があります。

0 : 未完了

1 : 完了

SAES_IER レジスタの CCFIE ビットがセットされている場合、このフラグがセットされると割込みが生成されます。

このフラグは、SAES_ICR レジスタの対応ビットをセットしてクリアします。

34.8.20 SAES 割込みクリアレジスタ (SAES_ICR)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
												w	w	w	w

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **RNGEIF** : RNG エラー割込みフラグクリア

SAES_ISR レジスタの RNGEIF ステータスビットをクリアするためには、アプリケーションはこのビットをセットする必要があります。

ビット 2 **KEIF** : キーエラー割込みフラグクリア

このビットをセットすると SAES_ISR レジスタの KEIF ステータスビットがクリアされます。

ビット 1 **RWEIF** : 読み出しまたは書き込みエラー割込みフラグクリア

このビットをセットすると、SAES_ISR レジスタの RWEIF ステータスビットがクリアされ、SAES_SR レジスタの RDERRF フラグと WRERRF フラグの両方がクリアされます。

ビット 0 **CCF** : 計算完了フラグクリア

このビットをセットすると SAES_ISR レジスタの CCF ステータスビットがクリアされます。

表 326. SAESレジスタマップとリセット値 (続き)

オフ セツ	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x044	SAES_SUSPR1	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x048	SAES_SUSPR2	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04C	SAES_SUSPR3	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x050	SAES_SUSPR4	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x054	SAES_SUSPR5	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x058	SAES_SUSPR6	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x05C	SAES_SUSPR7	SUSP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x060- 0x2FF	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x300	SAES_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGIE	KEIE	RWIE	CCFIE
	リセット値																													0	0	0	0	
0x304	SAES_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
	リセット値																													0	0	0	0	
0x308	SAES_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RNGEIF	KEIF	RWEIF	CCF
	リセット値																													0	0	0	0	

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

35 ハッシュプロセッサ (HASH)

35.1 概要

ハッシュプロセッサは、セキュアハッシュアルゴリズム (SHA-1、SHA-2 ファミリ)、および HMAC (鍵付きハッシュメッセージ認証コード) アルゴリズムに完全に準拠した処理系となっています。HMAC はメッセージ認証を必要とするアプリケーションに適しています。

ハッシュプロセッサは、 2^{64} ビット未満 (SHA-1、SHA-224、および SHA-256 の場合) または 2^{128} ビット未満 (SHA-384、SHA-512 の場合) の任意の長さのメッセージに対して、FIPS (連邦情報処理規格) で承認された 160、224、256 ビットの長さのダイジェストを生成します。

35.2 HASH の主な機能

- 以下に準拠したデータ認証アプリケーションに好適
 - 連邦情報処理規格公報 FIPS PUB 180-4、セキュアハッシュ標準 (SHA-1 および SHA-2 ファミリ)
 - 連邦情報処理規格公報 FIPS PUB 186-4、デジタル署名標準 (DSS)
 - インターネット技術タスクフォース (IETF) RFC 2104、HMAC : メッセージ認証のための鍵付きハッシング、および連邦情報処理規格公報 FIPS PUB 198-1、鍵付きハッシュメッセージ認証コード (HMAC)
- SHA-1、SHA2-224、SHA2-256、SHA2-384、および SHA2-512 の高速計算
 - SHA-1、SHA-256 それぞれのアルゴリズムを使用して 1 個の 512 ビットブロックのデータを処理する時間はそれぞれ 82、66 クロックサイクル
 - SHA2-384 または SHA2-512 アルゴリズムを使用して 1 個の 1024 ビットブロックのデータを処理する時間は 98 クロックサイクル
 - SHA-2 の切り捨てられた出力をサポート (SHA2-512/224、SHA2-512/256)
- 対応しているアルゴリズムについて HMAC モードをサポート
- 連続するメッセージブロックから、対応する 32 ビットワードのダイジェストが相互に追加され、メッセージ全体のダイジェストを形成
 - 入力ビット列の内部リトルエンディアン表現に準拠する自動 32 ビットワードスワッピング
 - サポートしているワードスワッピング形式 : ビット、バイト、ハーフワード、および 32 ビットワード
- 内部入力 FIFO に関連付けられた 32 ビットの書き込み専用シングル入力レジスタ (64 バイトのブロックサイズ (16 x 32 ビット)) に対応
- ダイジェスト最低ブロックサイズに適合するよう入力ビット列を補完する自動パディング
- 32 ビットワードによるアクセスでのみ (それ以外は AHB エラーが発生) アクセス可能な AHB スレーブペリフェラル。
- 出力メッセージダイジェストの場合 8 x 32 ビットワード (H0 から H15)
- 1 つのチャネルを使用したダイレクトメモリアクセス (DMA) をサポートする自動データフロー制御。
- 4 ワードのシングルおよび固定の両方の DMA パースト転送のサポート。
- ブロック単位での割込み可能なメッセージダイジェスト計算
 - 再ロード可能なダイジェストレジスタ
 - DMA を含むハッシュ計算サスペンド/レジュームメカニズム

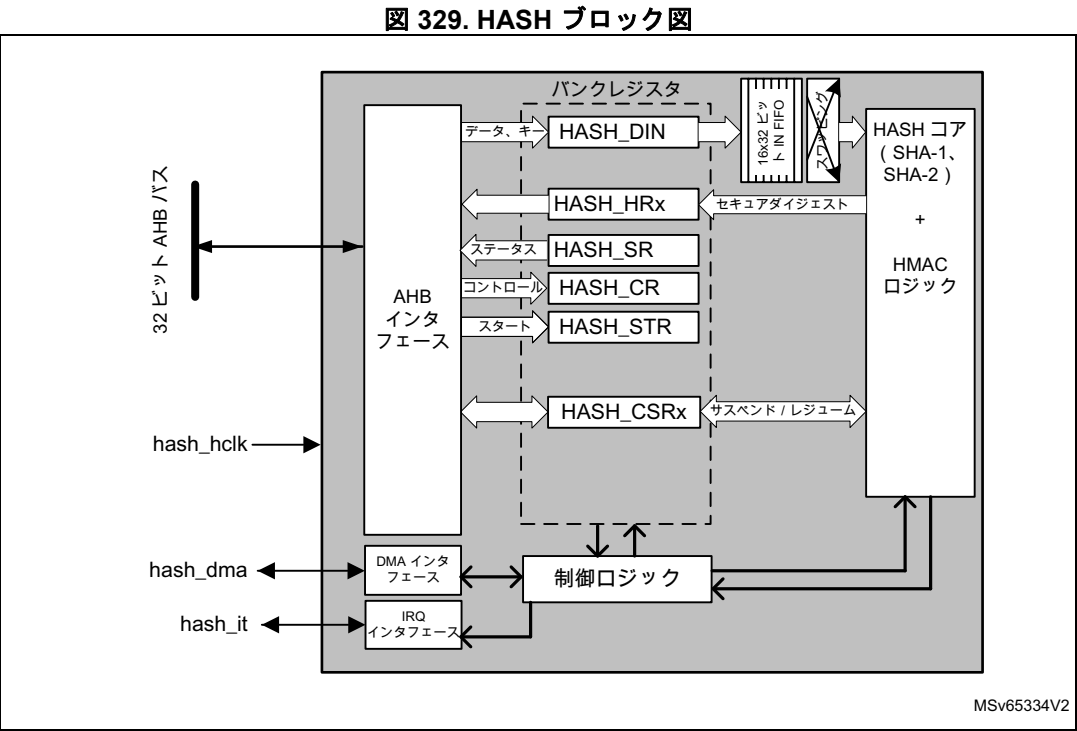
35.3 ハッシュの実装

デバイスは HASH ペリフェラルのシングルインスタンスを持ちます。

35.4 HASH の機能詳細

35.4.1 HASH ブロック図

図 329 に、ハッシュプロセッサのブロック図を示します。



35.4.2 HASH 内部信号

表 327 では、製品レベル（パッド上）ではなく、HASH レベルで使用可能な内部信号を知るのに役立つリストを記載しています。

表 327. HASH 内部入力／出力信号

信号名	信号タイプ	説明
hash_hclk	デジタル入力	AHB バスクロック
hash_it	デジタル出力	ハッシュプロセッサグローバル割込みリクエスト
hash_dma	デジタル入出力	DMA パーストリクエスト／確認

35.4.3 セキュアハッシュアルゴリズムについて

ハッシュプロセッサは、FIPS PUB 180-4 標準によって定義されたセキュアハッシュアルゴリズムに完全に準拠した実装となっています。

各アルゴリズムで、HASH はメッセージやデータファイルの圧縮表記を生成します。より具体的には、メッセージが入力に提供されると、HASH 処理コアは、メッセージダイジェストと呼ばれる固定長の出力文字列を生成します（表 328 を参照）。

表 328. サポートされているハッシュアルゴリズムの情報

アルゴリズム	メッセージ ダイジェストサイズ (ビット単位)	ブロックサイズ (バイト単位) ⁽¹⁾	メッセージ長	ビット列 メッセージ
SHA-1	160	64	< 2 ⁶⁴ ビット	はい
SHA2-224	224			
SHA2-256	256			
SHA2-384	384	128	< 2 ¹²⁸ ビット	はい
SHA2-512	512 ⁽²⁾			

1. ブロックサイズ = (NBWE-1) * 4 バイト。NBWE[4:0] ビットフィールドは、ALGO ビットと INIT ビットが HASH_CR レジスタに書き込まれた後、HASH_SR レジスタから読み出すことができます。
2. ダイジェストサイズは、SHA2-512/224 では 224 ビット、SHA2-512/256 では 256 ビット（切捨てモード）。

その後、メッセージの署名を作成したり検証したりするため、メッセージダイジェストはデジタル署名アルゴリズムを使用して処理することができるようになります。

メッセージ自体ではなくメッセージダイジェストに署名を行うことで、多くの場合処理効率が向上します。これは通常、メッセージダイジェストの方がメッセージよりもはるかにサイズが小さいからです。デジタル署名の検証者は、デジタル署名の作成者が使用したのと同じハッシュアルゴリズムを使用する必要があります。

ハッシュプロセッサがサポートしている SHA-2 機能は NIST によって「安全」とであるとされています。それはあるメッセージダイジェストに対応するメッセージの発見、もしくは同じメッセージダイジェストを生成する 2 つの異なるメッセージの発見が計算上不可能だからです（SHA-1 は 2017 年 2 月から安全ではないとされています）。メッセージの転送中に何らかの変更が加えられた場合、極めて高い確率で異なるメッセージダイジェストになり、その署名は検証に失敗します。

35.4.4 メッセージデータの供給

HASH で処理するメッセージ（またはデータファイル）はビット列とみなされなければなりません。FIPS PUB 180-4 標準により、このメッセージビット列はビッグエンディアン規約で表記された 16 進数ワードで左から右に表記されるため、最上位ビットは各ワード内で一番左のビット位置に格納されます。たとえば、「01100001 01100010 01100011」のビット列で表されるメッセージ文字列「abc」は、32 ビットワードで 0x00636261、8 ビットワードで 0x61626300 として表されます。

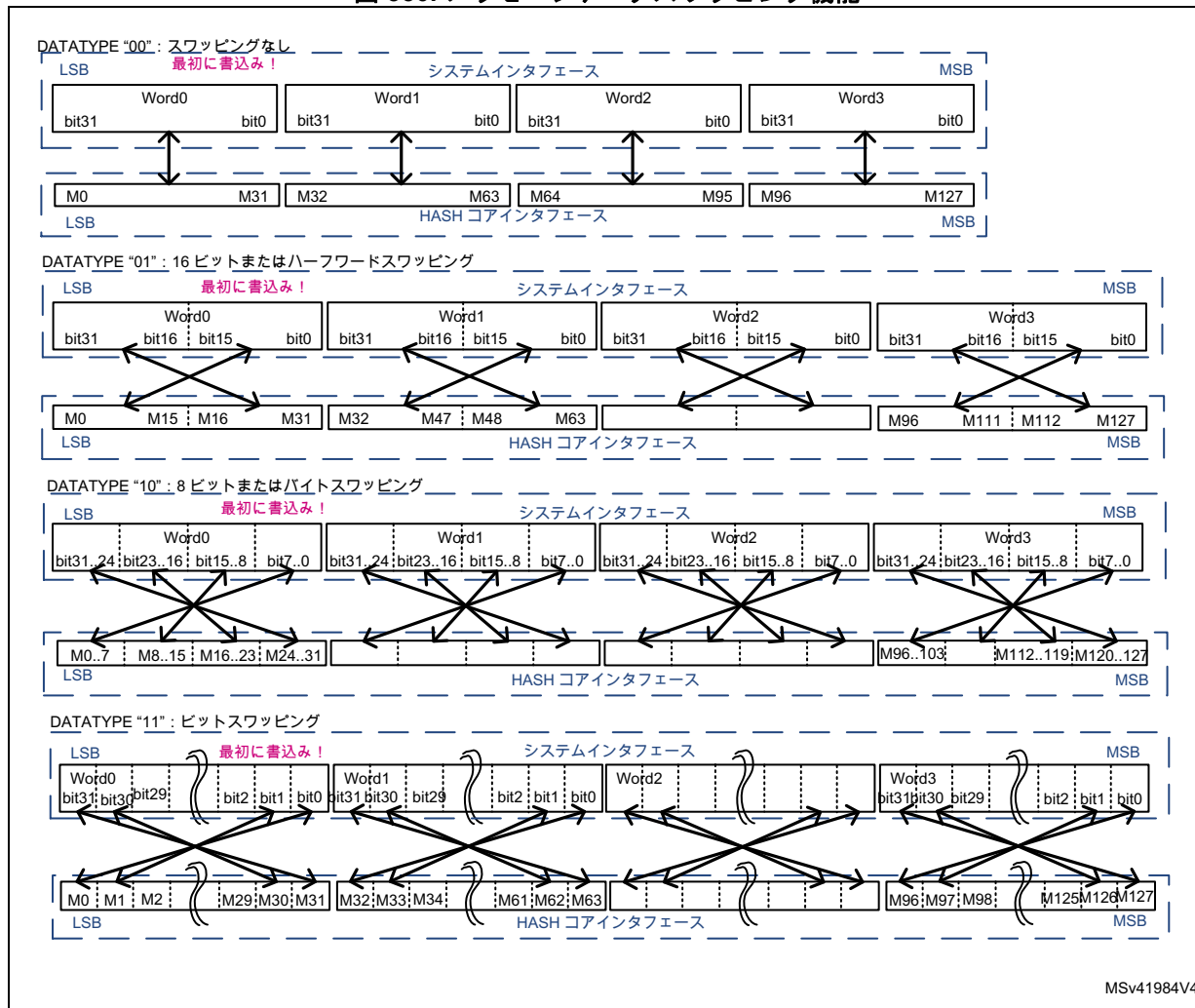
データは HASH_DIN レジスタに書き込むことにより 32 ビットワードごとに HASH に入力されます。HASH_DIN レジスタの現在の内容は、このレジスタに新しいデータが書き込まれるたびに、16 ワードの入力 FIFO に転送されます。そのため、HASH_DIN と FIFO で、17 の 32 ビットワード長の FIFO を形成します（IN バッファと呼びます）。

処理されるデータの種類（たとえば、データが ASCII テキストストリームの場合のバイトスワッピングなど）に応じて、入力 FIFO からのデータに対して、リトルエンディアンハッシュ処理コアに入力する前に、ビット、バイト、ハーフワード、またはスワッピングなしの操作を実行する必要があります。図 330 に、HASH 制御レジスタ（HASH_CR）の DATATYPE ビットフィールドに従って、ハッ

シュ処理コアの 32 ビットデータブロック M0~31 が、ドライバによって入力 FIFO に書き込まれた 1 つの 32 ビットワードからどのように構成されるかを示します

ビットスワッピングが無効 (DATATYPE = 00) になっている場合の HASH_DIN データのエンディアン形式は、次のように記載できます。メッセージの最下位ビットはハッシュプロセッサに入力される最初のワードの MSB の位置でなければならず、ビット列の 32 番目のビットはハッシュプロセッサに入力される 2 番目のワードの MSB の位置でなければなりません。それ以降は同様です。

図 330. メッセージデータスワッピング機能



35.4.5 メッセージダイジェストの計算

ハッシュプロセッサは、メッセージダイジェストを計算する際に、いくつかのブロックを順次処理します。ブロックサイズについては、表 328: サポートされているハッシュアルゴリズムの情報を参照してください。

DMA または CPU がブロックをハッシュプロセッサに書き込むたびに、HASH は自動的にメッセージダイジェストの計算を開始します。この操作は、部分ダイジェスト計算として知られています。

セクション 35.4.4: メッセージデータの供給に示すとおり、処理するメッセージは、HASH 32 ビットワードに入力され、入力 FIFO を満たすために HASH_DIN レジスタに書き込まれます。このデータでハッシュ計算を実行するために、アプリケーションは以下の手順に従う必要があります。

1. HASH_CR レジスタを使用してハッシュプロセッサを初期化します。
 - ALGO ビットフィールドを使用して正しいアルゴリズムを選択するように HASH_CR レジスタに書き込み、INIT ビットをセットします (他のビットは 0 に保ちます)。次に、HASH_SR レジスタから NBWE ビットフィールドを読み取り、 $(NBWE-1) * 4$ によって、アルゴリズムのブロックサイズを割り出します。ブロックサイズがすでにわかっている場合は (詳細は [表 328](#) を参照)、この手順は不要です。
 - ALGO[3:0] フィールドを使用して正しいアルゴリズムを選択します。必要に応じて、DATATYPE[1:0] ビットフィールドを使用してメッセージ入力ワードに正しいスワッピング操作をプログラムします。
 - HMAC モードが必要な場合、HMAC キーのサイズがアルゴリズムの既知のブロックサイズより大きい場合は、MODE ビットのほかに LKEY ビットもセットします (そうでない場合は、LKEY はクリアしたままにします)。詳細については、[セクション 35.4.7: HMAC 操作](#) を参照してください。
 - メッセージの最後のワードの有効ビット数が 32 ビットではない場合、HASH_STR レジスタの NBLW[4:0] を更新して有効ビット数を定義します。NBLW 情報は、最終的なメッセージダイジェストの計算の前に自動メッセージパディングを正しく実行するために使用されます。
2. HASH_CR レジスタの INIT ビットをセットして、初期化を完了します。データを DMA で転送する場合、DMAE ビットもセットします。

注意： 手順 2 をプログラミングするときは、その前または同時に正しい設定値 (ALGO、DATATYPE、HMAC モード、キーの長さ、NBLW) を設定することが重要です。

3. データが DMA で自動で転送されない場合は、HASH_DIN レジスタに書き込んでデータ入力を開始します。ブロックの処理はブロックの最終値が 入力 FIFO に入力された後で初めて開始できることに注意してください。部分または最終ダイジェスト計算の管理方法は、データのハッシュプロセッサへの供給方法によって異なります。
 - ソフトウェアでデータが入力される場合：

部分ダイジェスト計算は、アプリケーションが次のブロックの最初のワードを書き込むたびにトリガされ、ブロックサイズは HASH_SR の NBWE ビットによって定義されます。プロセッサが再びレディ (HASH_SR の DINIS = 1) になると、ソフトウェアは新しいデータを HASH_DIN に書き込めるようになります。このメカニズムによって、HASH によるウェイトステートの導入を回避します。

最終ダイジェスト計算は、最終ブロックが入力され、ソフトウェアが DCAL ビットをセットすると、トリガされます。メッセージ長がブロックサイズの整数倍ではない場合、DCAL ビットを書き込む前に HASH_STR レジスタの NBLW フィールドを書き込む必要があります (詳細については、[セクション 35.4.6](#) を参照)。
 - シングル DMA 転送 (MDMAT = 0) としてデータが入力される場合：

部分ダイジェスト計算は、FIFO がフルになるたびに自動的にトリガされます。

最終ダイジェスト計算は、最終ブロックが DMA によって HASH_DIN レジスタに転送された際に自動的にトリガされます (DCAL ビットがハードウェアによってセットされます)。メッセージ長がブロックサイズの整数倍ではない場合、DMA を有効にする前に HASH_STR レジスタの NBLW フィールドを書き込む必要があります (詳細については、[セクション 35.4.6](#) を参照)。
 - 多重 DMA 転送 (MDMAT = 1) によってデータが入力される場合：

部分ダイジェスト計算は、シングル DMA 転送の場合と同様にトリガされます (上記説明を参照)。ただし、最終ダイジェスト計算は、最終ブロックが DMA によって HASH_DIN レジスタに転送された際に自動的にトリガされません (DCAL ビットはハードウェアによってセットされません)。これにより、ハッシュプロセッサは、新しい DMA 転送をこのダイジェスト計算の一部として受信できます。最終ダイジェスト計算を開始するには、シ

ングル DMA 転送の場合と同様に最終ダイジェスト計算をトリガするために、最後の DMA 転送の前に、ソフトウェアで MDMAT ビットをクリアする必要があります。

4. いったんダイジェスト計算が完了すると (DCIS = 1)、結果のダイジェストは表 329 に記載しているとおり、出力レジスタから読み出すことができます。

表 329. ハッシュプロセッサ出力

アルゴリズム	有効な出力レジスタ	最上位ビット	ダイジェストサイズ (ビット)
SHA-1	HASH_H0 から HASH_H4	HASH_H0[31]	160
SHA2-224	HASH_H0 から HASH_H6	HASH_H0[31]	224
SHA2-256	HASH_H0 から HASH_H7		256
SHA2-384	HASH_H0 から HASH_H11	HASH_H0[31]	384
SHA2-512	HASH_H0 から HASH_H15		512 ⁽¹⁾

1. ダイジェストサイズは、SHA2-512/224 では 224 ビット、SHA2-512/256 では 256 ビット (切捨てモード)

HMAC の詳細な説明については、[セクション 35.4.7 : HMAC 操作](#)を参照してください。

35.4.6 メッセージのパディング

概要

メッセージの圧縮表記を計算する際に、データをハッシュプロセッサに (ブロックサイズ転送ごとの自動部分ダイジェスト計算で) 供給する処理は、オリジナルメッセージの最後のビットが HASH_DIN レジスタに書き込まれるまでループします。

メッセージの長さ (ビット数) は任意の整数値をとることができるため、ハッシュプロセッサに書き込まれる最後のワードの有効ビット数は 1 から 32 まで可能です。最終メッセージダイジェスト計算の前にメッセージのパディングを正しく行うため、最後のワードの有効ビット数 NBLW を HASH_STR レジスタに書き込む必要があります。

パディング処理

DMA が有効または無効な状態での詳細なパディング手順については、[セクション 35.4.5 : メッセージダイジェストの計算](#)に記載しています。

パディングの例

連邦情報処理規格 PUB 180-4 によって指定されているように、メッセージのパディングは、「1」の後に k 個の「0」が続き、その後にメッセージの長さ L (ビット単位) に等しい 64 ビットの整数値が続きます。この 3 つのパディング操作により、長さ $L + 1 + k + 64$ のパディングされたメッセージが生成され、512 ビットの倍数となります。

ハッシュプロセッサでは、HASH_DIN レジスタに対して最後に書き込まれたワードの中の NBLW ビットフィールドで定義された位置に、「1」が追加され、残りの上位ビットは「0」にクリアされます。

FIPS PUB180-4 からの例

オリジナルメッセージが ASCII バイナリコード形式の「abc」で、長さ $L = 24$ であると仮定します。

```
byte 0    byte 1    byte 2    byte 3
01100001 01100010 01100011 UUUUUUUU
<-- HASH_DINに書かれた最初のワード-->
```


NBLW に値 24 をロードする必要があります。ビット列のビット位置 24（上記ビット列で左から右へカウントします）に「1」が付加されます。これは HASH_DIN レジスタのビット 31 に対応します（リトルエンディアン規約）：

```
01100001 01100010 01100011 1UUUUUUU
```

L = 24 であるため、上記のビット列のビット数は 25 ビットであり、423 個の「0」ビットが付加されて 448 ビットになります。

その結果、16 進数（ビッグエンディアン形式のバイトワード）で以下のようにになります。

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

2 ワード形式のメッセージ長の値 L（00000000 00000018）が付加されます。したがって、パディング処理により最終的に得られるメッセージは、16 進数（ビッグエンディアン形式のバイトワード）で次のようになります。

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000018
```

ハッシュプロセッサが HASH_DIN 入力レジスタ内のバイトをスワップするようプログラムされている（HASH_CR で DATATYPE = 10）場合、次の手順を使って上記メッセージを入力する必要があります。

1. HASH_DIN レジスタに 0xUU636261 を書き込みます（ここで'U'は無視を意味します）。
2. HASH_STR レジスタに 0x18 を書き込みます（オリジナルメッセージの長さは 24 ビットなので HASH_DIN レジスタに最後に書き込まれたワードの有効ビット数は 24 です）。
3. HASH_STR レジスタに 0x10 を書き込んでメッセージのパディング処理（上記で説明）を開始し、ダイジェスト計算を実行します。
4. ハッシュ計算が完了すると、SHA-1 アルゴリズムのメッセージダイジェストが HASH_HRx レジスタ（x = 0~4）で利用できる状態になります。この FIPS の例の場合の期待値は次のとおりです。

```
HASH_HR0 = 0xA9993E36
HASH_HR1 = 0x4706816A
HASH_HR2 = 0xBA3E2571
HASH_HR3 = 0x7850C26C
HASH_HR4 = 0x9CD0D89D
```


35.4.7 HMAC 操作

概要

インターネット技術タスクフォース RFC2104 および NIST FIPS PUB 198-1 で指定されているように、HMAC アルゴリズムは、処理するメッセージをユーザが選択したキーに不可逆的に結合することによるメッセージ認証に使用します。このアルゴリズムは 2 つのネストされたハッシュ操作で構成されています。

$$\text{HMAC}(\text{message}) = \text{Hash}((\text{Key} \mid \text{pad}) \text{ XOR } \text{opad} \mid \text{Hash}((\text{Key} \mid \text{pad}) \text{ XOR } \text{ipad} \mid \text{message}))$$

ここで、

- $\text{opad} = [0x5C]_n$ (外側パッド)、 $\text{ipad} = [0x36]_n$ (内側パッド)
- $[X]_n$ は X を n 回繰り返すことを表し、 n は基礎となるハッシュ関数データブロックのバイトサイズと等しい値です (ブロックサイズが 512 ビットの場合は $n = 64$)。
- pad は、上記で定義された長さ n までキーを拡張するために必要な 0 の連続です。キー長が n より大きい場合、アプリケーションはまず $\text{Hash}()$ 関数を使用してキーをハッシュ処理して、その結果のバイト文字列を実際のキーとして HMAC に使用する必要があります。
- \mid は連結演算子を表します。

注： ハッシュプロセッサの HMAC モードは、サポートされているすべてのアルゴリズムで使用できます。

HMAC 処理

HMAC を計算するには、4 つの各手順が必要です。

1. ソフトウェアで、MODE ビットをセットし、ALGO ビットで目的のアルゴリズムを選択して、INIT ビットをセットします。使用するキーの長さが ブロックサイズを超える場合、LKEY ビットもセットする必要があります。この場合、HMAC の仕様で必要とされているように、ハッシュプロセッサは、実際のキーではなくキーのハッシュを使用します。
2. ソフトウェアで、メッセージ文字列のロードと同じメカニズムを使用して、内側ハッシュ関数に使用するキーを提供します。つまり、キーデータを HASH_DIN レジスタに書き込んでから、DCAL ビットと正しい NBLW を HASH_STR レジスタにセットして転送を完了します。
3. プロセッサが再びレディ (HASH_SR の DINIS = 1) になると、ソフトウェアはメッセージ列を HASH_DIN に書き込めるようになります。最後のブロックの最後のワードを入力し、ソフトウェアで HASH_STR レジスタの DCAL ビットをセットするとき、メッセージ長がブロックサイズの正確な倍数でない場合は、NBLW ビットフィールドを同時にゼロ以外の値にプログラムする必要があります。[セクション 35.4.5: メッセージダイジェストの計算](#)の説明のように、メッセージ列を提供するのに DMA を使用することもできます。
4. プロセッサが再びレディ (HASH_SR の DINIS = 1) になると、ソフトウェアで、外側のハッシュ関数に使用するキーのデータを HASH_DIN レジスタに書き込むことによってキーを提供し、その後、DCAL ビットをセットして正しい NBLW を HASH_STR レジスタにプログラムすることで転送を完了します。HMAC の結果は、DCIS ビットがセットされるとすぐに、有効な出力レジスタ (HASH_HRx) で確認できます。

注： HMAC 初期計算遅延は、[セクション 35.6: HASH 処理時間](#)に記載しているとおりキーとメッセージの長さによって変わります。

エンディアン形式の管理の詳細については、[セクション 35.4.4: メッセージデータの供給](#)を参照してください。

HMAC の例

以下は、NIST によって指定されている HMAC SHA-1 アルゴリズム (HASH_CR で ALGO = 00 および MODE = 1) の例です。SHA-1 のブロックサイズは 64 バイトです。

オリジナルメッセージが ASCII バイナリコード形式の「Sample message for keylen = blocklen」で、長さ L = 34 バイトであると仮定します。HASH がスワップなしモードでプログラムされている場合 (HASH_CR で DATATYPE = 00)、次のデータを HASH_DIN レジスタに連続でロードする必要があります。

1. NIST によって指定されている**内側ハッシュキー**入力 (長さ = 64、つまりパディング不要)。キーの長さ = 64 であるため、HASH_CR レジスタの LKEY ビットはクリアされます。

```
00010203 04050607 08090A0B 0C0D0E0F 10111213 14151617
18191A1B 1C1D1E1F 20212223 24252627 28292A2B 2C2D2E2F
30313233 34353637 38393A3B 3C3D3E3F
```

2. **メッセージ**入力 (長さ = 34、つまりパディング必須)。HASH_STR を 0x20 にセットして、メッセージのパディングと内側のハッシュ計算を開始する必要があります ('U'は無視とみなします)。

```
53616D70 6C65206D 65737361 67652066 6F72206B 65796C65
6E3D626C 6F636B6C 656EUUUU
```

3. **外側ハッシュキー**入力 (長さ = 64、つまりパディング不要)。ここで、内部ハッシュキーと同一のキーを入力します。

4. **最終外部ハッシュ計算**が、HASH によって実行されます。以下のように、HMAC-SHA1 ダイジェストの結果が HASH_HRx レジスタで入手できます。

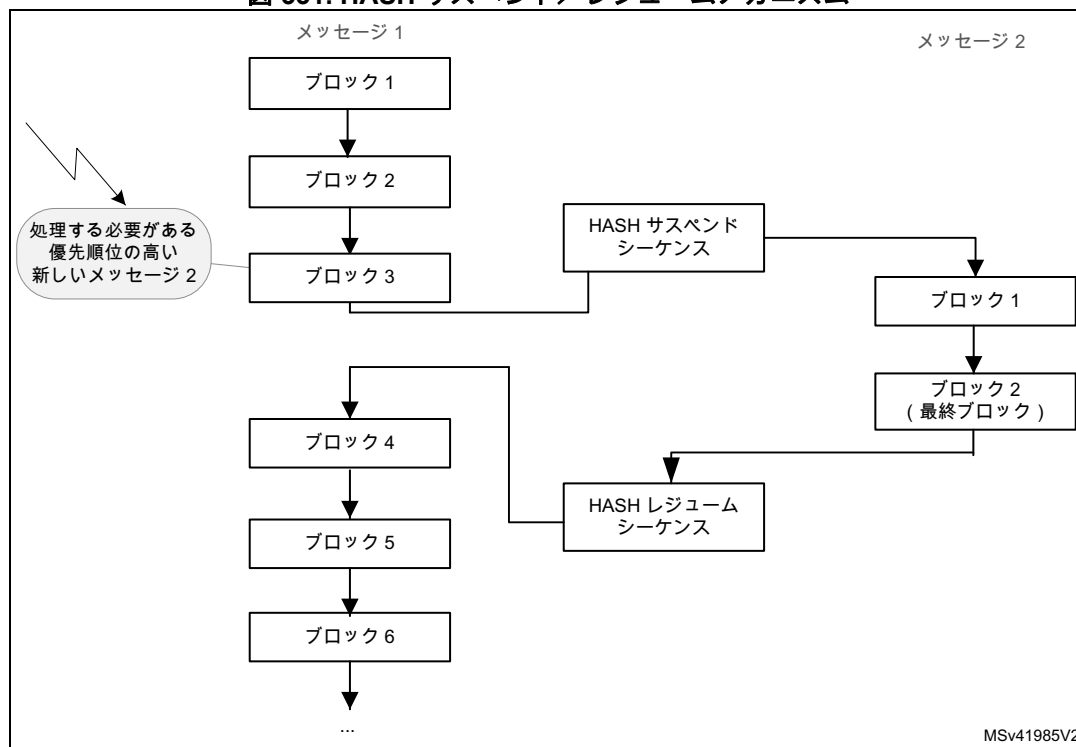
```
HASH_HR0 = 0x5FD596EE
HASH_HR1 = 0x78D5553C
HASH_HR2 = 0x8FF4E72D
HASH_HR3 = 0x266DFD19
HASH_HR4 = 0x2366DA29
```

35.4.8 HASH サスペンド／レジューム操作

概要

ハッシュ／HMAC 操作に割り込んでより優先度の高い別の処理を行うことができます。図 331 に示すように、割り込まれた処理は優先度の高いタスクが処理された後で完了します。

図 331. HASH サスペンド／レジュームメカニズム



これを行うには、割り込まれたタスクのコンテキストを HASH レジスタからメモリに保存し、その後 HASH レジスタからメモリに復元する必要があります。

データフローをソフトウェアまたは DMA によって制御する手順を以下に示します。

ソフトウェアによってロードされるデータ

DMA がハッシュプロセッサへのメッセージのロードに使用されていない場合、進行中のブロック処理がないときのみコンテキストを保存できます。

メッセージの処理をサスペンドするには、NBWE に定義されたワード数を書き込んだ後に次の手順に従います。

1. ポーリングモードでは、BUSY = 0 になるまで待ち、DINIS ステータスビットがセットされているかどうかをポーリングします。
割込みモードでは、DINIS 割込みハンドラーで次のステップを実施します（推奨）。
 - HASH_IMR
 - HASH_STR
 - HASH_CR
 - SHA-1 または SHA2-256 が選択されている場合は HASH_CSR0～HASH_CSR37、さらに HMAC 操作が進行中の場合は HASH_CSR38～HASH_CSR53 も
 - SHA2-384 または SHA2-512（切捨てでも、否でも）が選択されている場合は HASH_CSR0～HASH_CSR90、さらに HMAC 操作が進行中の場合は HASH_CSR91～HASH_CSR102 も
2. 次のレジスタの内容をメモリに保存します。

メッセージの処理をレジュームするには、次の手順に従います。

1. メモリに保存した値を、HASH_IMR、HASH_STR、HASH_CR のレジスタに書き込みます。
2. HASH_CR レジスタの INIT ビットをセットすることによって、ハッシュプロセッサを初期化します。
3. メモリに保存した値を、HASH_CSRx レジスタに書き込みます。
4. 処理の割り込まれたところから再開します。

DMA によってロードされるデータ

DMA を使用してメッセージをハッシュプロセッサにロードしている場合、以下に説明するように安全なダイジェスト計算をサスペンドして復元することをお勧めします。

このシーケンスでは、ハッシュペリフェラルに割り当てられた DMA チャンネルは、メッセージ 1 の処理に割り当てられたままになります（図 331 を参照）。

DMA を使用してメッセージの処理をサスペンドするには、次の手順に従います。

1. DMAE ビットをクリアして DMA インタフェースを無効にします。ハッシュペリフェラルによって、現在のバースト転送を完了するために十分なデータが DMA 経由で自動的に取得されます。
2. 最後の DMA 転送が完了するまで待ちます（HASH_SR で DMAS = 0 になるまで）。
3. DMA チャンネルを無効にします。
4. ポーリングまたは割込みモード（推奨）で、ハッシュプロセッサがレディになる（処理中のブロックがない状態）まで待ちます。つまり、HASH_SR で DINIS = 1 になるのを待ちます。HASH_SR で DCIS もセットされている場合、ハッシュ結果は利用可能であり、コンテキストスワッピングは無用です。そうではない場合には、ステップ 5 に進みます。
5. HASH_IMR、HASH_STR、および HASH_CR のレジスタを保存します。使用している SHA アルゴリズムに応じて以下のいくつかの HASH_CSRx レジスタも保存します。
 - SHA-1 または SHA2-256 が選択されている場合は HASH_CSR0～HASH_CSR37、さらに HMAC 操作が進行中の場合は HASH_CSR38～HASH_CSR53 も保存します。
 - SHA2-384 または SHA2-512 が選択されている場合は HASH_CSR0～HASH_CSR90、さらに HMAC 操作が進行中の場合は HASH_CSR91～HASH_CSR102 も保存します。

DMA を使用してメッセージの処理をレジュームするには、次の手順に従います。

1. 再度割り込まれないうちに、引き続きメッセージを最後まで転送するために DMA コントローラを再設定します。
2. メモリに保存した値を HASH_IMR、HASH_STR、および HASH_CR レジスタにプログラムします。
3. HASH_CR レジスタの INIT ビットをセットすることによって、ハッシュプロセッサを初期化します。
4. メモリに保存した値を HASH_CSRx レジスタにプログラムします。
5. DMAE ビットをセットすることにより処理が割り込まれたところから再開します。

35.4.9 HASH DMA インタフェース

HASH は4 ワードのシングルおよび固定の両方の DMA パースト転送をサポートしています。

ハッシュプロセッサには、DMA コントローラに接続するインタフェースが搭載されています。この DMA は、HASH_CR レジスタの DMAE ビットをセットすることによって HASH にデータを書き込むために使用できます。このビットがセットされると、HASH は、HASH_DIN レジスタにブロックを書き込む必要があるたびに、DMA リクエストを開始します。

4 つの 32 ビットワードを受信すると、HASH は自動的に DMA への新しいリクエストをトリガします。詳細については、[セクション 35.4.5：メッセージダイジェストの計算](#)を参照してください。

DMA 転送を開始する前に、ソフトウェアは HASH_DIN レジスタにコピーされる最後のワードでの有効ビット数をプログラムする必要があります。これは HASH_STR レジスタに次の値を書き込むことによって行います。

$$NBLW = Len(Message) \% 32$$
 ここで、「 $x \% 32$ 」は、 x を 32 で割った余りを示します。

HASH_SR レジスタの DMAS ビットから、DMA インタフェース動作に関する情報を得ることができます。このビットは DMAE と共にセットされ、DMAE がクリアされ、DMA 転送中ではない場合にクリアされます。

注： DMAS ビットに関連する割込みはありません。

MDMAT がセットされているとき、転送のサイズは4ワードの倍数でなければなりません。

35.4.10 HASH エラー管理

ハッシュプロセッサによって生成されるエラーフラグはありません。

35.5 HASH 割込み

マスク可能な 2 つの割込みソースが個別に HASH プロセッサによって生成され、次のイベントが通知されます。

- ダイジェスト計算完了 (DCIS)
- データ入力バッファレディ (DINIS)

両方の割込みソースは、同じグローバル割込みリクエスト信号 (hash_it) に接続されており、この信号は、デバイスの割込みコントローラに接続されています。HASH_IMR レジスタのマスクビットを変更することにより、それぞれの割込みソースを個別に有効にしたり無効にしたりすることができます。適切なマスクビットをセットすることで、その割込みが有効になります。

マスク可能な各割込みソースのステータスは HASH_SR レジスタから読み出すことができます。表 330 に、使用可能な機能の概要を示します。

表 330. HASH 割込みリクエスト

項目 (割込みの 略称)	割込みイベント	イベント フラグ	有効制御 ビット	割込みのクリア方法
HASH	ダイジェスト計算完了	DCIS	DCIE	DCIS のクリア、または INIT のセット
	新しいブロックを取得するためのデータ入力 バッファレディ	DINIS	DINIE	DINIS のクリア、または HASH_DIN へ の書き込み

35.6 HASH 処理時間

表 331 に、各動作モードで中間ブロックの処理に必要な時間を示します。

表 331. 処理時間 (クロックサイクル数)

動作モード	ブロックサイズ (バイト単位)	FIFO ロード ⁽¹⁾	計算フェーズ	合計
SHA-1	64	16	66	82
SHA2-224		16	50	66
SHA2-256				
SHA2-384	128	32	66	98
SHA2-512 ⁽²⁾				

1. ブロックをプロセッサにロードするために必要な時間を追加してください。

2. SHA2-512 には、SHA2-512/224 および SHA2-512/256 モードが含まれます。

メッセージ (または HMAC のキー) の最終ブロックを処理するのに必要な時間はこれより長い場合があります。この時間は最終ブロックの長さ (HMAC モードでの) によって異なります。

中間ブロックの処理と比べて以下の倍数増える場合があります。

- ハッシュメッセージは **1~2.5 倍**
- HMAC 入力キーは **~2.5 倍**
- HMAC メッセージは **1~2.5 倍**
- ショートキーの場合の HMAC 出力キーは **~2.5 倍**
- ロングキーの場合の HMAC 出力キーは **3.5~5 倍**

35.7 HASH レジスタ

HASH コアは、いくつかの制御レジスタ、ステータスレジスタ、およびいくつかのメッセージダイジェストレジスタと連携しています。これらのレジスタはすべて 32 ビットワード単位でのみアクセス可能です。それ以外は AHB エラーが発生します。

35.7.1 HASH 制御レジスタ (HASH_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[3:0]				LKEY
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MDMAT	DINNE	NBW[3:0]				Res.	MODE	DATATYPE[1:0]		DMAE	INIT	Res.	Res.
		rw	r	r	r	r	r		rw	rw	rw	rw	rw		

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:17 **ALGO[3:0]** : アルゴリズム選択

このビットは、ハッシュアルゴリズムを選択します。

0000 : SHA-1

0001 : 予約済み

0010 : SHA2-224

0011 : SHA2-256

1100 : SHA2-384

1101 : SHA2-512/224

1110 : SHA2-512/256

1111 : SHA2-512

この選択は、INIT ビットがセットされている場合にのみ有効です。計算中にこのビットフィールドを変更しても影響はありません。

ALGO ビットフィールドが更新され、INIT ビットがセットされると、HASH_SR の NBWE が自動的に 0x11 に更新されます。

ビット 16 **LKEY** : ロングキー選択

HMAC キーがハッシュアルゴリズムに対応するブロックサイズより大きい場合、アプリケーションはこのビットをセットする必要があります (詳細については表 328: サポートされているハッシュアルゴリズムの情報を参照)。たとえば、SHA2-256 の場合、ブロックサイズは 64 バイトです。

0 : HMAC のキーはブロックサイズ以下の長さです (ショートキー)。HASH_DIN に書き込まれた実際のキーの値は、HMAC の計算中に使用されます。

1 : HMAC のキーはブロックサイズより長い (ロングキー)。HMAC の計算中には、実際のキーの代わりにキーのハッシュが使用されます。

この選択は、INIT および MODE ビットがセット (HMAC モードが選択) された場合にのみ有効です。計算中にこのビットを変更しても影響はありません。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **MDMAT** : 多重 DMA 転送

多重 DMA 転送が必要な大容量のファイルをハッシングするときに、このビットをセットします。

0 : DCAL は DMA 転送の最後に自動的にセットされます。

1 : DCAL は DMA 転送の最後に自動的にセットされません。

ビット 12 **DINNE** : DIN は空ではありません。

DINNE ビットの説明については、HASH_SR の DINNE ビットを参照してください。
このビットは読み出し専用です。

ビット 11:8 **NBW[3:0]** : 既にプッシュされたワード数

NBW[3:0] ビットフィールドの説明については、HASH_SR の NBWP[3:0] ビットフィールドを参照してください。
このビットは読み出し専用です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **MODE** : モード選択

このビットによって、選択したアルゴリズムに対して通常のモードまたは鍵付き HMAC モードを選択します。

0 : ハッシュモードを選択します

1 : HMAC モードを選択します。使用するキーの長さがアルゴリズムのブロックサイズを超える場合、LKEY ビットをセットする必要があります。

この選択は、INIT ビットがセットされている場合にのみ有効です。計算中にこのビットを変更しても影響はありません。

ビット 5:4 **DATATYPE[1:0]** : データ型選択

このビットフィールドでは、HASH_DIN レジスタに入力するデータのフォーマットを定義します。

00: 32 ビットデータ。HASH_DIN に書き込まれたデータは順序を変えずに直接 HASH 処理に使用されます。

01: 16 ビットデータ、すなわちハーフワード。HASH_DIN に書き込まれたデータは 2 つのハーフワードとみなされ、HASH 処理に使用される前にスワップされます。

10: 8 ビットデータ、すなわちバイト。HASH_DIN に書き込まれたデータは 4 バイトとみなされ、HASH 処理に使用される前にスワップされます。

11: ビットデータ、またはビット列。HASH_DIN に書き込まれたデータは 32 ビット（ビット位置 0 におけるデータ列の最初のビット）とみなされ、HASH 処理（ビット位置 31 におけるデータ列の最初のビット）に使用する前にスワップされます。

ビット 3 **DMAE** : DMA 有効化

0 : DMA 転送は無効です

1 : DMA 転送は有効です。DMA リクエストは、ハッシュコアがデータを受け取る準備ができるとすぐに送られます。

このビットがセットされた後、メッセージの最後のデータがハッシュプロセッサに書き込まれている間、ハードウェアによってクリアされます。

DMA 転送中にこのビットをクリアしても、現在の転送は中止されません。その代わりに、転送が完了するか INIT がセットされるまで、HASH の DMA インタフェースは内部的に有効になっています。

INIT ビットをセットしても、DMAE ビットはクリアされません。

ビット 2 **INIT** : メッセージダイジェスト計算の初期化

このビットをセットするとハッシュプロセッサコアがリセットされるので、HASH は新しいメッセージのメッセージダイジェストを計算することが可能となります。

このビットをクリアしても影響はありません。このビットを読み出すと常に 0 が返されます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

35.7.2 HASH データ入力レジスタ (HASH_DIN)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

HASH_DIN はデータ入力レジスタです。このレジスタは 32 ビット幅です。このレジスタを使用して、ハッシュアルゴリズムによって定義された（詳細は、表 328 : サポートされているハッシュアルゴリズムの情報を参照）ブロック毎にメッセージを入力します。たとえば、SHA2-256 の場合、ブロックサイズは 64 バイトです。

HASH_DIN レジスタがプログラムされると、AHB バス上にある値がハッシュコアに「プッシュ」され、このレジスタには AHB バス上の新しい値が入ります。正しいメッセージ形式を取得するには、事前に HASH_CR レジスタの DATATYPE ビットを設定する必要があります。

HASH_DIN レジスタにブロック全体が書き込まれると、以下のように中間ダイジェストの計算が開始されます。

- DMA を使用していない場合は、次のブロックの最初のデータを HASH_DIN レジスタに書き込むことによって開始されます
- DMA を使用している場合は自動的に開始されます

HASH_DIN レジスタに最後のブロックが書き込まれると、HASH_STR レジスタの DCAL ビットをセットすること（最終ダイジェストの計算）によって最終ダイジェストの計算（パディングを含む）が開始されます。この動作は、DMA を使用し、かつ MDMAT ビットがクリアされている場合は自動的に開始されます。

HASH_DIN レジスタを読み出すと 0 が返されます。

注 : HASH がビジーの場合、ダイジェスト計算（中間または最終）が完了していないと、HASH_DIN レジスタへの書き込みアクセスによって AHB バスが停止する可能性があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAIN[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 DATAIN[31:0] : データ入力

このレジスタに書き込むと、レジスタの現在の内容が FIFO にプッシュされ、レジスタには AHB バス上の新しい値が入ります。
このレジスタを読み出すと、0 が返されます。

35.7.3 HASH スタートレジスタ (HASH_STR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

HASH_STR レジスタには 2 つの機能があります。

- ハッシュプロセッサに入力されたメッセージの最後のワードにおける有効ビット数を定義します (HASH_DIN レジスタに書き込まれた最後のデータの有効最下位ビット数)。
- DCAL ビットをセットすることによりメッセージの最後のブロックの処理を開始します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW[4:0]				
							rw				rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DCAL** : ダイジェスト計算

このビットをセットすると以前書き込まれた NBLW の値を使用してメッセージのパディングが開始され、INIT ビットが最後にセットされてから入力 FIFO に書き込まれたすべてのデータワードを使用して最終メッセージダイジェストの計算が開始されます。
このビットを読み出すと 0 が返されます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **NBLW[4:0]** : 最後のワードの有効ビット数

メッセージのビット列の最後のワードが HASH_DIN レジスタに書き込まれると、ハッシュプロセッサは内部データスワッピング後に、以下に指定された有効ビットのみを取得します。

0x00 : 書き込まれた最後のデータのすべての 32 ビット (すなわち、M[31:0]) が有効なメッセージビットです。

0x01 : 書き込まれた最後のデータ (スワッピング後) の 1 ビット (すなわち、M[0]) だけが有効です。

0x02 : 書き込まれた最後のデータ (スワッピング後) の 2 ビット (すなわち、M[1:0]) だけが有効です。

0x03 : 書き込まれた最後のデータ (スワッピング後) の 3 ビット (すなわち、M[2:0]) だけが有効です。

.....

0x1F : 書き込まれた最後のデータ (スワッピング後) の 31 ビット (すなわち、M[30:0]) だけが有効です。

上記メカニズムは、DCAL = 0 のときのみ有効です。DCAL がセットされているときに NBLW ビットが書き込まれた場合、NBLW ビットフィールドは変更されません。つまり、同時に NBLW を設定しつつ DCAL をセットすることはできません。

NBLW ビットを読み出すと最後に NBLW に書き込まれた値が返されます。

35.7.4 HASH ダイジェストレジスタ

これらのレジスタには、次のように定義されたメッセージダイジェストの計算結果が入っています。

- HASH_HR0、HASH_HR1、HASH_HR2、HASH_HR3、HASH_HR4 のレジスタは、SHA-1 ダイジェストの結果を返します。
- HASH_HR0 から HASH_HR6 のレジスタは、SHA2-224、SHA2-512/224 のダイジェスト結果を返します。
- HASH_HR0 から HASH_HR7 のレジスタは、SHA2-256、SHA2-512/256 のダイジェスト結果を返します。
- HASH_HR0 から HASH_HR11 のレジスタは、SHA2-384 のダイジェスト結果を返します。
- HASH_HR0 から HASH_HR15 のレジスタは、SHA2-512 のダイジェスト結果を返します。

すべてのケースで、ダイジェストの最上位ビットは HASH_HR0[31] に格納され、未使用の HASH_HRx レジスタは 0 として読み出されます。

ハッシュコアが中間ダイジェストまたは最終メッセージダイジェストを計算している (DCIS ビットが 0 に等しい) 最中に、これらのいずれかのレジスタへの読出しアクセスが実行されると、この読出し操作には 0 が返されます。

注： (INIT ビットをセットすることによって) 新しいメッセージのダイジェスト計算を開始するとき、HASH_HRx レジスタは強制的にリセット値になります。
HASH_HR0 から HASH_HR4 のレジスタには、2 つの異なるアドレスを使用してアクセスできます (レジスタエイリアシング)。

HASH エイリアスダイジェストレジスタ x (HASH_HRAx)

アドレスオフセット : $0x0C + x * 0x4$, ($x = 0 \sim 4$)

リセット値 : $0x0000\ 0000$

HASH_HRAx レジスタの内容は、アドレスオフセット $0x310$ にある HASH_HRx レジスタの内容と同じです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 Hx[31:0] : ハッシュデータ x

[セクション 35.7.4 : HASH ダイジェストレジスタ](#)の概要を参照してください。

HASH ダイジェストレジスタ x (HASH_HRx)アドレスオフセット : $0x310 + x * 0x4$ 、($x = 0 \sim 4$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 Hx[31:0] : ハッシュデータ x

[セクション 35.7.4 : HASH ダイジェストレジスタ](#)の概要を参照してください。**HASH 補助ダイジェストレジスタ x (HASH_HRx)**アドレスオフセット : $0x310 + x * 0x4$ 、($x = 5 \sim 15$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Hx[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hx[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 Hx[31:0] : ハッシュデータ x

[セクション 35.7.4 : HASH ダイジェストレジスタ](#)の概要を参照してください。

35.7.5 HASH 割込み有効レジスタ (HASH_IMR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DCIE** : ダイジェスト計算完了割込みイネーブル

0 : ダイジェスト計算完了割込みは無効です

1 : ダイジェスト計算完了割込みは有効です

ビット 0 **DINIE** : データ入力割込みイネーブル

0 : データ入力割込みは無効です

1 : データ入力割込みは有効です

35.7.6 HASH ステータスレジスタ (HASH_SR)

アドレスオフセット : 0x24

リセット値 : 0x0011 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBWE[4:0]				
											r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINNE	Res.	NBWP[4:0]					Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
r		r	r	r	r	r						r	r	rc_w0	rc_w0

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **NBWE[4:0]** : 期待されたワード数

このビットフィールドは、部分計算をトリガするために FIFO にプッシュする必要のあるメッセージ内のワード数を示します。NBWE は、HASH_DIN レジスタに書き込みアクセスが行われると値が 1 だけ減少します。

NBWE は、HASH_CR で INIT ビットがセットされると、期待されるブロックサイズ +1 (ワード単位) (0x11) に設定されます。そして、部分ダイジェスト計算が終了すると、期待されるブロックサイズ (0x10) に設定されます。

ビット 15 **DINNE** : DIN は空ではありません。

このビットは HASH_DIN レジスタに有効なデータがある場合 (少なくとも 1 度は書き込まれたあと) にセットされます。このビットは INIT ビット (初期化) または DCAL ビット (前回のメッセージ処理の完了) のどちらかがセットされるとクリアされます。

0 : データ入力バッファにデータがありません

1 : 入力バッファには少なくとも 1 ワードのデータがあります

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:9 **NBWP[4:0]** : 既にプッシュされたワード数

このビットフィールドは、既に FIFO にプッシュされているメッセージ内の正確なワード数です。NBWP は、HASH_DIN レジスタに書き込みアクセスが行われると値が 1 だけ増加します。

ダイジェスト計算が始まると、NBWP は NBWP - ブロックサイズ (ワード単位) に更新され、INIT ビットがセットされると NBWP は 0 になります。

ビット 8:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **BUSY** : ビジービット

0 : 現在処理中のブロックはありません。

1 : ハッシュコアが処理中のデータブロックがあります。

ビット 2 **DMAS** : DMA ステータス

このビットから、DMA インタフェース動作に関する情報を得ることができます。このビットは DMAE と共にセットされ、DMAE = 0、かつ DMA 転送中ではない場合にクリアされます。このビットに関連する割り込みはありません。

0 : DMA インタフェースは無効 (DMAE = 0)、かつ転送中ではありません。

1 : DMA インタフェースは有効 (DMAE = 1)、または転送中です。

ビット 1 **DCIS** : ダイジェスト計算完了割り込みステータス

このビットは、ダイジェストがレディになったとき (メッセージ全体が処理済)、ハードウェアによってセットされます。このビットは、そこに 0 を書き込むか HASH_CR レジスタの INIT ビットをセットすることによってクリアされます。

0 : HASH_HRx レジスタに利用可能なダイジェストはありません (ゼロが返されます)。

1 : ダイジェスト計算完了。HASH_HRx レジスタに利用可能なダイジェストがあります。HASH_IMR レジスタの DCIE ビットがセットされている場合、割り込みが生成されます。

ビット 0 **DINIS** : データ入力割り込みステータス

このビットは、FIFO が新しいブロックを取得する準備ができたときに (空き領域が 16 か所) ハードウェアによってセットされます。このビットは、そこに 0 を書き込むか HASH_DIN レジスタに書き込みを行うことによってクリアされます。

0 : 入力バッファの空き領域は 16 か所未満です。

1 : 入力バッファに新しいブロックを取り込むことができます。HASH_IMR レジスタの DINIE ビットがセットされている場合、割り込みが生成されます。

DINIS = 0 のとき、HASH_CSRx レジスタは 0 として読み出されます。

35.7.7 HASH コンテキストスワップレジスタ

ハッシュプロセッサの内部レジスタステータスはすべてこれらのレジスタ内にあります。ハッシュプロセッサがすでに別のタスクによって使用されているとき、優先度の高いタスクがハッシュプロセッサを使用する必要があるため、サスペンド/レジューム動作を実行する必要がある場合に役立ちます。

そのようなことが発生した場合、HASH_CSRx レジスタを読み出し、その値をシステムメモリ空間に保存する必要があります。そうすると、HASH プロセッサは優先権のあるタスクによって使用できるようになります。HASH 計算が完了すると、保存したコンテキストをメモリから読み出し、HASH_CSRx レジスタに書き戻すことができます。

HASH_CSRx レジスタは、DINIS が 1 に等しいときにのみ読み出せます。そうでないときは、ゼロが返されます。

HASH コンテキストスワップレジスタ x (HASH_CSRx)

アドレスオフセット : 0x0F8 + x * 0x4、(x = 0~102)

リセット値 : 0x0022 0002 (HASH_CSR0)

リセット値 : 0x0020 0000 (HASH_CSR2)

リセット値 : 0x0000 0000 (その他)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSx[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSx[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **CSx[31:0]** : コンテキストスワップ x
[セクション 35.7.7 : HASH コンテキストスワップレジスタ](#)の概要を参照してください。

表 332. HASH1 レジスタマップとリセット値

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

36 公開鍵アクセラレータ (PKA)

36.1 概要

PKA (Public Key Accelerator : 公開鍵アクセラレータ) は、特に RSA、ディフィー - ヘルマン、GF (p) (ガロア体) 上の ECC (楕円曲線暗号) に関連する暗号公開鍵の基本要素の計算を目的としています。手頃なコストで高いパフォーマンスを実現するために、これらの演算はモンゴメリ定義域で実行されます。

特定の演算に対して、必要な計算はすべてアクセラレータ内で実行されるため、入力や出力の処理のために追加でハードウェア/ソフトウェアで何かを行う必要はありません。

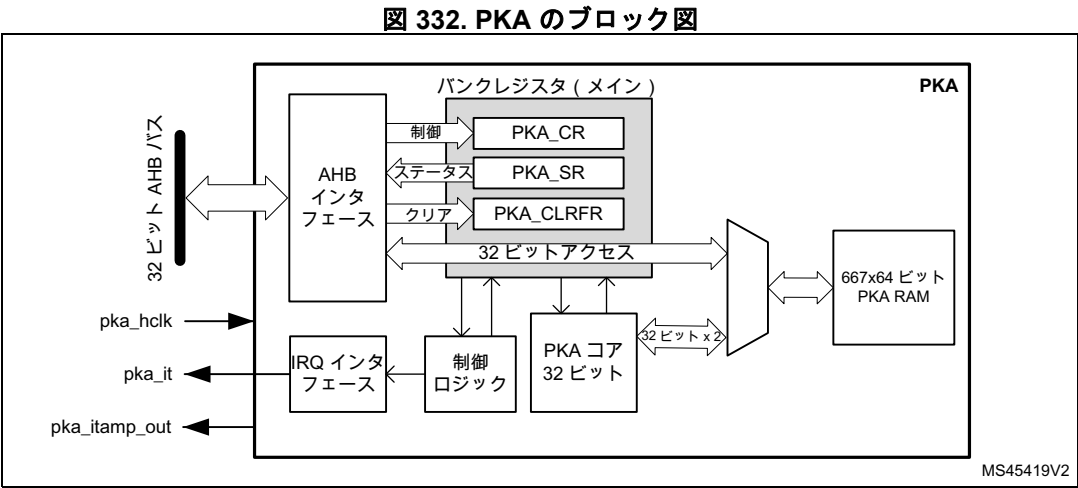
秘密情報を扱う場合に、PKA には差動電力解析 (DPA)、認定済みの SESIP、および PSA セキュリティ保証レベル 3 などの、サイドチャネル攻撃 (SCA) に対する保護が組み込まれています。

36.2 PKA の主な機能

- RSA、DH、GF (p) での ECC の演算の高速化は、高速剰余乗算に関するモンゴメリ法に基づきます。具体的には次のとおりです。
 - RSA べき剰余、RSA 中国剰余定理 (CRT) べき乗
 - ECC スカラー乗算、曲線上の点の確認、完全加算、ダブルベーススラダー、射影からアフィンへ
 - ECDSA 署名の生成および検証
- RSA/DH では最大 4160 ビット、ECC では最大 640 ビットの実数値を処理可能
- 秘密情報を扱う場合：差動電力解析 (DPA)、認定済みの SESIP、および PSA セキュリティ保証レベル 3 などの、サイドチャネル攻撃 (SCA) に対する保護
 - べき剰余、ECC スカラー乗算、ECDSA 署名生成に適用可能
- 加算、減算、乗算、モジュラリダクション、モジュラ逆数、比較、モンゴメリ乗算などの算術演算および剰余演算
- モンゴメリ定義域内外への変換を搭載
- 32 ビットワードのシングルアクセスでのみアクセス可能な AMBA AHB スレーブペリフェラル (それ以外は AHB バスエラーが発生し、書き込みアクセスは無視されます)

36.3 PKA の機能説明

36.3.1 PKA ブロック図



36.3.2 PKA 内部信号

表 333 に、PKA レベルで使用可能な内部信号をリストしています。これらの信号は製品接続パッドでは必ずしも使用できるわけではありません。

表 333. 内部入力／出力信号

信号名	信号タイプ	説明
pka_hclk	デジタル入力	AHB バスクロック
pka_it	デジタル出力	公開鍵アクセラレータ IP グローバル割込みリクエスト
pka_itamp_out	デジタル出力	TAMP への PKA 内部改ざんイベント信号（XOR 演算されます）。PKA が秘密情報を操作しているときに予期しない障害が発生した場合、またはプログラムされた入力ポイントが入力曲線上で見つからなかった（ECDSA 署名および ECC スカラー乗算のみ）場合にトリガされます。 この信号は、障害が検出されるとすぐにアサートされます。アサートされると、PKA レジスタへの読取りアクセスは 0 にリセットされ、書込みは無視されます。 PKA メモリがクリアされると、この信号はデアサートされます。

36.3.3 PKA のリセットおよびクロック

PKA は、AHB バスクロックでクロック供給されます。PKA ペリフェラルのリセット信号がリリースされると、PKA_RAM は 667 クロックサイクルをかけて自動的にクリアされます。この時間の間、PKA_CR レジスタの EN ビットのセットは無視されます。

デバイスに適用されるセキュリティポリシーに従って、改ざんイベントの後に PKA RAM をリセットすることもできます。「システムセキュリティ」セクションの「改ざんの検出と対応」を参照してください（本製品に該当する場合）。

36.3.4 PKA 公開鍵の高速化

概要

公開鍵アクセラレータ (PKA) は、リベスト、シャミア、エーデルマン (RSA)、ディフィー - ヘルマン (DH)、および素体上の ECC の演算を高速化するために使用されます。サポートされるオペランドサイズは、RSA および DH では最大 4160 ビット、ECC では最大 640 ビットです。

PKA は、素体上で定義されたすべての非特異楕円曲線をサポートします。これは、短いワイエルシュトラス方程式 $y^2 = x^3 + ax + b \pmod{p}$ で記述できます。詳細については、[セクション 36.5.1: サポートされている楕円曲線](#)を参照してください。

注： 2 値曲線、エドワーズ曲線、および Curve25519 は、PKA ではサポートされていません。

PKA RAM と呼ばれる 5336 バイト (64 ビットの 667 ワード) のメモリが、PKA への初期データ供給と、計算完了後の結果の保持に使用されます。PKA AHB インタフェースでアクセスします。

PKA 演算モード

PKA が実行できる演算のリストを、整数算術機能と素体 (Fp) 楕円曲線機能についてそれぞれ [表 334](#) および [表 335](#) に示します。

表 334. PKA 整数算術機能リスト

PKA_CR.MODE[5:0]		実行される演算	参照
16 進数	2 進数		
0x01	000001	モンゴメリパラメータ計算 $R2 \bmod n$	セクション 36.4.2
0x0E	001110	剰余加算 $(A+B) \bmod n$	セクション 36.4.3
0x0F	001111	剰余減算 $(A-B) \bmod n$	セクション 36.4.4
0x10	010000	モンゴメリ乗算 $(Ax B) \bmod n$	セクション 36.4.5
0x00	000000	べき剰余 $A^e \bmod n$	セクション 36.4.6
0x02	000010	べき剰余 $A^e \bmod n$ (高速モード)	
0x03	000011	べき剰余 $A^e \bmod n$ (保護)	セクション 36.4.6
0x08	001000	モジュラ逆数 $A^{-1} \bmod n$	セクション 36.4.7
0x0D	001101	モジュラリダクション $A \bmod n$	セクション 36.4.8
0x09	001001	算術加算 $A+B$	セクション 36.4.9
0x0A	001010	算術減算 $A-B$	セクション 36.4.10
0x0B	001011	算術乗算 AxB	セクション 36.4.11
0x0C	001100	算術比較 ($A=B$ 、 $A>B$ 、 $A<B$)	セクション 36.4.12
0x07	000111	RSA CRT べき乗	セクション 36.4.13

表 335. PKA 素体 (Fp) 楕円曲線機能リスト

PKA_CR.MODE[5:0]		実行される演算	参照
16 進数	2 進数		
0x28	101000	楕円曲線 Fp 上の点の確認	セクション 36.4.14
0x20	100000	ECC スカラー乗算 kP (保護)	セクション 36.4.15
0x23	100011	ECC 完全加算	セクション 36.4.18
0x24	100100	ECDSA 署名 (保護)	セクション 36.4.16
0x26	100110	ECDSA 検証	セクション 36.4.17
0x27	100111	ECC ダブルベーススラダー	セクション 36.4.19
0x2F	101111	ECC 射影からアフィンへ	セクション 36.4.20

これらの演算モードにはそれぞれ関連するコードがあり、PKA_CR レジスタの MODE フィールドに書き込む必要があります。アプリケーションが上記に記載されていない値を選択すると、MODE ビットフィールドへの書き込みは無視され、演算エラー (OPERRF) がトリガされます。これが起きた場合、エラーをクリアした後、新しい演算を選択する必要があります。

[表 334](#) および [表 335](#) の一部の演算は保護されていると表示されます。これらの演算は、秘密鍵を操作するとき (RSA 復号でのべき剰余、ECC でのスカラー乗算および署名) に使用されます。これらの秘密情報 (サイドチャネル攻撃から保護されている) は、保護された演算の終了 (BUSY がローになる) 時に PKA RAM から自動的に消去されます。また、それらはサイドチャネル攻撃から保護されます。

注意 : セキュリティ上の理由から、RSA 復号を実行する場合は、保護されたべき剰余演算 (MODE = 0x3) を選択することが非常に重要です。

モンゴメリ空間と高速モードの演算

効率面の理由で、PKA はモンゴメリ定義域でモジュラ乗算演算を内部で実行し、自動的に内外への変換を実行します。

モンゴメリパラメータ計算は時間がかかるため、アプリケーションでは演算開始前に、事前に計算されたモンゴメリパラメータを供給する高速な演算モードを使用することを決定できます。パフォーマンスの改善については、[セクション 36.5.2 : 計算時間](#)で説明します。

高速モードを使用する演算は、べき剰余 (MODE = 0x02) だけです。

36.3.5 PKA の標準的なアプリケーション

概要

PKA は、多数の公開鍵暗号化機能の高速化に使用できます。特に、

- RSA の暗号化と復号
- RSA のキー完成
- CRT-RSA の復号
- DSA と ECDSA 署名の生成および検証
- DH および ECDH のキーの合意

上記の機能の仕様については、次の公報に示されています。

- FIPS PUB 186-4、デジタル署名標準 (DSS)、2013 年 7 月、NIST 発行
- PKCS #1、RSA 暗号標準 v1.5、v2.1 および v2.2、RSA ラボ発行
- IEEE1363-2000、公開鍵暗号向け IEEE 標準仕様、2000 年 1 月
- ANSI X9.62-2005、金融サービス業界向け公開鍵暗号、楕円曲線デジタル署名アルゴリズム (ECDSA)、2005 年 11 月

主な機能の原理については、このセクションで説明しています。詳細については、上記に引用した資料を参照してください。

RSA のキーペア

次の RSA 演算では、公開鍵と秘密鍵の情報が以下のように定義されています。

- アリスが公開鍵 (n , e) をボブに送信します。数値 n および e は非常に大きな正の整数です。
- アリスは自身の秘密鍵 d (これも非常に大きな正の整数) を公開しないでおきます。その代わりに、この秘密鍵を 5 つのエレメント (p , q , dp , dq , $qInv$) で表すこともできます。

上記の表現の詳細については、RSA の仕様を参照してください。

RSA の暗号化と復号の原理

PKCS#1 仕様で推奨されるように、ボブはアリスの公開鍵 (n , e) を使用してメッセージ M を暗号化するために、次の手順で進める必要があります。

1. エンコードされたメッセージ $EM = \text{ENCODE}(M)$ を計算します。ここで ENCODE はエンコードメソッドを指します。
2. EM を整数 m ($0 \leq m < n$) に変換します。(m , n) は互いに素です。
3. 暗号文 $c = m^e \bmod n$ を計算します。
4. 整数 c を文字列の暗号文 C に変換します。

アリスは自身の秘密鍵 d を使用して暗号文 c を復号するために、以下に示した手順に従います。

1. 暗号文 C を整数の暗号文を表す c に変換します。
2. 必要に応じて、(n , e , d) の情報を使用して素因数 (p , q) を取得し、 $\phi = (p - 1) * (q - 1)$ を計算します。詳細は、NIST SP800-56B 付録 C を参照してください。
3. 平文 $m = c^d \bmod n = (m^e)^d \bmod n$ を復元します。秘密鍵が 5 つのエレメント (p , q , dp , dq , $qInv$) である場合、平文 m は、次の演算を実行して得られます。
 - a) $m_1 = c^{dp} \bmod p$
 - b) $m_2 = c^{dq} \bmod q$
 - c) $h = qInv(m_1 - m_2) \bmod p$
 - d) $m = m_2 + hq$
4. 整数メッセージを表す m を、エンコードされたメッセージ EM に変換します。
5. メッセージ $M = \text{DECODE}(EM)$ を復元します。ここで DECODE はデコードメソッドを指します。

上記の操作は、PKA で **べき剰余** $A^e \bmod n$ (秘密鍵が d である場合)、または **RSA CRT べき乗** (秘密鍵が 5 つのエレメント (p , q , dp , dq , $qInv$) である場合) を使用することにより高速化できます。

注: メッセージと整数のデコード操作と変換操作は、PKCS#1 標準で指定されています。

注: 復号プロセスでは、セキュリティ上の理由から、べき剰余演算の保護されたバージョン (MODE = 0x3) を強くお勧めします。暗号化プロセスでは秘密キーの知識が必要なため、MODE = 0x3 は使用できません。

楕円曲線の選択

次の ECC 演算では、曲線パラメータが以下のように定義されます。

- 曲線は関係者（アリスとボブ）が合意した楕円曲線フィールドに対応します。サポートされる曲線パラメータの概要を[セクション 36.5.1：サポートされている楕円曲線](#)に示しています。
- G は、大きな素数位数 n を持つ選択された楕円曲線の基点（ジェネレータとも言います）です（例： $n \times G = \text{単位元 } O$ ）。

ECDSA メッセージの署名生成

ECDSA（楕円曲線デジタル署名アルゴリズム）署名の生成機能の原理は次のとおりです。アリスはメッセージ m に自身の秘密鍵の整数 d_A を使用して署名するために、以下の手順を実行します。

1. $e = \text{HASH}(m)$ を計算します。ここで HASH は暗号ハッシュ関数を指します。
2. e の左端の L_n ビットを z とします。ここで、 L_n は群位数 n のビット長を指します。
3. 暗号として安全なランダムな整数 k を選択します ($0 < k < n$)。
4. 曲線の点 $(x_1, y_1) = k \times G$ を計算します。
5. $r = x_1 \bmod n$ を計算します。 $r = 0$ の場合、手順 3 に戻ります。
6. $s = k^{-1}(z + rd_A) \bmod n$ を計算します。 $s = 0$ の場合、手順 3 に戻ります。
7. 署名はペア (r, s) となります。

手順 4～7 は PKA が次のいずれかを使用して高速化します。

- [ECDSA 署名](#)
- 以下の演算すべて：
 - [ECC Fp スカラー乗算](#) $k \times P$
 - [モジュラリダクション](#) $A \bmod n$
 - [モジュラ逆数](#) $A^{-1} \bmod n$
 - [剰余加算および剰余乗算とモンゴメリ乗算](#)

ECDSA 署名の検証

ECDSA（楕円曲線デジタル署名アルゴリズム）署名の検証機能の原理は次のとおりです。ボブはアリスの署名を認証するために、彼女の公開鍵の曲線の点 Q_A のコピーを手に入れる必要があります。

ボブは、次の手順で Q_A が有効な曲線の点であることを検証できます。

1. Q_A が単位元 O と等しくないことを確認します。
2. Q_A が合意した曲線上にあることを確認します。
3. $n \times Q_A = O$ であることを確認します。

続いて、ボブは以下に示した手順に従います。

1. r と s が $[1, n-1]$ 内で整数であることを検証します。
2. $e = \text{HASH}(m)$ を計算します。ここで HASH は合意した暗号ハッシュ関数を指します。
3. e の左端の L_n ビットを z とします。
4. $w = s^{-1} \bmod n$ を計算します。
5. $u_1 = zw \bmod n$ および $u_2 = rw \bmod n$ を計算します。
6. 曲線の点 $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$ を計算します。
7. $r = x_1 \bmod n$ であれば、署名は有効です。そうでなければ、無効です。

手順 4～7 は PKA が [ECDSA 検証](#) を使用して高速化します。

36.3.6 演算を行うための PKA の手順

PKA 有効化／無効化

PKA_CR レジスタで EN ビットを 1 にセットすると、PKA ペリフェラルが有効になります。PKA_SR の INITOK ビットがセットされると、PKA は使用可能になります。EN = 0 の場合、PKA ペリフェラルはリセット状態で維持され、PKA メモリはアプリケーションによって AHB インタフェース経由で引き続きアクセス可能です。

注： PKA がメモリをクリアしている最中は、EN ビットをセットできません。

注： PKA_CR の EN ビットをセットするときは、MODE ビットフィールドの値が許可された PKA 演算に対応していることを確認してください（[セクション 36.3.7 の OPERRF](#) を参照）。

計算中に EN ビットを 0 にクリアすると、演算がアボートされます。この場合、PKA モードが 0x03、0x20、0x24 の場合を除き、PKA メモリの内容は保証されません。これらの演算では、PKA メモリはアボート後にクリアされ、メモリは 667 サイクルの間、使用できなくなります。このクリア時間中は PKA レジスタのみにアクセスでき、EN ビットへの書き込みは無視されます。

INITOK ビットが 0 のままの場合は、RNG ペリフェラルがクロック供給され、適切に初期化されていることを確認してから、PKA を再度有効にしてみてください。

データフォーマット

PKA RAM での入力データと結果のフォーマットは[セクション 36.4](#)に各演算に対して指定されています。

PKA 演算の実行

サポートされる各 PKA 演算は、次の手順によって実行されます。

1. アドレスオフセット 0x400 にある PKA 内部 RAM に初期データをロードします。
2. PKA_CR レジスタの MODE フィールドに実行する演算を指定して書き込んでから、同じく PKA_CR レジスタで START ビットをアサートします。
3. PKA_SR レジスタの PROCENDF ビットが 1 にセットされ、計算が完了するまで待ちます。
4. PKA 内部 RAM から結果データを読み出し、PKA_CLRFR の PROCENDFC ビットをセットすることによって、PROCENDF ビットをクリアします。

注： PKA がビジー（BUSY = 1）の場合、アプリケーションによる PKA RAM へのアクセスはすべて無視され、PKA_SR のフラグ RAMERRF がセットされます。

手順 2 で不正または不明な演算を選択すると、OPERRF エラーがトリガされ、手順 3（PROCENDF = 1）は発生しません。詳細については、[セクション 36.3.7](#)を参照してください。

事前に計算されたモンゴメリパラメータの使用（PKA 高速モード）

[セクション 36.3.4](#)に示すとおり、同じ係数で多数の演算を行う場合、アプリケーションでは対応するモンゴメリパラメータを一度だけ計算することが有益と考えられます（例：[セクション 36.4.5](#)を参照）。これは「高速モード」と言われています。

高速モードの使用を管理するためには、以下に説明する手順に従うことを推奨します。

1. PKA RAM で係数のサイズと値の情報をロードします。これらの情報は[セクション 36.5.1](#)にまとめられています。
2. PKA_CR レジスタで PKA をモンゴメリパラメータ計算モード（MODE="0x1"）にプログラムして、START ビットをアサートします。

3. PKA_SR レジスタの PROCENDF ビットが 1 にセットされるまで待ってから、PKA メモリから対応するモンゴメリパラメータを読み出します。その後、PKA_CLRFR の PROCENDFC ビットをセットすることにより PROCENDF ビットをクリアします。
4. 必要な PKA 操作を進め、レギュラ入力データの最上部でモンゴメリの情報 $R2 \bmod m$ をロードします。すべてのアドレスは[セクション 36.4](#) に示しています。

36.3.7 PKA エラー管理

PKA の使用時に、次のいくつかのエラーが発生することがあります。

- PKA RAM へのアクセスが予想の範囲外です。この場合、PKA_SR レジスタのアドレスエラーフラグ (ADDRERRF) がセットされます。
- PKA コアが PKA RAM を使用しているときに、PKA RAM への AHB アクセスが発生しました。この場合、RAM エラーフラグ (RAMERRF) が PKA_SR レジスタにセットされ、PKA RAM の読出しにはゼロが返され、書込みは無視されます。
- MODE ビットフィールドを使用して選択した演算モードが PKA 演算モード (またはビットフィールドの説明) にリストされていないか、または PKA が制限モードで実行されています (PKA_SR の LMF ビットを参照)。この場合、演算エラーフラグ (OPERRF) が PKA_SR レジスタにセットされ、MODE ビットフィールドへの書込みは無視されます。

上記の各エラーフラグに対して、アプリケーションが PKA_CR レジスタの対応ビットをセットしていれば、PKA によって割込みが生成されます (詳細は[セクション 36.6](#) を参照してください)。

ADDRERRF、OPERRF、および RAMERRF エラーは、PKA_CLRFR の対応するビットをセットすることによってクリアされます。

PKA は、PKA_CR レジスタで EN ビットをリセットすることでいつでも再初期化できます。

新しい演算を PKA_CR レジスタに書き込む前に、PKA_CLRFR の OPERRFC ビットを使用して OPERRF エラーをクリアする必要があります。

36.4 PKA 演算モード

36.4.1 概要

PKA でサポートされるさまざまな演算について、以下のサブセクションで説明し、PKA RAM に格納される入力データおよび結果のフォーマットを定義します。

警告：	PKA では、すべての入力パラメータが有効であり、相互に一貫性があると想定しているため、演算を開始する前に PKA へのすべての入力パラメータの有効性をチェックする必要があります。入力パラメータは、演算テーブルで指定されたオペランドのサイズを超えてはなりません。
------------	---

以下の情報は、すべての PKA 演算に適用されます。

- PKA コアではその RAM 内で 64 ビットワードを処理します。したがって、これ以降、すべてのワードサイズは 64 ビットです。
- エレメントを PKA の RAM に入力として書き込む場合、すべてのビットを 0 にした追加ワードを最上位の入力ワードの後に追加する必要があります。このルールは、オペランドのサイズが 1 に固定されている場合には適用されません。
- 報告されるすべての RAM ストレージアドレスはデータの最下位ワードを指しており、アプリケーションで使用する実際のアドレスを取得するには、示されたオフセットに PKA のベースアドレスを加える必要があります。
- サポートされるオペランドの「サイズ」は次のとおりです。
 - ROS (RSA のオペランドサイズ) : データサイズは $(rsa_size/64 + 1)$ ワードです。 rsa_size は選択した係数の長さ (ビット単位) です。たとえば、1024 ビットのオペランドサイズで RSA を計算する場合、ROS は 17 ワード (1088 ビット) となります。
 - EOS (ECC のオペランドサイズ) : データサイズは $(ecc_size/64 + 1)$ ワードです。 ecc_size は選択した素数の係数の長さ (ビット単位) です。たとえば、192 ビットのオペランドサイズで ECC を計算する場合、EOS は 4 ワード (256 ビット) となります。
 - ROS および EOS の値には、必要なオール 0 の追加ワードが含まれています。
- 特に指定がない限り、表のすべてのオペランドは整数です。

注 : PKA コアは 64 ビットの複数ワードを処理するため、上記の計算式で端数の結果となった場合は、最も近い整数に繰り上げる必要があります。

注 : 最大 ROS は 66 ワード (4160 ビットの最大指数サイズ) であり、最大 EOS は 11 ワード (640 ビットの最大オペランドサイズ) です。

最初の例として (PKA メモリのエンディアンをよりよく理解するために)、ECC Fp スカラー乗算の演算を準備するために、アプリケーションが ECC P256 曲線の点 P の x 座標 (EOS = 5 ワード) を書き込むときには、最下位ビットをアドレスオフセット 0x578 のビット 0 に、最上位ビットをアドレスオフセット 0x590 のビット 63 に配置する必要があります。次に、上述したように、アプリケーションは空のワード 0x0000000000000000 をアドレスオフセット 0x598 に書き込む必要があります。

2 番目の例として、やはり ECC Fp スカラー乗算の演算を準備するために、アプリケーションが 224 ビットの長さの係数 (つまり 切り上げて 4 つの 64 ビットワードに 1 を加えたもの) を持つ曲線に、 $a = -3$ という情報を書く必要があるとき、以下のデータを PKA メモリに書き込む必要があります。

@RAM+410	0x0000000000000001	/* 余分なワードを含まない曲線係数「a」の符号 */
@RAM+418	0x0000000000000011	/* a の LSB の値 */
@RAM+420	0x0000000000000000
@RAM+428	0x0000000000000000
@RAM+430	0x0000000000000000	/* a の MSB の値 */
@RAM+438	0x0000000000000000	/* すべて 0 の追加のワード */

36.4.2 モンゴメリパラメータ計算

この機能は、PKA で使用されるモンゴメリパラメータ ($R^2 \bmod n$) の計算に使用され、オペランドをモンゴメリ剰余系表現に変換します。この演算は、高速モード演算が使用される場合に非常に役立ちます。この場合、モンゴメリパラメータが入力として渡され、計算時間が節約されるためです。

注 : この演算は ECC 曲線でも使用できます。この場合、素数の係数の長さと EOS サイズを使用する必要があります。

モンゴメリパラメータ計算に関する演算方法の概要を表 336 に示しています。

表 336. モンゴメリパラメータ計算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x01	PKA_CR	6 ビット
	係数の長さ	(ビット単位、 $0 \leq \text{値} < 4160$)	RAM@0x408	64 ビット
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0x1088	ROS
OUT	結果 : $R^2 \bmod n$	-	RAM@0x620	

36.4.3 剰余加算

剰余加算の演算は、 $A + B \bmod n$ の計算式となります。演算方法の概要を表 337 に示しています。

表 337. 剰余加算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0E	PKA_CR	6 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	($0 \leq A < n$)	RAM@0xA50	ROS
	オペランド B	($0 \leq B < n$)	RAM@0xC68	
	係数の値 n	($n < 2^{4160}$)	RAM@0x1088	
OUT	結果 : $A+B \bmod n$	($0 \leq \text{結果} < n$)	RAM@0xE78	

36.4.4 剰余減算

剰余減算の演算は、次の計算式となります。

- $A \geq B$ の場合 : $A - B \bmod n$
- $A < B$ の場合 : $A + n - B \bmod n$

演算方法の概要を表 338 に示しています。

表 338. 剰余減算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0F	PKA_CR	6 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	($0 \leq A < n$)	RAM@0xA50	ROS
	オペランド B	($0 \leq B < n$)	RAM@0xC68	
	係数の値 n	($n < 2^{4160}$)	RAM@0x1088	
OUT	結果 : $A-B \bmod n$	($0 \leq \text{結果} < n$)	RAM@0xE78	

36.4.5 剰余乗算とモンゴメリ乗算

一連の乗算を実行する場合に効率性を高めるため、PKA はモンゴメリ定義域に少なくとも 1 つの入力がある乗算を高速化します。この演算の主な 2 つの使用法は次のとおりです。

- 自然な定義域からモンゴメリ定義域へ、およびその逆の値のマッピング
- 剰余乗算 $A \times B \bmod n$ の実行

上記の演算を実行する方法について、以下に記載します。「x」はこの演算を示し、A、B、C のオペランドは自然な定義域内にあります。

- モンゴメリ定義域への（からの）変換
 - A は自然な定義域内の整数と仮定して：
 - $r2modn$ をモンゴメリパラメータ計算を使って計算します。
 - 結果 $AR = A \times r2modn \bmod n$ は、モンゴメリ定義域内の A となります。
 - BR はモンゴメリ定義域内の整数と仮定して：
 - 結果 $B = BR \times 1 \bmod n$ は、自然な定義域内の B となります。
 - 同様に、a) で計算された値 AR は、 $A = AR \times 1 \bmod n$ を計算することで自然な定義域に変換できます。
- 単純な剰余乗算 $A \times B \bmod n$
 - $r2modn$ をモンゴメリパラメータ計算を使って計算します。
 - $AR = A \times r2modn \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - $AB = AR \times B \bmod n$ を計算します。出力は自然な定義域内にあります。
- 複数の剰余乗算 $A \times B \times C \bmod n$
 - $r2modn$ をモンゴメリパラメータ計算を使って計算します。
 - $AR = A \times r2modn \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - $BR = B \times r2modn \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - $ABR = AR \times BR \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - $CR = C \times r2modn \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - $ABCR = ABR \times CR \bmod n$ を計算します。出力はモンゴメリ定義域内にあります。
 - （オプション）乗算するオペランドがまだある場合は、上の 2 つの手順を繰り返します。
 - $ABC = ABCR \times 1 \bmod n$ を計算して、自然な定義域内で結果を取得します。

モンゴメリ乗算に関する演算方法の概要を表 339 に示しています。

表 339. モンゴメリ乗算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x10	PKA_CR	6 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	($0 \leq A < n$)	RAM@0xA50	ROS
	オペランド B	($0 \leq B < n$)	RAM@0xC68	
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0x1088	
OUT	結果 : $A \times B \bmod n^{(1)}$	-	RAM@0xE78	

1. 入力の性質に応じて、モンゴメリ定義域内または自然な定義域内の結果になります (例 2 と 3 を参照)。

36.4.6 べき剰余

べき剰余の演算は、シングルステップの RSA 演算を実行するために通常使用します。これは、 $A^e \bmod n$ の計算式となります。

公開情報を含む RSA 演算 (RSA 暗号化) では、表 340 および 表 341 に詳述する通常モードまたは高速モードを使用できます。秘密情報を含む RSA 演算 (RSA 復号) では、セキュリティ上の理由から 表 342 に詳述する保護モードを使用する必要があります。

注： この演算が開始されると、PKA 制御レジスタと PKA メモリは使用できなくなります。PKA によって BUSY ビットが 0 にセットされると、アクセスは回復します。

この演算が予期しないハードウェアイベントによるエラーで完了すると、TAMP ペリフェラルに対して PKA 改ざんイベントがトリガされ、ハードウェアによって消去されるまで PKA RAM へのアクセスがブロックされます。

注： $MODE = 0x03$ のとき、エラー出力が $0xD60D$ と異なる場合、PKA によってすべてのメモリの内容がクリアされ、秘密鍵に関する情報の漏えいが回避されます。

べき剰余に関する演算方法の概要を、表 340 (通常モード)、表 341 (高速モード)、および表 342 (保護モード) に示します。高速モードの使用方法は、セクション 36.3.6 に示しています。

表 340. べき剰余 (通常モード)

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x00	PKA_CR	6 ビット
IN	指数の長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	
IN/OUT	オペランド A (べき乗の底)	$(0 \leq A < n)$	RAM@0xC68	ROS
IN	指数 e	$(0 \leq e < n)$	RAM@0xE78	
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0x1088	
OUT	結果 : $A^e \bmod n$	$(0 \leq \text{結果} < n)$	RAM@0x838	

表 341. べき剰余 (高速モード)

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x02	PKA_CR	6 ビット
IN	指数の長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	
IN/OUT	オペランド A (べき乗の底)	$(0 \leq A < n)$	RAM@0xC68	ROS
IN	指数 e	$(0 \leq e < n)$	RAM@0xE78	
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0x1088	
IN/OUT	モンゴメリパラメータ $R2 \bmod n$	(必須)	RAM@0x620	
OUT	結果 : $A^e \bmod n$	$(0 \leq \text{結果} < n)$	RAM@0x838	

表 342. べき剰余 (保護モード)

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x03	PKA_CR	6 ビット
	指数 e の長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	係数またはオペランドの長さ	(ビット単位、null 以外)	RAM@0x408	
	オペランド A (べき乗の底)	($0 \leq A < n$)	RAM@0x16C8	ROS
	指数 e	($0 \leq e < n$)	RAM@0x14B8	
	係数の値 n	(奇数の整数のみ、 $n < 2^{4096}$)	RAM@0x0838	
	Phi 値 ⁽¹⁾	-	RAM@0x0C68	
OUT	結果 : $A^e \bmod n$	($0 \leq \text{結果} < n$)	RAM@0x838	
ERROR	$A^e \bmod n$ のエラー	- エラーなし : 0xD60D - エラー : 0xCBC9	RAM@0x1298	64 ビット

1. $\phi = (p-1) * (q-1)$ による n^1 のオイラーのトーシェント関数。ここで、p と q は係数 n の素因数です (詳細については、NIST SP800-56B 付録 C または [RSA の暗号化と復号の原理](#) を参照)。最適化として、キーペア生成の一部として phi 情報を保持することをお勧めします。別の方法は、キーを (N, e, d, ϕ) ではなく (p, q, e, d) として保存することです。この場合、PKA 算術乗算 $N = p * q$ 、および $\phi = (p-1) * (q-1)$ を使用して N と phi を導出します。

36.4.7 モジュラ逆数

モジュラ逆数演算は、逆数 $A^{-1} \bmod n$ の計算となります。係数 n が素数である場合、A のすべての値 ($1 \leq A < n$) に対してモジュラ逆数の出力は有効です。係数 n が素数ではない場合、A と n の最大公約数が 1 の場合のみ A が逆数を持ちます。

オペランド A が係数 n の約数である場合、結果は n の因数の倍数になります。

モジュラ逆数に関する演算方法の概要を表 343 に示しています。

表 343. モジュラ逆数

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x08	PKA_CR	6 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	($0 \leq A < n$)	RAM@0xA50	ROS
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0xC68	
OUT	結果 : $A^{-1} \bmod n$	$0 < \text{結果} < n$	RAM@0xE78	

36.4.8 モジュラリダクション

モジュラリダクションの演算は、 A を n で割った余りで計算されます。演算方法の概要を表 344 に示しています。

表 344. モジュラリダクション

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0D	PKA_CR	6 ビット
	オペランドの長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	係数の長さ	(ビット単位、 $8 < \text{値} < 4160$)	RAM@0x408	
	オペランド A	$(0 \leq A < 2n < 2^{4160})$	RAM@0xA50	ROS
	係数の値 n	(奇数の整数のみ、 $n < 2^{4160}$)	RAM@0xC68	
OUT	結果 $A \bmod n$	$(0 < \text{結果} < n)$	RAM@0xE78	

36.4.9 算術加算

算術加算の演算は、 $A + B$ の計算式となります。演算方法の概要を表 345 に示しています。

表 345. 算術加算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x09	PKA_CR	6 ビット
	オペランドの長さ M	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	オペランド B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	結果 : $A+B$	$(0 \leq \text{結果} < 2^{M+1})$	RAM@0xE78	ROS + 1

36.4.10 算術減算

算術減算の演算は、次の計算式となります。

- $A \geq B$ の場合 : $A - B$
- $A < B$ かつ $M/32$ の剰余が 0 を超える場合 : $A + 2^{\text{int}(M/32)*32+1} - B$
- $A < B$ かつ $M/32$ の剰余が 0 となる場合 : $A + 2^{\text{int}(M/32)*32} - B$

最後の 2 つの箇条書きでは、結果が負であるため、出力の最上位ワードに続く 32 ビットワードは 0xFFFF FFFF となります。

演算方法の概要を表 346 に示しています。

表 346. 算術減算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0A	PKA_CR	6 ビット
	オペランドの長さ M	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	オペランド B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	結果 : $A-B$	$(0 \leq \text{結果} < 2^M)$	RAM@0xE78	

36.4.11 算術乗算

算術乗算の演算は、 $A \times B$ の計算式となります。演算方法の概要を表 347 に示しています。

表 347. 算術乗算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0B	PKA_CR	6 ビット
	オペランドの長さ M	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	オペランド B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	結果 : $A \times B$	$(0 \leq \text{結果} < 2^M)$	RAM@0xE78	2xROS

36.4.12 算術比較

算術比較の演算は、次の計算式となります。

- $A = B$: 結果 = 0xED2C
- $A > B$: 結果 = 0x7AF8
- $A < B$: 結果 = 0x916A

算術比較に関する演算方法の概要を表 348 に示しています。

表 348. 算術比較

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x0C	PKA_CR	6 ビット
	オペランドの長さ M	(ビット単位、null 以外)	RAM@0x408	64 ビット
	オペランド A	$(0 \leq A < 2^M)$	RAM@0xA50	ROS
	オペランド B	$(0 \leq B < 2^M)$	RAM@0xC68	
OUT	結果 $A ? B$	0xED2C、0x7AF8 または 0x916A	RAM@0xE78	64 ビット

36.4.13 RSA CRT べき乗

効率性のために、OpenSSL RSA のような人気のある数多くの暗号ライブラリでは、中国剰余定理 (CRT) に基づき、復号および署名に次の最適化を使用しています。

- p および q は事前に計算された素数 (秘密鍵の一部として格納)
- $d_p = d \bmod (p-1)$
- $d_q = d \bmod (q-1)$
- $q_{inv} = q^{-1} \bmod p$

これらの値によって、次のようにべき乗 $m = A^d \bmod pq$ をより効率的に計算できます。

- $m_1 = A^{d_p} \bmod p$
- $m_2 = A^{d_q} \bmod q$
- $h = q_{inv} (m_1 - m_2) \bmod p$ (ここでは $m_1 > m_2$)
- $m = m_2 + hq \bmod pq$

CRT べき乗 $A^d \bmod pq$ に関する演算方法の概要を表 349 に示しています。

表 349. CRT べき乗

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x07	PKA_CR	6 ビット
IN	オペランドの長さ	(ビット単位、null 以外)	RAM@0x408	64 ビット
IN	オペランド d_p	$(0 < d_p < 2^{M/2})$	RAM@0x730	ROS/2
	オペランド d_Q	$(0 < d_Q < 2^{M/2})$	RAM@0xE78	
	オペランド q_{inv}	$(0 < q_{inv} < 2^{M/2})$	RAM@0x948	
	素数 $p^{(1)}$	$(0 < p < 2^{M/2})$	RAM@0xB60	
	素数 $q^{(1)}$	$(0 < q < 2^{M/2})$	RAM@0x1088	
IN	オペランド A	$(0 < A < 2^M)$	RAM@0x12A0	ROS
OUT	結果 : $A^d \bmod pq$	$(0 \leq \text{結果} < pq)$	RAM@0x838	

1. 2 以外にする必要があります。

36.4.14 楕円曲線 F_p 上の点の確認

この演算では、指定された点 $P(x, y)$ が素体の式 $y^2 = (x^3 + ax + b) \bmod p$ (ここで a および b は曲線のエレメントです) の曲線に当てはまるものかどうかを確認します。

楕円曲線 F_p 上の点の確認に関する演算方法の概要を表 350 に示しています。

表 350. 楕円曲線 F_p 上の点の確認

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x28	PKA_CR	6 ビット
	係数の長さ	(ビット単位、null 以外、 $8 < \text{値} < 640$)	RAM@0x408	64 ビット
	曲線係数 a の符号	0x0 : 正 0x1 : 負	RAM@0x410	
	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x418	
	曲線係数 b	($ b < p$)	RAM@0x520	EOS
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2640$)	RAM@0x470	
	点 P の座標 x	($x < p$)	RAM@0x578	
	点 P の座標 y	($y < p$)	RAM@0x5D0	
	モンゴメリパラメータ $R2 \bmod n$	-	RAM@0x4C8	
OUT	結果 : 曲線上の点 P	<ul style="list-style-type: none"> 0xD60D : 曲線上の点 0xA3B7 : 曲線上にない点 0xF946 : x または y 座標が係数 p より小さくない 	RAM@0x680	64 ビット

36.4.15 ECC Fp スカラー乗算

この演算は、 $a \times P (x_P, y_P)$ の計算式となります。ここで、 P は素体上の曲線の点であり、「 x 」は楕円曲線のスカラーと点の乗算です。計算結果は、同じ曲線に属する点または無限遠点となります。

ECC Fp スカラー乗算に関する演算方法の概要を表 351 に示しています。

注： この演算が開始されると、PKA 制御レジスタと PKA メモリは使用できなくなります。PKA によって BUSY ビットが 0 にセットされると、アクセスは回復します。

この演算が予期しないハードウェアイベントによるエラーで完了すると、TAMP ペリフェラルに対して PKA 改ざんイベントがトリガされ、ハードウェアによって消去されるまで PKA RAM へのアクセスがブロックされます。PKA 改ざんは、プログラムされた入力ポイントが入力した ECC 曲線上に見つからない場合にもトリガされます。これは、PKA の演算「楕円曲線上の点」を使用することで回避できます。

表 351. ECC Fp スカラー乗算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x20	PKA_CR	6 ビット
IN	曲線の素数位数 n の長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	曲線の係数 p の長さ	(ビット単位、null 以外、 $8 < \text{値} < 640$)	RAM@0x408	
	曲線係数 a の符号	– 0x0 : 正 – 0x1 : 負	RAM@0x410	
IN	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x418	EOS
	曲線係数 b	(正の整数)	RAM@0x520	
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x1088	
	スカラー倍数 k	($0 \leq k < 2^{640}$)	RAM@0x12A0	
	点 P の座標 x_P	($x < p$)	RAM@0x578	
	点 P の座標 y_P	($y < p$)	RAM@0x470	
	曲線の素数位数 n	(整数素数)	RAM@0xF88	
OUT	結果 : $k \times P$ の座標 x'	(結果 $< p$)	RAM@0x578	EOS
	結果 : $k \times P$ の座標 y'	(結果 $< p$)	RAM@0x5D0	
ERROR	$k \times P$ のエラー	– エラーなし : 0xD60D – エラー : 0xCBC9	RAM@0x680	64 ビット

この演算を実行する場合、次の特殊なケースに注意する必要があります。

- $k = 0$ のとき、曲線パラメータ b が 0 以外の場合、この関数は $(0, 0)$ の無限遠点を返し、そうでない場合は $(0, 1)$ を返します。 k が 0 以外の場合、無限遠点が返される場合があります。アプリケーションがこの動作を検出したとき、新しい計算を実行する必要があります。
- $k < 0$ の場合 (すなわち、負のスカラー乗算が必要な場合)、倍数の絶対値 $k = |-k|$ を PKA に提供する必要があります。計算完了後、 $-P = (x, -y)$ の式を使用して、有効な最終結果の y 座標を計算できます (x 座標は同じままです)。

注： エラー出力が 0xD60D と異なる場合、PKA によってすべてのメモリの内容がクリアされ、秘密鍵に関する情報の漏えいが回避されます。

36.4.16 ECDSA 署名

ECDSA 署名演算（[セクション 36.3.5](#) で説明）の概要は [表 352](#)（入力パラメータ）と [表 353](#)（出力パラメータ）に示します。

アプリケーションで出力エラーが 0xD60D かどうかを確認し、異なっている場合は、新しい k を生成して、ECDSA 署名演算を繰り返す必要があります。

注： この演算が開始されると、PKA 制御レジスタと PKA メモリは使用できなくなります。PKA によって BUSY ビットが 0 にセットされると、アクセスは回復します。

この演算が予期しないハードウェアイベントによるエラーで完了すると、TAMP ペリフェラルに対して PKA 改ざんイベントがトリガされ、ハードウェアによって消去されるまで PKA RAM へのアクセスがブロックされます。PKA 改ざんは、プログラムされた入力ポイントが入力した ECC 曲線上に見つからない場合にもトリガされます。これは、PKA の演算「楕円曲線上の点」を使用することで回避できます。

表 352. ECDSA 署名 - 入力

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x24	PKA_CR	6 ビット
	曲線の素数位数 n の長さ ($nlen$)	(ビット単位、null 以外)	RAM@0x400	64 ビット
	曲線の係数 p の長さ	(ビット単位、 $8 < \text{値} < 640$)	RAM@0x408	
	曲線係数 a の符号	0x0 : 正 0x1 : 負	RAM@0x410	
	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x418	EOS
	曲線係数 b	(正の整数)	RAM@0x520	
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x1088	
	整数 $k^{(1)}$	($0 \leq k < 2^{640}$)	RAM@0x12A0	
	曲線の基点 G の座標 x	($x < p$)	RAM@0x578	
	曲線の基点 G の座標 y	($y < p$)	RAM@0x470	
	メッセージのハッシュ z	($nlen$ に等しいハッシュサイズ) ⁽²⁾	RAM@0xFE8	
	秘密鍵 d	($0 < d$)	RAM@0xF28	
	曲線の素数位数 n	(整数素数)	RAM@0xF88	

- 通常、この整数は暗号として安全な乱数ですが、場合によっては決定論的に生成されることがあります。
- ハッシュパラメータのサイズを曲線の素数位数 n の長さと等しくするには、ゼロによるパディングまたはハッシュの切捨てを使用する必要があります。

表 353. ECDSA 署名 - 出力

方向とパラメータ		値 (注)	ストレージ	サイズ
OUT	署名部分 r	($0 < r < n$)	RAM@0x730	EOS
	署名部分 s	($0 < s < n$)	RAM@0x788	
ERROR	署名の結果	- 0xD60D : 計算成功、エラーなし - 0xCBC9 : 計算失敗 - 0xA3B7 : 署名部分 r が 0 - 0xF946 : 署名部分 s が 0	RAM@0xFE0	64 ビット

注： エラー出力が 0xD60D または 0xCBC9 に等しい場合、PKA によってすべてのメモリの内容がクリアされ、秘密鍵に関する情報の漏えいが回避されます。エラー出力が 0xA3B7 または 0xF946 に等しい場合、PKA のメモリの内容は部分的に消去され、エラーコードは読取り可能に保たれます。

拡張 ECDSA サポート

PKA は、拡張 ECDSA 署名もサポートしており、入力および出力は ECDSA 署名と同じ（それぞれ表 352 および表 353）ですが、点 kG の座標が追加されます。この追加出力は表 354 に定義されています。

表 354. 拡張 ECDSA 署名 - 追加出力

方向とパラメータ		値 (注)	ストレージ	サイズ
OUT	曲線の点 kG の座標 x_1	$(0 \leq x_1 < p)$	RAM@0x1400	EOS
	曲線の点 kG の座標 y_1	$(0 \leq y_1 < p)$	RAM@0x1458	

36.4.17 ECDSA 検証

ECDSA 検証演算（セクション 36.3.5 で説明）の概要は表 355（入力パラメータ）と表 356（出力パラメータ）に示します。

アプリケーションで出力エラーが 0xD60D かどうかを確認する必要があり、異なっていた場合は、署名は検証されていません。

表 355. ECDSA 検証 - 入力

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x26	PKA_CR	6 ビット
	曲線の素数位数 n の長さ ($nlen$)	(ビット単位、null 以外)	RAM@0x408	64 ビット
	曲線の係数 p の長さ	(ビット単位、null 以外、 $8 < \text{値} < 640$)	RAM@0x4C8	
	曲線係数 a の符号	0x0 : 正 0x1 : 負	RAM@0x468	
	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x470	EOS
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x4D0	
	曲線の基点 G の座標 x	$(x < p)$	RAM@0x678	
	曲線の基点 G の座標 y	$(y < p)$	RAM@0x6D0	
	公開鍵曲線の点 Q の座標 x_Q	$(x_Q < p)$	RAM@0x12F8	
	公開鍵曲線の点 Q の座標 y_Q	$(y_Q < p)$	RAM@0x1350	
	署名部分 r	$(0 < r < n)$	RAM@0x10E0	
	署名部分 s	$(0 < s < n)$	RAM@0xC68	
	メッセージのハッシュ z	($nlen$ に等しいハッシュサイズ) ⁽¹⁾	RAM@0x13A8	
	曲線の素数位数 n	(整数素数)	RAM@0x1088	

1. ハッシュパラメータのサイズを曲線の素数位数 n の長さと等しくするには、ゼロによるパディングまたはハッシュの切捨てを使用する必要があります。

表 356. ECDSA 検証 - 出力

方向とパラメータ		値 (注)	ストレージ	サイズ
OUT	結果 : ECDSA 検証	- 0xD60D : 有効な署名 - 0xA3B7 : 無効な署名	RAM@0x5D0	64 ビット
	計算された署名部分 r	- ($0 < r < n$)	RAM@0x578	EOS

36.4.18 ECC 完全加算

ECC 完全加算では、楕円曲線上の 2 つの指定された点の加算を計算します。

演算方法の概要を表 357 に示しています。

注 : 2 つの入力点と結果の点は、ヤコビアン座標 (X, Y, Z) で表されます。アフィン座標 (x, y) で点を
入力するには、 $(X, Y, Z) = (x, y, 1)$ 変換を使用することができます。結果の点をヤコビアン座標に変
換するには、 $(x, y) = (X/Z^2, Y/Z^3)$ 変換を使用することができます。

表 357. ECC 完全加算

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x23	PKA_CR	6 ビット
	曲線の係数 p の長さ	(ビット単位、null 以外、 $8 < \text{値} < 640$)	RAM@0x408	64 ビット
	曲線係数 a の符号	- 0x0 : 正 0x1 : 負	RAM@0x410	
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x470	EOS
	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x418	
	最初の点 P の座標 X	$(x < p)$	RAM@0x628	
	最初の点 P の座標 Y	$(y < p)$	RAM@0x680	
	最初の点 P の座標 Z	$(z < p)$	RAM@0x6D8	
	2 番目の点 Q の座標 X	$(x < p)$	RAM@0x730	
	2 番目の点 Q の座標 Y	$(y < p)$	RAM@0x788	
	2 番目の点 Q の座標 Z	$(z < p)$	RAM@0x7E0	
OUT	結果の座標 X	$(x < p)$	RAM@0xD60	
	結果の座標 Y	$(y < p)$	RAM@0xDB8	
	結果の座標 Z	$(z < p)$	RAM@0xE10	

36.4.19 ECC ダブルベースラダー

ECC ダブルベースラダー演算は、 $k \cdot P + m \cdot Q$ の計算で構成されます。ここで、 (P, Q) は楕円曲線上の 2 点であり、 (k, m) は 2 つのスカラーです。演算方法の概要を表 358 に示しています。

結果の点が無限遠点である場合（エラーコード 0xA3B7）、結果の座標は $(0, 0)$ となります。

注： 2 つの入力点はヤコビアン座標 (X, Y, Z) で表されます。アフィン座標 (x, y) で点を入力するには、 $(X, Y, Z) = (x, y, 1)$ 変換を使用することができます。結果はアフィン座標 (x, y) で表されます。

表 358. ECC ダブルベースラダー

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x27	PKA_CR	6 ビット
	曲線の素数位数 n の長さ	(ビット単位、null 以外)	RAM@0x400	64 ビット
	曲線の係数 p の長さ	(ビット単位、null 以外、 $8 < \text{値} < 640$)	RAM@0x408	
	曲線係数 a の符号	– 0x0 : 正 – 0x1 : 負	RAM@0x410	EOS
	曲線係数 $ a $	(絶対値、 $ a < p$)	RAM@0x418	
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x470	
	整数 k	($0 < k < 2^{640}$)	RAM@0x520	
	整数 m	($0 < m < 2^{640}$)	RAM@0x578	
	最初の点 P の座標 X	($x < p$)	RAM@0x628	
	最初の点 P の座標 Y	($y < p$)	RAM@0x680	
	最初の点 P の座標 Z	($z < p$)	RAM@0x6D8	
	2 番目の点 Q の座標 X	($x < p$)	RAM@0x730	
	2 番目の点 Q の座標 Y	($y < p$)	RAM@0x788	
	2 番目の点 Q の座標 Z	($z < p$)	RAM@0x7E0	
OUT	結果の座標 x	($x < p$)	RAM@0x578	EOS
	結果の座標 y	($y < p$)	RAM@0x5D0	
	エラーコード	– 点は無限遠点ではない : 0xD60D – 点は無限遠点 : 0xA3B7	RAM@0x520	64 ビット

36.4.20 ECC 射影からアフィンへ

ECC 射影からアフィンへの演算では、同次射影座標での点 P の表現とアフィン座標での点 P の表現の間の変換を計算します。つまり、点が (X, Y, Z) のトリプルで表される場合、アフィン座標 $(x, y) = (X/Z, Y/Z)$ を計算します。

すべての演算は、点が属する曲線の係数 p をモジュロとして行われます。結果の点が無限遠点である場合（エラーコード 0xA3B7）、結果の座標は $(0, 0)$ となります。

演算方法の概要を表 359 に示しています。

表 359. ECC 射影からアフィンへ

方向とパラメータ		値 (注)	ストレージ	サイズ
IN	MODE	0x2F	PKA_CR	6 ビット
	曲線の係数 p の長さ	(ビット単位、 $8 < \text{値} < 640$)	RAM@0x408	64 ビット
	曲線の係数の値 p	(奇数の整数素数、 $0 < p < 2^{640}$)	RAM@0x470	EOS
	点 P の座標 X (射影)	$(x < p)$	RAM@0xD60	
	点 P の座標 Y (射影)	$(y < p)$	RAM@0xDB8	
	点 P の座標 Z (射影)	$(z < p)$	RAM@0xE10	
	モンゴメリパラメータ $R2 \bmod n$	-	RAM@0x4C8	
OUT	点 P の座標 x (アフィン)	$(x < p)$	RAM@0x578	EOS
	点 P の座標 y (アフィン)	$(y < p)$	RAM@0x5D0	
ERROR	エラーコード	<ul style="list-style-type: none"> 点は無限遠点ではない : 0xD60D 点は無限遠点 : 0xA3B7 	RAM@0x680	64 ビット

36.5 設定と処理時間の例

36.5.1 サポートされている楕円曲線

PKA は、素体上で定義されたすべての非特異楕円曲線をサポートしています。これらの曲線は、短いワイエルシュトラス方程式、 $y^2 = x^3 + ax + b \pmod{p}$ で記述できます。

注： 2 値曲線、エドワーズ曲線、および Curve25519 は、PKA ではサポートされていません。ECC 演算でサポートされる最大オペランドサイズは 640 ビットです。

これらの楕円曲線の ECC 定義域パラメータを公開する場合、標準化団体では以下のようなパラメータを定義しています。

- 有限体 $GF(p)$ のすべての点の算術演算における係数として使用される素数整数 p
- 以下に定義する G によって生成される群の位数である（通常は素数の）整数 n
- 座標 (G_x, G_y) で定義される曲線 G の基点
- 短いワイエルシュトラス方程式の係数である整数 a と b

最後の箇条書きについて、標準化団体が a を負であると定義している場合、PKA は以下の 2 つの表現をサポートしています。

1. 有限体 $GF(p)$ において $p-|a|$ として定義された a 、たとえば $p-3$:
 曲線係数 $p = 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF$
 曲線係数 a の符号 = $0x0$ (正)
 曲線係数 $a = 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF$
2. 負として定義された a 、たとえば -3 :
 曲線係数 $p = 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF$
 曲線係数 a の符号 = $0x1$ (負)
 曲線係数 $a = 0x00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000003$

表 360 に、ECC 演算で PKA がサポートする曲線群をまとめます。

表 360. ECC 演算でサポートされる曲線群

曲線名	標準	参照	
P-192	NIST	デジタル署名標準（DSS）、NIST FIPS 186-4	
P-224			
P-256			
P-384			
P-521			
brainpoolP224r1, brainpoolP224t1	IETF	– Brainpool 楕円曲線、IETF RFC 5639 – インターネット鍵交換（IKE）グループ記述レジストリの Brainpool 楕円曲線、IETF RFC 6932	https://tools.ietf.org
brainpoolP256r1, brainpoolP256t1			
brainpoolP320r1, brainpoolP320t1			
brainpoolP384r1, brainpoolP384t1			
brainpoolP512r1, brainpoolP512t1			
secp192k1, secp192r1	SEC	効率的な暗号化の標準 SEC 2 曲線	https://www.secg.org
secp224k1, secp224r1			
secp256k1, secp256r1			
secp384r1			
secp521r1			
公開鍵暗号化アルゴリズム SM2 の推奨曲線パラメータ	OSCCA	– 楕円曲線に基づいた公開鍵暗号アルゴリズム SM2、中国国家商務管理機構 OSCCA の SM2、2010 年 12 月 – デジタル署名 - パート 3 離散対数ベースのメカニズム、ISO/IEC 14888-3、2018 年 11 月	

36.5.2 計算時間

次の表に、AHB クロックサイクルで表した PKA の計算時間の概要を示します。

表 361. べき剰余

指数の長さ (ビット単位)	モード	係数の長さ (ビット単位)			
		1024	2048	3072	4096
3	ノーマル	124600	491000	684000	1133200
	高速	22700	82000	178000	311000
17	ノーマル	135700	531400	772400	1288000
	高速	33800	122500	266500	465800
$2^{16} + 1$	ノーマル	180000	693700	1126200	1907200
	高速	78200	284700	620400	1085000
1024	保護	9958000	-	-	-
	ノーマル	5850000	-	-	-
	高速	5748000	-	-	-
	CRT ⁽¹⁾	1775000	-	-	-
2048	保護	-	63886000	-	-
	ノーマル	-	42240000	-	-
	高速	-	41832000	-	-
	CRT ⁽¹⁾	-	11670000	-	-
3072	保護	-	-	199403000	-
	ノーマル	-	-	136830000	-
	高速	-	-	136325000	-
	CRT ⁽¹⁾	-	-	36886000	-
4096	保護	-	-	-	454318000
	ノーマル	-	-	-	316000000
	高速	-	-	-	315226000
	CRT ⁽¹⁾	-	-	-	84577000

1. CRT は中国剰余定理最適化の略です (MODE ビットフィールド = 0x07)。

表 362. ECC スカラー乗算⁽¹⁾

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
-	1590000	3083000	5339000	8518000	17818000	21053000	31826000

1. これらの時間は、スカラーパラメータに含まれる 1 の数に影響され、モンゴメリパラメータ R2 の計算が含まれています。

表 363. ECDSA 署名の平均計算時間^{(1) (2)}

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
-	1500000	2744000	4579000	7184000	14455000	16685000	24965000

- これらの値は、指定された長さのランダムな係数による平均実行時間です。係数の長さや値に影響されます。
- NIST 楕円曲線の有限体を定義する係数の実行時間は、Brainpool 楕円曲線に使用する係数や同じサイズのランダムな係数に対してかかる時間より短くなります。

表 364. ECDSA 検証の平均計算時間

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
1011000	1495000	2938000	5014000	7979000	16804000	19254000	29582000

表 365. ECC ダブルベーススラダーの平均計算時間

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
967000	1419000	2768000	4784000	7547000	15854000	18257000	28257000

表 366. ECC 射影からアフィンへの平均計算時間

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
47600	78000	148300	253000	419000	838400		1049300

表 367. ECC 完全加算の平均計算時間

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
10000	12000	18000	26000	39000	53000		89000

表 368. 楕円曲線 Fp 上の点の確認の平均計算時間

係数の長さ (ビット単位)							
160	192	256	320	384	512	521	640
3400	4200	6100	8300	10900	17200	-	-

表 369. モンゴメリパラメータの平均計算時間⁽¹⁾

係数の長さ (ビット単位)							
192	256	320	512	1024	2048	3072	4096
8600	8710	11870	17000	102000	410000	506000	822000

1. 計算時間は、係数の長さと値に依存します。したがって、これらの値は、与えられた長さのランダム係数の平均実行時間です。

36.6 PKA 割込み

公開鍵アクセラレータによって生成され個別にマスク可能な割込みソースは 4 つあり、次のイベントが通知されます。

1. PKA でサポートされていない演算のエラー (OPERRF) (セクション 36.3.7 を参照)
2. マップされていないアドレスへのアクセス (ADDRERRF) (セクション 36.3.7 を参照)
3. PKA 演算中の PKA RAM のアクセス (RAMERRF) (セクション 36.3.7 を参照)
4. PKA 演算終了 (PROCENDF)

割込みソースは、同じグローバル割込みリクエスト信号 pka_it に接続されます。

PKA 制御レジスタ (PKA_CR) のマスクビットを変更することにより、上記の割込みソースを個別に有効にしたり無効にしたりすることができます。適切なマスクビットを 1 にセットすることで、その割込みが有効になります。個別の割込みイベントのステータスは PKA ステータスレジスタ (PKA_SR) から読み出すことができ、PKA_CLRFR レジスタでクリアされます。

表 370 に、使用可能な機能の概要を示します。

表 370. PKA 割込みリクエスト

項目 (略称)	イベント	イベントフラグ	有効制御ビット	クリア方法
PKA	サポートされていない演算	OPERRF	OPERRIE	OPERRFC ビットのセット
	マップされていないアドレスへのアクセスエラー	ADDRERRF	ADDRERRIE	ADDRERRFC ビットのセット
	PKA RAM アクセスエラー	RAMERRF	RAMERRIE	RAMERRFC ビットのセット
	PKA 演算終了	PROCENDF	PROCENDIE	PROCENDFC ビットのセット

36.7 PKA レジスタ

36.7.1 PKA 制御レジスタ (PKA_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRIE	ADDR ERRIE	RAM_ ERRIE	Res.	PROC ENDIE	Res.
										rW	rW	rW		rW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MODE[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	START	EN
		rW	rW	rW	rW	rW	rW							rW	rW

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **OPERRIE** : 演算エラー割込みイネーブル

0 : PKA_SR の OPERRF フラグがセットされても、割込みは生成されません。

1 : PKA_SR の OPERRF フラグがセットされると、割込みが生成されます。

ビット 20 **ADDRERRIE** : アドレスエラー割込みイネーブル

0 : PKA_SR の ADDRERRF フラグがセットされても、割込みは生成されません。

1 : PKA_SR の ADDRERRF フラグがセットされると、割込みが生成されます。

ビット 19 **RAMERRIE** : RAM エラー割込みイネーブル

0 : PKA_SR の RAMERRF フラグがセットされても、割込みは生成されません。

1 : PKA_SR の RAMERRF フラグがセットされると、割込みが生成されます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **PROCENDIE** : 操作終了割込みイネーブル

0 : PKA_SR の PROCENDF フラグがセットされても、割込みは生成されません。

1 : PKA_SR の PROCENDF フラグがセットされると、割込みが生成されます。

ビット 16:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **MODE[5:0]** : PKA 演算コード

000000 : モンゴメリパラメータ計算後にべき剰余

000001 : モンゴメリパラメータ計算のみ

000010 : べき剰余のみ (モンゴメリパラメータを最初にロードする必要があります)

000011 : べき剰余 (保護、秘密情報を扱う場合に使用)

100000 : モンゴメリパラメータ計算後に ECC スカラー乗算 (保護)

100100 : ECDSA 署名 (保護)

100110 : ECDSA 検証

101000 : 楕円曲線 Fp 上の点の確認

000111 : RSA CRT べき乗

001000 : モジュラ逆数

001001 : 算術加算

001010 : 算術減算

001011 : 算術乗算

001100 : 算術比較

001101 : モジュラリダクション

001110 : 剰余加算

001111 : 剰余減算

010000 : モンゴメリ乗算

100011 : ECC 完全加算

100111 : ECC ダブルベースラダー

101111 : ECC 射影からアフィンへ

ここにリストされていない演算が、EN ビットがセットされた状態でアプリケーションによって書き込まれると、OPERRF ビットが PKA_SR レジスタにセットされ、MODE ビットフィールドへの書込みは無視されます。PKA が制限モード (PKA_SR の LMF = 1) に設定されている場合、EN ビットが 1 の状態で MODE に 0x26 とは異なる値を書き込むと、OPERRF ビットがセットされ、MODE ビットへの書込みは無視されます。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **START** : 演算開始

このビットに 1 を書き込むと、PKA RAM に既書き込まれたオペランドとデータを使用して、MODE[5:0] で選択された演算を開始します。このビットは常に 0 として読み出されます。

START ビットがセットされているときに不正な演算を選択すると、演算は開始されず、PKA_SR で OPERRF ビットがセットされます。

注 : PKA がビジーであると、START は無視されます。

ビット 0 **EN** : PKA 有効化

0 : PKA を無効にします。

1 : PKA を有効化します。ハードウェアによって PKA_SR の INITOK がセットされると、PKA は動作可能になります。

EN = 1 のときに不正な演算を選択すると、PKA_SR で OPERRF ビットがセットされます。詳細については、PKA_CR.MODE ビットフィールドを参照してください。

注 : EN = 0 の場合、PKA RAM はアプリケーションでアクセスできます。

36.7.2 PKA ステータスレジスタ (PKA_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRF	ADDR ERRF	RAM_ ERRF	Res.	PROC ENDF	BUSY
										r	r	r		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LMF	INITOK
														r	r

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **OPERRF** : 演算エラーフラグ

0 : イベントエラーはありません。

1 : PKA_CR レジスタで不正または不明な演算が選択されました。

このビットは、PKA_CLRFR の OPERRFC ビットを使用してクリアします。

ビット 20 **ADDRERRF** : アドレスエラーフラグ

0 : アドレスエラーはありません。

1 : アドレスアクセスが範囲外 (マップされていないアドレス) です。

このビットは、PKA_CLRFR の ADDRERRFC ビットを使用してクリアします。

ビット 19 **RAMERRF** : PKA RAM エラーフラグ

0 : PKA RAM アクセスエラーはありません。

1 : PKA コアが計算中に内部 RAM を使用して、PKA RAM への AHB アクセスが発生しました (AHB PKA_RAM は PKA 演算中にアクセスできません)。

このビットは、PKA_CLRFR の RAMERRFC ビットを使用してクリアします。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **PROCENDF** : PKA 演算終了フラグ

0 : 演算が進行中です。

1 : PKA 演算が完了しました。このフラグは、BUSY ビットがネゲートされたときにセットされます。

ビット 16 **BUSY** : ビジーフラグ

このビットは、PKA 演算が進行中 (PKA_CR の START = 1) に必ずセットされます。計算が完了すると自動的にクリアされ、PKA RAM が再びアクセス可能になります。

0 : 進行中の演算がありません (デフォルト)。

1 : 演算が進行中です。

PKA が誤った OP コードで開始された場合、数サイクルの間ビジー状態に留まり、その後、演算を自動的にアポートしてレディ状態 (BUSY = 0) に戻ります。

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **LMF** : 制限モードフラグ

このビットは、PKA_CR の EN ビットがセットされると更新されます。

0 : MODE ビットフィールドに記載されているすべての値が使用できます。

1 : PKA によって ECDSA 検証 (MODE = 0x26) のみがサポートされます。

ビット 0 **INITOK** : PKA 初期化 OK

このビットは、PKA の初期化が完了するとアサートされます。RNG が適切な乱数を出力できなかった場合、INITOK は 0 のままとなります。

0 : PKA は正しく初期化されません。START ビットはセットできません。

1 : PKA は正しく初期化され、正常に使用できます。

36.7.3 PKA クリアフラグレジスタ (PKA_CLRFR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OP ERRFC	ADDR ERRFC	RAM_ ERRFC	Res.	PROC ENDFC	Res.
										w	w	w		w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **OPERRFC** : 演算エラーフラグのクリア

0 : 影響なし。

1 : PKA_SR の OPERRF フラグをクリアします。

ビット 20 **ADDRERRFC** : アドレスエラーフラグのクリア

0 : 影響なし。

1 : PKA_SR の ADDRERRF フラグをクリアします。

ビット 19 **RAMERRFC** : PKA RAM エラーフラグのクリア

0 : 影響なし。

1 : PKA_SR の RAMERRF フラグをクリアします。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **PROCENDFC** : PKA 演算終了フラグのクリア

0 : 影響なし。

1 : PKA_SR の PROCENDF フラグをクリアします。

ビット 16:0 予約済みであり、リセット値に保持する必要があります。

注 : **PKA_CLRFR** を読み出すとすべて 0 で返されます。

36.7.4 PKA RAM

PKA RAM は、PKA ベースアドレスと比較して 0x0400 のオフセットアドレスでマップされています。PKA.AHB インタフェースでは、32 ビットワードのシングルアクセスのみがサポートされています。

RAM サイズは 5336 バイトです (最大ワードオフセット : 0x14D0)

注 : [セクション 36.3.3 : PKA のリセットおよびクロック](#)で説明したように、PKA RAM は、PKA リセットまたは製品リセットの直後には使用できません。

36.7.5 PKA レジスタマップ

表 371. PKA レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PKA_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRIE	ADDRERRIE	RAMERRIE	Res.	PROCENDIE	Res.	Res.	Res.	MODE[5:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	START	EN
	リセット値											0	0	0		0				0	0	0	0	0	0							0	0
0x004	PKA_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRF	ADDRERRF	RAMERRF	Res.	PROCENDF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LMF	INITOK
	リセット値											0	0	0		0	0															0	0
0x008	PKA_CLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPERRFC	ADDRERRFC	RAMERRFC	Res.	PROCENDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0	0	0		0																	

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

37 オンザフライ復号エンジン (OTFDEC)

37.1 概要

オンザフライ復号エンジン (On-The-Fly DEcryption engine : OTFDEC) により、読出しリクエストのアドレス情報に基づいて AHB トラフィックをオンザフライで復号できます。OTFDEC では、4 つの独立して重複のない暗号化領域を定義できます。

OTFDEC はカウンタモードで AES-128 を使用して、遅延を最小限に抑えます。その結果、暗号化された領域の内容が変更されるたびに、領域全体を異なる暗号化コンテキスト (キーまたは初期化ベクタ) で再度暗号化する必要があります。この制約により、OTFDEC が外部 NOR Flash に格納された読出し専用データまたはコードの復号に適したものになっています。

注 : OTFDEC を OCTOSPI と連携して使用する場合、Flash メモリコントローラのメモリマップドモードで Flash メモリにアクセスする必要があります。

製品でセキュリティが有効になっている場合、安全なホストでのみ OTFDEC をプログラムできます。

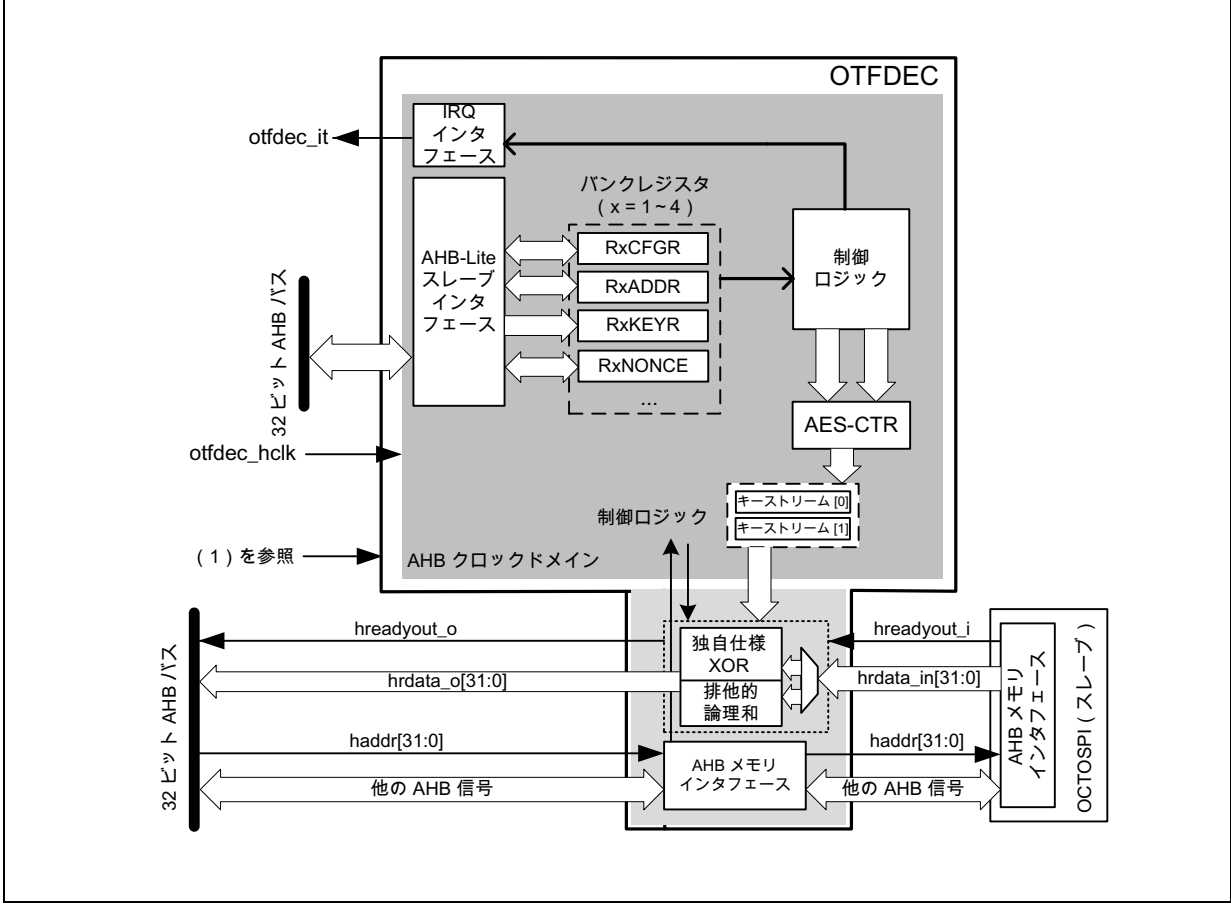
37.2 OTFDEC の主な機能

- OCTOSPI のメモリマップド読出し操作 (1 回または複数回) 時のオンザフライ 128 ビット復号
 - カウンタ (CTR) モードでの AES の使用 (2 つの 128 ビットキーストリームバッファ)
 - あらゆる読出しサイズのサポート
 - 暗号化/復号に読出しの物理アドレスを使用
- 最大 4 つの独立した暗号化領域
 - 領域定義の単位 : 4096 バイト
 - 領域設定の書込みロックメカニズム
 - 各領域につき、独自の 128 ビットキー、2 バイトのファームウェアバージョン、8 バイトのアプリケーション定義ノンスあり。このうち最低 1 つはアプリケーションで暗号化するたびに変更が必要
- 暗号化キーの機密性と整合性による保護
 - 書込み専用レジスタ (ソフトウェアロックメカニズム付き)
 - 公開鍵情報として 8 ビット CRC が使用可能
- OCTOSPI プリフェッチメカニズムのサポート
- AES ストリーム暗号の最上部に独自仕様の保護レイヤを追加するための高度な暗号化モードを選択可能 (実行専用)
- 32 ビットワードのシングルアクセスでのみアクセス可能な特権対応の AMBA AHB スレーブペリフェラル (それ以外は AHB バスエラーが発生し、書込みアクセスは無視される)
- TrustZone セキュリティが製品で有効になっている場合のみ安全なプログラミングが可能
- 暗号化モード

37.3 OTFDEC の機能説明

37.3.1 OTFDEC ブロック図

図 333. OTFDEC ブロック図



1. otfdec_tzen

37.3.2 OTFDEC 内部信号

表 372 では、製品レベル（パッド上）ではなく、OTFDEC レベルで使用可能な内部信号を知るのに役立つリストを記載しています。

表 372. OTFDEC 内部入力／出力信号

信号名	信号タイプ	説明
otfdec_hclk	デジタル入力	AHB バスクロック
otfdec_it	デジタル出力	OTFDEC グローバル割込みリクエスト
otfdec_tzen	デジタル入力	OTFDEC TrustZone 有効、ペリフェラルの TrustZone 機能を制御 (TZEN)

FLASHの TZEN オプションビットは、デバイスで TrustZone をアクティブにするために使用します。

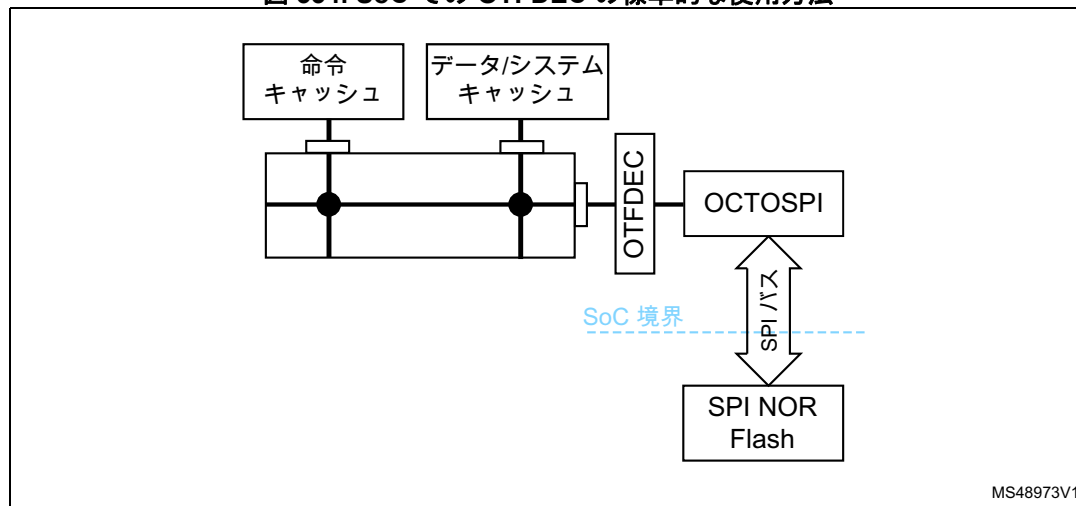
- TZEN = 1 : TrustZone セキュリティが製品で有効になっています。
- TZEN = 0 : TrustZone セキュリティが製品で無効になっています。

37.3.3 OTFDEC オンザフライ復号

概要

OTFDEC の標準的な使用方法を図 334 に示します。

図 334. SoC での OTFDEC の標準的な使用方法



OTFDEC の本来の目的は、外部 SPI NOR Flash デバイスに格納されている読み出し専用ファームウェアライブラリの機密性を保護することです。

特別なロック処理が OTFDEC で利用でき、復号キーの整合性を保護し、ソフトウェアの DoS 攻撃に対するその他の設定も保護します。ほとんどのレジスタへの OTFDEC アクセスは、OTFDEC_PRIVCFGR レジスタの PRIV ビットをセットすることで、特権のみにすることができます。OTFDEC は、TrustZone セキュリティが有効化されている場合、TrustZone CPU でのみ書き込むことができます。

OTFDEC を OCTOSPI と連携して使用する場合、Flash コントローラのメモリマップドモードで Flash メモリに読み出す必要があります。

オンザフライでの復号に加え、OTFDEC は一度に 32 ビットワードずつ暗号化することもできます (詳細については、[セクション 37.5.3 : OTFDEC での暗号化](#)を参照)。

OTFDEC アーキテクチャ

OTFDEC では、関連する AHB バスの AHB 読み出し転送を解析します。読み出しリクエストが OTFDEC でプログラムされた 4 つの領域のいずれかにある場合、カウンタモードでの AES アルゴリズムに基づき、制御ロジックによってキーストリームの計算がトリガされます。このキーストリームを使用して、OCTOSPI AHB マスタからの読み出し転送に存在するデータをオンザフライで復号し、キーストリーム情報を計算している間にこのマスタの HREADYOUT 信号をローレベルに保持します (これには最大 11 サイクルかかります)。有効化された OTFDEC 領域以外へのアクセスは、すべて非暗号化領域に属します。

各 OTFDEC 領域は、 $x = 1 \sim 4$ の OTFDEC_RxCFGR、OTFDEC_RxSTARTADDR、OTFDEC_RxENDADDR、OTFDEC_RxNONCER、および OTFDEC_RxKEYR レジスタでプログラムされます。OTFDEC_RxCFGR で、MODE ビットは OTFDEC 動作モード (標準または高度な暗号化) を定義します。

領域判定の単位は 4096 バイトです。

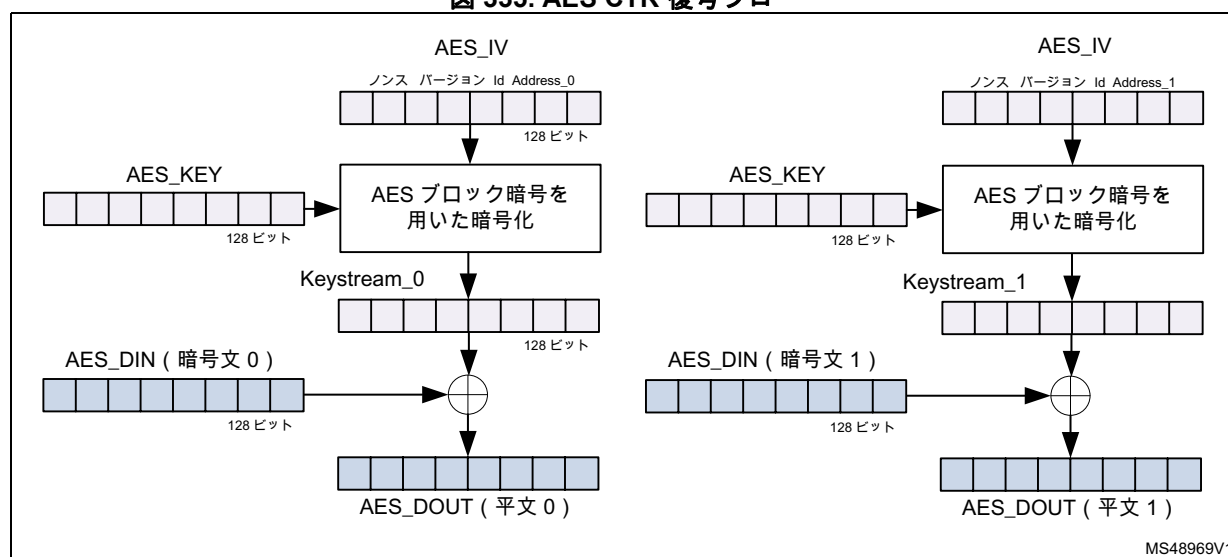
注： OTFDEC で領域の重複を防ぐことはできませんが、これは有効なプログラミングとならないため、アプリケーションソフトウェアで回避する必要があります。

OTFDEC は、4096 バイト整列のアドレス境界にまたがらない限り、インクリメンタルバーストやラップバーストを復号できます。

37.3.4 OTFDEC を使用したカウンタモードでの AES 復号

図 335 に、OTFDEC がカウンタ連鎖モードで業界標準の高度暗号化標準 (AES) アルゴリズムを使用する流れについて示します。このモードは、NIST によって特別公報 800-38A 『ブロック暗号の推奨動作モード』で指定されています。

図 335. AES CTR 復号フロー



特別なキーストリーム情報である 128 ビットデータブロックは、それぞれ以下の定義に従って AES ブロック暗号を使用して生成されます。

- 初期化ベクタ AES_IV[127:0] = RxNONCER1[31:0] || RxNONCER0[31:0] || 0b0000 0000 0000 0000 || RxCFGR[31:16] || 0b00 || (x-1) || ReadAddress[31:4]
- キーマテリアル AES_KEY[127:0] = RxKEYR3[31:0] || RxKEYR2[31:0] || RxKEYR1[31:0] || RxKEYR0[31:0]

注： 上記の x は、選択した暗号化領域の RegionID です (x = 1~4)。
ReadAddress は暗号化されたデータブロックの AHB アドレスで、モジュロ 128 ビットです。

結果の 128 ビットキーストリームは、128 ビット暗号文データとの排他的論理和がとられ、128 ビットの明文データが生成されます。

- AES_DIN および AES_DOUT のデータブロックは次の規則に従って構成されます (「|」はバイナリ連結を表します)。

$$\text{AES_Dx}[127:0] = \text{AHB_word}(@ | 0xC)[31:0] | \text{AHB_word}(@ | 0x8)[31:0] | \text{AHB_word}(@ | 0x4)[31:0] | \text{AHB_word}(@ | 0x0)[31:0]$$
 ここで、@ はキーストリーム (上記 ReadAddress[31:4]) の計算に使用される 16 進数アドレスを指します。

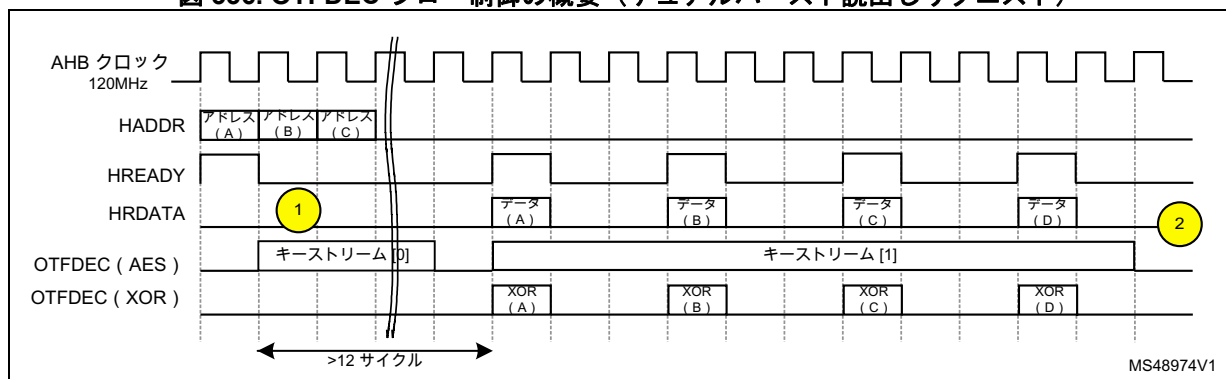
読出しリクエストが暗号化領域内にない場合や、復号がこの領域で有効になっていない場合、AHB データは変更されません。

注： アプリケーションによって OTFDEC_RxCFGR の MODE ビットフィールドが 11 にセットされると、さらに保護レイヤが AES ストリーム暗号の最上部に追加されます。この高度な暗号化モードは指示がある場合에만使用できます（実行専用領域）。

37.3.5 フロー制御管理

図 336 に、1 つの 128 ビット AES データブロックに対応する INCR4 AHB バーストを OTFDEC で管理する方法について示します。

図 336. OTFDEC フロー制御の概要（デュアルバースト読出しリクエスト）



以下の注意点があります。

1. OTFDEC は、データの復号（キーストリームの計算）の準備が整っていないため、AHB マスタからの HREADY 信号を強制的にローにします。
2. キーストリームバッファにより、OTFDEC がこの設定（120MHz AHB クロック、SPI クロックごとに 2 バイトを伝達する 104MHz SPI バス）で 12 サイクル以内に新しいデータ一式を処理する準備が整います。

37.3.6 OTFDEC エラー管理

OTFDEC は以下のとおり定義されたエラーを自動的に管理します。

- OTFDEC_RxKEYR レジスタに対する不正読出し。
- OTFDEC_RxCFGR の CONFIGLOCK または KEYLOCK が「1」の場合の OTFDEC_RxKEYR レジスタに対する不正書込み（アクセスがセキュアである場合）。セキュリティが製品で無効になっている場合は、アクセスが非セキュアである場合に同じエラーが発生します。
- OTFDEC_RxCFGR の CONFIGLOCK が「1」の場合の OTFDEC_RxKEYR、OTFDEC_RxSTARTADDR、OTFDEC_RxENDADDR、または OTFDEC_RxNONCER のレジスタに対する不正書込み（ $x = 1 \sim 4$ ）（アクセスがセキュアである場合）。セキュリティが製品で無効になっている場合は、アクセスが非セキュアである場合に同じエラーが発生します。
- 実行専用領域に対する不正読出し（MODE[1:0] = 11）。このような不正リクエストはバスエラーなしで 0x0 が返されます。
- 暗号化有効時の領域に対する実行リクエスト（ENC = 1）。リクエストはバスエラーなしで 0x0 が返されます。
- キーエラー：キーレジスタが null または適切に初期化されていない場合（KEYCRC = 0x0）の暗号化領域に対する読出しリクエスト。エラーの原因は、誤ったキーのローディングシーケンス（OTFDEC_RxCFGR の KEYCRC を参照）であるか、アポルトイベント（タンパ検出、許可されていないデバッグ接続、信頼できないブート、RDP のレベルのリグレーション）が考えられます。このような読出しリクエストはバスエラーなしで 0x0 が返されます。

- TrustZone セキュリティが製品で有効になっており、アクセスが非セキュアである場合のあらゆるレジスタに対する書込み。

この最後のエラーは、製品リファレンス・マニュアルの GTZC セクションに記載されているとおり、TrustZone 割込みコントローラで管理およびクリアされます。

これらのエラーには（最後の 1 つを除く）、OTFDEC_IER レジスタに SEIE、XONEIE、または KEIE ビットがセットされている場合は、割込みを生成できます。（[セクション 37.4](#) を参照）。

注： キーエラーの後、再度 OTFDEC キーを適切に初期化する必要があり、レジスタがロックされている場合は、OTFDEC のリセットが必要になる場合があります。

37.4 OTFDEC 割込み

OTFDEC によって生成される個別にマスク可能な割込みソースは 3 つあり、次のセキュリティイベントが通知されます。

- キーに対する不正な読出しまたは書込みアクセス（SEIF フラグ）。[セクション 37.3.6](#) を参照してください。
- CONFIGLOCK = 1 の場合の領域の設定に対する不正書込み（SEIF フラグ）。[セクション 37.3.6](#) を参照してください。
- XONEIF フラグをトリガする、実行専用領域（MODE[1:0] = 11）への読出しアクセス。
- 暗号化有効時の実行（XONEIF フラグ）。
- KEIF フラグをトリガするキーエラー（0 で読み出される暗号化領域）。[セクション 37.3.6](#) を参照してください。

割込みソースは、同じグローバル割込みリクエスト信号に接続されます。

[表 373](#) に示すように、OTFDEC_IER の対応する SEIE、XONEIE、または KEIE ビットをセットすることにより、OTFDEC 割込みソースを有効／無効にすることができます。割込みイベントのステータスは OTFDEC_ISR で確認でき、このイベントは OTFDEC_ICR でクリアできます。

表 373. OTFDEC 割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ ⁽¹⁾	有効制御ビット	割込みのクリア方法
OTFDEC	セキュリティエラー	SEIF	SEIE	OTFDEC_ICR に SEIF をセット
	実行専用 暗号化中に実行	XONEIF	XONEIE	OTFDEC_ICR に XONEIF をセット
	キーエラー	KEIF	KEIE	OTFDEC_ICR に KEIF をセット

1. イベントフラグは、OTFDEC_ISR レジスタ内にあります。

37.5 OTFDEC アプリケーション情報

37.5.1 OTFDEC 初期化処理

概要

OTFDEC の主な側面の 1 つは、秘密鍵に関わる、信頼性のあるレジスタの初期化です。以下に示す 2 種類の信頼性のある初期化処理を推奨します。

注： これらのシーケンスは、ファームウェア開発時にキーや領域設定のロックを推奨するわけではないため、製品コード向けとなります。

設定レジスタへの書込みは、領域が有効化されていても、設定のロックによって許可されている場合は有効です。

初期化処理 1：すべての領域に対する 1 つのキー

この処理では、1 つのエントリティが 4 つの保護された領域の復号に使用する秘密鍵を所有します。推奨される OTFDEC 設定シーケンスは、以下のとおりです。

1. $x = 1 \sim 4$ について、OTFDEC_RxCFGR に正しい MODE[1:0] の値を書き込みます。
2. $x = 1 \sim 4$ について、(有効な CRC を得るために) KEYCRC に記載されているシーケンスで OTFDEC_RxKEYR レジスタをプログラムします。キーレジスタは書込み専用ということに注意してください。
3. $x = 1 \sim 4$ について、キー CRC をチェックします。問題なければ、OTFDEC_RxCFGR レジスタの KEYLOCK ビットをセットします。このビットはクリアできません (この領域のキーレジスタ x は書き込めなくなります)。
4. 領域 x を復号するには (必ずしも復号キーを所有するエントリティで実行する必要がないタスク):
 - a) キー CRC が領域に格納された暗号化バイナリに対応しているかどうか確認します。
 - b) このバイナリに対応する詳細情報 (ノンス、開始アドレス、終了アドレス、バージョン番号) を入力します。
 - c) REG_EN でこの領域の復号を有効にします。
 - d) OTFDEC_RxCFGR の CONFIGLOCK ビットをセットします。このビットはクリアできません (領域設定は書き込めなくなります)。

注意： 指定した領域について、MODE ビットが変更されると、キーレジスタと関連 CRC がハードウェアによってクリアされます。そのため、上記の手順 1 は手順 2 の前に実行する必要があり、手順 2 以降は MODE ビットを変更しないでください。

初期化処理 2 : 領域ごとに 1 つのキー

この処理では、1 つのエンティティが 1 つ（または複数）の保護された領域の復号に使用する秘密鍵を所有できます。推奨される OTFDEC 設定シーケンスは、以下のとおりです。

1. 領域 x を復号するには（このタスクは、対応するキーを所有するエンティティで実行する**必要があります**）：
 - a) OTFDEC_RxCFGR に正しい MODE[1:0] の値を書き込みます。
 - b) （有効な CRC を得るために）KEYCRC に記載されているシーケンスで OTFDEC_RxKEYR レジスタをプログラムします。キーレジスタは書き込み専用ということに注意してください。
 - c) キー CRC をチェックします。問題なければ、OTFDEC_RxCFGR レジスタの KEYLOCK ビットをセットします。このビットはクリアできません（キーレジスタは書き込めなくなります）。
 - d) 保護されたファームウェアに対応する詳細情報（ノンス、開始アドレス、終了アドレス、バージョン番号）を入力します。
 - e) REG_EN でこの領域の復号を有効にします。
 - f) OTFDEC_RxCFGR の CONFIGLOCK ビットをセットします。このビットはクリアできません（領域設定は書き込めなくなります）。

注意： 指定した領域について、MODE ビットが変更されると、キーレジスタと関連 CRC がハードウェアによってクリアされます。そのため、上記の手順 a) を手順 b) の前に実行する必要があり、手順 b) 以降は MODE ビットを変更しないでください。

37.5.2 OTFDEC および電力管理

OTFDEC がリセットされるたびに、[セクション 37.5.1](#) に示した正しいキーのローディングシーケンスを実行する必要があります（この場合では、OTFDEC_RxCFGR で KEYCRC = 0）。

OTFDEC がハードウェアによってリセットされるたびに、アプリケーションソフトウェアでこのポイントを確認することを推奨します。

37.5.3 OTFDEC での暗号化

コードおよびデータ標準の暗号化

OTFDEC は、MODE[1:0] = 10 で保護された領域に格納されたバイナリを処理するとき、カウンタモードで標準の AES を使用します。このモードを選択すると、AES 互換のハードウェアアクセラレータまたはライブラリを使用して、これらの保護されたライブラリを暗号化できます。以下の高度な暗号化セクションに示すように、OTFDEC を使用することもできます（MODE[1:0] = 10）。

AES の入力および出力の定義とエンディアンは、[セクション 37.3.4 : OTFDEC を使用したカウンタモードでの AES 復号](#)で定義されています。

OTFDEC の高度な暗号化

OTFDEC は、MODE[1:0] = 11 で保護された領域に格納されたコードを処理するとき、カウンタモードで標準の AES の最上位にある独自仕様の保護レイヤを使用します。

高度な暗号化モードを使用してタンパに対する堅牢性を高めることができます。

推奨される OTFDEC を使用した暗号化シーケンスは、以下のとおりです。

1. 暗号化用のアプリケーションで OTFDEC_CR の ENC ビットをセットします。このアプリケーションは、製品の TrustZone セキュリティを有効にする際に、TrustZone セキュアモードで実行する必要があります。OTFDEC_PRIVCFGR に PRIV ビットがセットされている場合、このアプリケーションには特権がなければなりません。

2. [セクション 37.5.1: OTFDEC 初期化処理](#)に示すように、暗号化アプリケーションでは OTFDEC を初期化します。OCTOSPI も、OTFDEC が暗号化モードで完全に機能するように、適切にクロック供給する必要があります。この手順は、手順 1 の前に行うこともできます。
3. 暗号化アプリケーションでは、所定の保護されたアドレスに 32 ビットの平文データを書き込み、その後暗号化されたデータを RAM に格納するために同じアドレスに読み戻します。このデータは、暗号化モードで OTFDEC によって傍受されるため、デバイスの外に出さないください。
4. 暗号化アプリケーションは、すべてのバイナリが処理されるまで、前の手順（アドレスの変更）に戻ります。
5. 暗号化アプリケーションは OTFDEC_CR の ENC ビットをクリアします。その後、別のアプリケーションで暗号化バイナリを外部 Flash の正しいアドレスに書き込めるようになります。

この手順について、いくつかの重要な注意点があります。

- 暗号化粒度は 32 ビットです（32 ビットのシングルアクセスは必須です）。
- ENC ビットがセットされている場合は、非暗号化領域の読出しは通常のデータを返します（暗号化も復号もされていないデータなど）。暗号化モードでは、OCTOSPI（レジスタを含む）にアクセスしてはなりません。これは、OTFDEC が ENC ビットがセットされている間の OCTOSPI との通信を遮断するためです。
- OTFDEC は、ENC = 1 の間は実行をサポートしません（暗号化されたデータの読出しのみ）。不正実行検出時に、XONEIF フラグが立てられ、ゼロが返されます。

37.5.4 OTFDEC キー CRC のソースコード

以下は、OTFDEC_RxKEYR レジスタのキーをロードした後に KEYCRC ビットフィールドの OTFDEC によって示される計算結果との比較に使用できる CRC ソースコードです。

```
uint8_t getCRC(uint32_t * keyin)
{
    const uint8_t CRC7_POLY = 0x7;
    const uint32_t key_strobe[4] = {0xAA55AA55, 0x3, 0x18, 0xC0};
    uint8_t i, j, k, crc = 0x0;
    uint32_t keyval;

    for (j = 0; j < 4; j++)
    {
        keyval = *(keyin+j);
        if (j == 0)
        {
            keyval ^= key_strobe[0];
        }
        else
        {
            keyval ^= (key_strobe[j] << 24) | (crc << 16) | (key_strobe[j] << 8)
| crc;
        }

        for (i = 0, crc = 0; i < 32; i++)
        {
            k = (((crc >> 7) ^ (keyval >> (31-i)) & 0xF)) & 1;
            crc <<= 1;
            if (k)
            {
                crc ^= CRC7_POLY;
            }
        }
        crc ^= 0x55;
    }
    return crc;
}
```


37.6 OTFDEC レジスタ

37.6.1 OTFDEC 制御レジスタ (OTFDEC_CR)

アドレスオフセット : 0x0

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **ENC** : 暗号化モードビット

このビットがセットされると、OTFDEC は暗号化モードで使用されます。この間、アプリケーションは平文データを書き、暗号化されたデータを読み戻すことができます。このビットがクリアされると (デフォルト)、OTFDEC は復号モードで使用されます。この間、アプリケーションは復号されたデータのみを読み戻します。両方のモードで、暗号化コンテキスト (キー、ノンス、ファームウェアバージョン) は適切に初期化する必要があります。

このビットをセットすると、データアクセスのみが許可されます (それ以外の場合はゼロが返され、XONEIF がセットされます)。MODE = 11 の場合は、高度な暗号化モードが自動的に選択されます。

0 : OTFDEC が復号モードで動作しています。

1 : OTFDEC が暗号化モードで動作しています。

注 : ENC ビットがセットされると、OCTOSPI にはアクセスできなくなります (レジスタおよびメモリマップド領域)。

37.6.2 OTFDEC 特権アクセス制御設定レジスタ (OTFDEC_PRIVCFGR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **PRIV** : 特権アクセス保護

0 : OTFDEC レジスタアクセスで保護が追加されません。

1 : OTFDEC_PRIVCFGR 以外のすべてのレジスタにアクセスする際に、保護が追加されます。

— レジスタに対する非特権読出しアクセスは 0 を返します。

— レジスタに対する非特権書込みアクセスは無視されます。

注 : このビットは、特権モードでのみ書き込むことができます。読出しに制限はありません。

37.6.3 OTFDEC 領域 x 設定レジスタ (OTFDEC_RxCFGR)

アドレスオフセット : 0x20 + 0x30 * (x1)、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

CONFIGLOCK ビットが 1 にセットされている場合、書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_VERSION[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYCRC[7:0]								Res.	Res.	MODE[1:0]		Res.	KEYLOCK	CONFIG LOCK	REG_EN
r	r	r	r	r	r	r	r			rw	rw		rs	rs	rw

ビット 31:16 **REG_VERSION[15:0]** : 領域のファームウェアバージョン

領域に対応する OTFDEC_RxCFGR の REG_EN ビットがセットされる前に、この 16 ビットのビットフィールドを正しく初期化する必要があります。

ビット 15:8 **KEYCRC[7:0]** : 領域キーの 8 ビット CRC

KEYLOCK = 0 のとき、次のシーケンスで正確にこの領域のキーをロードしている間に、KEYCRC ビットフィールドがハードウェアによって自動計算されます。KEYR0、KEYR1、KEYR2、最後に KEYR3 の順です (すべて 1 回の書き込み)。新しい有効なシーケンスが開始するとすぐに新しい計算が開始され、有効なシーケンスが完了するまで、KEYCRC は 0 として読み出されます。

KEYLOCK = 1 のとき、KEYCRC は次のリセットまで変わりません。

CRC 計算は、標準の CRC-8-CCITT アルゴリズム

$X^8 + X^2 + X + 1$ (規則による) を使用した 8 ビットチェックサムです。ソースコードは、[セクション 37.5.4](#) で確認可能です。

このフィールドは読み出し専用です。

注： CRC 情報は、キーの最後のビットが書き込まれた後にのみ更新されます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **MODE[1:0]** : 動作モード

このビットフィールドで、この領域での OTFDEC 動作モードを選択します。

10 : すべての読み出しアクセスが復号されます (命令およびデータ)。

11 : 高度な暗号化モードが有効になり、命令アクセスのみが復号されます。

その他 : 予約済み

MODE ≠ 11 の場合、標準の AES 暗号化モードが有効になります。

MODE ビットのいずれかが変更されると、領域キーと関連 CRC が 0 になります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEYLOCK** : 領域キーのロック

0 : この領域の KEYRx レジスタへの書き込みが許可されます。

1 : この領域の KEYRx レジスタへの書き込みが、次の OTFDEC リセットまで無視されます。KEYCRC ビットフィールドはロックされます。

注： このビットは 1 回セットされます。すなわち、このビットがセットされると、OTFDEC がリセットされた場合のみ 0 にリセットできます。

ビット 1 **CONFIGLOCK** : 領域設定のロック

0 : この領域の OTFDEC_RxCFGR、OTFDEC_RxSTARTADDR、OTFDEC_RxENDADDR、および OTFDEC_RxNONCERy のレジスタへの書き込みが許可されます。

1 : この領域の OTFDEC_RxCFGR、OTFDEC_RxSTARTADDR、OTFDEC_RxENDADDR、および OTFDEC_RxNONCERy のレジスタへの書き込みが、次の OTFDEC リセットまで無視されます。

注： このビットは 1 回セットされます。すなわち、このビットがセットされると、OTFDEC がリセットされた場合のみ 0 にリセットできます。このビットをセットすると、KEYLOCK ビットが強制的に 1 になります。

ビット 0 **REG_EN** : 領域のオンザフライ復号イネーブル

0 : この領域でのオンザフライ復号が無効になります。

1 : この領域でのオンザフライ復号が有効になります。データは対応するキーストリームとの排他的論理和がとられます。

注： このビットをセットしたときに領域のコンテキスト (バージョン、キー、ノンス) が無効であれば、不要なデータが復号されます。

37.6.4 OTFDEC 領域 x 開始アドレスレジスタ (OTFDEC_RxSTARTADDR)

アドレスオフセット : $0x24 + 0x30 * (x1)$ 、($x = 1$ から 4)

リセット値 : $0x0000\ 0000$

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_START_ADDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_START_ADDR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **REG_START_ADDR[31:0]** : 領域 AHB 開始アドレス

このレジスタは、OTFDEC_RxCFGR レジスタで領域に対応する REG_EN ビットがセットされる前に書き込む必要があります。

このレジスタへの書込みは、OTFDEC_RxCFGR レジスタの領域 CONFIGLOCK ビットがセットされている間に行われた場合、破棄されます。

注 : 領域を決定すると、最初の 12 ビット (LSB) と最後の 4 ビット (MSB) は無視されます。

このレジスタに読出しでアクセスする場合、4 MSB ビットと 12 LSB ビットは 0 を返します。

37.6.5 OTFDEC 領域 x 終了アドレスレジスタ (OTFDEC_RxENDADDR)

アドレスオフセット : $0x28 + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x00000FFF

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_END_ADDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_END_ADDR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **REG_END_ADDR[31:0]** : 領域 AHB 終了アドレス

このレジスタは、OTFDEC_RxCFGR レジスタで領域に対応する REG_EN ビットがセットされる前に書き込む必要があり、OTFDEC_RxENDADDR は厳密に OTFDEC_RxSTARTADDR より大きい有効な値にする必要があります。

このレジスタへの書込みは、OTFDEC_RxCFGR の領域 CONFIGLOCK ビットがセットされている間に行われた場合、破棄されます。

注： 領域を決定すると、最初の 12 ビット (LSB) と最後の 4 ビット (MSB) は無視されます。

このレジスタに読出しでアクセスする場合、4 MSB ビットは 0 を返し、12 LSB ビットは 1 を返します。

37.6.6 OTFDEC 領域 x ノンスレジスタ 0 (OTFDEC_RxNONCER0)

アドレスオフセット : $0x2C + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_NONCE[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_NONCE[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **REG_NONCE[31:0]** : 領域のノンス、ビット [31:0]

このレジスタは、OTFDEC_RxCFGR で領域に対応する REG_EN ビットがセットされる前に書き込む必要があります。

このレジスタでは、OTFDEC_RxCFGR の領域 CONFIGLOCK ビットがセットされている間に行われた書込みは破棄されます。

37.6.7 OTFDEC 領域 x ノンスレジスタ 1 (OTFDEC_RxNONCER1)

アドレスオフセット : $0x30 + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_NONCE[63:48]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_NONCE[47:32]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **REG_NONCE[63:32]** : 領域のノンス、ビット [63:32]

NONCE[63:0] ビットフィールドの説明については、OTFDEC_RxNONCER0 レジスタを参照してください。

37.6.8 OTFDEC 領域 x キーレジスタ 0 (OTFDEC_RxKEYR0)

アドレスオフセット : $0x34 + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **REG_KEY[31:0]** : 領域のキー、ビット [31:0]

このレジスタは、OTFDEC_RxCFGR で領域に対応する REG_EN ビットがセットされる前に書き込む必要があります。

このレジスタを読み出すと 0 の値が返されます。このレジスタへの書込みは、OTFDEC_RxCFGR で領域 CONFIGLOCK または KEYLOCK ビットがセットされている間に行われた場合、破棄されます。

注 : アプリケーションによって正常に OTFDEC_RxCFGR および OTFDEC_RxKEYR の MODE ビットが変更されると、関連する KEYCRC は消去されます。

37.6.9 OTFDEC 領域 x キーレジスタ 1 (OTFDEC_RxKEYR1)

アドレスオフセット : $0x38 + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_KEY[63:48]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_KEY[47:32]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **REG_KEY[63:32]** : 領域のキー、ビット [63:32]

KEY[127:0] ビットフィールドの説明については、OTFDEC_RxKEYR0 レジスタを参照してください。

37.6.10 OTFDEC 領域 x キーレジスタ 2 (OTFDEC_RxKEYR2)

アドレスオフセット : $0x3C + 0x30 * (x1)$ 、(x = 1 から 4)

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_KEY[95:80]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_KEY[79:64]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **REG_KEY[95:64]** : 領域のキー、ビット [95:64]

KEY[127:0] ビットフィールドの説明については、OTFDEC_RxKEYR0 レジスタを参照してください。

37.6.11 OTFDEC 領域 x キーレジスタ 3 (OTFDEC_RxKEYR3)

アドレスオフセット : $0x40 + 0x30 * (x1)$ 、($x = 1$ から 4)

リセット値 : $0x0000\ 0000$

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REG_KEY[127:112]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_KEY[111:96]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **REG_KEY[127:96]** : 領域のキー、ビット [127:96]

KEY[127:0] ビットフィールドの説明については、OTFDEC_RxKEYR0 レジスタを参照してください。

37.6.12 OTFDEC 割込みステータスレジスタ (OTFDEC_ISR)

アドレスオフセット : $0x300$

リセット値 : $0x0000\ 0000$

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権読出しは 0 を返します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIF	XONEIF	SEIF
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEIF** : キーエラー割込みフラグステータス

このビットはハードウェアでセットされ、アプリケーションによってのみ読み出されます。キーレジスタが null または適切に初期化されていない場合 (KEYCRC = 0x0) に暗号化領域で読出しアクセスが発生すると、ビットがセットされます。

ビットは、アプリケーションが OTFDEC_ICR で対応するビットを 1 にセットするとクリアされます。
0 : OTFDEC が適切に動作します。

1 : キーレジスタが null または適切に初期化されていない有効な暗号化領域 (KEYCRC = 0x0) で読出しアクセスを検出しました。OTFDEC は、読出しに対して 0 の値を返し、OTFDEC_IER で KEIE ビットが 1 にセットされている場合はオプションで割込みを生成します。

KEIF がセットされると、誤ったキーレジスタでの領域に対する読出しは 0 の値を返します。これらのキーレジスタが適切に初期化されるまで、この状態が続きます (KEYCRC が 0 ではない場合)。

ビット 1 XONEIF : 実行専用、実行なしエラー割込みフラグステータス

このビットはハードウェアでセットされ、アプリケーションによってのみ読み出されます。MODE ビットが 11 にセットされた暗号化領域で、命令フェッチではない読出しアクセスが検出されると、ビットがセットされます。最後に、暗号化モードが有効で実行アクセスが検出された場合は、XONEIF もセットされます。

ビットは、アプリケーションが OTFDEC_ICR で対応するビットを 1 にセットするとクリアされます。

0 : 実行専用エラーステータスがありません。ペンディング状態の割込みはありません。

1 : MODE ビットが 11 にセットされた ある領域で読出しアクセスが検出されたか、ENC = 1 で実行アクセスが検出されました。OTFDEC は、不正アクセスに対して 0 の値を返し、OTFDEC_IER で XONEIE ビットが 1 にセットされている場合はオプションで割込みを生成します。

ビット 0 SEIF : セキュリティエラー割込みフラグステータス

このビットはハードウェアでセットされ、アプリケーションによってのみ読み出されます。1 つ以上のセキュリティエラーが検出された場合に、ビットがセットされます。

ビットは、アプリケーションが OTFDEC_ICR で対応するビットを 1 にセットするとクリアされます。

0 : セキュリティエラーステータスがありません。ペンディング状態の割込みはありません。

1 : セキュリティエラーフラグステータスと、ペンディング状態の割込みがあります。OTFDEC_IER の対応する SEIE ビットに応じて、実際の割込みが生成されます。

37.6.13 OTFDEC 割込みクリアレジスタ (OTFDEC_ICR)

アドレスオフセット : 0x304

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

PRIV ビットが OTFDEC_PRIVCFGR でセットされている場合、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIF	XONEIF	SEIF
													w	w	w

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 KEIF : キーエラー割込みフラグクリア

このビットはアプリケーションによって書き込まれ、常に 0 が読み出されます。

0 : KEIF フラグステータスは影響を受けません。

1 : KEIF フラグステータスが OTFDEC_ISR でクリアされます。

注 : KEIF をクリアしても、問題の原因 (誤ったキーレジスタ) は解決しません。暗号化領域に再びアクセスできるようにするには、OTFDEC キーレジスタを再度適切に初期化する必要があります。

ビット 1 XONEIF : 実行専用、実行なしエラー割込みフラグクリア

このビットはアプリケーションによって書き込まれ、常に 0 が読み出されます。

0 : XONEIF フラグステータスは影響を受けません。

1 : XONEIF フラグステータスが OTFDEC_ISR でクリアされます。

ビット 0 **SEIF** : セキュリティエラー割込みフラグクリア

このビットはアプリケーションによって書き込まれ、常に 0 が読み出されます。

0 : SEIF フラグステータスは影響を受けません。

1 : SEIF フラグステータスが OTFDEC_ISR でクリアされます。

37.6.14 OTFDEC 割込み有効レジスタ (OTFDEC_IER)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

TrustZone セキュリティが製品で有効になっている場合、非セキュア AHB 書込みアクセス (HNONSEC = 1) は破棄されます。

OTFDEC_PRIVCFGR で PRIV ビットがセットされている場合、非特権読出しは 0 を返し、非特権書込みは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEIE	XONEIE	SEIE
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **KEIE** : キーエラー割込みイネーブル

このビットはアプリケーションによって書込み／読出しされます。KEIF フラグステータスがセットされている場合に、OTFDEC 割込み生成を制御します。

0 : キーエラーフラグ KEIF での割込み生成は無効（マスク済み）です。

1 : キーエラーフラグ KEIF での割込み生成は有効（未マスク）です。

ビット 1 **XONEIE** : 実行専用、実行なしエラー割込みイネーブル

このビットはアプリケーションによって書込み／読出しされます。XONEIF フラグステータスがセットされている場合に、OTFDEC 割込み生成を制御します。

0 : 実行専用エラー XONEIF での割込み生成は無効（マスク済み）です。

1 : 実行専用エラー XONEIF での割込み生成は有効（未マスク）です。

ビット 0 **SEIE** : セキュリティエラー割込みイネーブル

このビットはアプリケーションによって書込み／読出しされます。SEIF フラグステータスがセットされている場合に、OTFDEC 割込み生成を制御します。

0 : セキュリティエラー SEIF での割込み生成は無効（マスク済み）です。

1 : セキュリティエラー SEIF での割込み生成は有効（未マスク）です。

37.6.15 OTFDEC レジスタマップ

表 374. OTFDEC レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	OTFDEC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ENC
	リセット値																																0
0x04 - 0x0C	予約済み	予約済み																															
0x10	OTFDEC_PRIVCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PRIV
	リセット値																																0
0x14- 0x1C	予約済み	予約済み																															
0x20	OTFDEC_R1CFGR1	REG1_VERSION [15:0]															KEYCRC[7:0]							Res	Res	MODE[1:0]		KEYLOCK	CONFIGLOCK	REG_EN			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x24	OTFDEC_R1STARTADDR	REG1_START_ADDR [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	OTFDEC_R1ENDADDR	REG1_END_ADDR [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	OTFDEC_R1NONCER0	REG1_NONCE [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	OTFDEC_R1NONCER1	REG1_NONCE [63:32]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	OTFDEC_R1KEYR0	REG1_KEY [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	OTFDEC_R1KEYR1	REG1_KEY [63:32]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	OTFDEC_R1KEYR2	REG1_KEY [95:64]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	OTFDEC_R1KEYR3	REG1_KEY [95:64]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44 - 0x4C	予約済み	予約済み																															
0x50	OTFDEC_R2CFGR	REG2_VERSION [15:0]															KEYCRC[7:0]							Res	Res	MODE[1:0]		KEYLOCK	CONFIGLOCK	REG_EN			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x54	OTFDEC_R2STARTADDR	REG2_START_ADDR [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 374. OTFDEC レジスタのマップとリセット値 (続き)

[illegible]

表 374. OTFDEC レジスタのマップとリセット値 (続き)

[illegible]

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

38 高機能制御タイマ (TIM1/TIM8)

38.1 TIM1/TIM8 の概要

高機能制御タイマ (TIM1/TIM8) は、プログラム可能なプリスケアラによって駆動される 16 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

高機能制御タイマ (TIM1/TIM8) と汎用タイマ (TIMy) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 38.3.30 : タイマの同期](#)に示すように、相互に同期させることができます。

38.2 TIM1/TIM8 の主な特徴

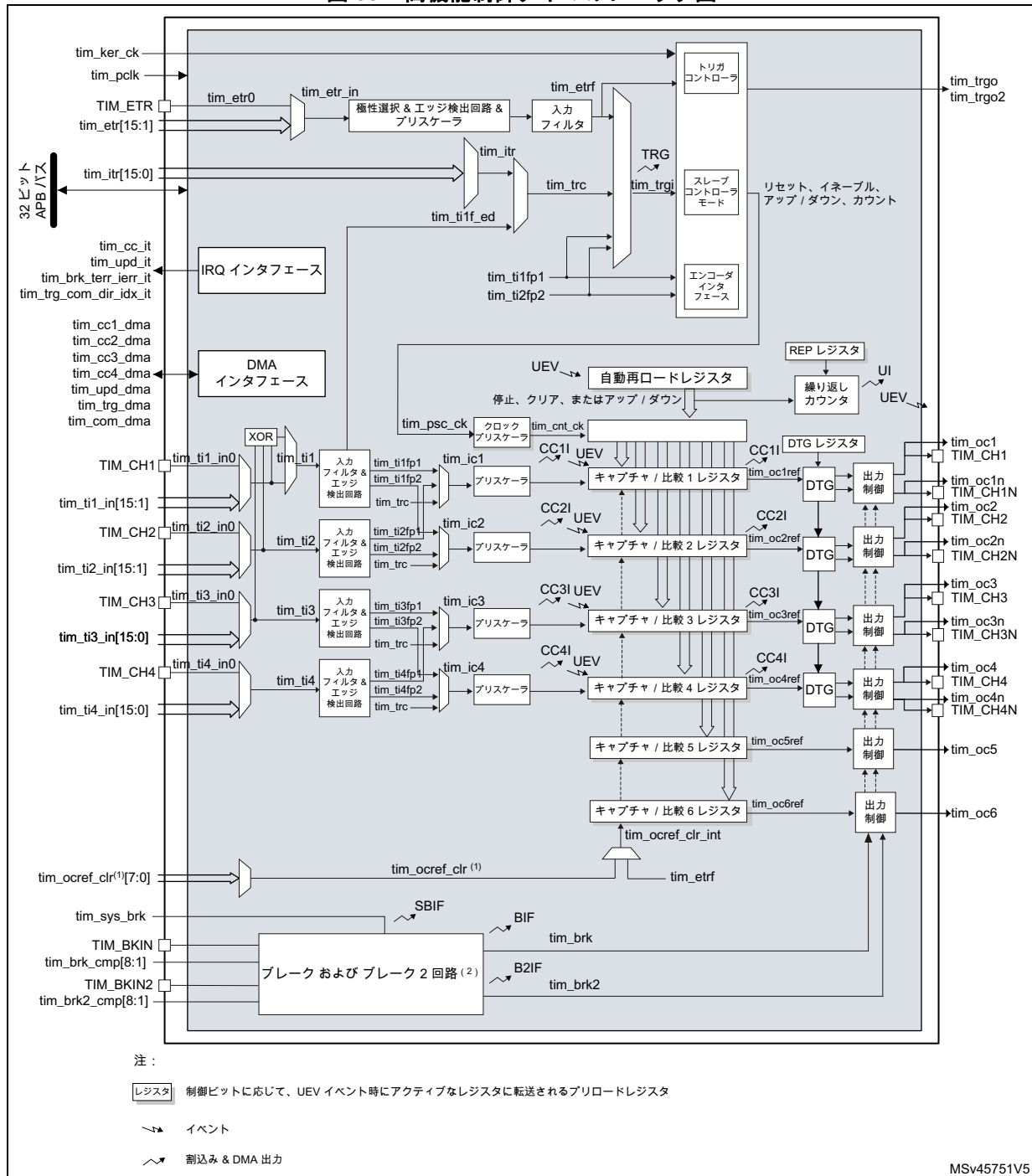
TIM1/TIM8 タイマの主な機能は、次のとおりです。

- 16 ビットのアップカウンタ、ダウンカウンタ、アップ/ダウン自動再ロードカウンタ。
- 16 ビットのプログラム可能なプリスケアラ。カウンタクロック周波数を 1 から 65536 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 6 つの独立チャネル：
 - 入力キャプチャ (ただしチャネル 5 および 6)
 - 出力比較
 - PWM 生成 (エッジアラインモードとセンターアラインモード)
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をユーザが選択可能な安全な設定にする 2 つのブレイク入力。
- 以下のイベント時の割込み/DMA 生成：
 - 更新：カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - 入力キャプチャ
 - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

38.3 TIM1/TIM8機能詳細

38.3.1 ブロック図

図 337. 高機能制御タイマのブロック図



- この機能は一部のタイマでは利用できません (セクション 38.3.2: TIM1/TIM8 ピンおよび内部信号を参照)。
- 詳細については、図 384: ブレイクおよびブレイク2回路の概要を参照してください。

38.3.2 TIM1/TIM8 ピンおよび内部信号

このセクションの以下の表は、TIM の入力および出力の一覧です。

表 375. TIM の入出力ピン

ピン名	信号タイプ	説明
TIM_CH1 TIM_CH2 TIM_CH3 TIM_CH4	入力／出力	タイマ多目的チャネル。 各チャネルは、キャプチャ、比較、または PWM で使用できます。 TIM_CH1 と TIM_CH2 は、外部クロック (tim_ker_ck クロックの 1/4 未満)、外部トリガ、および直交エンコーダの入力としても使用できます。 TIM_CH1、TIM_CH2、および TIM_CH3 は、デジタルホール効果センサとのインターフェースに使用できます。
TIM_CH1N TIM_CH2N TIM_CH3N TIM_CH4N	出力	デッドタイム挿入の可能性がある TIM_CHx 出力から導き出されるタイマ相補出力。
TIM_ETR	入力	外部トリガ入力。この入力 は、外部トリガまたは外部クロックソースとして使用できます。tim_etr_in プリスケーラが使用されている場合、この入力は tim_ker_ck よりも高い周波数のクロックを受信できます。
TIM_BKIN TIM_BKIN2	入力／出力	ブレークおよびブレーク 2 入力。これらの入力は双方向モードにも設定することができます。

表 376. TIM 内部入力／出力信号

内部信号名	信号タイプ	説明
tim_ti1_in [15:0] tim_ti2_in [15:0] tim_ti3_in [15:0] tim_ti4_in [15:0]	入力	内部タイマ入力バス。tim_ti1_in[15 : 0] と tim_ti2_in[15 : 0] の入力は、キャプチャ用に、または外部クロック (tim_ker_ck クロックの 1/4 未満) として、および直交エンコーダの信号に使用できます。
tim_etr[15:0]	入力	外部トリガ内部入力バス。これらの入力は、トリガ、外部クロックとして、またはハードウェアのサイクルごとのパルス幅制御に使用できます。tim_etr_in プリスケーラが使用されている場合、これらの入力は tim_ker_ck よりも高い周波数のクロックを受信できます。
tim_itr[15:0]	入力	内部トリガ入力バス。これらの入力は、スレーブモードコントローラ用に、または入力クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_trgo/tim_trgo2	出力	内部トリガ出力。これらのトリガは、他のタイマおよび／またはその他のペリフェラルによって使用されます。

表 376. TIM 内部入力／出力信号 (続き)

内部信号名	信号タイプ	説明
tim_ocref_clr[7:0]	入力	タイマの tim_ocref_clr 入力バス。これらの入力 は、通常ハードウェアのサイクルごとのパルス幅 制御のために、tim ocxref 信号をクリアするた めに使用できます。
tim_brk_cmp[8:1]	入力	内部信号のブレーク入力
tim_brk2_cmp [8:1]	入力	内部信号のブレーク 2 入力
tim_sys_brk[n:0]	入力	システムブレーク入力この入力は MCU のシス テムレベルのエラーを集めます。
tim_pclk	入力	タイマ APB クロック
tim_ker_ck	入力	タイマカーネルクロック
tim_cc_it	出力	タイマのキャプチャ／比較割込み
tim_upd_it	出力	タイマ更新イベント割込み
tim_brk_terr_ierr_it	出力	タイマブレーク、ブレーク 2、遷移エラー、およ びインデックスエラー割込み
tim_trg_com_dir_idx_it	出力	タイマトリガ、転流、方向、およびインデックス 割込み
tim_cc1_dma tim_cc2_dma tim_cc3_dma tim_cc4_dma	出力	タイマキャプチャ／比較 1~4 DMA リクエスト
tim_upd_dma	出力	タイマ更新 DMA リクエスト
tim_trg_dma	出力	タイマトリガ DMA リクエスト
tim_com_dma	出力	タイマ転流 DMA リクエスト

表 377、表 378、表 379、および表 380 に、tim_ti[4:1] 入力マルチプレクサに接続されるソースを列挙します。

表 377. tim_ti1 入力マルチプレクサへの相互接続

tim_ti1 入力	ソース	
	TIM1	TIM8
tim_ti1_in0	TIM1_CH1	TIM8_CH1
tim_ti1_in [15:1]	予約済み	予約済み

表 378. tim_ti2 入力マルチプレクサへの相互接続

tim_ti2 入力	ソース	
	TIM1	TIM8
tim_ti2_in0	TIM1_CH2	TIM8_CH2
tim_ti2_in [15:1]	予約済み	予約済み

表 379. tim_ti3 入力マルチプレクサへの相互接続

tim_ti3 入力	ソース	
	TIM1	TIM8
tim_ti3_in0	TIM1_CH3	TIM8_CH3
tim_ti2_in [15:1]	予約済み	予約済み

表 380. tim_ti4 入力マルチプレクサへの相互接続

tim_ti4 入力	ソース	
	TIM1	TIM8
tim_ti4_in0	TIM1_CH4	TIM8_CH4
tim_ti4_in [15:1]	予約済み	予約済み

表 381 に、tim_itr 入力マルチプレクサに接続される内部ソースを列挙します。

表 381. 内部トリガ接続

タイマ内部トリガ入力信号	TIM1	TIM8
tim_itr0	予約済み	tim1_trgo
tim_itr1	tim2_trgo	tim2_trgo
tim_itr2	tim3_trgo	tim3_trgo
tim_itr3	tim4_trgo	tim4_trgo
tim_itr4	tim5_trgo	tim5_trgo
tim_itr5	tim8_trgo	予約済み
tim_itr6	tim12_trgo	tim12_trgo
tim_itr7	tim13_oc1	tim13_oc1

表 381. 内部トリガ接続 (続き)

タイマ内部トリガ入力信号	TIM1	TIM8
tim_itr8	tim14_oc1	tim14_oc1
tim_itr9	tim15_trgo	tim15_trgo
tim_itr10	tim16_oc1	tim16_oc1
tim_itr11	tim17_oc1	tim17_oc1
tim_itr[15:12]	予約済み	予約済み

表 382 に、tim_etr 入力マルチプレクサに接続される内部ソースを列挙します。

表 382. tim_etr 入力マルチプレクサへの相互接続

タイマ外部トリガ入力信号	タイマ外部トリガ信号割り当て	
	TIM1	TIM8
tim_etr0	TIM1_ETR	TIM8_ETR
tim_etr[2:1]	予約済み	予約済み
tim_etr3	adc1_awd1	adc2_awd1
tim_etr4	adc1_awd2	adc2_awd2
tim_etr5	adc1_awd3	adc2_awd3
tim_etr[15:6]	予約済み	予約済み

表 383、表 384、および表 385 に、tim_brk および tim_brk2 入力に接続されるソースを列挙します。

表 383. タイマブレイク相互接続

tim_brk 入力	TIM1	TIM8
TIM_BKIN	TIM1_BKIN ピン	TIM8_BKIN ピン
tim_brk_cmp[8:1]	予約済み	予約済み

表 384. タイマ break2 相互接続

tim_brk2 入力	TIM1	TIM8
TIM_BKIN2	TIM1_BKIN2 ピン	TIM8_BKIN2 ピン
tim_brk2_cmp [8:1]	予約済み	予約済み

表 385. システムブレイク相互接続

tim_sys_brk 入力	TIM1/TIM8	SBS_CFGR2 レジスタのイネーブルビット
tim_sys_brk0	FLASH ダブル ECC エラー	ECCL
tim_sys_brk1	プログラム可能な電圧検出器 (PVD)	PVDL
tim_sys_brk2	SRAM ダブル ECC エラー	SEL
tim_sys_brk3	Cortex®-M33 LOCKUP	CLL
CSS	クロックセキュリティシステム	なし (常に有効)

38.3.3 タイムベースユニット

プログラム可能な高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、カウンタが動作中でも、ソフトウェアで読み書きができます。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ有効な、プリスケアラ出力 tim_cnt_ck から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

注： TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があります。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周します。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。



 338 と  339 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 338. プリスケーラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

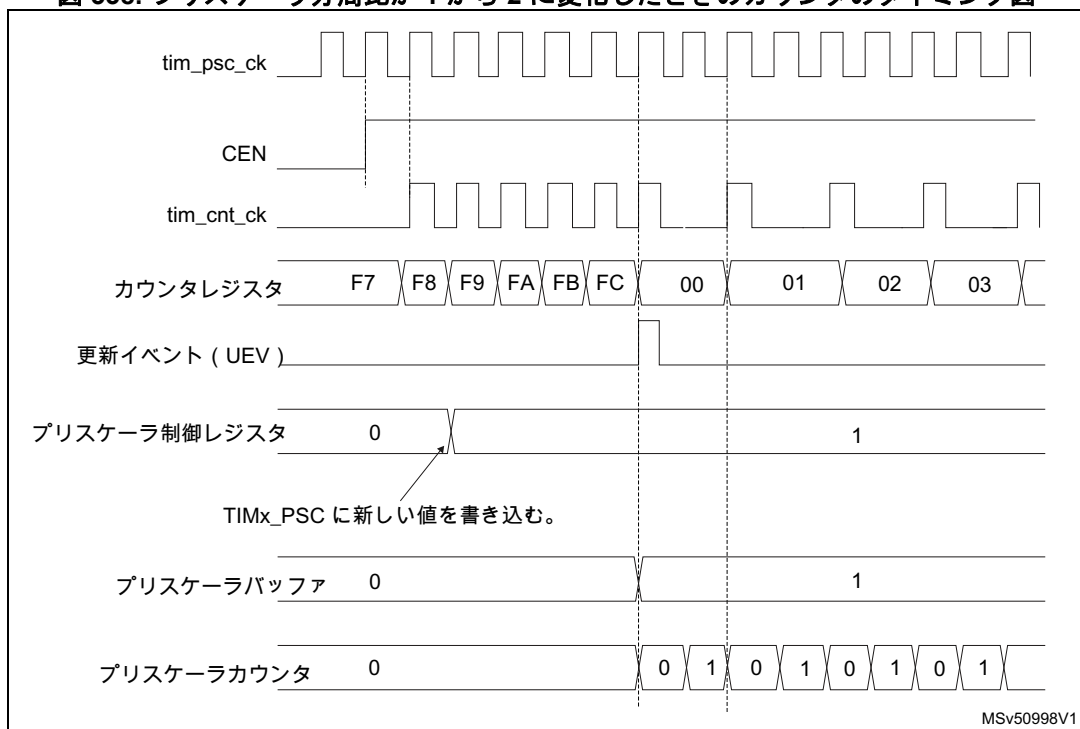
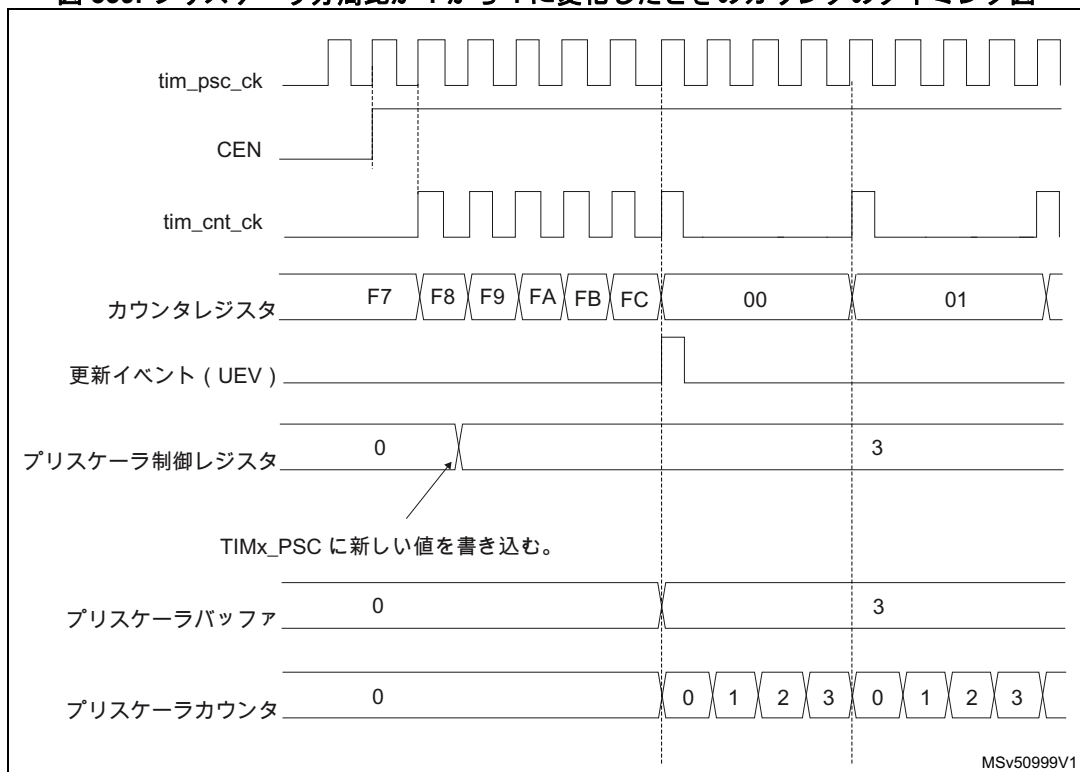


図 339. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



38.3.4 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) + 1 までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みが発生するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのパッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 340. 内部クロック分周比が 1 の場合のカウンタのタイミング図

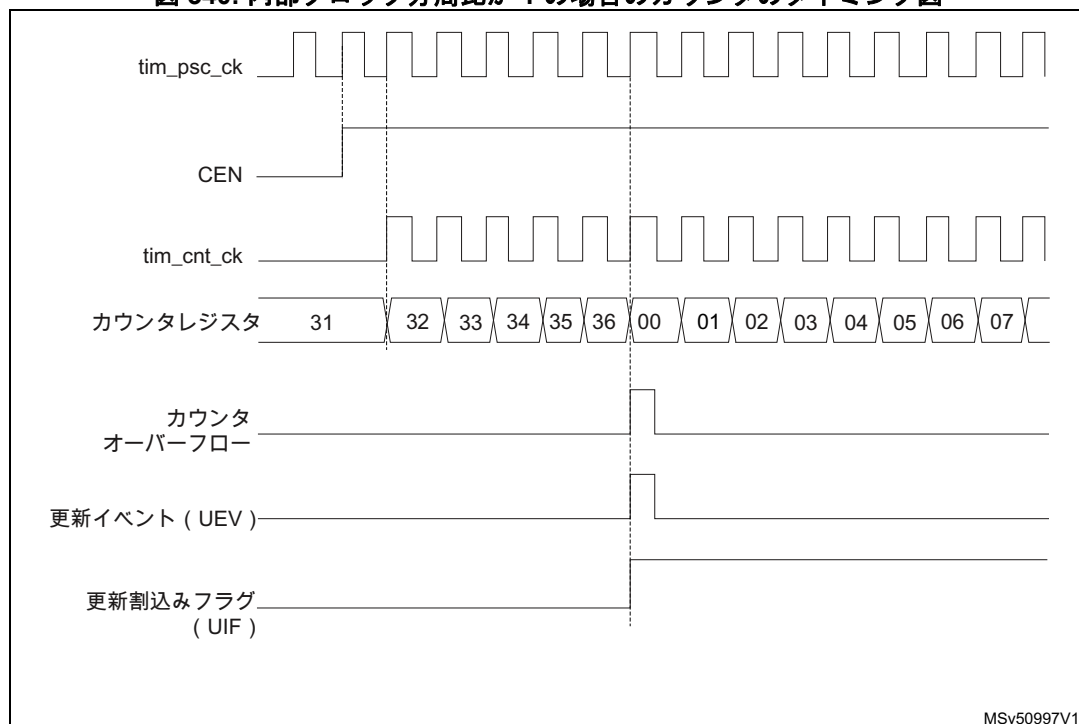


図 341. 内部クロック分周比が 2 の場合のカウンタのタイミング図

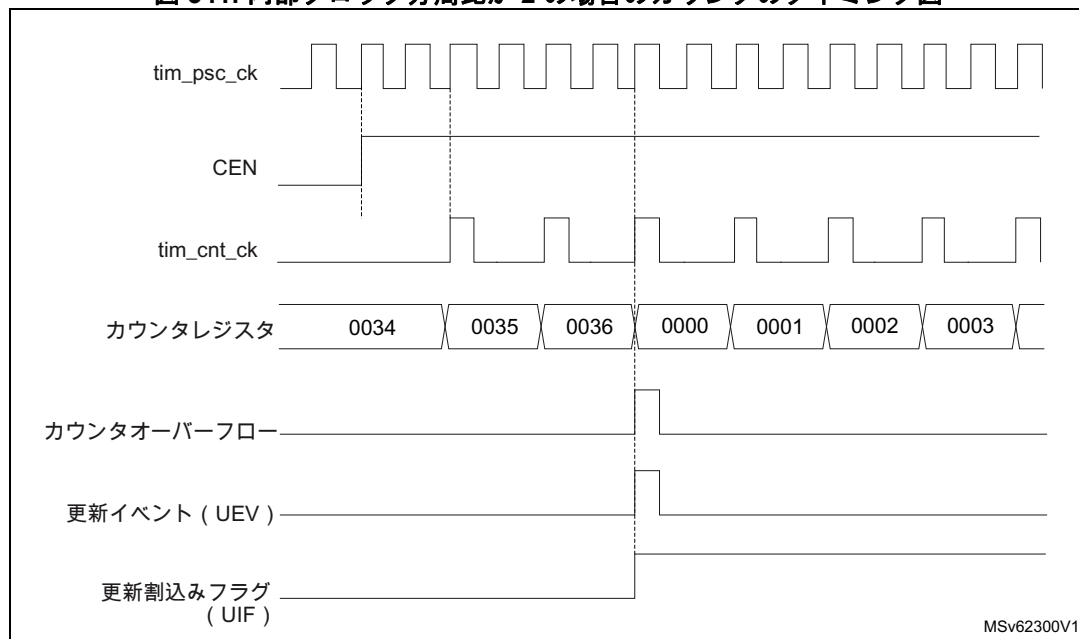


図 342. 内部クロック分周比が 4 の場合のカウンタのタイミング図

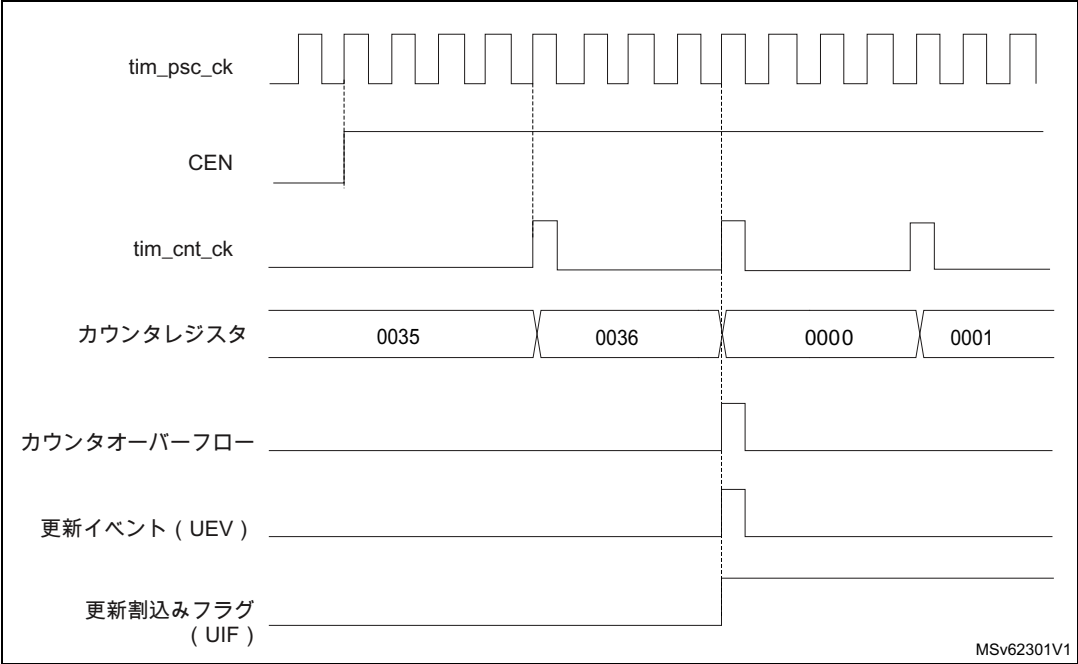


図 343. 内部クロック分周比が N の場合のカウンタのタイミング図

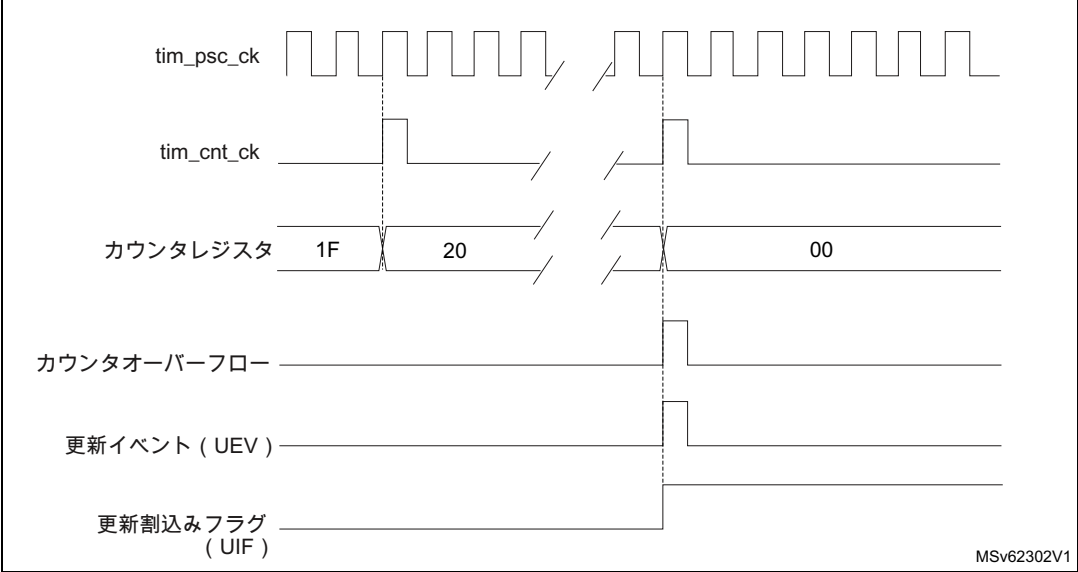


図 344. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

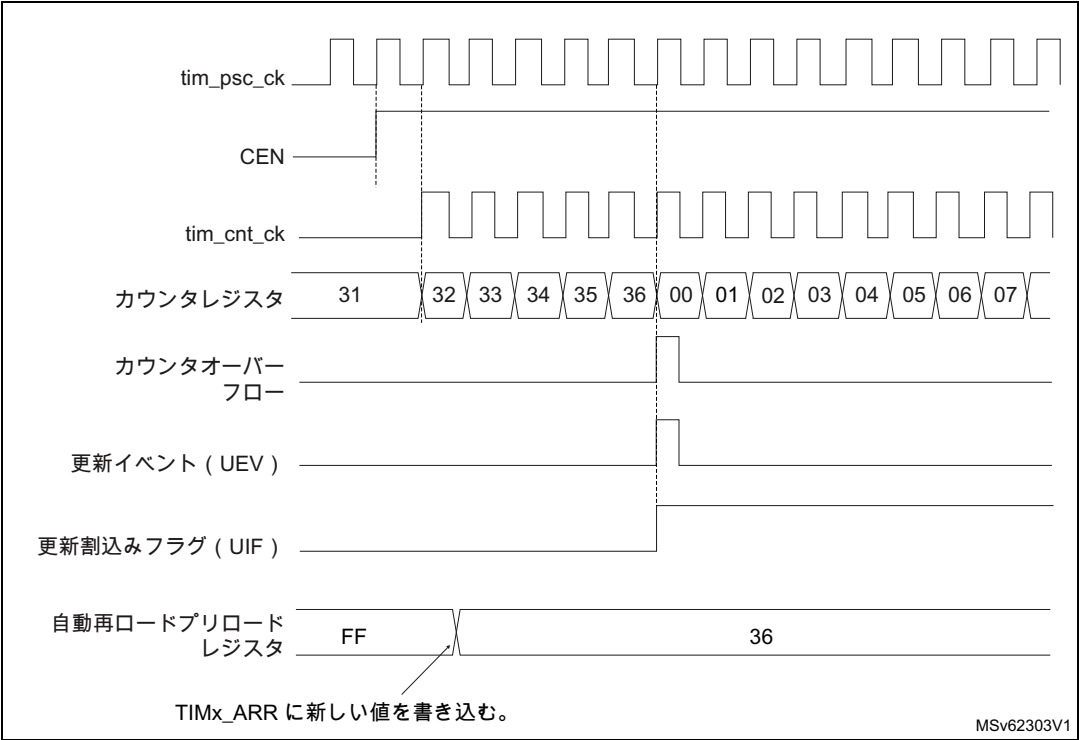
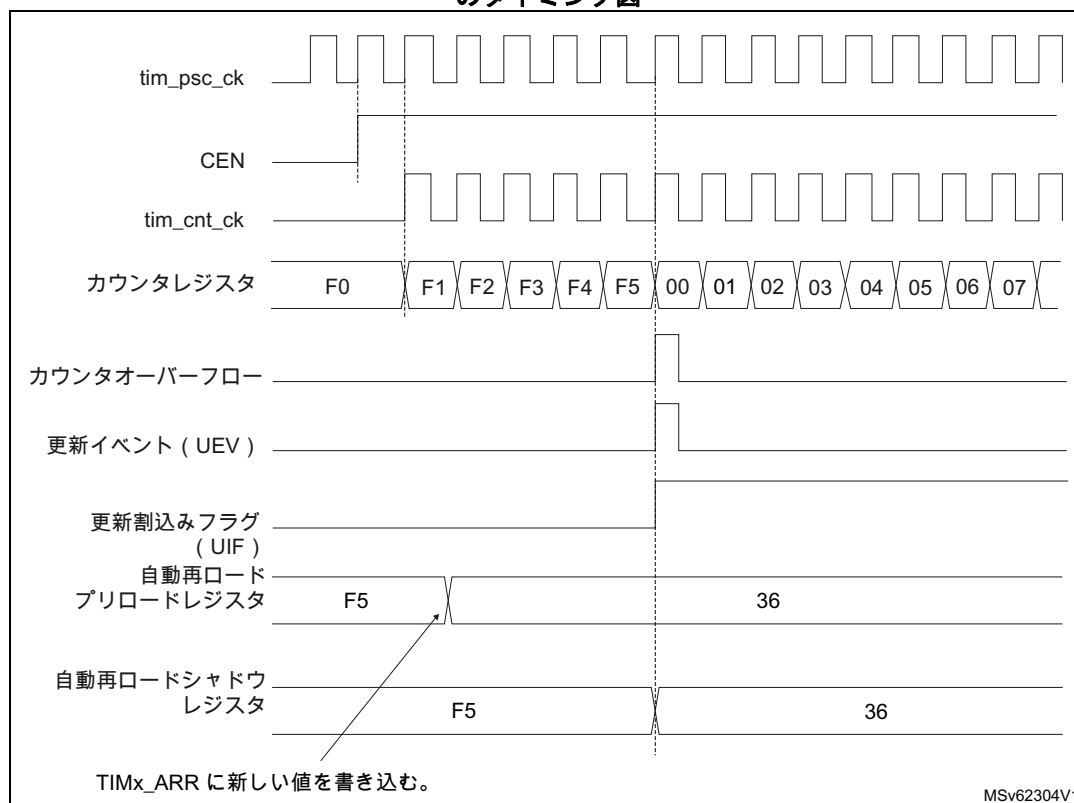


図 345. ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) + 1 までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みが発生するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 346. 内部クロック分周比が 1 の場合のカウンタのタイミング図

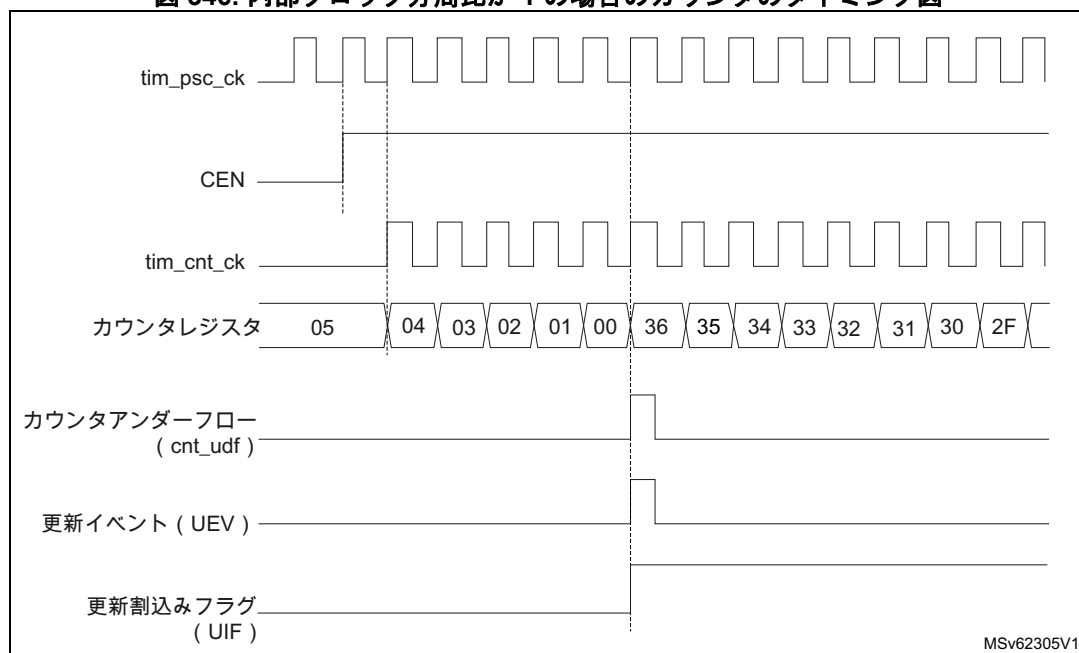


図 347. 内部クロック分周比が 2 の場合のカウンタのタイミング図

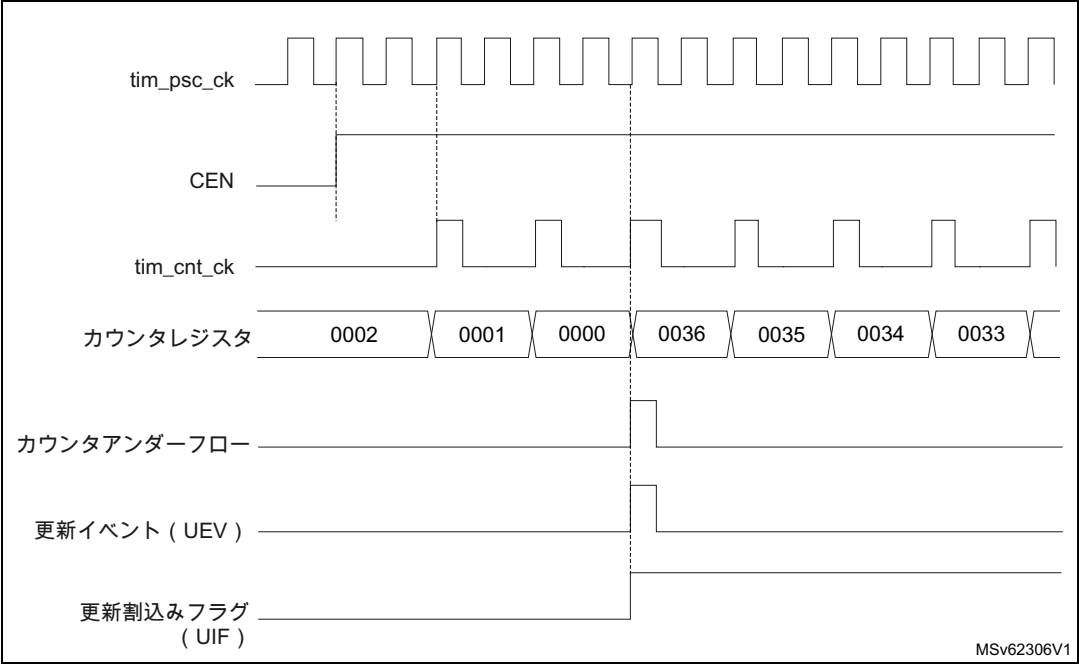


図 348. 内部クロック分周比が 4 の場合のカウンタのタイミング図

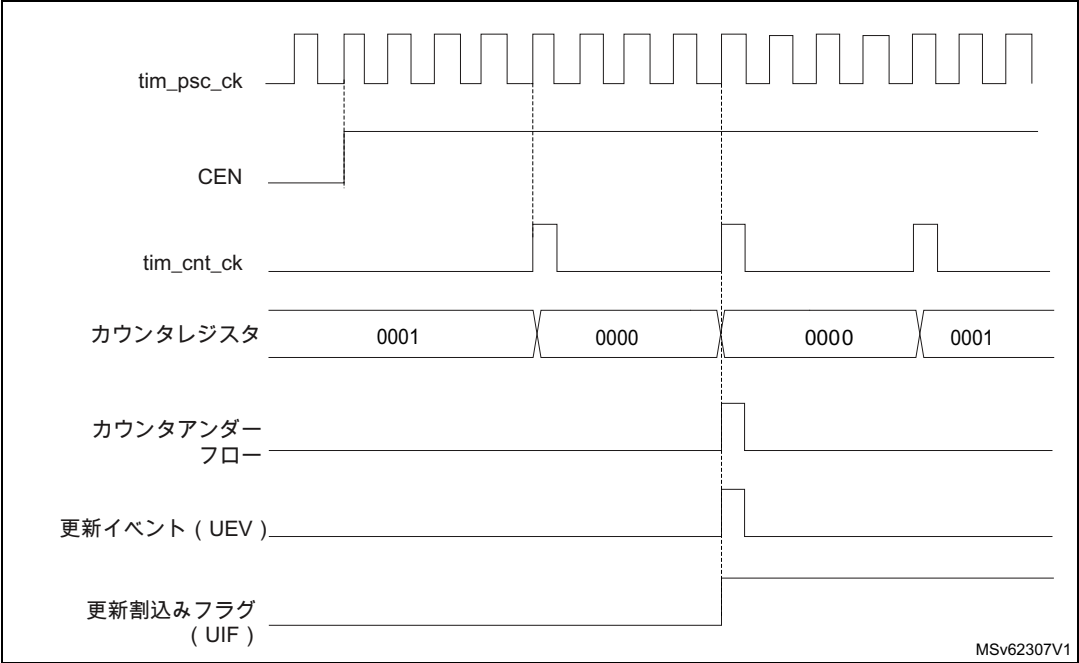


図 349. 内部クロック分周比が N の場合のカウンタのタイミング図

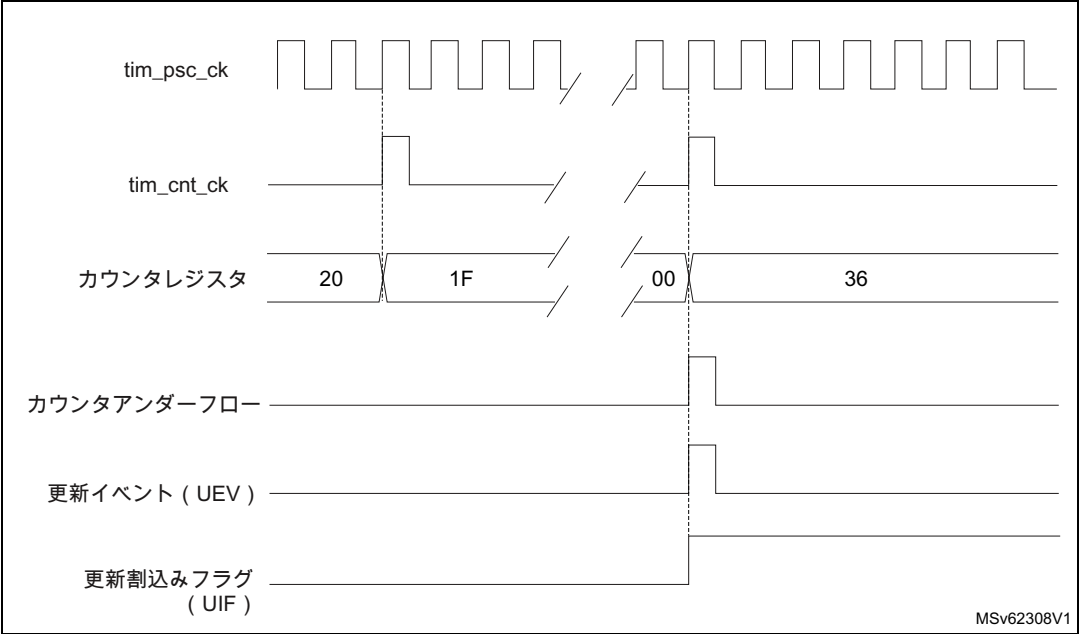
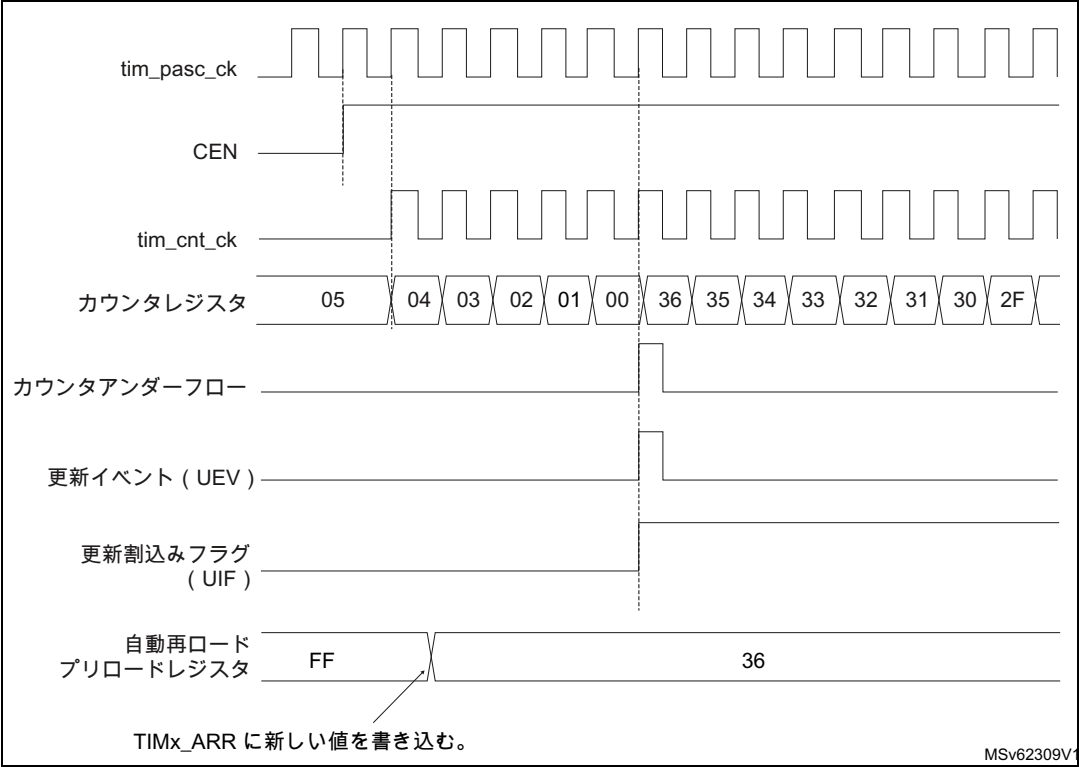


図 350. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図



センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、TIMx_CR1 レジスタの方向ビット (DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

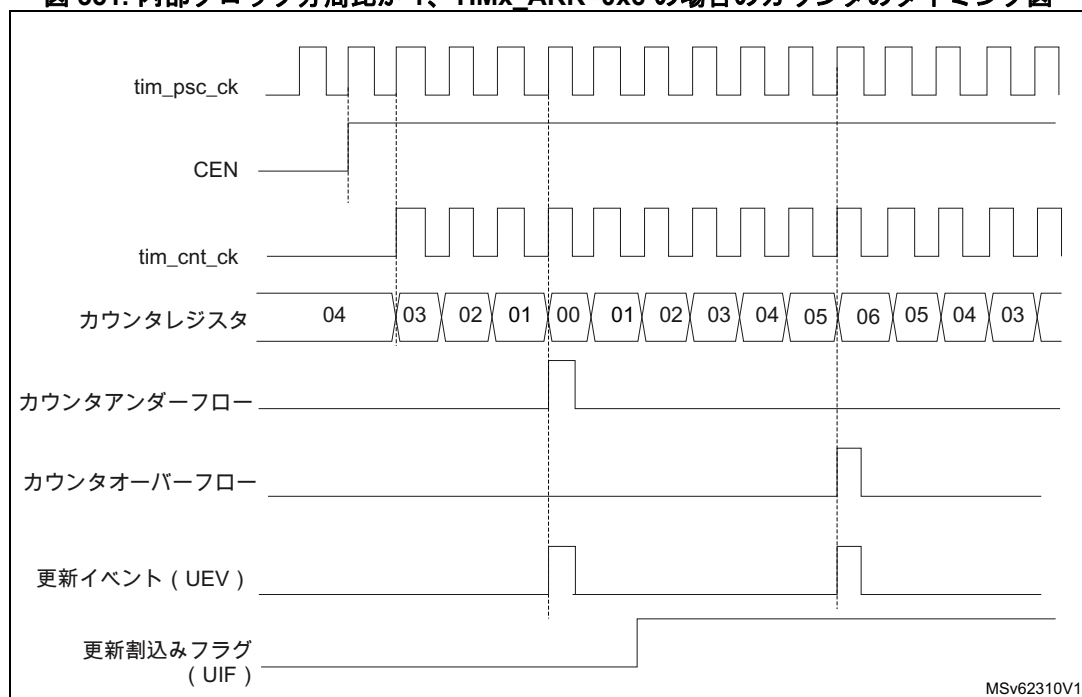
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みが発生するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 351. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 38.6 : TIM1/TIM8 レジスタ](#)を参照)。

図 352. 内部クロック分周比が 2 の場合のカウンタのタイミング図

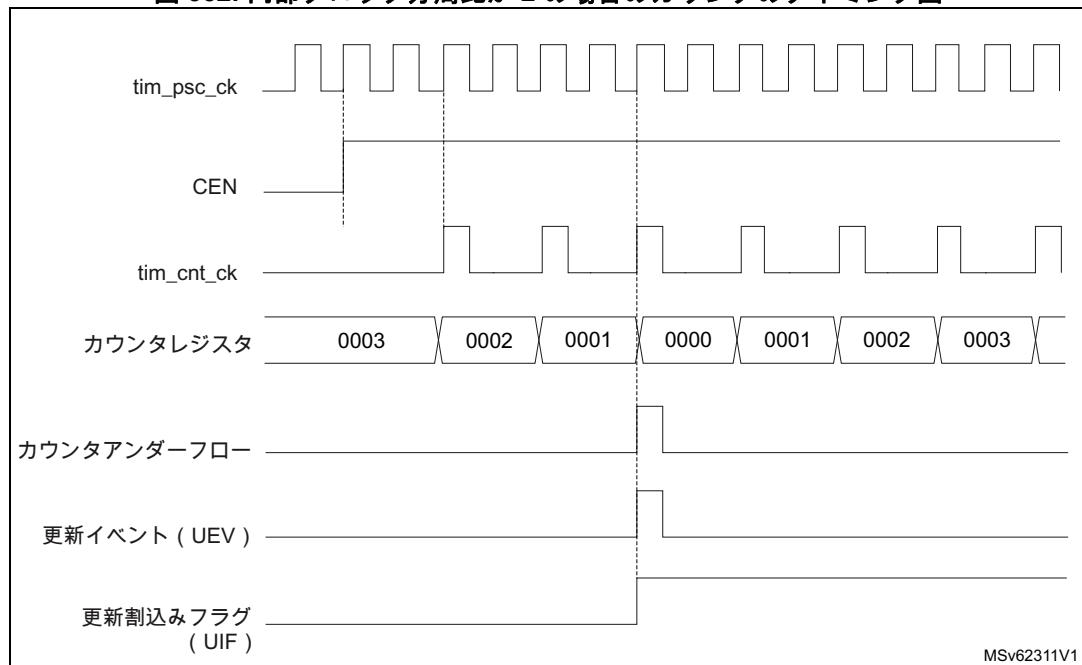


図 353. 内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図

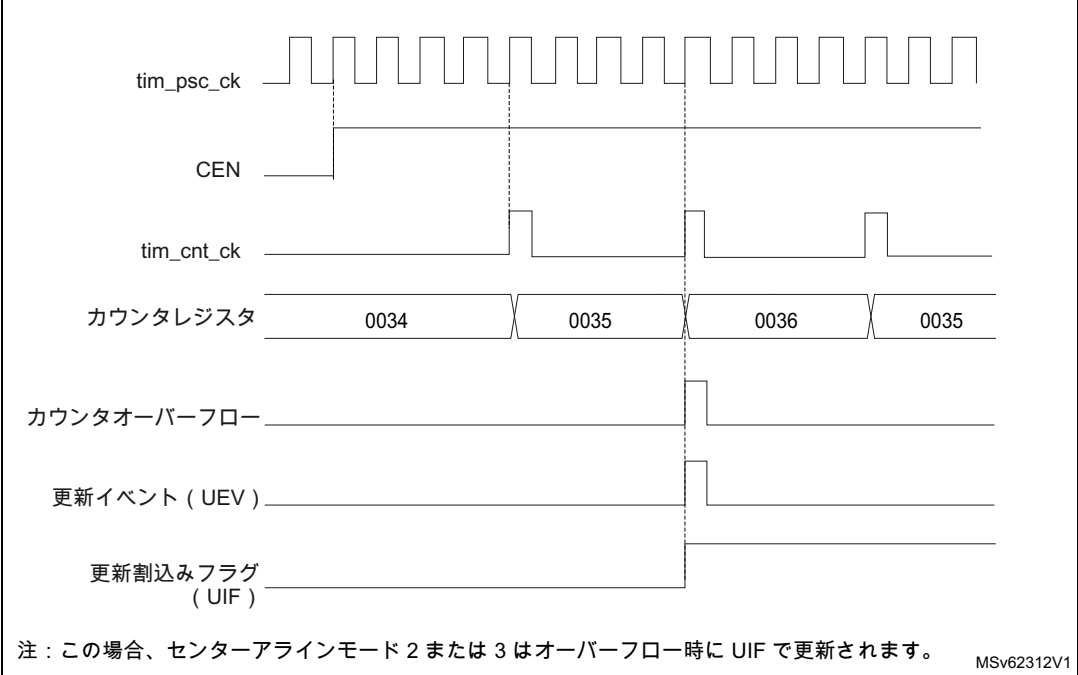


図 354. 内部クロック分周比が N の場合のカウンタのタイミング図

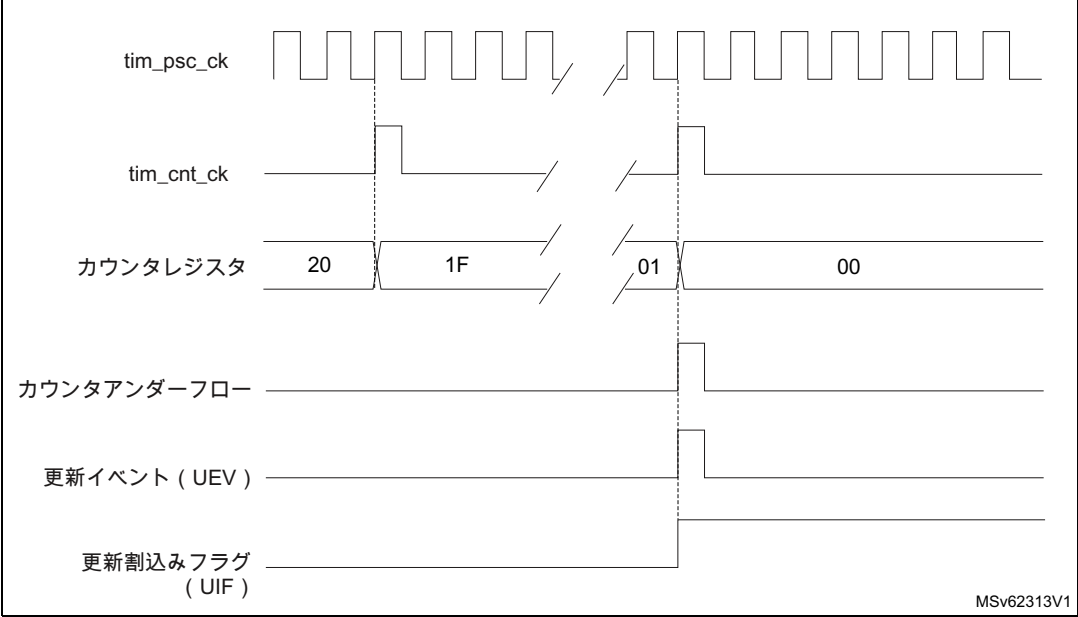


図 355. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

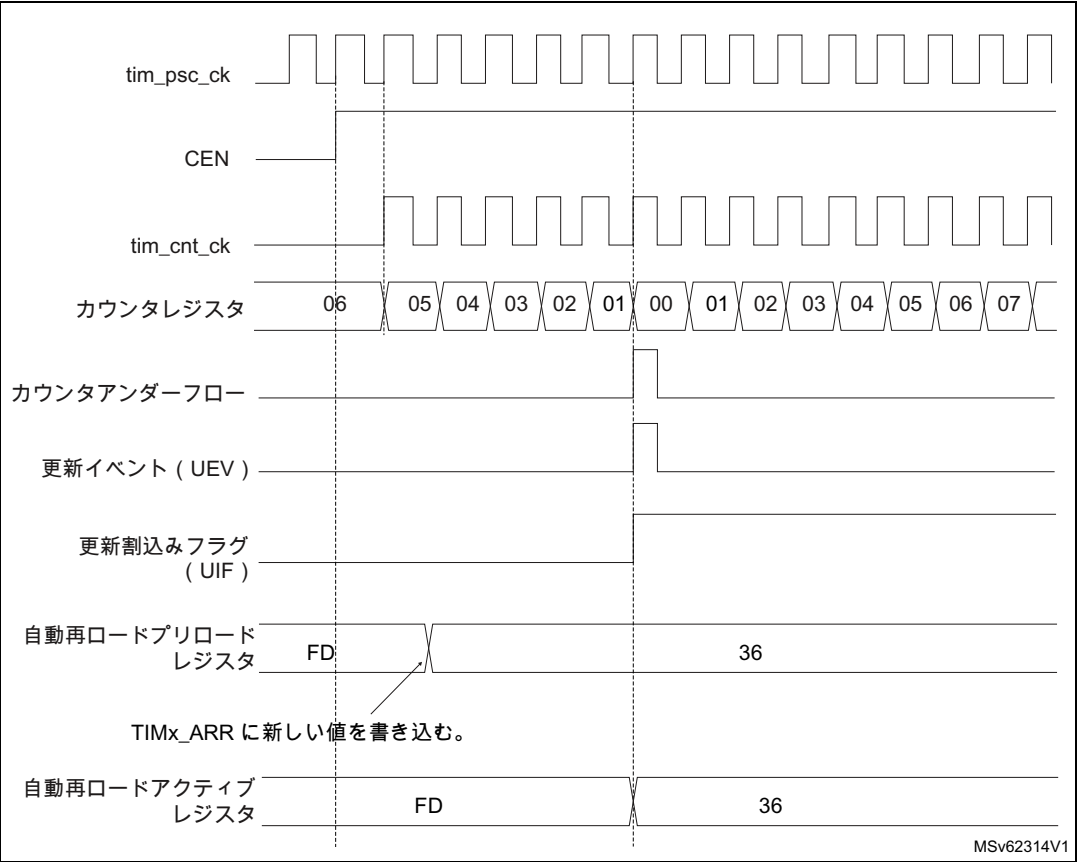
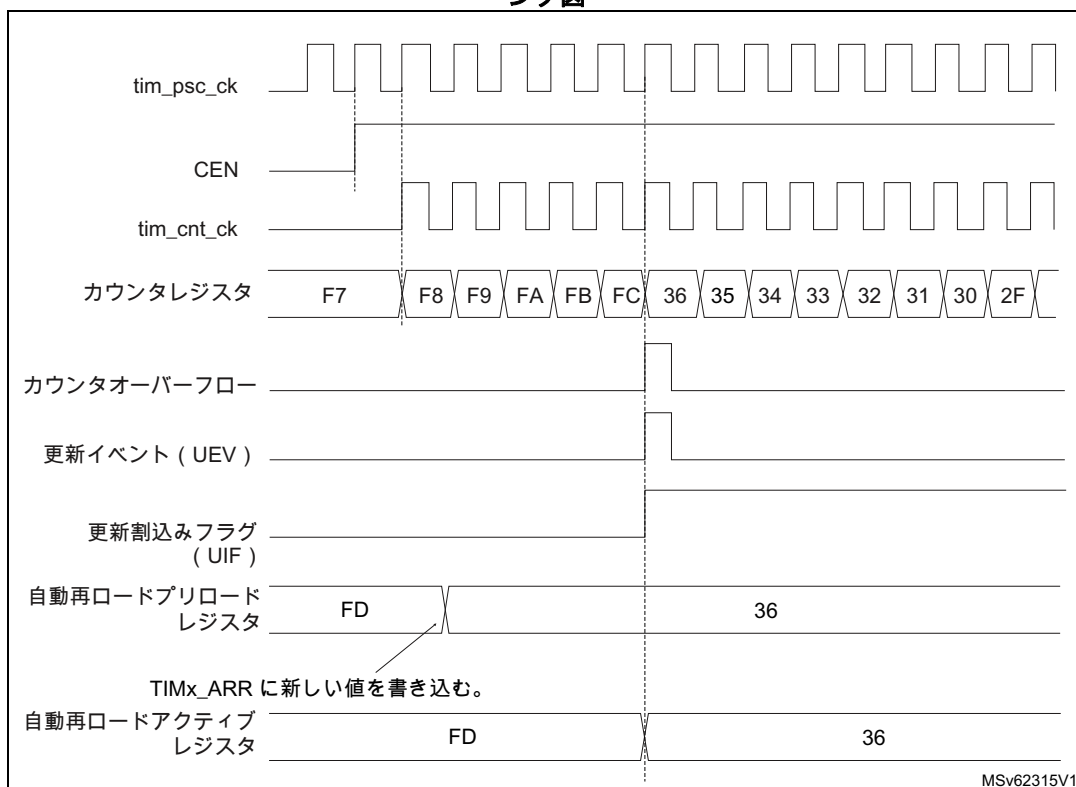


図 356. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



38.3.5 繰返しカウンタ

セクション 38.3.3: タイムベースユニットに、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx_RCR 繰返しカウンタレジスタの値を N とすると、N+1 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx_ARR 自動再ロードレジスタ、TIMx_PSC プリスケアラレジスタ、比較モードの TIMx_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰返しカウンタは、次の場合にデクリメントします。

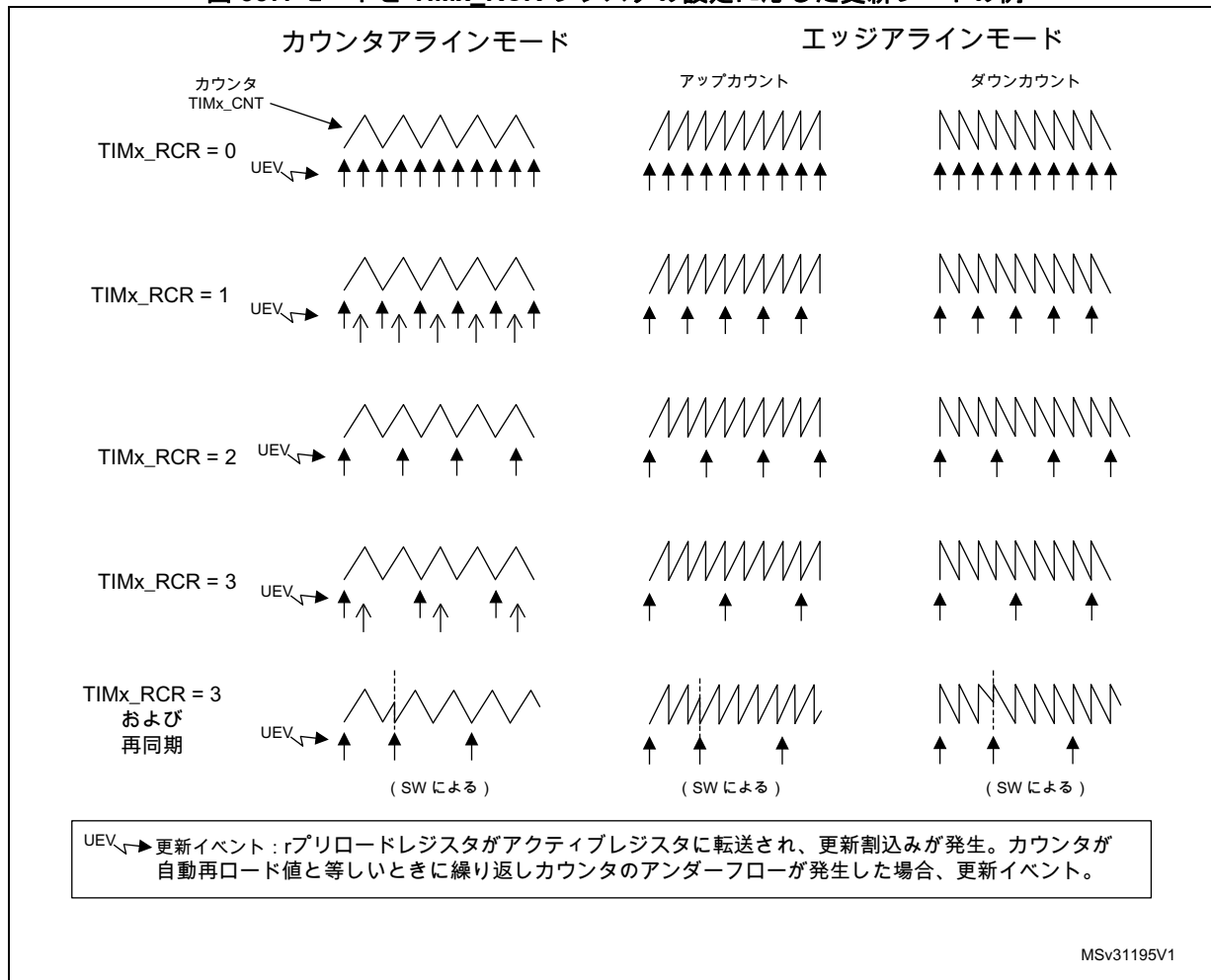
- アップカウントモードで、カウンタオーバーフローごと
- ダウンカウントモードで、カウンタアンダーフローごと
- センターアラインモードで、カウンタオーバーフローとカウンタアンダーフローごと最大繰返し回数は 32768 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センターアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は $2 \times T_{ck}$ です。

繰返しカウンタは自動再ロードタイプです。繰返しの回数は、TIMx_RCR レジスタの値によって定義されたとおりに維持されます (図 357 を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰返しカウンタの値にかかわらず直ちにイベントが発生し、繰返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

センターラインモードでは、RCR が奇数の場合、RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベントが発生します。カウンタの開始前に RCR が書き込まれた場合は、アンダーフローで、UEV が発生します。カウンタの開始後に RCR が書き込まれた場合は、オーバーフローで UEV が発生します。

たとえば、TIMx_RCR = 3 の場合、TIMx_RCR の書き込みタイミングに応じて 4 回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 357. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



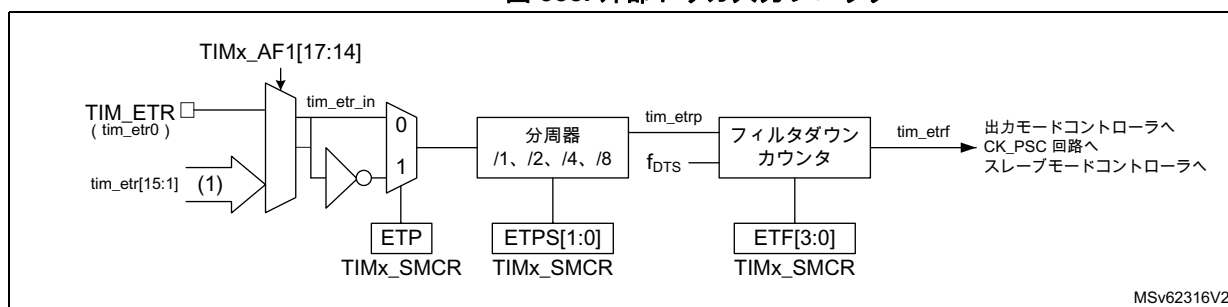
38.3.6 外部トリガ入力

タイマには外部トリガ入力 `tim_etr_in` 機能があります。以下の目的で使用できます。

- 外部クロック（外部クロックモード 2、[セクション 38.3.7](#) を参照）
- スレーブモードのトリガ（[セクション 38.3.30](#) を参照）
- サイクルごとの電流調整の PWM リセット入力（[セクション 38.3.9](#) を参照）

以下の [図 358](#) は、`tim_etr_in` 入力の前処理について説明しています。入力の極性は、TIMxSMCR レジスタの ETP ビットで定義されています。トリガは ETPS[1:0] ビットフィールドでプログラムされた分周器でプリスケールし、ETF[3:0] ビットフィールドでデジタル的にフィルタリングすることができます。結果の信号 (`tim_etrf`) は、次の 3 つの目的に使用できます。すなわち、外部クロックとして、出力を調整する（通常は電流制限のために PWM 出力をリセットする）ため、およびスレーブモードコントローラのトリガとしてです。

図 358. 外部トリガ入力ブロック



MSv62316V2

`tim_etr_in` 入力は、入力ピン（デフォルト設定）、内部ソースといった複数のソースから入力されます。この選択は、TIMx_AF1 レジスタの ETRSEL[3:0] ビットフィールドで行います。

製品の `etr_in` 入力へ接続されるソースの一覧については[セクション 38.3.2: TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

38.3.7 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

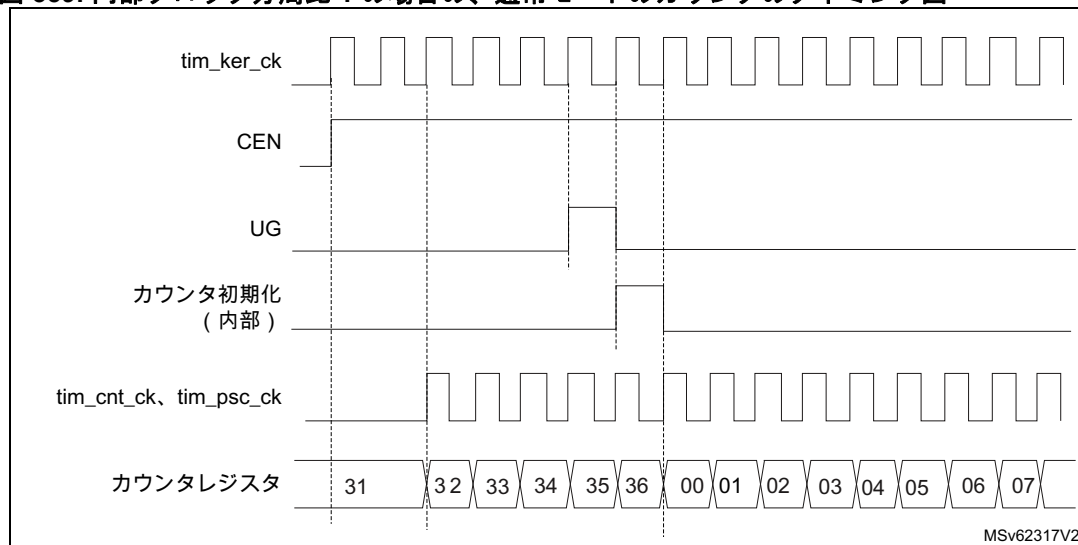
- 内部クロック (`tim_ker_ck`)
- 外部クロックモード 1 : 外部入力ピン (`tim_ti1` または `tim_ti2`)
- 外部クロックモード 2 : 外部トリガ入力 (`tim_etr_in`)
- エンコーダモード

内部クロックソース (`tim_ker_ck`)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます（自動的にクリア状態に保たれる UG ビットを除きます）。CEN ビットに 1 が書き込まれるとすぐに、プリスケアラには内部クロック `tim_ker_ck` が供給されます。

[図 359](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

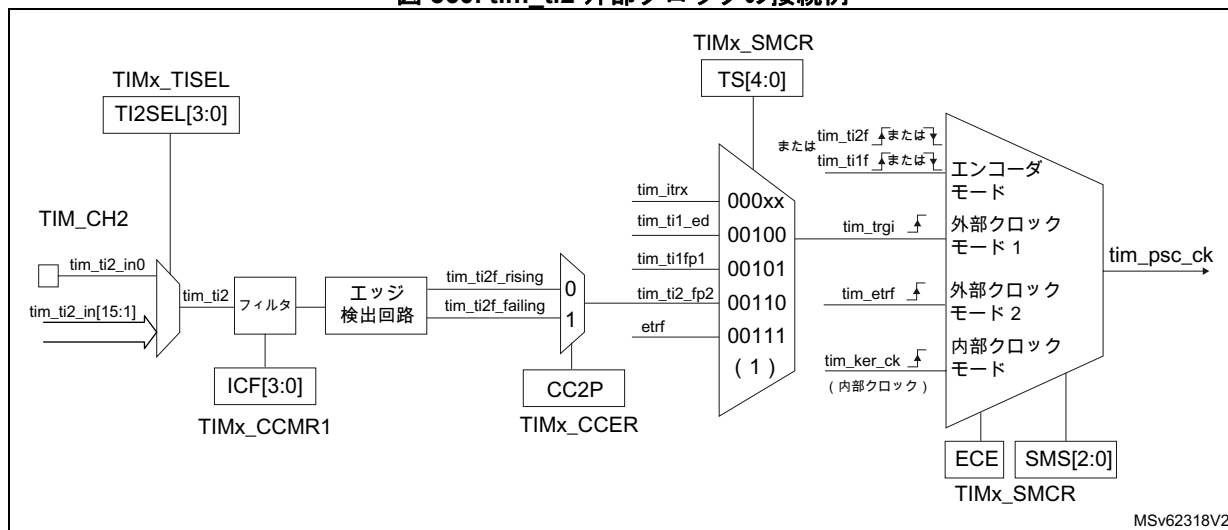
図 359. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立上がりまたは立下がりエッジでカウントすることができます。

図 360. tim_ti2 外部クロックの接続例



1. 01000 から 11111 の範囲のコードは予約済み。

たとえば、tim_ti2 入力の立上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

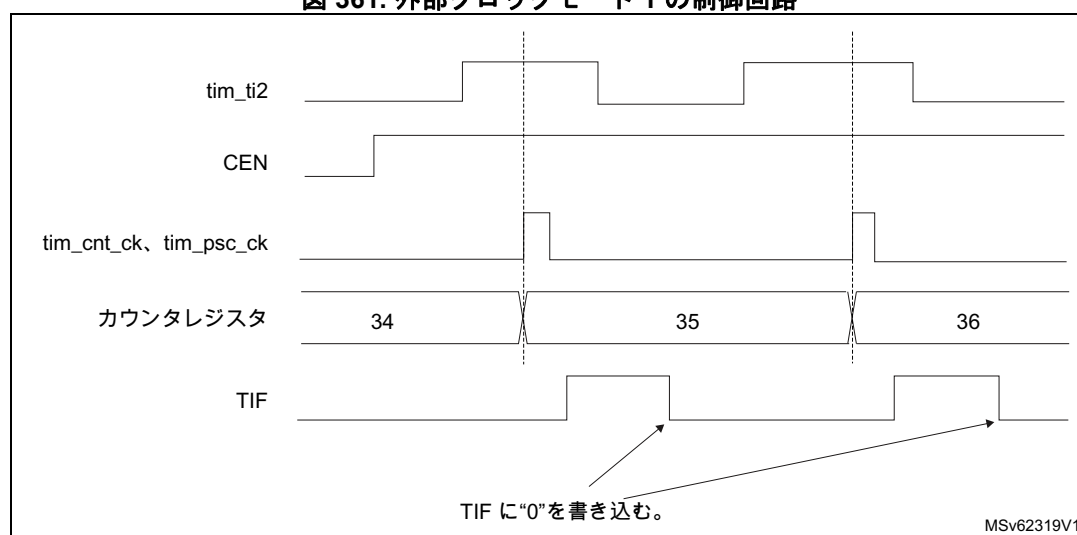
1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が tim_ti2 入力の立上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。
3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=00110 を書き込むことによって、トリガ入力ソースとして tim_ti2 を選択します。*/
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケアラはトリガには使用されないので、設定する必要はありません。

tim_ti2 の立上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際のカウンタクロックまでの間には、tim_ti2 入力の再同期回路による遅延があります。

図 361. 外部クロックモード 1 の制御回路



MSv62319V1

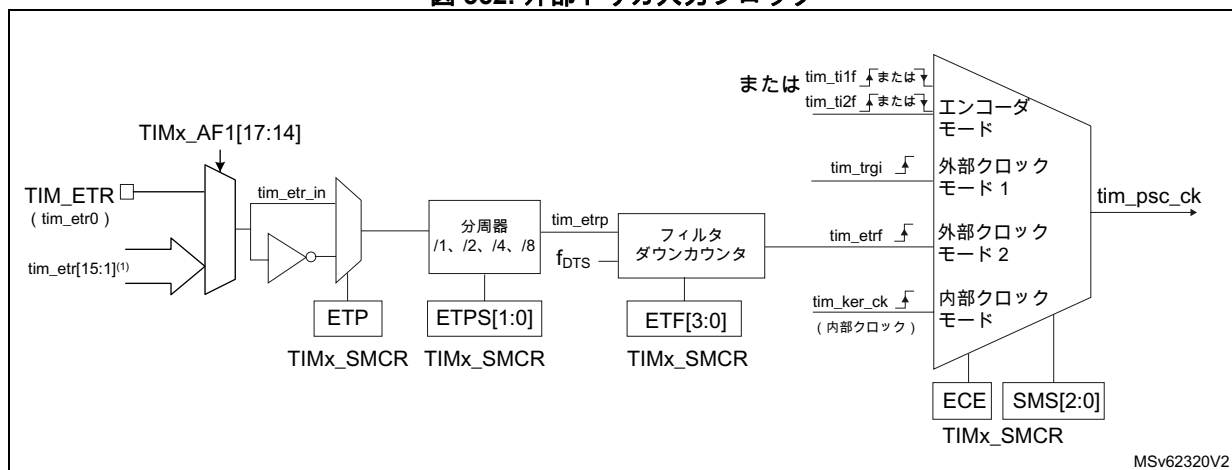
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 tim_etr_in の立上がりまたは立下がりエッジごとにカウントします。

図 362 に、外部トリガ入力ブロックの概要を示します。

図 362. 外部トリガ入力ブロック



1. セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号 を参照してください。

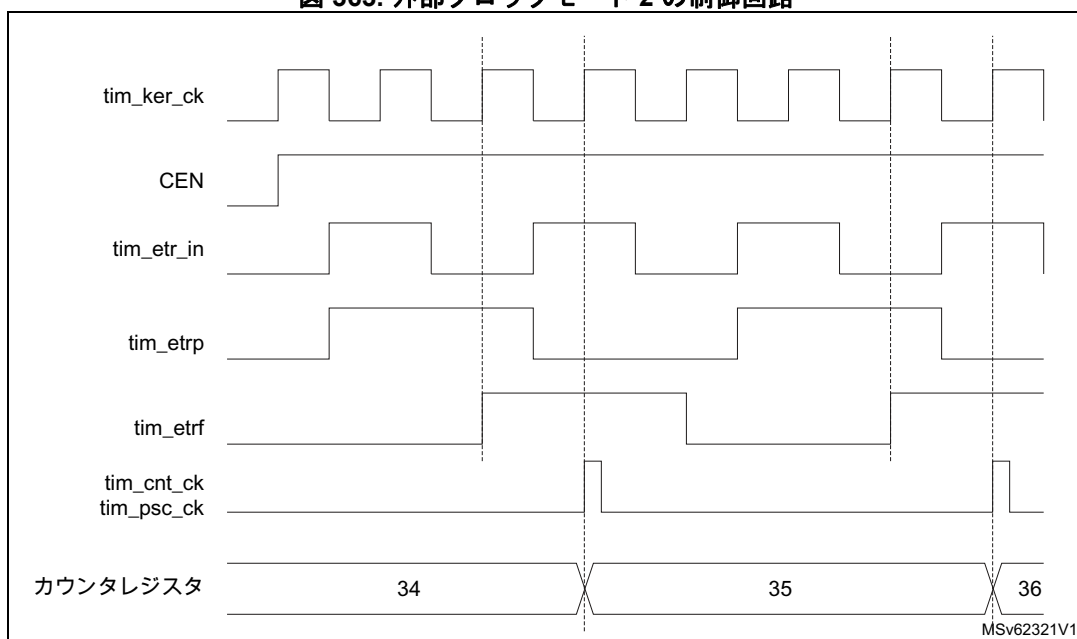
たとえば、tim_etr_in の 2 回の立上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケラを設定します。
3. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、tim_etr_in 入力の立上がりエッジ検出を選択します。
4. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の tim_etr_in 立上がりエッジごとに 1 回カウントします。

tim_etr_in の立上がりエッジから実際のカウンタクロックまでの間に、tim_etrp 信号の再同期回路による遅延があります。結果として、カウンタで正しくキャプチャできる最大周波数は、最大で tim_ker_ck 周波数の 1/4 です。ETRP 信号が高速の場合、ユーザは適切な ETPS プリスケラ設定によって外部信号の分周比を適用する必要があります。

図 363. 外部クロックモード 2 の制御回路



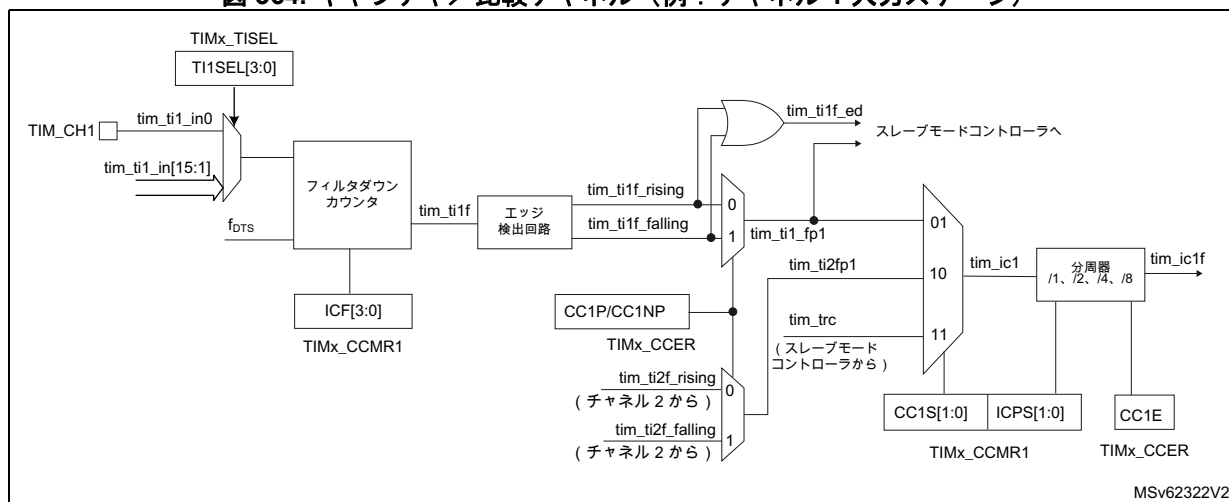
38.3.8 キャプチャ／比較チャンネル

各キャプチャ／比較チャンネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（チャンネル 5 および 6 を除くデジタルフィルタ、マルチプレクス、プリスケアラ）、および出力カステージ（コンパレータと出力制御）から構成されています。

図 364 から 図 367 に、1 つのキャプチャ／比較チャンネルの概要を示します。

入カステージは、対応する `tim_tix` 入カをサンプリングして、フィルタリングを行った信号 `tim_tixf` を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入カとして、またはキャプチャコマンドとして使用される信号（`tim_tixfpy`）を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ（`ICxPS`）に渡されます。

図 364. キャプチャ／比較チャンネル（例：チャンネル 1 入カステージ）



出カステージは、その後、基準として使用される中間波形 `tim_ocxref` (アクティブハイ) を生成します。信号の極性は最終出力に影響を与えます。

図 365. キャプチャ／比較チャンネル 1 メイン回路

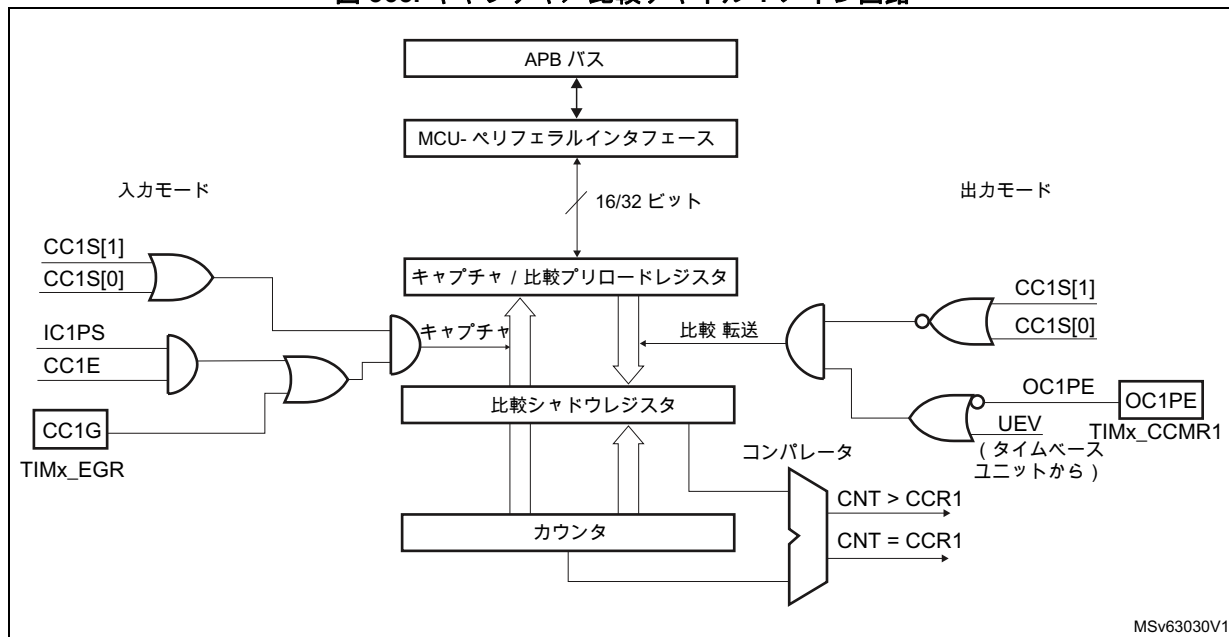
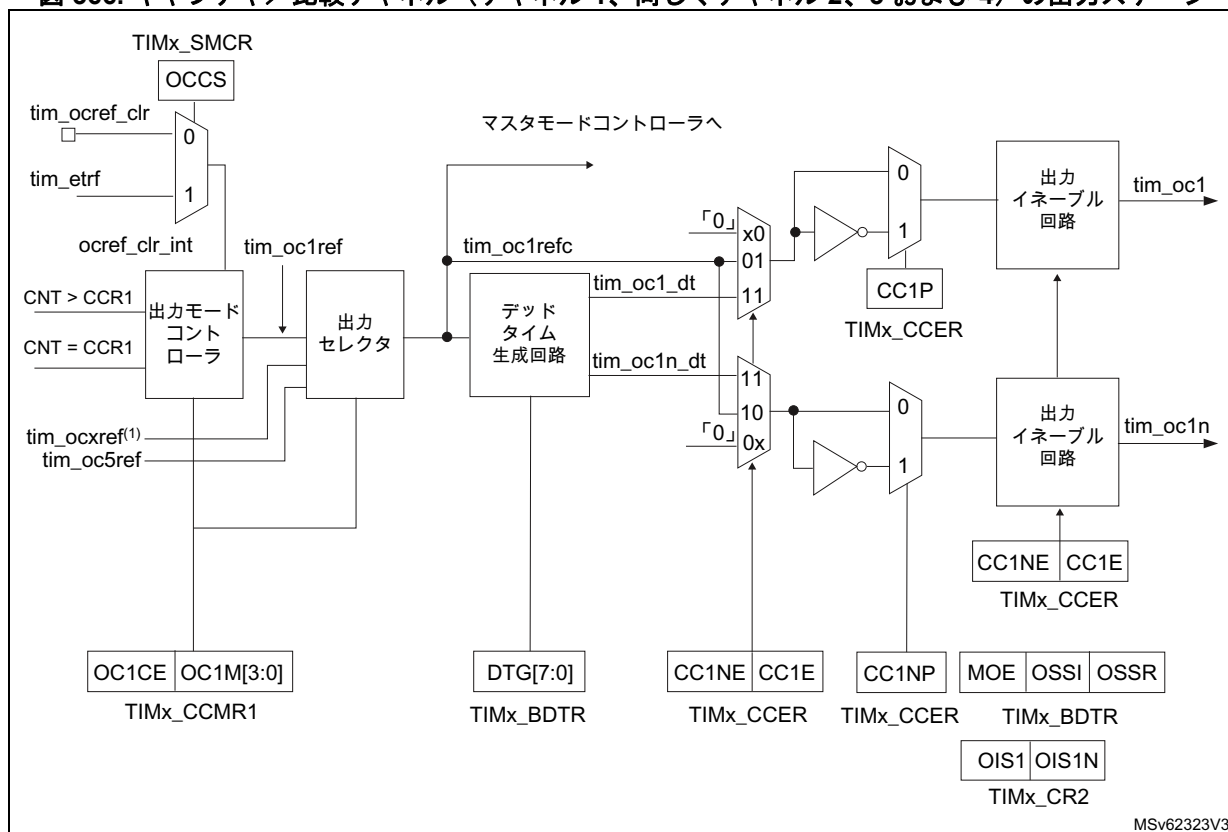
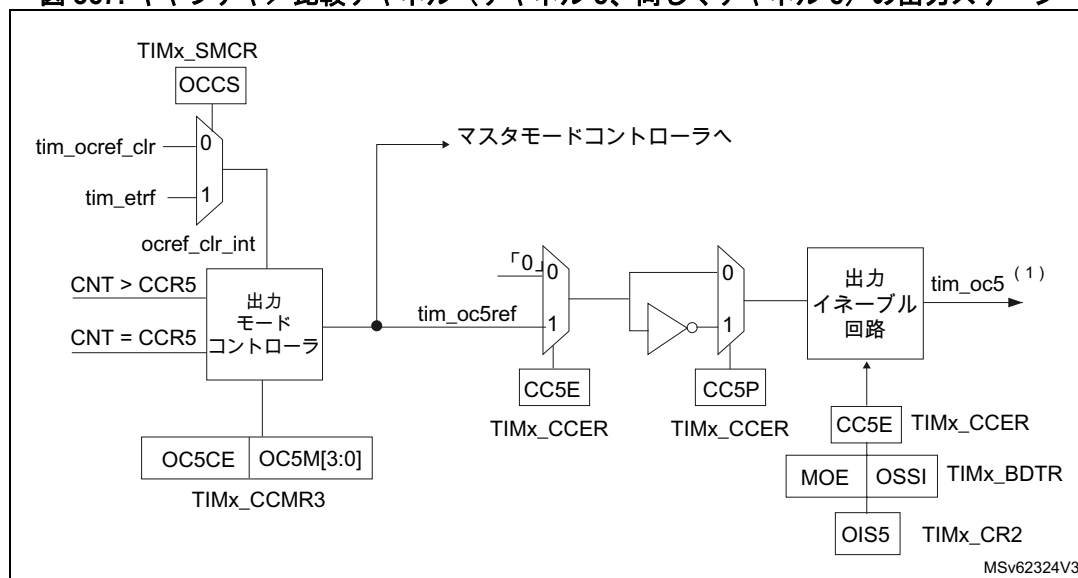


図 366. キャプチャ／比較チャネル（チャネル 1、同じくチャネル 2、3 および 4）の出力ステージ



1. tim ocxref、ここで x は相補チャネルのランク

図 367. キャプチャ／比較チャネル（チャネル 5、同じくチャネル 6）の出力ステージ



1. 外部的には使用できません。

キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書込みおよび読出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

38.3.9 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって遷移が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、0 を書き込むとクリアされます。

次の例は、tim_ti1 入力が増加したときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は tim_ti1 入力とリンクされていなければならないので、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続される信号に関して、適切な入力フィルタ時間をプログラムします (入力が tim_tix の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットによって)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、tim_ti1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- tim_ti1 チャネルのアクティブ変化のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます (この場合、立上がりエッジの選択)。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要の場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE ビットをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これは、フラグ読み出し後、データ読み出し前に発生するかもしれないオーバキャプチャの見落としを避けるためです。

注： IC 割込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

38.3.10 PWM 入力モード

このモードは、単一の `tim_tix` 入力に接続された PWM 信号の周期とデューティサイクルの両方を以下のように測定するために使用されます。

- `TIMx_CCR1` レジスタは周期値（2 つの連続した立上がりエッジ間の間隔）を保持します。
- `TIMx_CCR2` レジスタはパルス幅（2 つの連続した立上がりと立下がりエッジ間の間隔）を保持します。

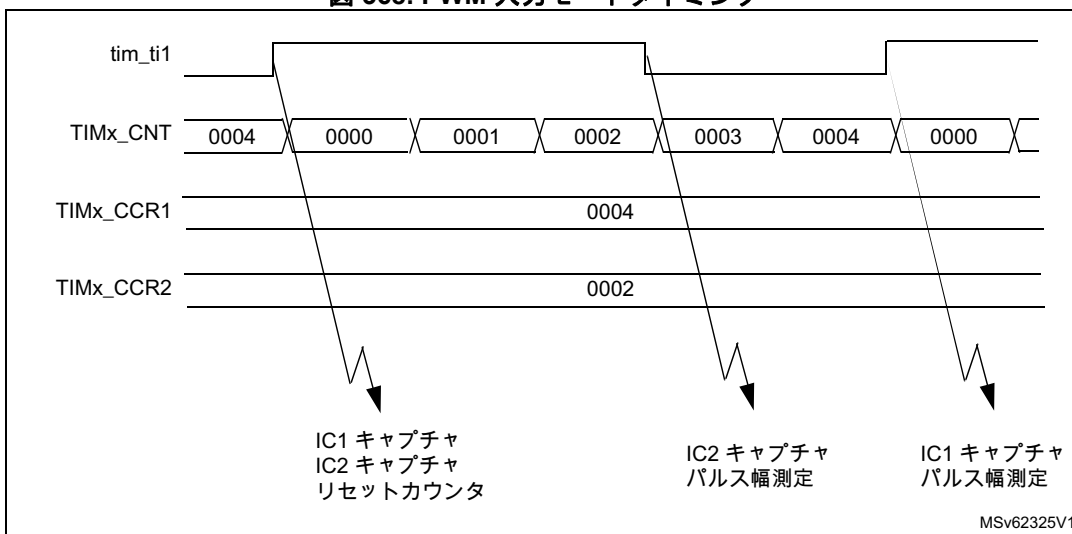
このモードは、入力キャプチャモードの特殊ケースです。設定手順は似ていますが、以下の違いがあります。

- 2 つの `ICx` 信号が同じ `tim_tixfp1` 入力にマッピングされます。
- この 2 つの `ICx` 信号は、逆の極性のエッジでアクティブです。
- 2 つの `tim_tixfp` 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

`tim_ti1` に適用される PWM 信号の周期とパルス幅は、以下の手順を使用して測定できます。

- `TIMx_CCMR1` レジスタの `CC1S` ビットに“01”を書き込むことによって（`tim_ti1` を選択）、`TIMx_CCR1`のアクティブ入力を選択します。
- `CC1P` ビットと `CC1NP` ビットに“0”を書き込むことによって（立上がりエッジでアクティブ）、`tim_ti1fp1` のアクティブな極性を選択します（`TIMx_CCR1` のキャプチャとカウンタクリアの両方に使用）。
- `TIMx_CCMR1` レジスタの `CC2S` ビットに“10”を書き込むことによって（`tim_ti1` を選択）、`TIMx_CCR2`のアクティブ入力を選択します。
- `CC2P` ビットと `CC2NP` ビットに `CC2P/CC2NP=“10”` を書き込むことによって（立下がりエッジでアクティブ）、`tim_ti1fp2` のアクティブ極性を選択します（`TIMx_CCR2` のキャプチャに使用されます）。
- `TIMx_SMCR` レジスタの `TS` ビットに 00101 を書き込むことによって（`tim_ti1fp1` を選択）、有効なトリガ入力を選択します。
- `TIMx_SMCR` レジスタの `SMS` ビットに 0100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- `TIMx_CCER` レジスタの `CC1E` と `CC2E` ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 368. PWM 入力モードタイミング



38.3.11 強制出力モード

このモード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、各出力比較信号 (tim_ocxref、そして tim_ocx/tim_ocxn) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (tim_ocxref/tim_ocx) を強制的にアクティブレベルにするには、対応する TIMx_CCMRx レジスタの OCxM ビットに 0101 を書き込む必要があります。これにより、tim_ocxref は強制的にハイレベルになり (tim_ocxref は常にアクティブハイ)、tim_ocx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (tim_ocx アクティブハイ) => tim_ocx は強制的にハイレベルになります。

tim_ocxref 信号は、TIMx_CCMRx レジスタの OCxM ビットに 0100 を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

38.3.12 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。マイクロコントロール内でチャンネル 5 および 6 のみが使用可能である場合に、チャンネル 1 から 4 を出力できます (たとえば、合成波形生成または ADC トリガのため)。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=0000)、アクティブにセットされるか (OCxM=0001)、非アクティブにセットされるか (OCxM=0010)、または反転されます (OCxM=0011)。
- 割込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割込みマスク (TIMx_DIER レジスタの CCXIE ビット) がセットされている場合は、割込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

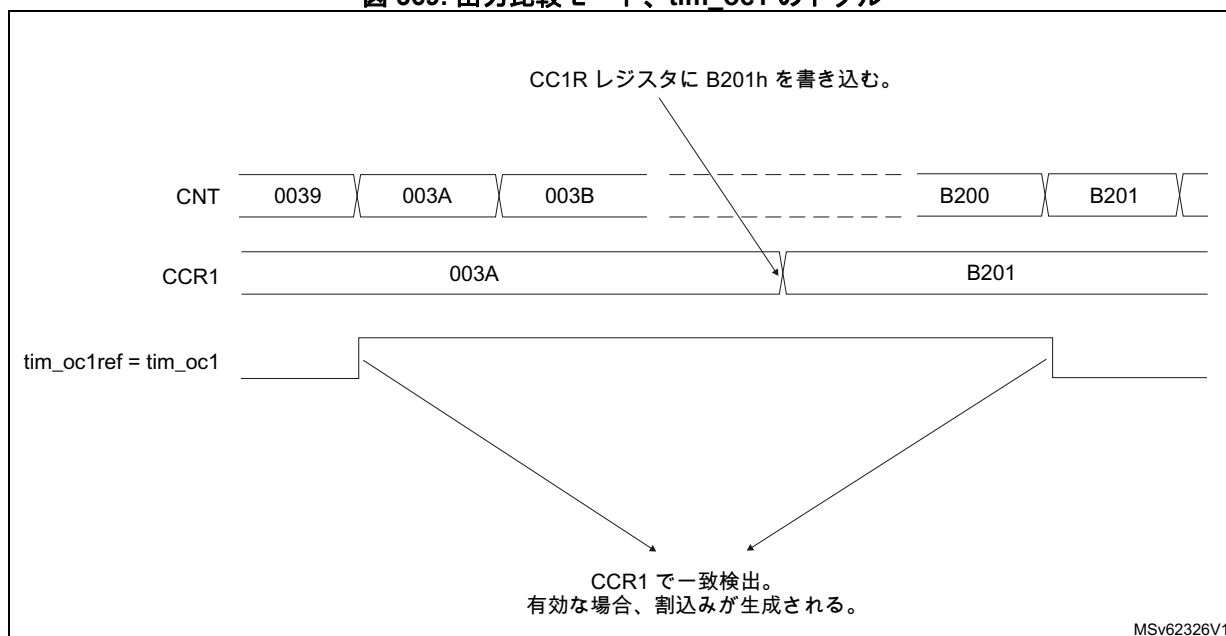
出力比較モードでは、更新イベント UEV は tim_ocxref および tim_ocx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順

1. カウンタクロックを選択します（内部、外部、プリスケアラ）。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
 - CNT と CCRx が一致したときに tim_ocx 出力ピンを反転するには、OCxM ビットに 0011 を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り（OCxPE=0）。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を図 369 に示します。

図 369. 出力比較モード、tim_oc1 のトグル



38.3.13 PWM モード

パルス幅変調（PWM）モードは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルの信号を生成するために使用されます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“0110”（PWM モード 1）または“0111”（PWM モード 2）を書き込むことによって、チャンネルごとに選択できます（tim_ocx 出力ごとに 1 つの PWM）。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも（アップカウントまたはセンターアラインモードで）有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

tim_ocx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。これは、アクティブハイまたはアクティブローとしてプログラムできます。tim_ocx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCER レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx かどうかが判断されます (カウンタの方向によります)。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

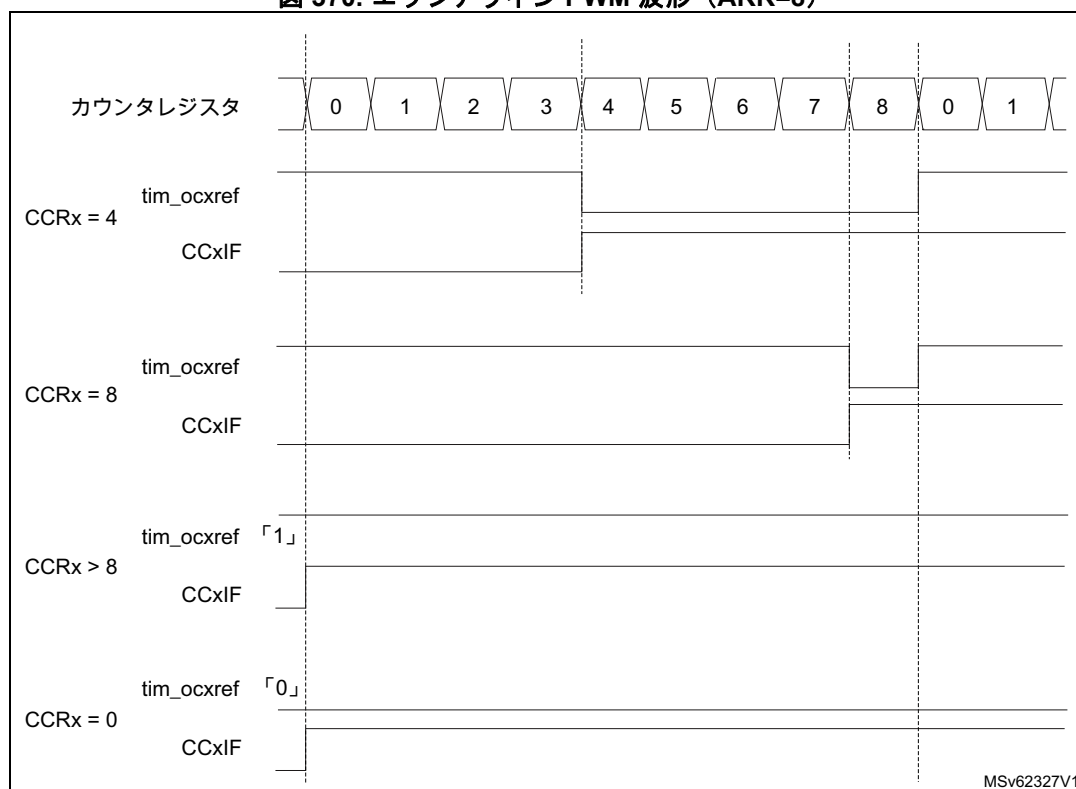
PWM エッジアラインモード

- アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[1416 ページのアップカウントモード](#)を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 tim_ocxref は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、tim_ocxref は“1”に保持されます。比較値が 0 の場合、tim_ocxref は“0”に保持されます。[図 370](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 370. エッジアライン PWM 波形 (ARR=8)



- ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。1420 ページのダウンカウントモードを参照してください。

PWM モード 1 では、基準信号 tim_ocxref は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、tim_ocxref は「1」に保持されます。このモードでは、0 % の PWM 信号を生成することはできません。

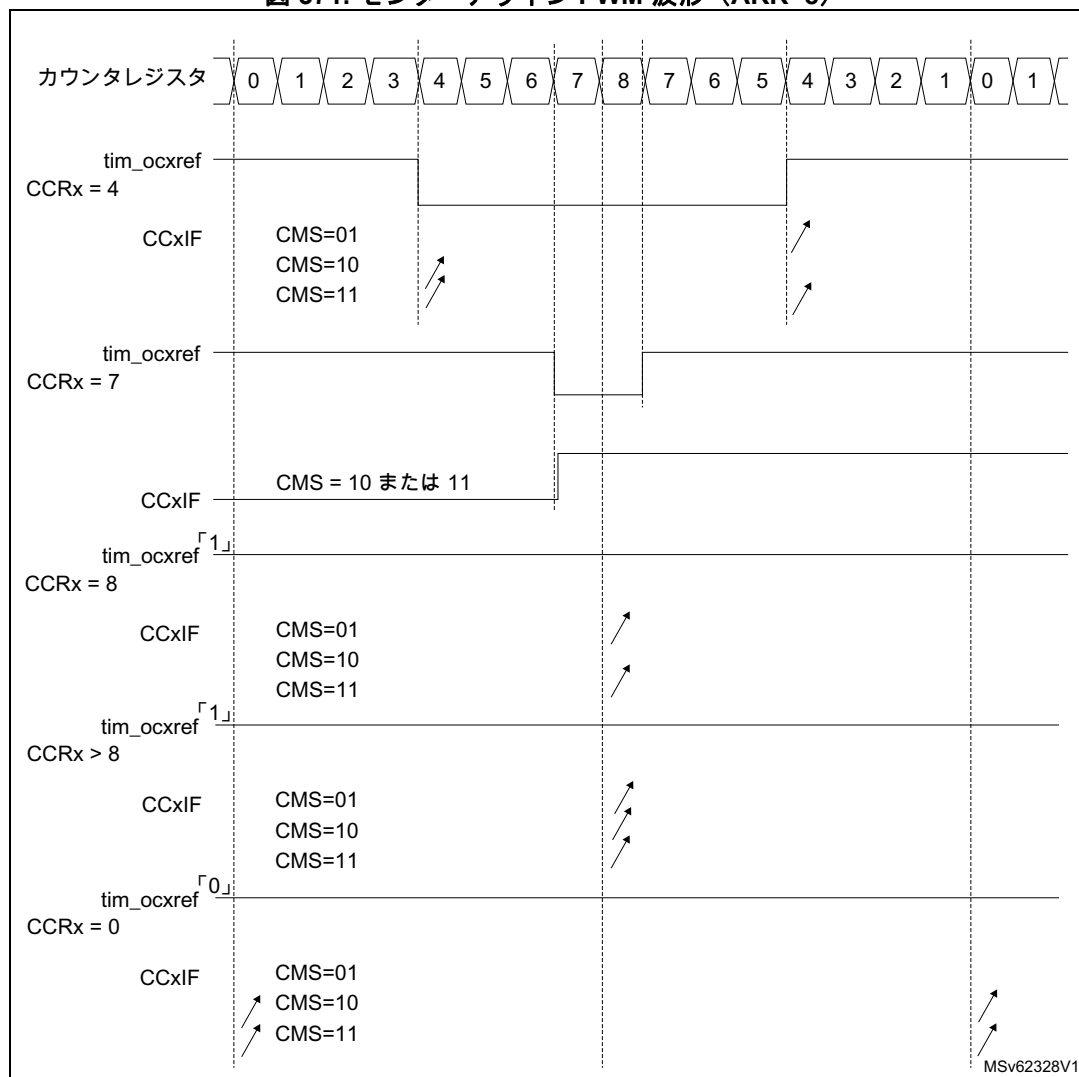
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが「00」でないときにアクティブです（その他すべての設定は、tim_ocxref/tim_ocx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。1424 ページのセンターアラインモード (アップ/ダウンカウント) を参照してください。

図 371 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 371. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

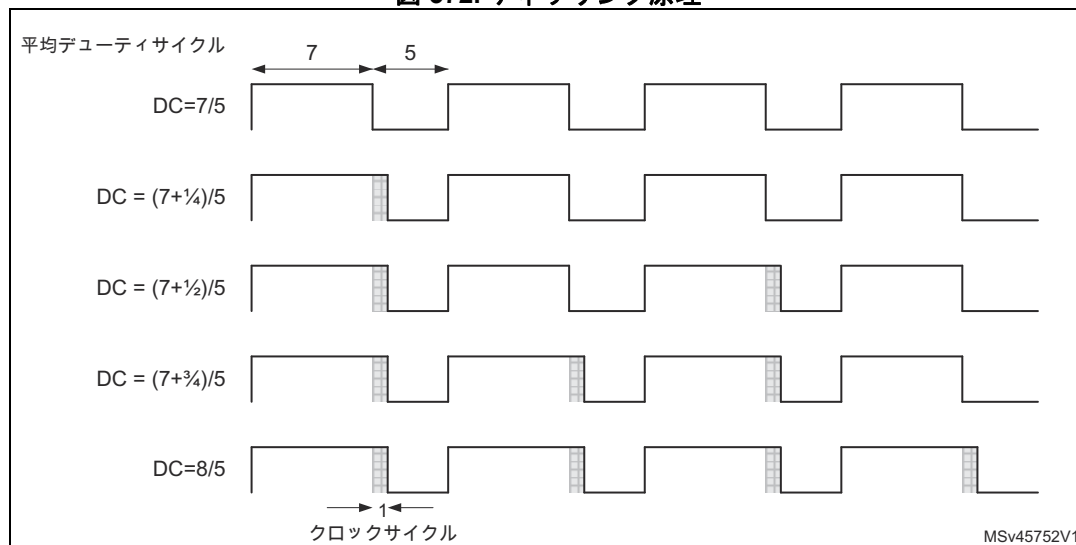
- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - 自動再ロード値より大きい値がカウンタに書き込まれた場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

ディザリングモード

PWM モードの有効分解能は、TIMx_CR1 レジスタの DITHEN ビットを使用して、ディザリングモードを有効にすることで増加できます。これは、CCR（デューティサイクル分解能の増加）と ARR（PWM 周波数分解能の増加）の両方に適用されます。

動作原理は、事前に定義されたパターンで、16 個の連続した PWM 周期にわたって実際の CCR（または ARR）値をわずかに変更する（1 タイマクロック周期を追加する、またはしない）ことです。これにより、平均のデューティサイクルまたは PWM 周期を考えると、16 倍の分解能の向上が可能になります。下記の図 372 は、4 つの連続する PWM サイクルに適用されたディザリング原理を示します。

図 372. ディザリング原理



ディザリングモードが有効になっている場合、レジスタのコーディングは以下のように変更されます（たとえば、図 373 を参照）。

- 下位 4 ビットは、拡張分解能部分（分数部分）用です。
- 上位ビットは、ビット 19:4 に左シフトされ、ベース値用のコーディングになります。

注：

DITHEN ビットをリセットするときは、次の手順に従う必要があります。

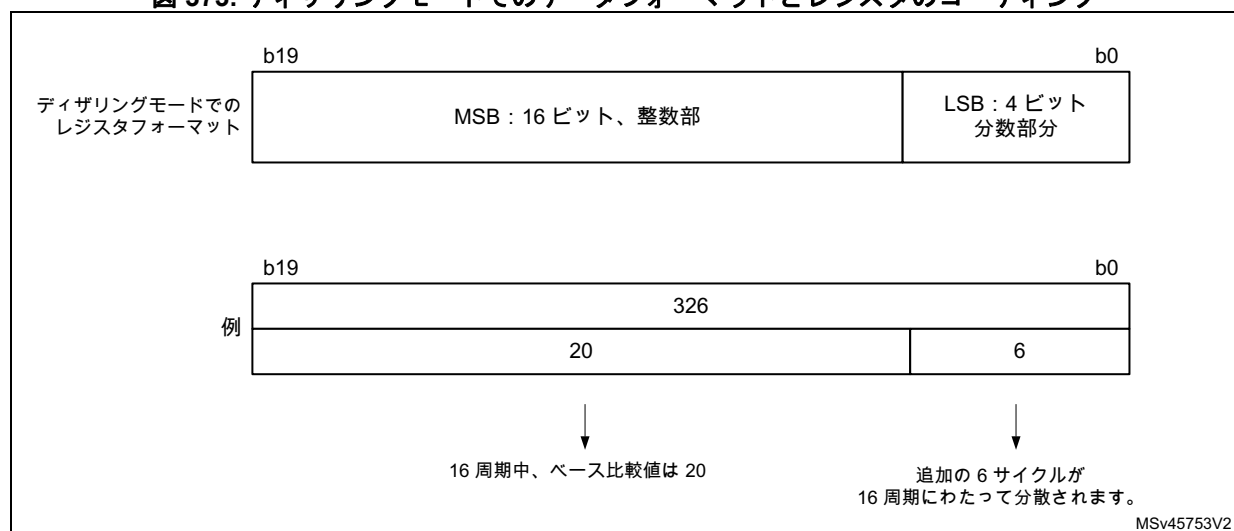
1を参照してください。CEN および ARPE ビットをリセットする必要があります。

2を参照してください。DITHEN ビットをリセットする必要があります。

3を参照してください。CCIF フラグをクリアする必要があります。

4を参照してください。CEN ビットをセットすることができます（最終的に ARPE = 1 とともに）。

図 373. ディザリングモードでのデータフォーマットとレジスタのコーディング



最小周波数は次の計算式で与えられます。

$$\text{分解能} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

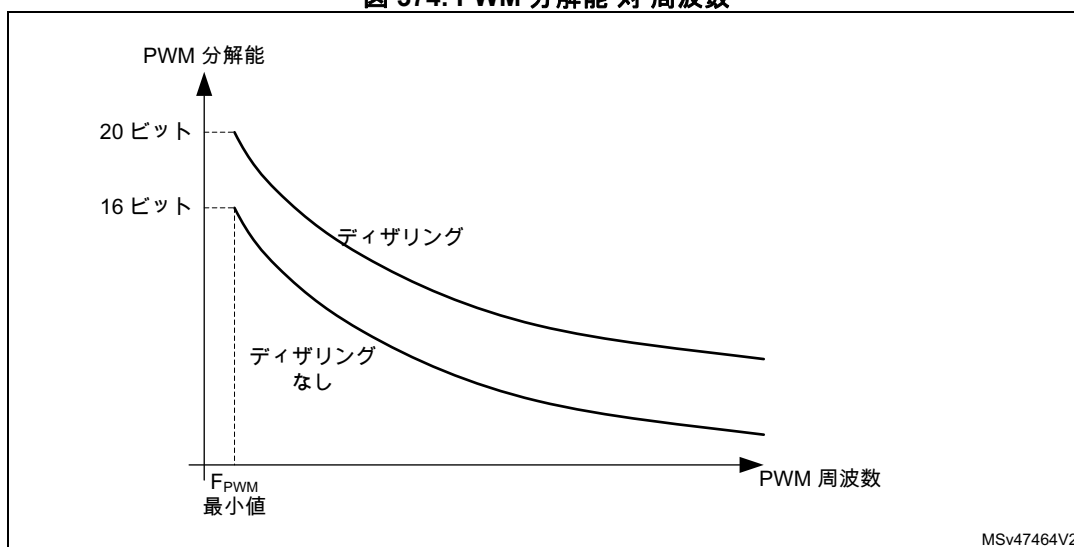
$$\text{ディザリングモード無効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

$$\text{ディザリングモード有効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

注 : TIMx_ARR および TIMx_CCRy の最大値は、ディザリングモードでは 0xFFFFF に制限されています (整数部では 65534、ディザリング部分では 15 に対応します)。

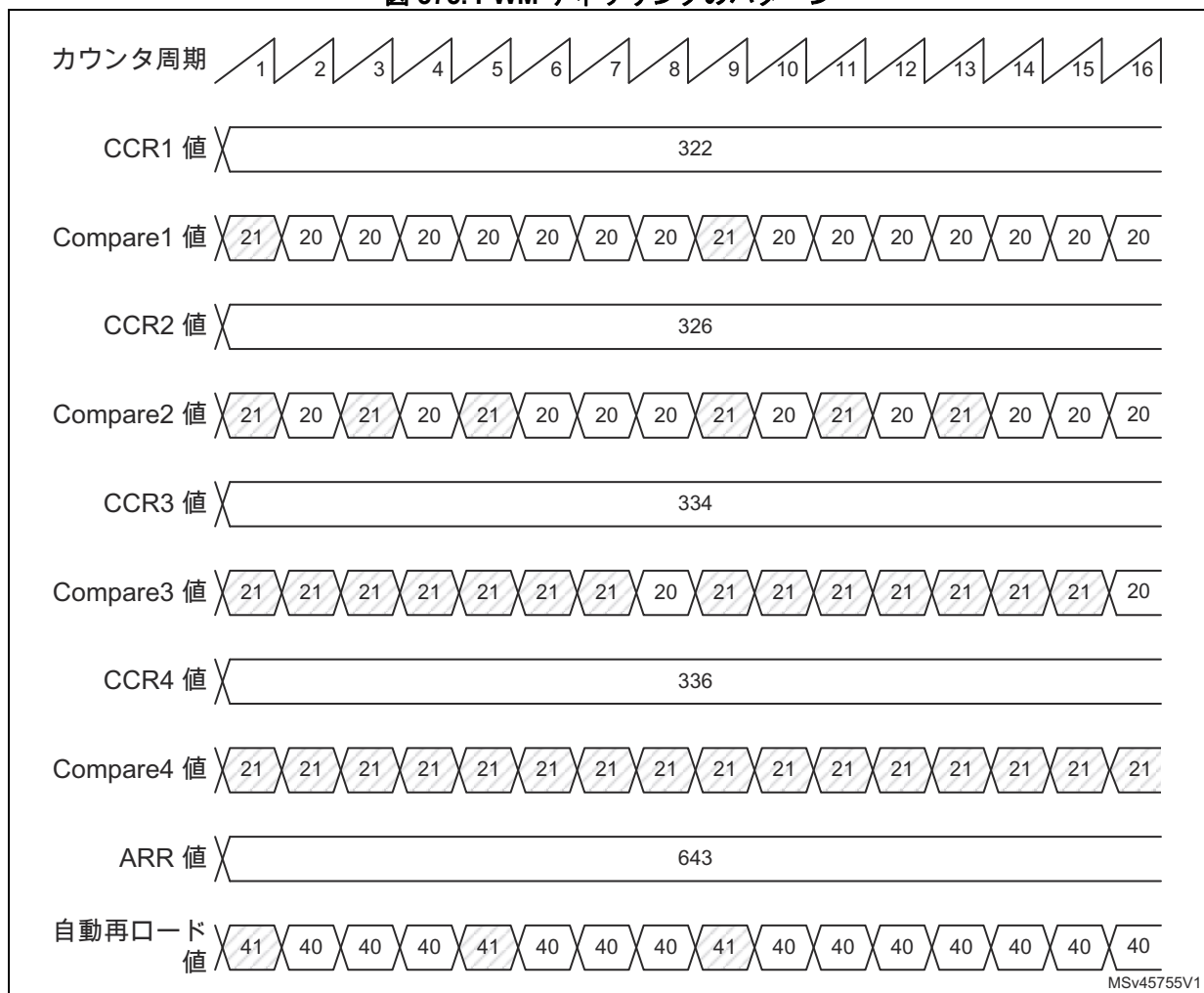
下記の図 374 に示すように、ディザリングモードは PWM 周波数に関係なく、PWM 分解能を増加するために使用されます。

図 374. PWM 分解能 対 周波数



下記の図 375 で説明するように、デューティサイクルや周期の変更は、連続した 16 周期にわたって広がります。

図 375. PWM ディザリングのパターン



自動再ロード値と比較値の増分は、下記の表 386 に記載された特定のパターンに従って分布します。ディザリングシーケンスは、増分が可能な限り均等に分散され、全体のリップルが最小になるように行われます。

表 386. CCR および ARR レジスタの変更のディザリングパターン

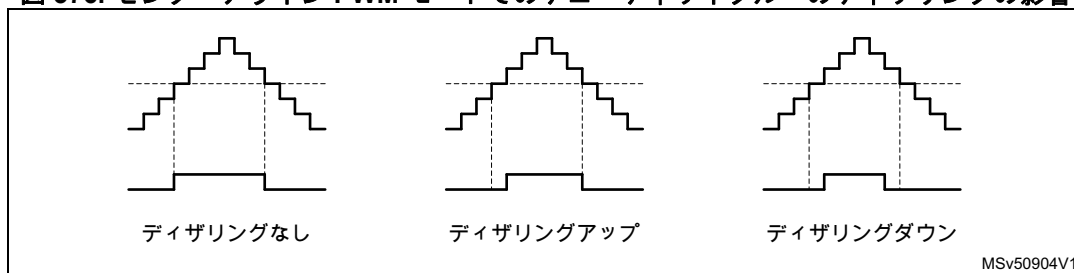
LSB 値	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-

表 386. CCR および ARR レジスタの変更のディザリングパターン (続き)

LSB 値	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

ディザリングモードはセンターアライン PWM モード (TIMx_CR1 レジスタの CMS ビットが“00”に等しくないとき) でも使用できます。この場合、ディザリングパターンは、以下の図 376 に示すように、アップおよびダウンカウンTFーズを考慮して、8 つの連続した PWM 周期にわたって適用されます。

図 376. センターアライン PWM モードでのデューティサイクルへのディザリングの影響



下記の表 387 は、センターアライン PWM モードでどのようにディザリングパターンが追加されるかを示しています。

表 387. センターアライン PWM モードでの CCR レジスタ変更のディザリングパターン

LSB 値	PWM 周期															
	1		2		3		4		5		6		7		8	
	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-

表 387. センターアライン PWM モードでの CCR レジスタ変更のディザリングパターン (続き)

LSB 値	PWM 周期															
	1		2		3		4		5		6		7		8	
	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

38.3.14 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって 2 つのセンターアライン PWM 信号の生成を可能にします。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx_CCRx レジスタペアで決定されます。1 つ目のレジスタがアップカウント時の PWM を制御し、2 つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- tim_oc1refc (または tim_oc2refc) は、TIMx_CCR1 および TIMx_CCR2 によって制御されます。
- tim_oc3refc (または tim_oc4refc) は、TIMx_CCR3 および TIMx_CCR4 によって制御されます。

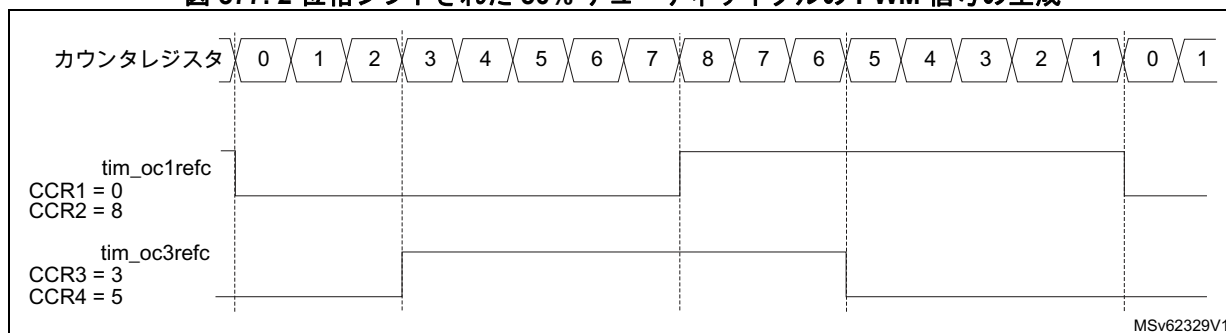
非対称 PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2 チャンネルごとに選択できます (CCR レジスタペアごとに 1 つの tim_ocx 出力)。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

特定のチャンネルが非対称の PWM チャンネルとして使用されると、その相補チャンネルも使用できます。たとえば、tim_oc1refc 信号がチャンネル 1 (非対称 PWM モード 1) に生成されると、チャンネル 2 の tim_oc2refc 信号、または非対称 PWM モード 1 の結果として得られる tim_oc2refc 信号を出力できます。

図 377 は、非対称 PWM モードを使用して生成される信号の例を表します (チャンネル 1 から 4 は非対称 PWM モード 2 として設定されます)。これにより、デッドタイムジェネレータとともにフルブリッジ位相シフト DC-DC コンバータを制御できます。

図 377. 2 位相シフトされた 50% デューティサイクルの PWM 信号の生成



38.3.15 組合せ PWM モード

組み合わせ PWM モードでは、2 つのエッジラインまたはセンターライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2 つの TIMx_CCRx レジスタで決定されず。結果として得られる信号 tim_ocxrefc は、2 つの PWM 基準信号の OR または AND による論理結合から成ります。

- tim_oc1refc (または tim_oc2refc) は、TIMx_CCR1 および TIMx_CCR2 によって制御されます。
- tim_oc3refc (または tim_oc4refc) は、TIMx_CCR3 および TIMx_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの tim_ocx 出力)。

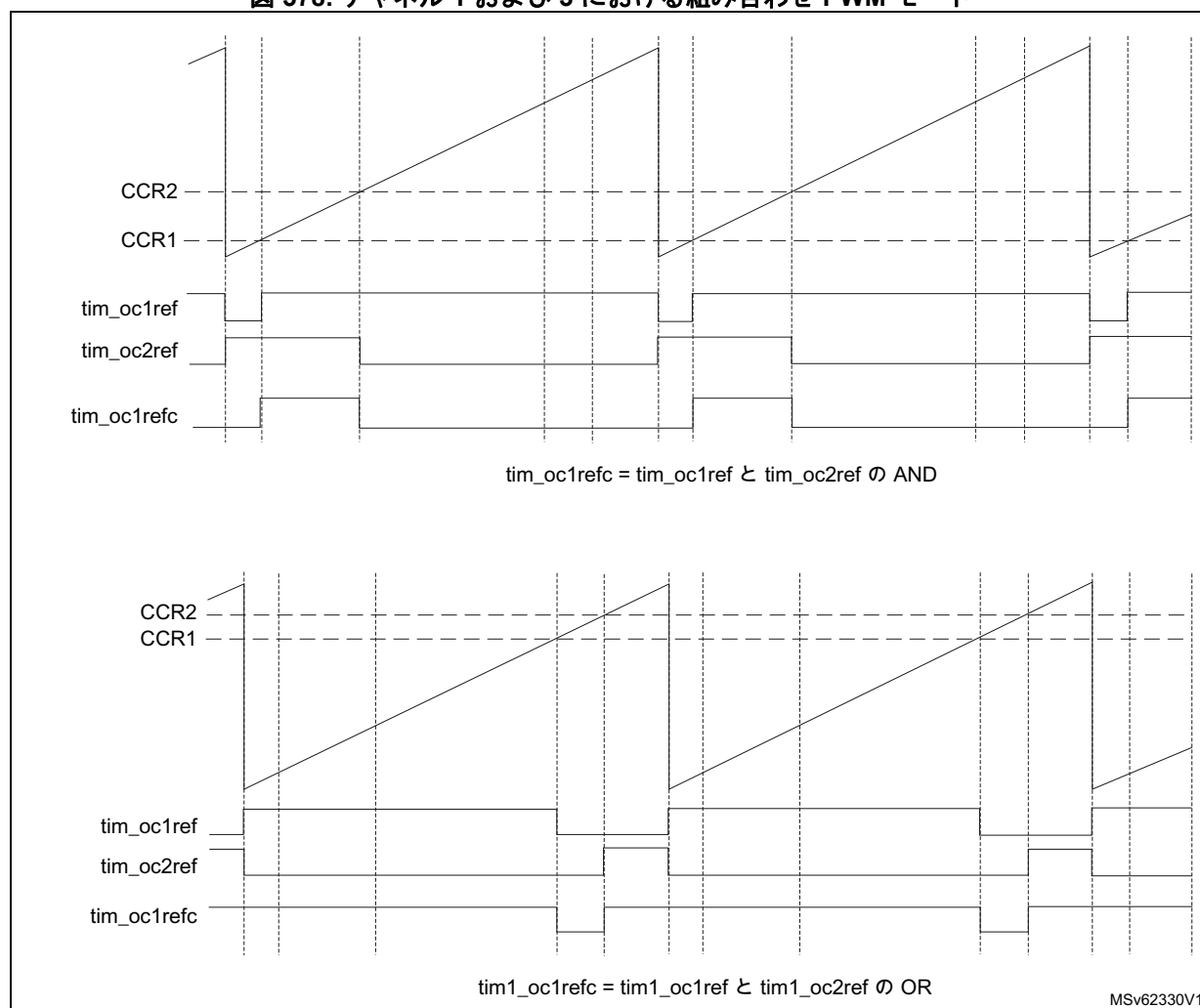
特定のチャネルが組み合わせ PWM チャネルとして使用されている場合、相補チャネルを反対の PWM モードに設定する必要があります (たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします)。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

図 378 は、次の設定で取得可能な組み合わせ PWM モードを使用して生成される信号の例を表します。

- チャネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャネル 2 が PWM モード 1 で設定されている場合
- チャネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャネル 4 が PWM モード 1 で設定されている場合

図 378. チャンネル 1 および 3 における組み合わせ PWM モード



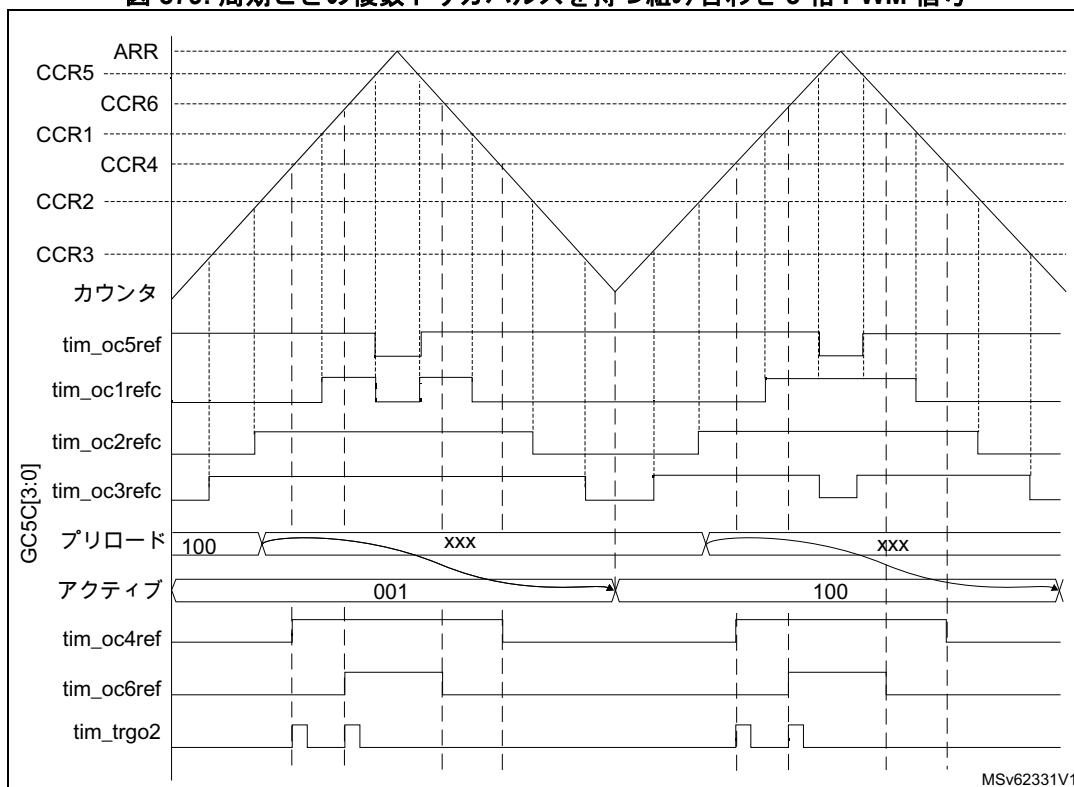
38.3.16 組合せ 3 相 PWM モード

組み合わせ 3 相 PWM モードでは、パルスの中で論理積を取った単一のプログラム可能な信号とともに 1 つから 3 つのセンターアライン PWM 信号を生成できます。結果として得られる組み合わせ信号の定義には、tim_oc5ref 信号が使用されます。TIMx_CCR5 の 3 ビット GC5C[3:1] では、tim_oc5ref を組み合わせる基準信号を選択できます。結果として得られる信号 tim_ocxrefc は、2 つの PWM 基準信号の AND による論理結合から生成されます。

- GC5C1 がセットされると、tim_oc1refc は TIMx_CCR1 および TIMx_CCR5 によって制御されます。
- GC5C2 がセットされると、tim_oc2refc は TIMx_CCR2 および TIMx_CCR5 によって制御されます。
- GC5C3 がセットされると、tim_oc3refc は TIMx_CCR3 および TIMx_CCR5 によって制御されます。

組み合わせ 3 相 PWM モードは、少なくとも 3 ビット GC5C[3:1] の 1 つをセットすることでチャンネル 1 から 3 で個別に選択できます。

図 379. 周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号



tim_trgo2 波形は、特定の 3 相 PWM 信号での ADC の同期方法を示します。詳細については、[セクション 38.3.31 : ADC トリガ](#)を参照してください。

38.3.17 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1/TIM8) は、2 つの相補信号を出力して、出力時のスイッチオンおよびスイッチオフを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性（レベルシフタの内在的な遅延、電源スイッチによる遅延など）に応じて調整する必要があります。

出力（主出力 tim_ocx または相補 tim_ocxn）の極性は出力ごとに独自に選択できます。これは TIMx_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 tim_ocx および tim_ocxn は、TIMx_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx_BDTR レジスタと TIMx_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、[表 395 : 1522 ページのブレイク機能を持つ相補 tim_ocx および tim_ocxn チャンネルの出力制御ビット](#)を参照してください。特に、IDLE 状態に切り替わるとき（MOE が 0 になるとき）に、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレイク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッドタイムジェネレータがあります。この回路は、基準波形 tim_ocxref から tim_ocx と tim_ocxn の 2 つの出力を生成します。tim_ocx と tim_ocxn がアクティブハイの場合、

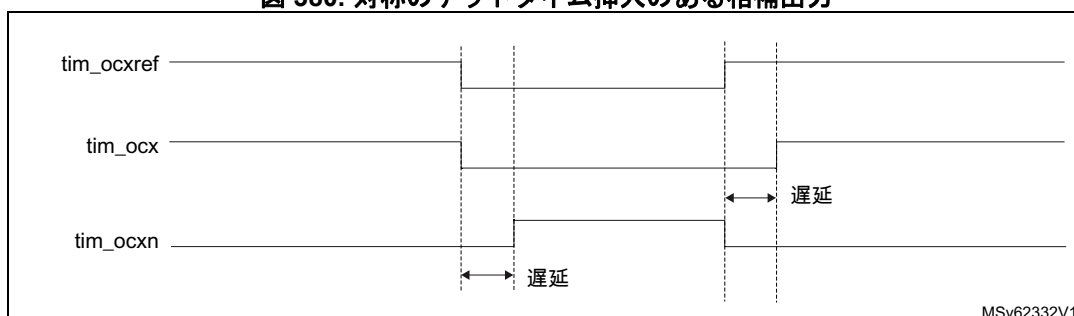
- tim_ocx 出力信号は基準信号と同じですが、立上がりエッジが基準の立上がりエッジより遅い点が異なります。

- tim_ocxn 出力信号は、立上がりエッジが基準波形の立下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (tim_ocx または tim_ocxn) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 tim_ocxref との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 380. 対称のデッドタイム挿入のある相補出力



TIMx_DTR2 の DTAE ビットは、図 381 に示すように、デッドタイム値を、基準信号の立上がりエッジと立下がりエッジに対して変えるために使用されます。

非対称モード (DTAE = 1) では、立上がりエッジ基準デッドタイムは TIMx_BDTR レジスタの DTG[7:0] ビットフィールドによって定義され、立下がりエッジ基準は TIMx_DTR2 レジスタの DTGF[7:0] ビットフィールドによって定義されます。DTAE ビットは、カウンタを有効にする前に書き込む必要があり、CEN=1 の間には変更してはなりません。

プリロードメカニズムを使用して、PWM 運転中にデッドタイム値を動作中に更新することができます。デッドタイムビットフィールド DTG[7:0] と DTGF[7:0] は、TIMx_DTR2 レジスタで DTPE ビットがセットされるとプリロードされます。プリロード値は、次の更新イベントでアクティブレジスタにロードされます。

注： カウンタが有効なときに DTPE ビットを有効にした場合、最後の更新以降に書き込まれた新しい値は破棄され、以前の値が使用されます。

図 381. 非対称のデッドタイム

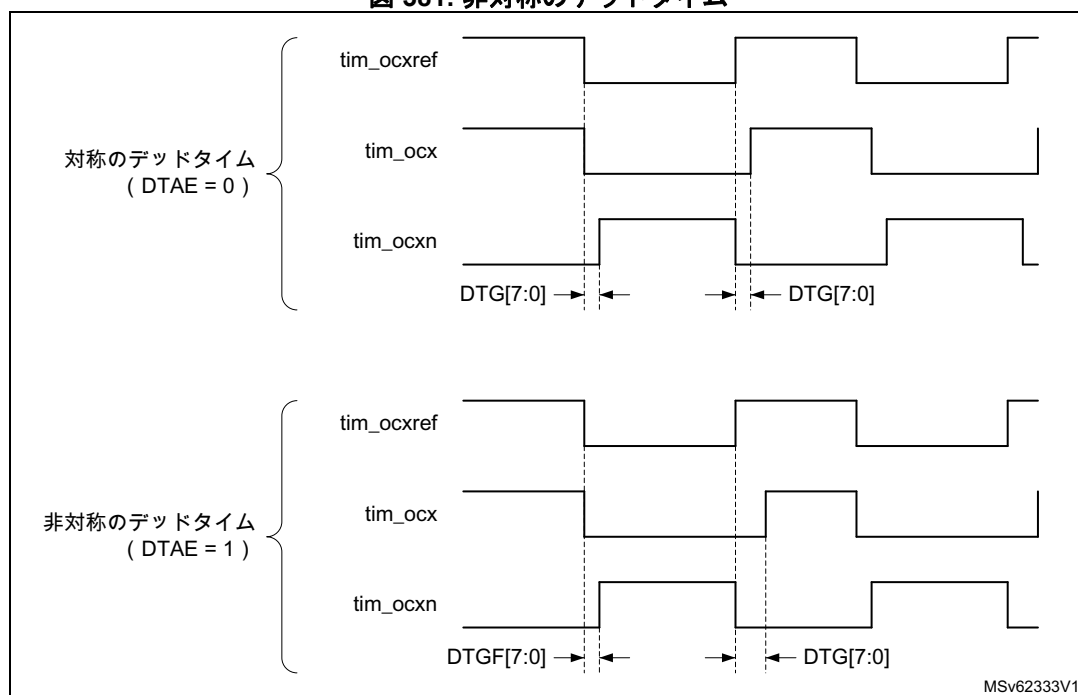


図 382. 負のパルスより長い遅延があるときのデッドタイムの波形

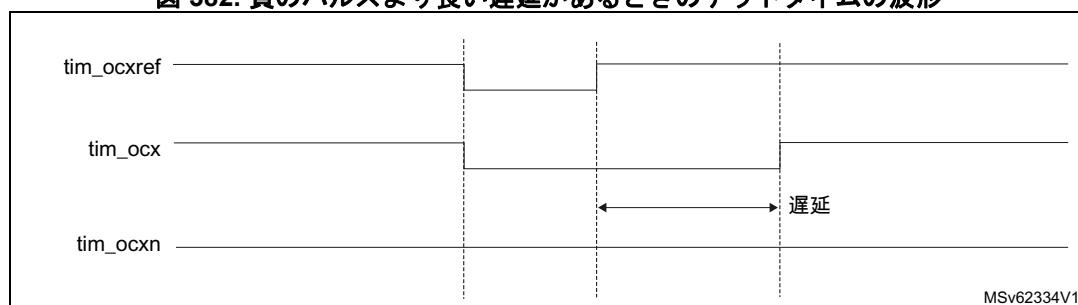
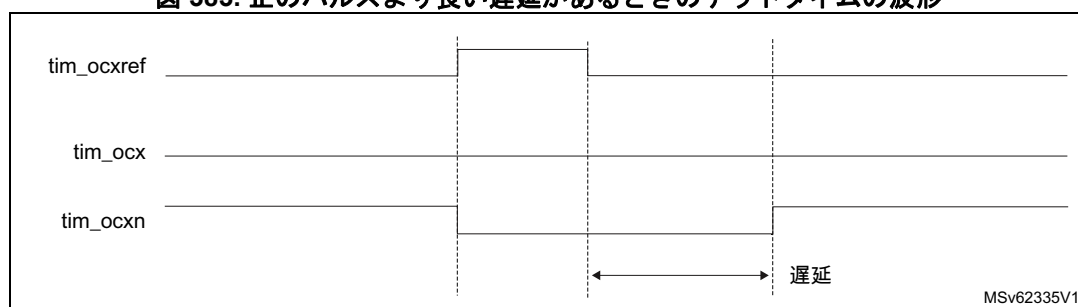


図 383. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx_BDTR レジスタの DTG ビットでプログラム可能です。遅延計算については、[セクション 38.6.20 : TIMx ブレークおよびデッドタイムレジスタ \(TIMx_BDTR\) \(x = 1、8\)](#) を参照してください。

tim_ocxref 信号の tim_ocx または tim_ocxn へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx_CCER レジスタの CCxE ビットおよび CCxNE ビットを設定することによって、tim_ocxref 信号を tim_ocx 出力または tim_ocxn 出力にリダイレクトできます。

これは、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定するために使用されます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができま

注： tim_ocxn のみが有効なときには（CCxE=0、CCxNE=1）、相補にならず、tim_ocxref がハイレベルになるとすぐにアクティブになります。たとえば、CCxNP=0 の場合は、tim_ocxn=tim_ocxref になります。他方、tim_ocx と tim_ocxn の両方が有効なときには（CCxE=CCxNE=1）、tim_ocxref がハイになると tim_ocx はアクティブになり、tim_ocxn は補信号となり、tim_ocxref がローのときにアクティブになります。

38.3.18 ブレーク機能の使用

ブレーク機能の目的は、タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。2 つのブレーク入力通常、パワーステージおよび 3 相インバータの異常出力に接続されています。アクティブ化すると、ブレーク回路は PWM 出力を遮断し、強制的に事前定義された安全な状態に移行させます。出力の遮断をトリガするために、いくつかの内部 MCU イベントを選択することも可能です。

このブレークには、2 つのチャンネルがあります。システムレベル障害（クロック障害、ECC/パリティエラーなど）とアプリケーション障害（入力ピンおよび内蔵コンパレータからの障害）の両方を集め、出力をデッドタイムの持続時間経過後に事前定義されたレベル（アクティブまたはインアクティブ）に強制できるブレークチャンネル。アプリケーション障害のみを含み、出力をインアクティブ状態に強制できるブレーク2 チャンネル。

ブレーク時の出力有効信号および出力レベルは、いくつかの制御ビットに依存しています。

- TIMx_BDTR レジスタの MOE ビット。ソフトウェアによって出力を有効/無効にするために使用され、ブレーク または ブレーク 2 イベント時にリセットされます。
- TIMx_BDTR レジスタの OSSR ビット。出力をインアクティブ状態で制御するか、GPIO コントローラへの制御を解除するかについて、タイマを定義します（通常、ハイインピーダンスモードにするため）。
- TIMx_CR2 レジスタの OISx および OISxN ビット。アクティブまたはインアクティブな出力遮断レベルをセットします。OISx および OISxN の値にかかわらず、一度に tim_ocx 出力と tim_ocxn 出力を両方ともアクティブレベルにセットすることはできません。詳細については、[表 395：1522 ページのブレーク機能を持つ相補 tim_ocx および tim_ocxn チャンネルの出力制御ビット](#)を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。ブレーク機能は、TIMx_BDTR レジスタの BKE ビットおよび BK2E ビットをセットすることによって有効にできます。ブレーク入力の極性は、同じレジスタの BKP および BK2P ビットを設定することによって選択できます。BKEx と BKPx は、同時に変更できます。BKEx および BKPx ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立下がりエッジは非同期のことがあるので、実際の信号（出力に作用する信号）と同期制御ビット（TIMx_BDTR レジスタからアクセスできる）の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 に設定された場合、MOE を正しく読み出すためには、遅延（ダミー命令）を挿入する必要があります。これは、書き込みは非同期信号に対応しますが、読出しは同期信号を反映するからです。

ブレーク (tim_brk) チャンネルのソースは以下のいずれかです。

- (GPIO オルタネート機能選択レジスタでの選択に従って) TIMx_BKIN ピンの 1 つに接続された外部ソースで、極性選択およびオプションのデジタルフィルタリングあり
- 内部ソース :
 - tim_brk_cmpx 入力から着信するもの (製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照)
 - システムブレークリクエストから着信するもの (製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照)

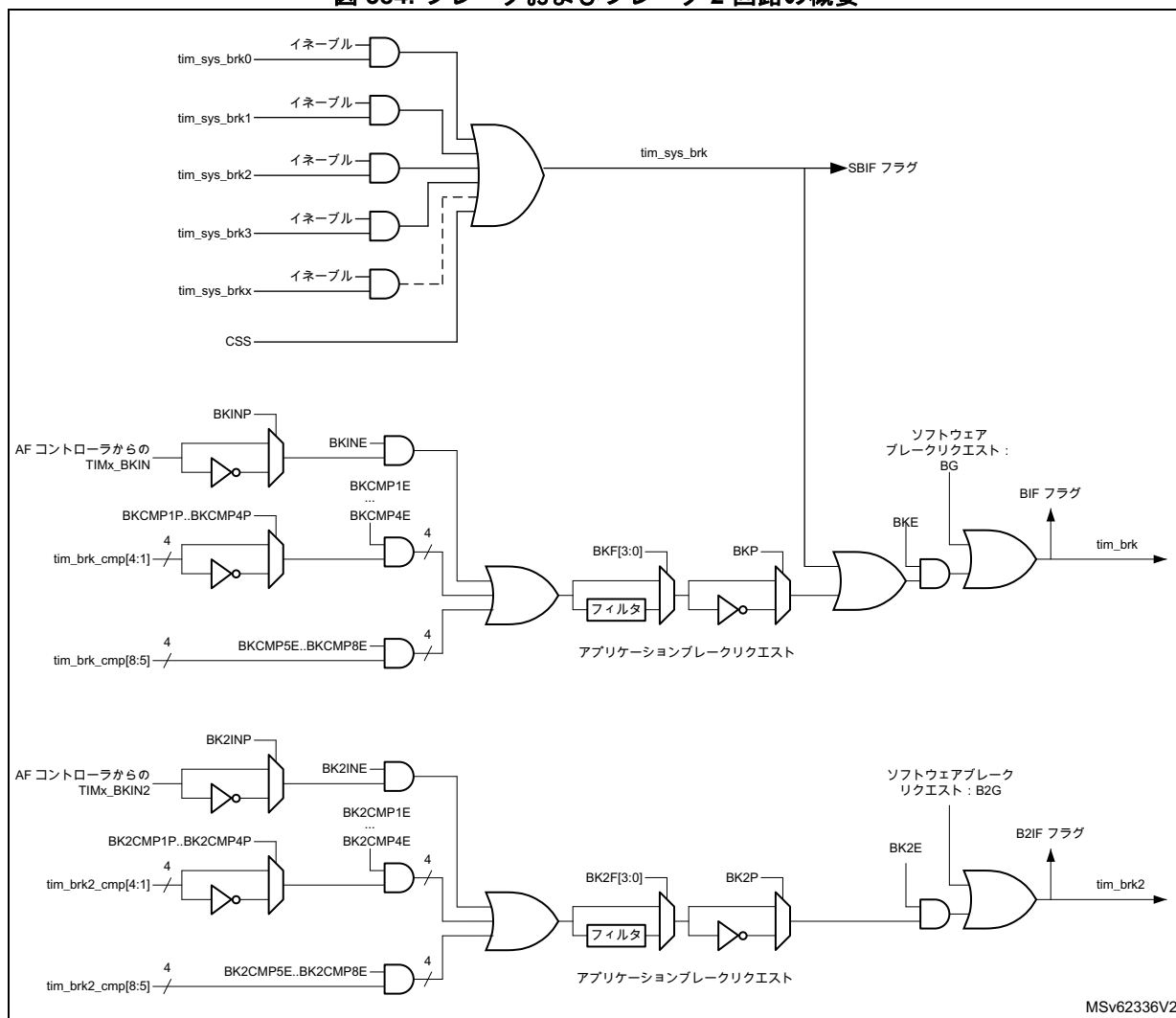
ブレーク 2 (tim_brk2) のソースは以下のいずれかです。

- (GPIO オルタネート機能選択レジスタでの選択に従って) TIMx_BKIN2 ピンの 1 つに接続された外部ソースで、極性選択およびオプションのデジタルフィルタリングあり
- tim_brk2_cmpx 入力から着信する内部ソース (製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照)

ブレークイベントは、TIMx_EGR レジスタで BG および B2G ビットを使用して、ソフトウェアによって生成することもできます。

以下の図 384 のとおり、すべてのソースはタイマ tim_brk または tim_brk2 入力に入る前に論理和がとられます。

図 384. ブレークおよびブレーク 2 回路の概要



注： 非同期（クロックなし）動作は、プログラム可能なフィルタが無効な場合にのみ保証されます。有効になっている場合は、必ずブレークイベントが処理されるように、フェイルセーフクロックモード（たとえば、内部 PLL や CSS を使用）を使用する必要があります。

ブレークが 1 つ発生すると（ブレーク入力の 1 つで選択されたレベル）、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態またはアイドル状態になるか、GPIO コントローラへの制御が解除されます（OSSI ビットで選択）。これは、MCU オシレータがオフの場合に有効です。
- 各出力チャネルは、MOE=0 になったとき、TIMx_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマは出力の制御（GPIO コントローラによって引き継がれた）を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
 - 出力は、まずインアクティブ状態に置かれます（極性に依存します）。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、tim_ocx と tim_ocxn を同時にアクティブレベルに駆動することはできません。

きません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください (約 2 tim_ker_ck クロックサイクル)。

- OSSI=0 の場合、タイマは出力の制御 (ハイインピーダンス状態を強制する GPIO コントローラによって引き継がれた) を解除し、そうでない場合、イネーブル出力はハイのままか、CCxE または CCxNE ビットのどちらかがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx_SR レジスタの SBIF、BIF、および B2IF ビット) がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合、割込みが生成されます。TIMx_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント (UEV) で再び自動的にセットされます。たとえば、これを使用してレギュレーションを行うことができます。そうでない場合、MOE ビットはアプリケーションが再び "1" をセットするまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

注 : AOE ビットが設定されている状態で CPU により MOE がリセットされると、出力はアイドル状態になり、OSSI 値に応じてインアクティブレベルまたはハイインピーダンスに強制されます。MOE ビットと AOE ビットの両方が CPU によってリセットされた場合、出力は無効状態になり、TIMx_CR2 レジスタの OISx ビットでプログラムされたレベルで駆動されます。

注 : ブレーク入力は、信号レベルに対してアクティブです。このため、ブレーク入力がアクティブな間は、MOE をセットできません (自動的に、ソフトウェアによっても)。この間、ステータスフラグ BIF および B2IF をクリアできません。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書込み保護機能を設けてあります。これは、いくつかのパラメータ (デッドタイムの長さ、tim_ocx/tim_ocxn 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性) の設定を固定するために使用されます。TIMx_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 38.6.20 : TIMx ブレークおよびデッドタイムレジスタ \(TIMx_BDTR\) \(x = 1, 8\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。


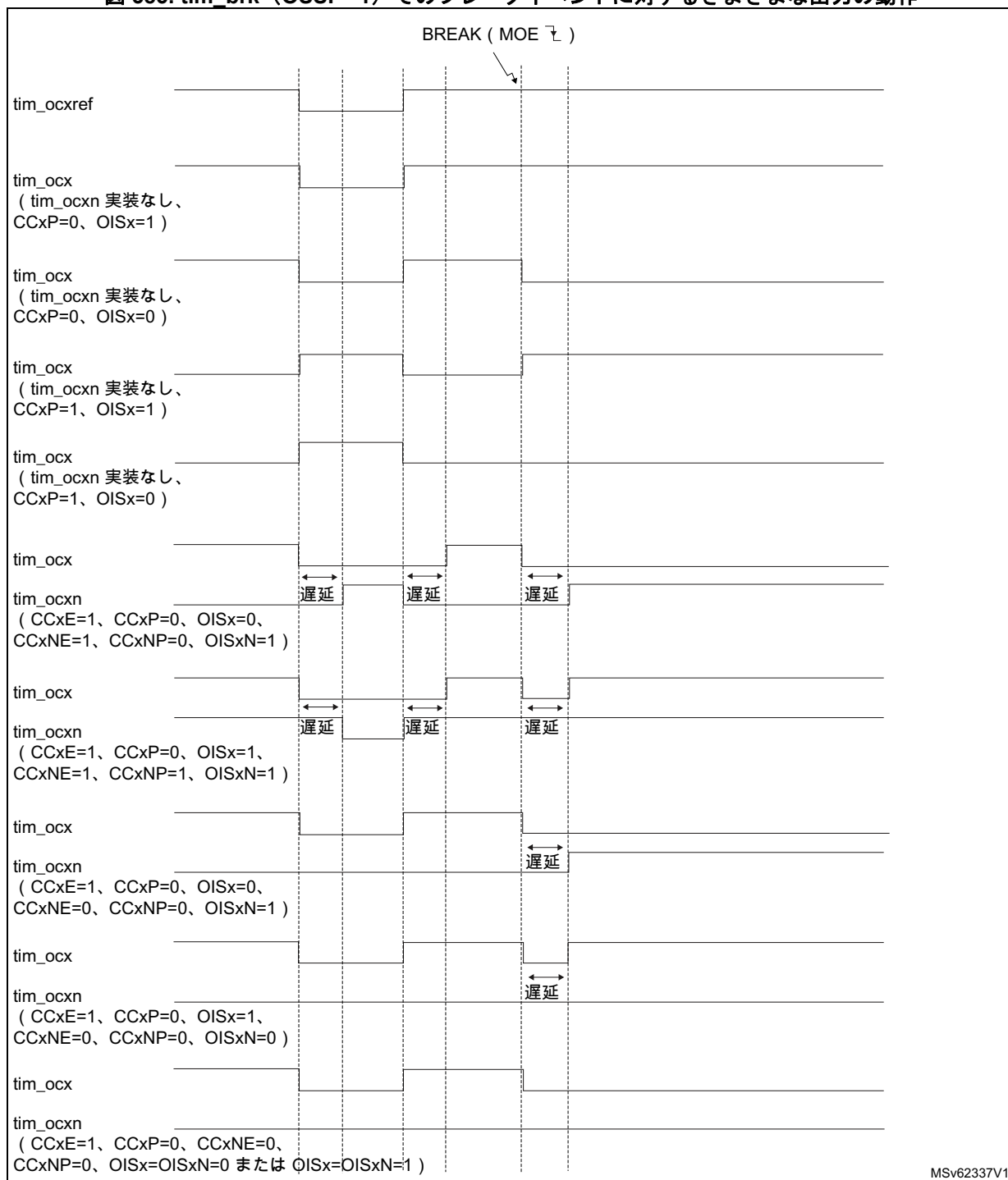
 [図 385](#) に、ブレークに対する出力の動作例を示します。

図 385. tim_brk (OSSI = 1) でのブレーキイベントに対するさまざまな出力の動作



2 つのブレーキ入力、タイマ出力で異なる動作を示します。

- tim_brk 入力は、無効化（インアクティブ状態）するか、PWM 出力を強制的に事前定義した安全な状態に移行できます。
- tim_brk2 は、PWM 出力の無効化（インアクティブ状態）のみ可能です。

表 388 に示すように、tim_brk の優先順位は tim_brk2 入力よりも高いです。

注 : tim_brk2 は OSSR = OSSl = 1 の場合にのみ使用してください。

表 388. タイマ出力と tim_brk/tim_brk2 入力の動作

tim_brk	tim_brk2	タイマ出力状態	通常の使用例	
			tim_ocxn 出力 (ローサイド・スイッチ)	tim_ocx 出力 (ハイサイド・スイッチ)
アクティブ	X	<ul style="list-style-type: none">– 非アクティブから強制される出力状態 (デッドタイム後)– OSSl = 0 の場合、出力は無効 (GPIO ロジックが制御を引き継ぐ)	デッドタイム挿入後 ON	OFF
非アクティブ	アクティブ	非アクティブ	OFF	OFF

図 386 では、tim_brk および tim_brk2 入力で信号がアクティブな場合の、tim_ocx および tim_ocx 出力の動作の例を示します。この場合、両方の出力がアクティブハイ極性になります (TIMx_CCER レジスタの CCxP = CCxNP = 0)。

図 386. tim_brk および tim_brk2 のアサート後の PWM 出力状態 (OSSl=1)

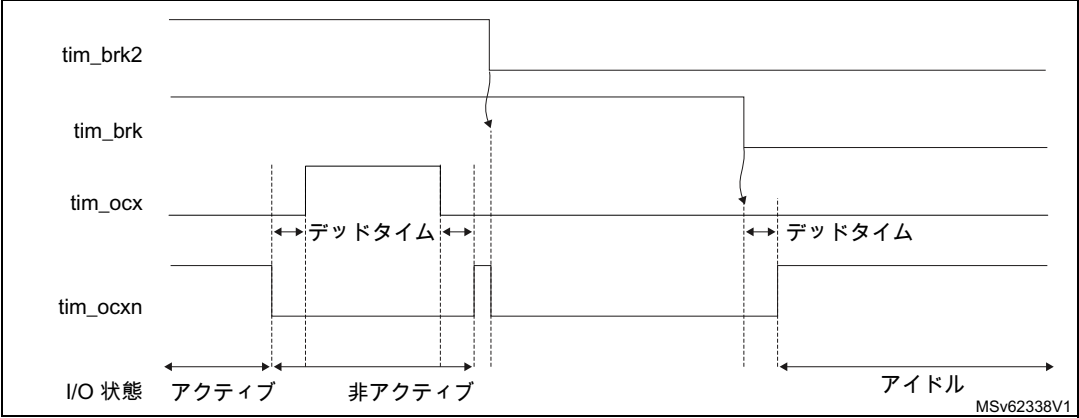
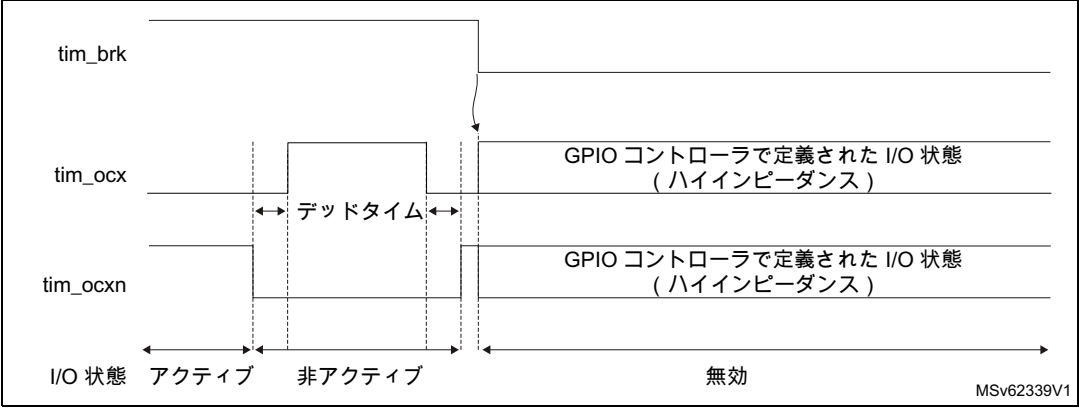


図 387. tim_brk アサート後の PWM 出力状態 (OSSl=0)



38.3.19 双方向ブレーク入力

TIM1/TIM8 には、図 388 に示すとおり、双方向ブレーク I/O が搭載されています。

これによって、以下がサポートされます。

- 入力および出力ステータスピンドちらにもなるユニークなピンを持ち、外部の MCU やゲートドライバに障害を通知するために使用できる幅広いレベルのグローバルブレーク信号
- 複数の内部および外部ブレークソースを統合する必要がある場合、互いに論理和をとり、ユニークなブレークイベントをトリガする内部ブレークソースおよび複数の外部オープンドレインソース

tim_brk および tim_brk2 入力は、TIMxBDTR レジスタの BKBID および BK2BID ビットを使用して双方向モードに設定されます。BKBID プログラミングビットは、TIMxBDTR レジスタの LOCK ビットを使用して読み出し専用モードにロックできます (LOCK レベル 1 以上)。

双方向モードは、tim_brk と tim_brk2 入力の両方で使用でき、I/O をアクティブロー極性でオープンドレインモードに設定する必要があります (BKINP、BKP、BK2INP、BK2P ビットを使用)。システム (たとえば CSS)、内部ペリフェラル、またはブレーク入力からのブレークリクエストによって、障害イベントを通知するブレーク入力のレベルが強制的にローにされます。安全性のため、極性ビットが正しくセット (アクティブハイ極性) されていない場合の双方向モードは禁止されています。

ブレークソフトウェアイベント (BG および B2G) も、ブレーク I/O を強制的に 0 にして、タイマがブレーク状態に移行されたことを外部コンポーネントに示します。ただし、これはブレークが有効になっている場合 (BKE または B2KE = 1) にのみ有効です。ソフトウェアブレークイベントが BKE または B2KE = 0 で生成された場合、出力が安全な状態になり、ブレークフラグはセットされますが、TIMx_BKIN および TIMx_BKIN2 I/O には効果がありません。

安全解除メカニズムは、システムが完全にロックされることを防ぎます (ブレーク入力でレベルがローになると、同じ入力でレベルをローに強制するブレークがトリガされます)。

BKDSRM (BK2DSRM) ビットが 1 にセットされると、この機能によりブレーク出力が解放され、障害信号をクリアしてシステムを再設定できます。

ブレーク保護回路はどの時点においても無効にできません。

- ブレーク入力パスは常にアクティブです。BKDSRM (BK2DSRM) ビットがセットされており、オープンドレイン制御が解放されている場合でもブレークイベントはアクティブです。これによって、ブレーク条件が存在する限り、PWM 出力の再開を防ぐことができます。
- 出力が有効になっている (MOE ビットがセットされている) 限り、BKDSRM (BK2DSRM) ビットでブレーク保護を解除できません (表 389 を参照)。

表 389. ブレーク保護解除条件

MOE	BKBID (BK2BID)	BKDSRM (BK2DSRM)	ブレーク保護状態
0	0	X	設定
0	1	0	設定
0	1	1	解除
1	X	X	設定

ブレーク回路の設定および再設定

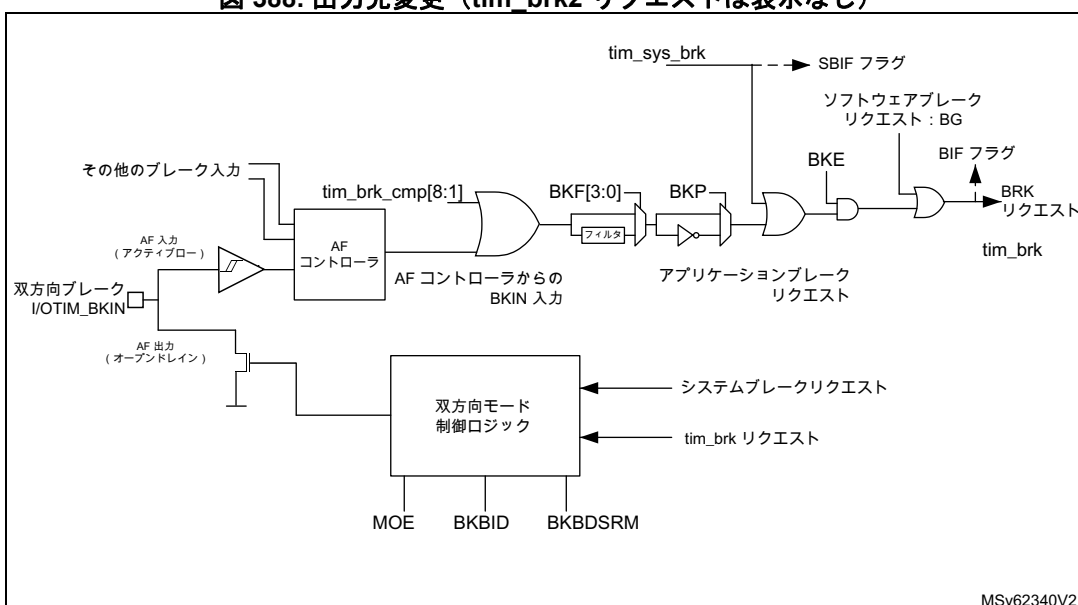
ブレーク回路（入力モードまたは双方向モード）はデフォルト（ペリフェラルリセット設定）で設定されています。

ブレーク（ブレーク 2）イベント後に保護を再び設定するには次の手順に従う必要があります。

- BKDSRM (BK2DSRM) ビットをセットして、出力制御を解放する必要があります。
- ソフトウェアは、システムブレーク条件（ある場合）がなくなるまで待ってから、SBIF ステータスフラグをクリアする必要があります（または再設定の前にシステムでクリアします）。
- ソフトウェアは、BKDSRM (BK2DSRM) ビットをハードウェアによってクリアされるまでポーリングする必要があります（アプリケーションブレーク条件がなくなるとき）。

この時点から、ブレーク回路は設定されアクティブになり、MOE ビットをセットして、PWM 出力を再び有効にできます。

図 388. 出力先変更 (tim_brk2 リクエストは表示なし)

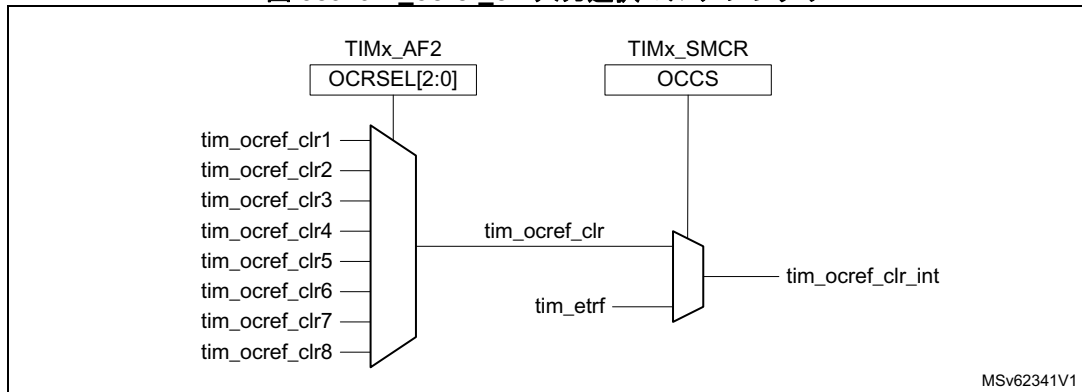


38.3.20 外部イベントによる tim_ocxref 信号のクリア

特定のチャンネルの tim_ocxref 信号は tim_ocref_clr_int 入力にハイレベルを適用するとクリアされます（対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを 1 にセットする）。tim_ocxref は、次の PWM サイクルでアクティブ状態に移移するまでローのままです。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。tim_ocref_clr_int 入力は、TIMx_SMCR レジスタで OCCS ビットを設定することで、tim_ocref_clr 入力と tim_etr (フィルタ後の tim_etr_in) の間で選択できます。

tim_ocref_clr 入力は、以下の図 389 に示すように、TIMx_AF2 レジスタの OCRSEL[2:0] ビットフィールドを使用して複数の入力から選択できます。製品で利用できるソースの一覧については、[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#) を参照してください。

図 389. tim_ocref_clr 入力選択マルチプレクサ

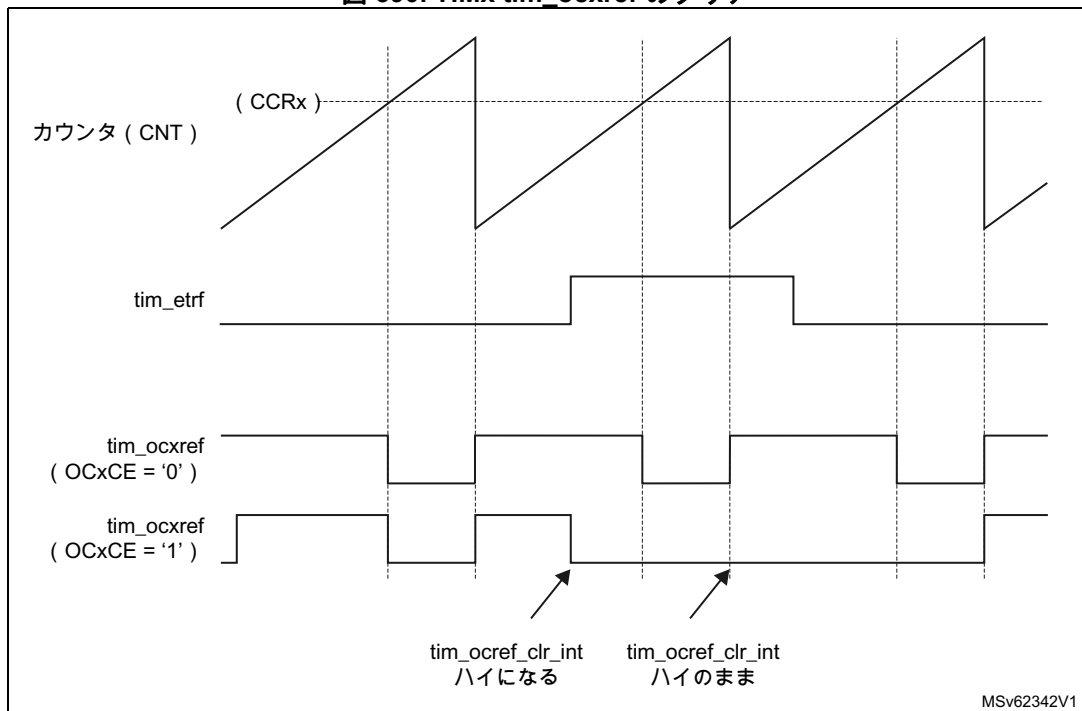


tim_etrf が選択された場合、tim_etrf_in は次のように設定する必要があります。

1. 外部トリガプリスケラはオフ状態に維持する必要があります (TIMx_SMCR レジスタの ETPS[1:0] ビットを"00"にセット)。
2. 外部クロックモード 2 を無効にします (TIMx_SMCR レジスタの ECE ビットを"0"にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて (トリガに接続されたソースの極性、およびフィルタを使ってノイズを除去する最終的なニーズに従って) 設定できます。

図 390 に、イネーブルビット OCxCE の両方の値について、tim_etrf 入力がハイレベルになったときの tim_ocxref 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 390. TIMx tim_ocxref のクリア



注： 100% デューティサイクルの PWM の場合 (CCR_x > ARR の場合)、次のカウンタオーバーフローで tim_ocxref が再度有効になります。

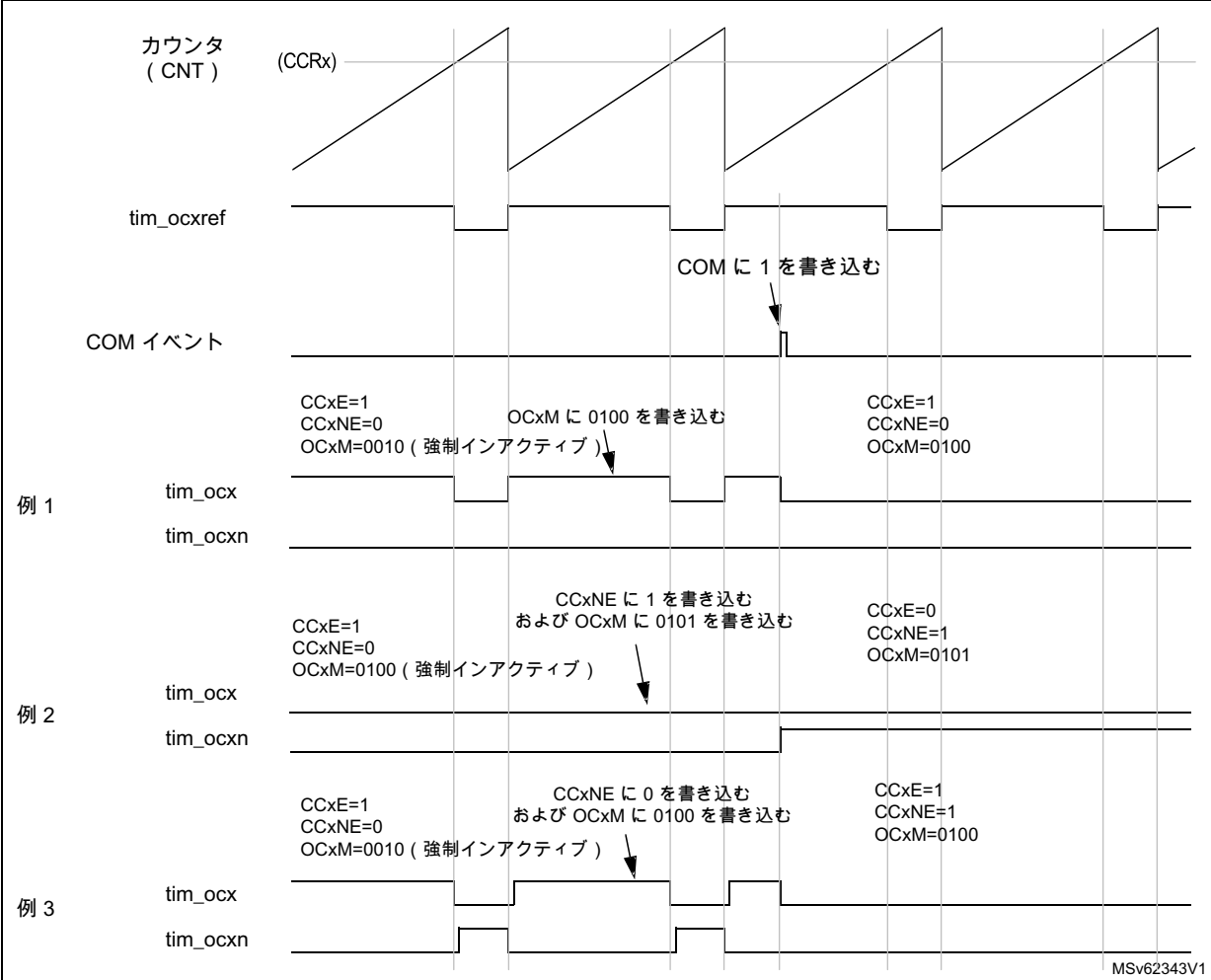
38.3.21 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転流イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (tim_trgi 立上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx_SR レジスタの COMIF ビット)、これによって割り込み (TIMx_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 391 に、COM イベントが発生したときの tim_ocx と tim_ocxn 出力の動作を、3 種類のプログラミング構成の例で示します。

図 391. 6 ステップ生成 COM の例 (OSSR=1)



38.3.22 ワンパルスモード

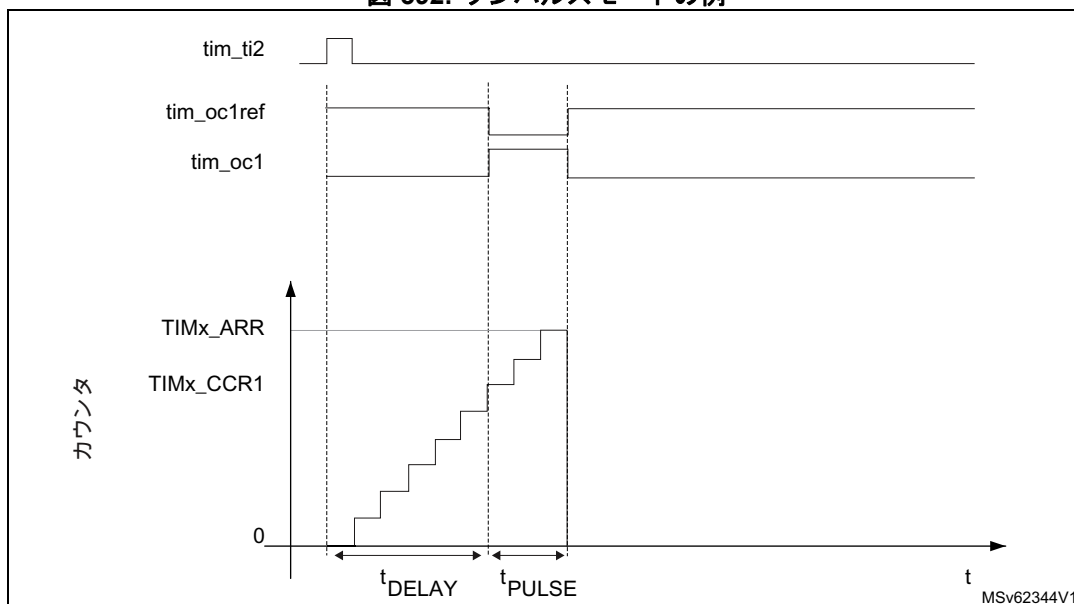
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 392. ワンパルスモードの例



たとえば、tim_ti2 入力ピンで立上がりエッジが検出されたときに、tim_oc1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

tim_ti2fp2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタの CC2S ビットに "01" を書き込むことによって、tim_ti2fp2 を tim_ti2 に配置します。
- tim_ti2fp2 は、立上がりエッジを検出して、TIMx_CCER レジスタで CC2P="0" と CC2NP="0" を書き込みます。
- tim_ti2fp2 をスレーブモードコントローラのトリガ (tim_trgi) として構成するために、TIMx_SMCR レジスタの TS="00110" を書き込みます。
- tim_ti2fp2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます（クロック周波数とカウンタプリスケアラを考慮に入れて）。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタに OC1M=111 を書き込むことによって、PWM モード 2 を有効にする必要があります。オプションで、TIMx_CCMR1 レジスタに OC1PE='1' を書き込み、TIMx_CR1 レジスタの ARPE ビットを書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込む必要があります。次に、UG ビットをセットすることによって更新を生成し、tim_ti2 で外部トリガイベントを待つ必要があります。この例では、CC1P に"0"を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので（シングルモード）、TIMx_CR1 レジスタの OPM ビットに"1"を書き込みます。こうすると、カウンタは次の更新イベント時に停止します（カウンタが自動再ロード値に達して、"0"に戻る時点）。TIMx_CR1 レジスタの OPM ビットが"0"にセットされると、繰り返しモードが選択されます。

特殊ケース : tim_ocx 高速イネーブル :

ワンパルスモードでは、tim_tix 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、tim_ocxref（および tim_ocx）は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

38.3.23 再トリガ可能なワンパルスモード

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 38.3.22](#) で説明した再トリガ不可能なワンパルスモードとは、次のような違いがあります。

- パルスはトリガが発生し次第開始します（プログラム可能な遅延はありません）。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

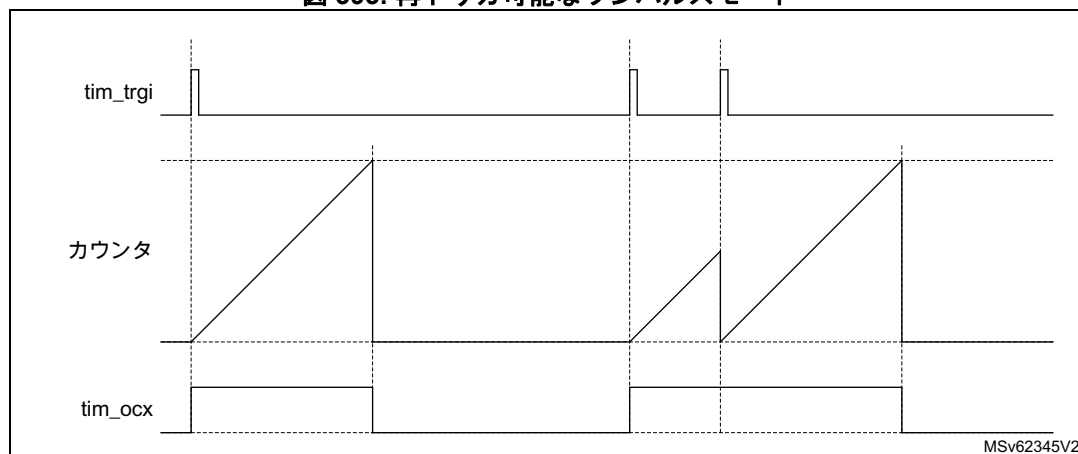
タイマはスレーブモードである必要があり、このときビットは TIMx_SMCR レジスタで SMS[3:0] = "1000"（リセットモードとトリガモードの組み合わせ）、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が "1000"または"1001"にセットされています。

タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります（ARR レジスタによってパルス長がセットされます）。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

注 : OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

このモードをセンターアライン PWM モードと組み合わせて使用することはできません。TIMx_CR1 では、CMS[1:0] = 00 にする必要があります。

図 393. 再トリガ可能なワンパルスモード

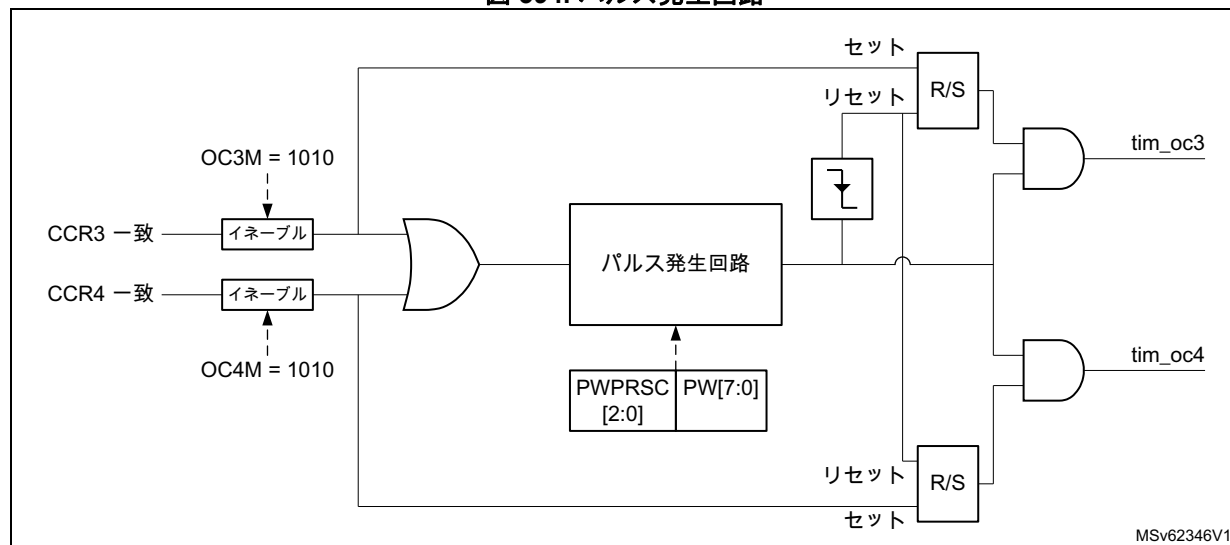


38.3.24 比較モードでのパルス

比較一致イベント時にパルスを生成できます。デバッグまたは同期のために、カウンタ値が特定の比較値と等しいときに、プログラム可能なパルス幅を持つ信号が生成されます。

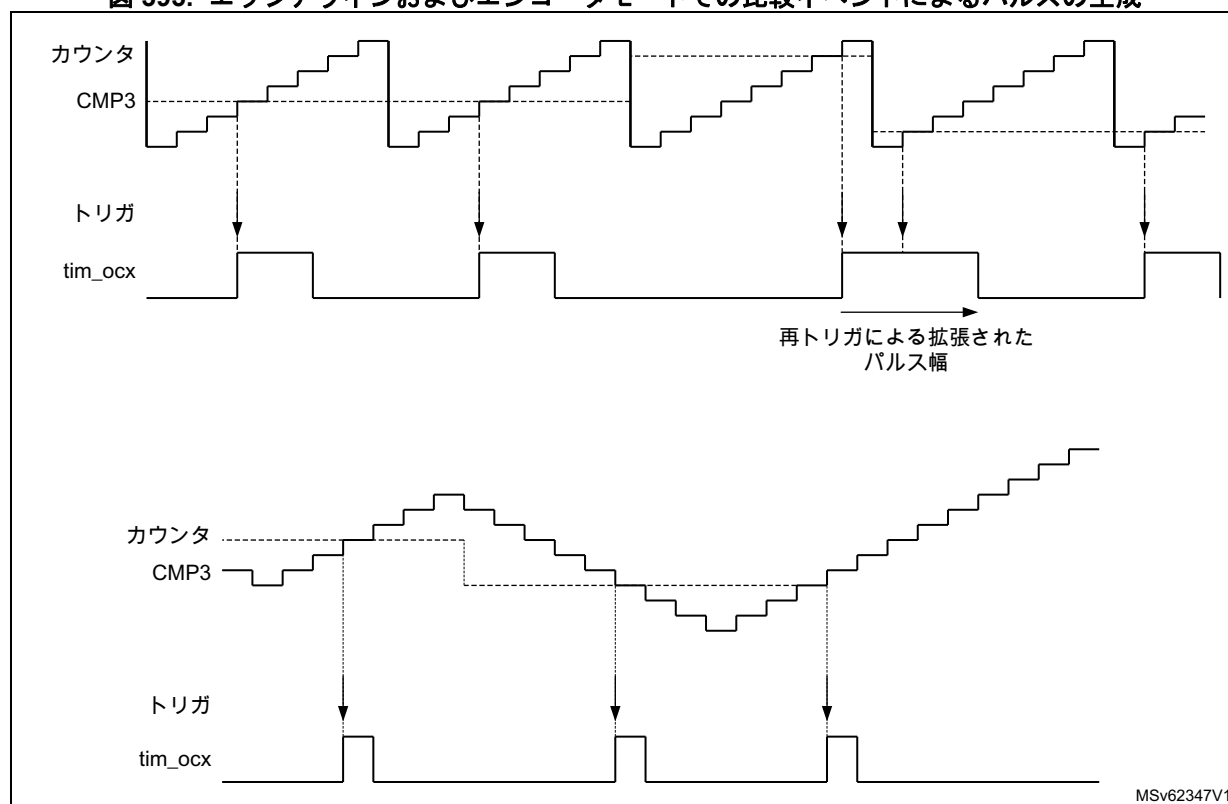
このモードは、エンコーダモードを含むすべてのスレーブモードの選択で、エッジアラインおよびセンタアラインのカウントモードで使用できます。これは、チャンネル 3 とチャンネル 4 のみで使用できます。以下の図 394 に示すように、パルス発生回路はただ 1 つであり、2 つのチャンネルで共有されます。

図 394. パルス発生回路



下記の図 395 は、エッジアラインおよびエンコーダ動作モードでパルスがどのように生成されるかを示しています。

図 395. エッジアラインおよびエンコーダモードでの比較イベントによるパルスの生成



この出力比較モードは、TIMx_CCMR2 レジスタの OC3M[3:0] および OC4M[3:0] ビットフィールドを使用して選択されます。

パルス幅は、次のように、PWPRSC[2:0] ビットに従ってプリスケールされた特定のクロックを使用し、レジスタの PW[7:0] ビットフィールドを使用してプログラムします。

$$t_{PW} = PW[7:0] \times t_{PWG}$$

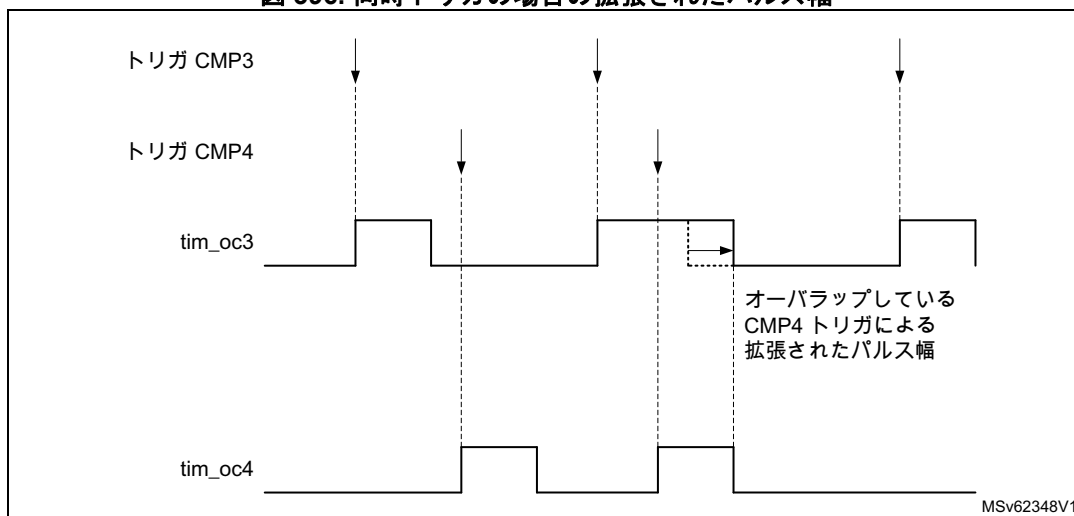
$$\text{ここで、} t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$$

上記は、プリスケラの値に応じた分解能と最大値を示します。

パルスは再トリガ可能です。パルスが進行中に新しいトリガが入ると、パルスは拡張されます。

注： 2つのチャンネルが同時に有効になっている場合、1つのチャンネルのトリガが並行出力上に生成されたパルスとオーバーラップしない限り、パルスは独立して発行されます。反対に、2つのトリガがオーバーラップしている場合、最初に到着したトリガに関連するパルス幅は（再トリガのために）拡張されますが、後に到着したトリガのパルス幅は正しくなります（以下の図 396 に示すとおり）。

図 396. 同時トリガの場合の拡張されたパルス幅



38.3.25 エンコーダインタフェースモード

直交エンコーダ

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが tim_ti1 エッジのみをカウントしている場合は SMS="0001" を、tim_ti2 エッジのみをカウントしている場合は SMS="0010" を、tim_ti1 と tim_ti2 の両方のエッジをカウントしている場合は SMS="0011" を書き込みます。

tim_ti1 と tim_ti2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2つの入力 tim_ti1 と tim_ti2 は、直交エンコーダとのインタフェースに使用されます。表 390 を参照してください。カウンタのクロック供給は、tim_ti1fp1 または tim_ti2fp2 (入力フィルタおよび極性選択の後には tim_ti1 と tim_ti2、フィルタされず反転されない場合は tim_ti1fp1=tim_ti1、フィルタされず反転されない場合は tim_ti2fp2=tim_ti2) での有効な遷移ごとに行われます。ただし、カウンタは有効である (TIMx_CR1 レジスタの CEN ビットに "1" が書き込まれている) ことが前提です。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが tim_ti1 のみ、tim_ti2 のみ、または tim_ti1 と tim_ti2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (tim_ti1 または tim_ti2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機能を持続します。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (tim_ti1 と tim_ti2 は同時に切り替わらないと想定しています)。

表 390. カウント方向とエンコーダ信号 (CC1P = CC2P = 0)

アクティブエッジ	SMS[3:0]	他方の信号の レベル (tim_ti2 に対する tim_ti1fp1、 tim_ti1 に対する tim_ti2fp2)	tim_ti1fp1 信号		tim_ti2fp2 信号	
			立上がり	立下がり	立上がり	立下がり
tim_ti1 のみカウント x1 モード	1110	高	ダウン	アップ	カウントなし	カウントなし
		ロー	カウントなし	カウントなし	カウントなし	カウントなし
tim_ti2 のみカウント x1 モード	1111	高	カウントなし	カウントなし	アップ	ダウン
		ロー	カウントなし	カウントなし	カウントなし	カウントなし
tim_ti1 のみカウント x2 モード	0001	高	ダウン	アップ	カウントなし	カウントなし
		ロー	アップ	ダウン	カウントなし	カウントなし
tim_ti2 のみカウント x2 モード	0010	高	カウントなし	カウントなし	アップ	ダウン
		ロー	カウントなし	カウントなし	ダウン	アップ
tim_ti1 と tim_ti2 の 両方をカウント x4 モード	0011	高	ダウン	アップ	アップ	ダウン
		ロー	アップ	ダウン	ダウン	アップ

直交エンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部トリガ入力に接続して、カウンタのリセットをトリガできます。

図 397 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx_CCMR1 レジスタ、tim_ti1fp1 は tim_ti1 に配置)
- CC2S="01" (TIMx_CCMR2 レジスタ、tim_ti1fp2 は tim_ti2 に配置)
- CC1P="0"、CC1NP="0" (TIMx_CCER レジスタ、tim_ti1fp1 非反転、tim_ti1fp1=tim_ti1)
- CC2P="0"、CC2NP="0" (TIMx_CCER レジスタ、tim_ti1fp2 非反転、tim_ti1fp2=tim_ti2)
- SMS="0011" (TIMx_SMCR レジスタ、両方の入力が立上がりと立下がりの両エッジでアクティブ)
- CEN="1" (TIMx_CR1 レジスタ、カウンタ有効)

図 397. エンコーダインタフェースモードにおけるカウンタの動作例

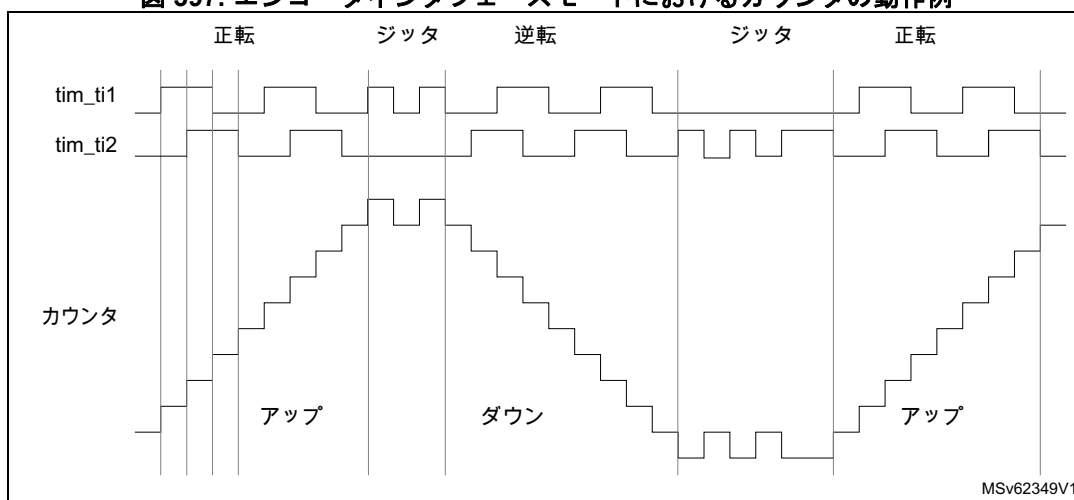
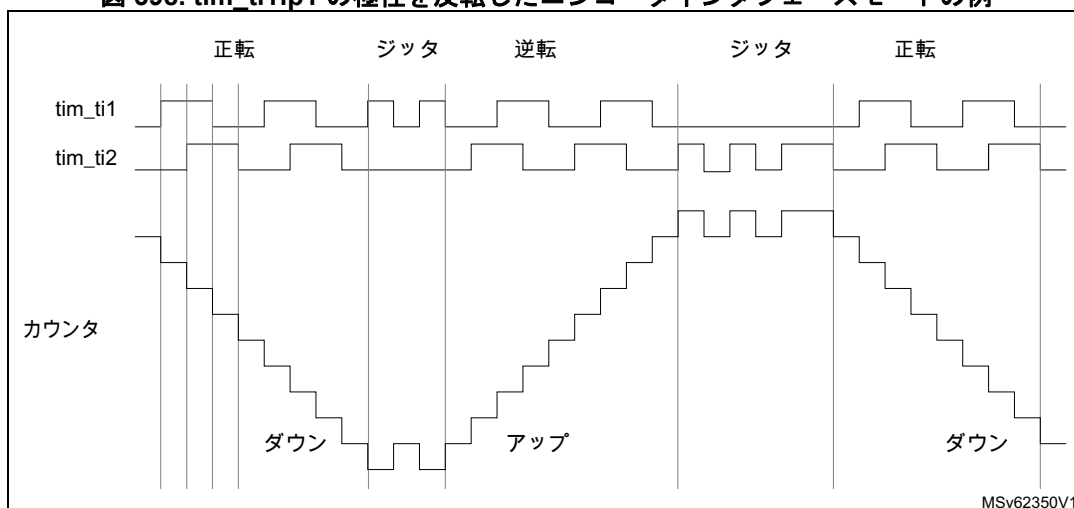


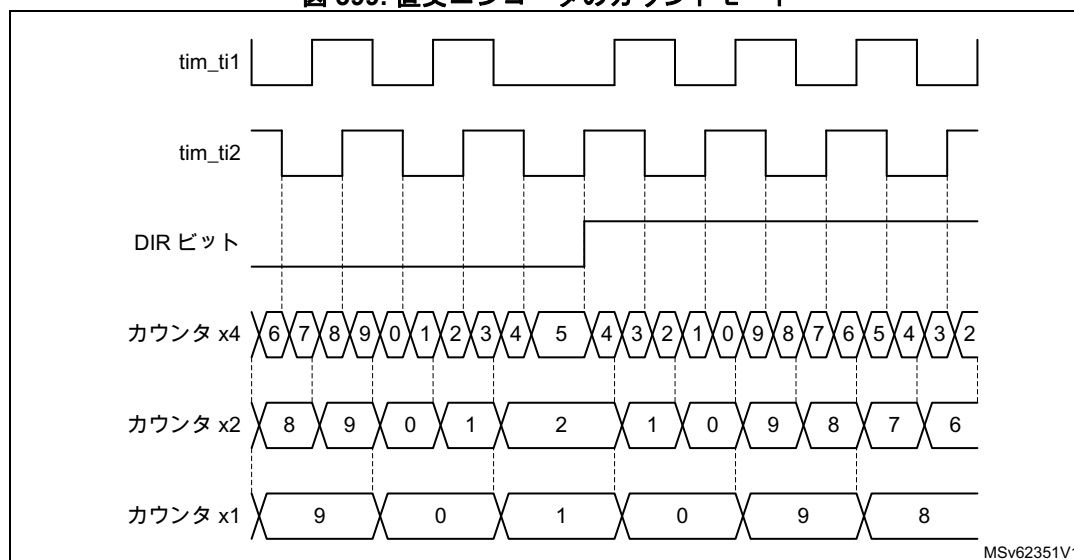
図 398 に、tim_ti1fp1 の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、CC1P="1"）。

図 398. tim_ti1fp1 の極性を反転したエンコーダインタフェースモードの例



以下の図 399 は、さまざまなカウントモードでの速度反転中のタイマカウンタ値を示しています。

図 399. 直交エンコーダのカウントモード



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、DMA リクエストを通じて値を読み出すことも可能です。

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。バックグラウンドタスク（カウンタの読出し）と割り込み（更新割り込み）との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読出しアクセス時に UIFCPY フラグによって上書きされます（カウンタの最上位ビットには書き込みモード時のみアクセス可能）。

クロックプラス方向エンコーダモード

直交エンコーダモードのほか、タイマは他のタイプのエンコーダもサポートしています。

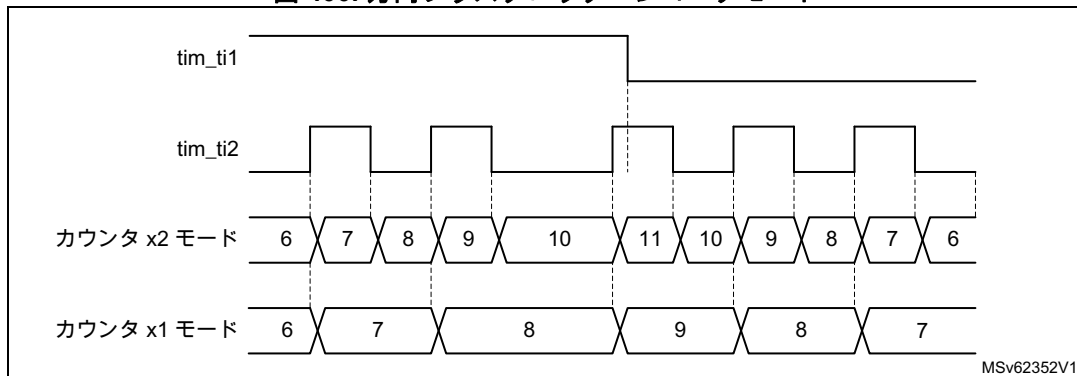
図 400 に示す「クロックプラス方向」モードでは、`tim_ti2` の単一ラインでクロックが提供され、`tim_ti1` 入力を使用して方向が強制されます。

このモードは TIMx_SMCR レジスタの SMS[3:0] ビットフィールドによって、以下のように有効にされます。

- 1010 : X2 モード。カウンタはクロックの立上がりりと立下がりの両エッジで更新されます。
- 1011 : X1 モード。カウンタは 次の CC2P ビット値に従って単一のクロックエッジで更新されます。CC2P = 0 は立上がりエッジ検出に対応し、CC2P = 1 は立下がりエッジ検出に対応します。

tim_ti1 の方向信号の極性は、次の CC1P ビットで設定されます。0 は正の極性 (tim_ti1 がハイのときアップカウントで、tim_ti1 がローのときダウンカウント) に対応し、CC1P = 1 は負の極性 (tim_ti1 がローのときアップカウント) に対応します。

図 400. 方向プラスクロックエンコーダモード



方向クロックエンコーダモード

図 401 の「方向クロック」モードでは、クロックは 2 本のラインで提供され、1 本のアップカウントクロックラインと 1 本のダウンカウントクロックラインを持つというように、方向に応じて一度に 1 つが提供されます。

このモードは TIMx_SMCR レジスタの SMS[3:0] ビットフィールドによって、以下のように有効にされます。

- 1100 : x2 モード。カウンタは、2 つのクロックラインのいずれかの立上がりで立上がり、立下がり更新されます。CC1P および CC2P ビットは、クロックアイドル状態用のコーディングです。CCxP = 0 はハイレベルアイドル状態に対応し (下記図 401 を参照)、CCxP = 1 はローレベルアイドル状態に対応します (下記図 402 を参照)。
- 1101 : x1 モード。カウンタは、CC1P および CC2P ビット値に従って、単一のクロックエッジで更新されます。CCxP = 0 は立下がり検出とハイレベルアイドル状態に対応し (下記図 401 を参照)、CCxP = 1 は立上がり検出とローレベルアイドル状態に対応します (下記図 402 を参照)。

図 401. 方向クロックエンコーダモード (CC1P = CC2P = 0)

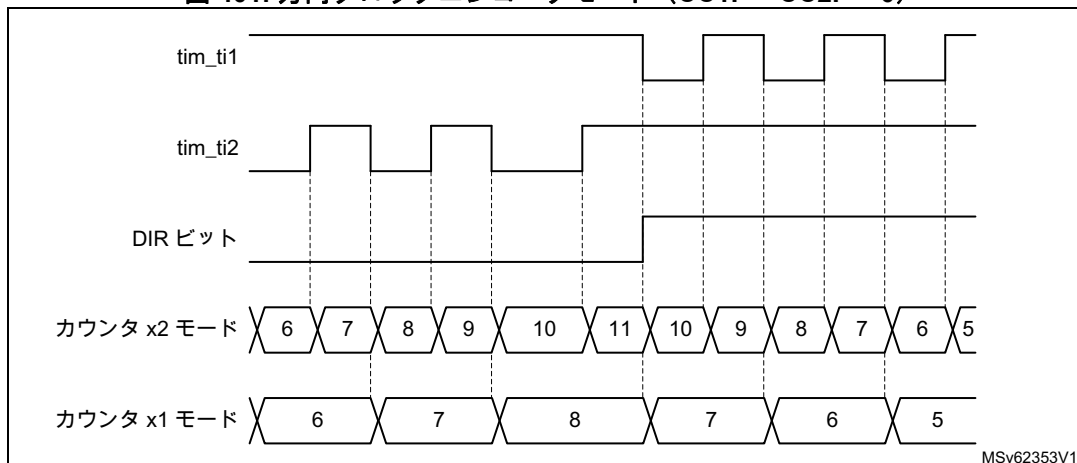
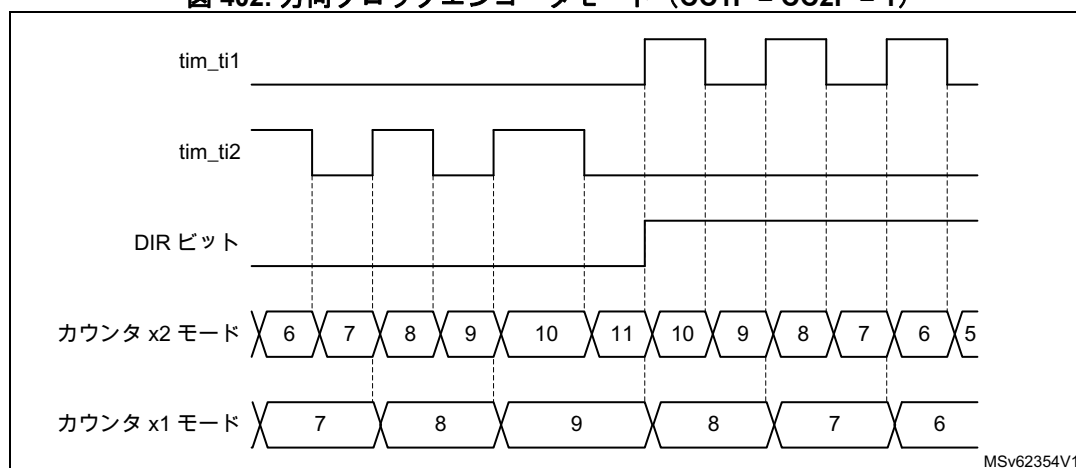


図 402. 方向クロックエンコーダモード (CC1P = CC2P = 1)



以下の表 391 に、任意の入力遷移に対して方向クロックモードがどのように動作するかを詳しく説明します。

表 391. カウント方向 対 エンコーダ信号および極性設定

方向クロック モード	SMS[3:0]	他方の信号の レベル (tim_ti2 に対する tim_ti1fp1、 tim_ti1 に対する tim_ti2fp2)	tim_ti1fp1 信号		tim_ti2fp2 信号	
			立上がり	立下がり	立上がり	立下がり
x2 モード CCxP=0	1100	高	ダウン	ダウン	アップ	アップ
		ロー	カウントなし	カウントなし	カウントなし	カウントなし
x2 モード CCxP=1	1100	高	カウントなし	カウントなし	カウントなし	カウントなし
		ロー	ダウン	ダウン	アップ	アップ
x1 モード CCxP=0	1101	高	カウントなし	ダウン	カウントなし	アップ
		ロー	カウントなし	カウントなし	カウントなし	カウントなし
x1 モード CCxP=1	1101	高	カウントなし	カウントなし	カウントなし	カウントなし
		ロー	ダウン	カウントなし	アップ	カウントなし

インデックス入力

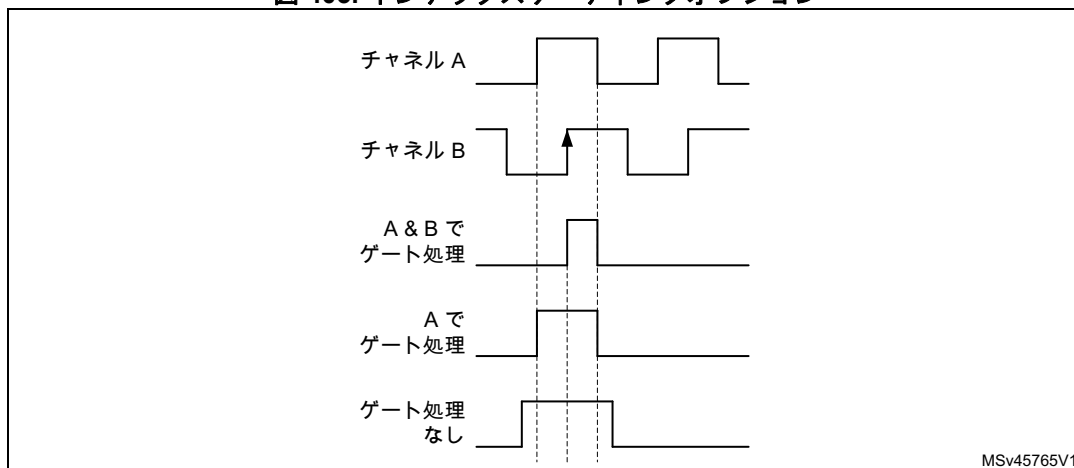
カウンタは、絶対基準位置を示すエンコーダからのインデックス信号によってリセットできます。インデックス信号は、tim_etr_in 入力に接続する必要があります。これは、デジタル入力フィルタを使用してフィルタリングできます。

インデックス機能は、TIMX_ECR レジスタの IE ビットで有効にされます。IE ビットは、SMS[3:0] ビットフィールドが次の値であるときに、エンコーダモードでのみセットする必要があります。0001、0010、011、1010、1011、1100、1101、1110、1111。

以下の図 403 に示すように、インデックスパルス調整のいくつかのオプションを備えた市販のエンコーダが提案されています。

- A および B でゲート処理：パルス幅は 1 つのチャンネル周期の 1/4 であり、A および B の両方のエッジにそろえられます。
- A でゲート処理（または B でゲート処理）：パルス幅は 1 つのチャンネル周期の 1/2 であり、チャンネル A（または チャンネル B）の 2 つのエッジにそろえられます。
- ゲート処理なし：パルス幅は最大 1 チャンネル周期であり、どのエッジにもそろえられません。

図 403. インデックスゲーティングオプション

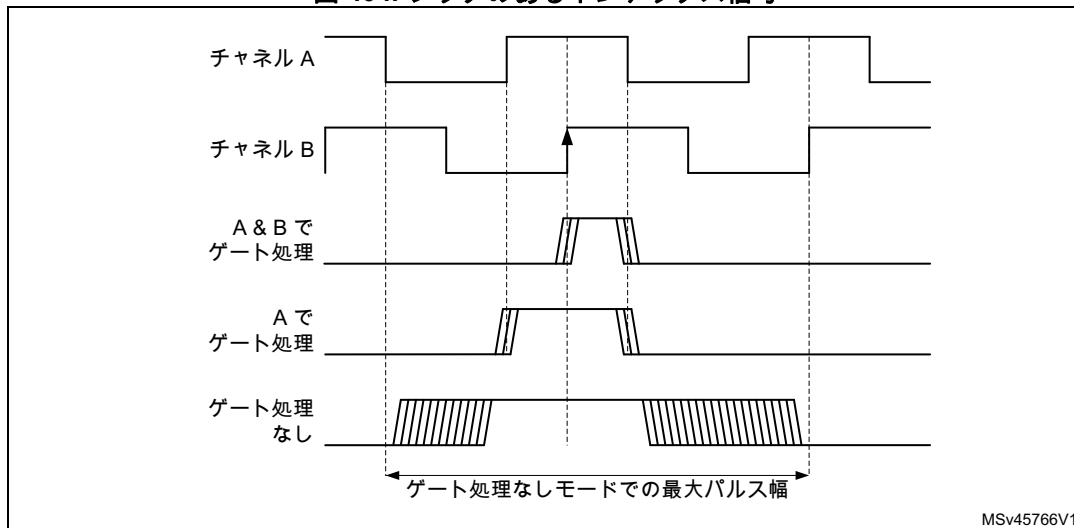


MSv45765V1

以下の図 404 に示すように、ゲーティングモードに関係なく、回路はインデックス信号のジッタを許容します。

ゲート処理なしモードでは、信号はエンコーダの 2 周期より厳密に小さくしなければなりません。パルス幅がエンコーダの 2 周期以上の場合、カウンタは複数回リセットされます。

図 404. ジッタのあるインデックス信号



MSv45766V1

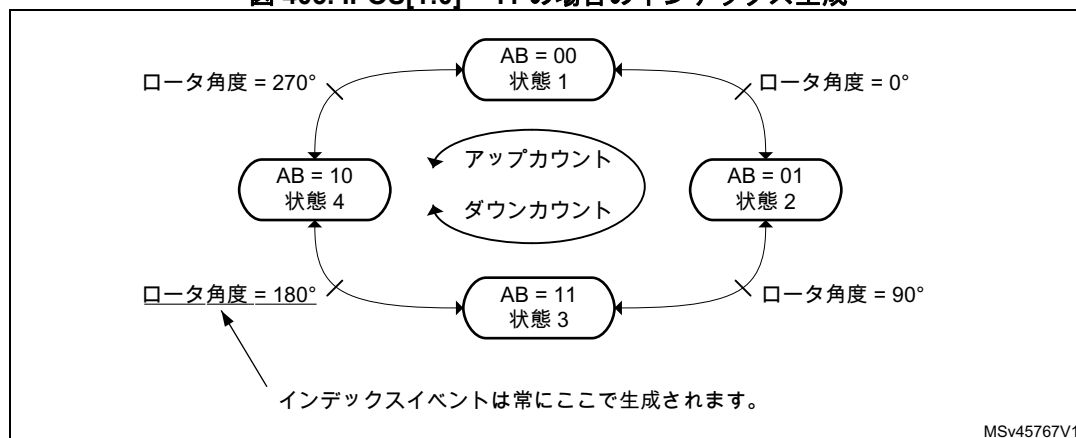
タイマは、特定のプログラミングを必要とせずに、3 つのゲーティングオプションを同様にサポートします。TIMx_ECR レジスタの IPOS[1:0] ビットフィールドを使用して、インデックスを同期する必要があるエンコーダ状態（チャンネル A とチャンネル B の状態の組み合わせ）を定義するだけです。

インデックス検出イベントは、速度の反転中に対称動作を確保するために、カウント方向に応じて異なる動作をします。

- カウンタはアップカウント中 (DIR ビット = 0) に、リセットされます。
- カウンタはダウンカウント時に、TIMx_ARR に設定されます。

これにより、カウント方向に関係なく、まったく同じ機械的角度位置でインデックスを生成できます。以下の図 405 は、単純な例 (機械的回転ごとに 4 つのエッジを提供するエンコーダ) について、生成されるインデックスの位置を示しています。

図 405. IPOS[1:0] = 11 の場合のインデックス生成



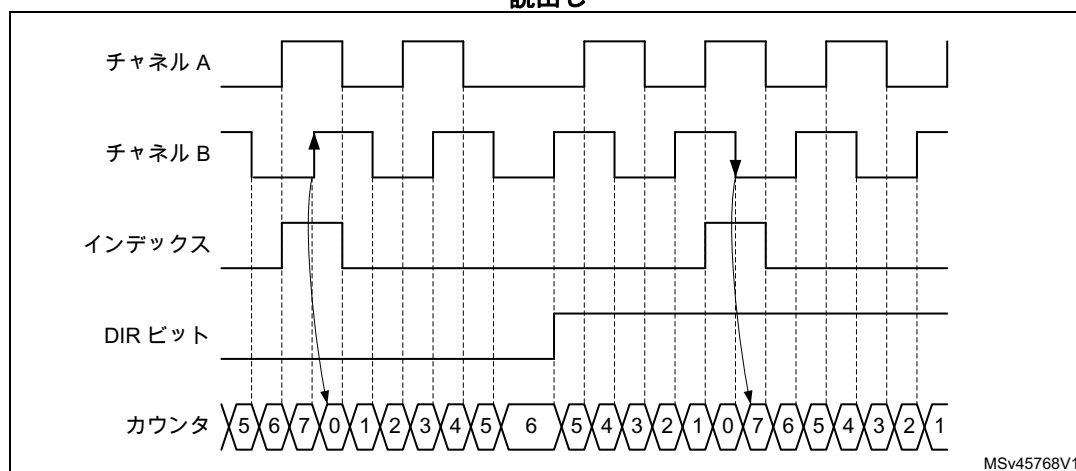
以下の図 406 に、IPOS[1:0] = 11 の場合の波形と対応する値を示します。カウンタの値が強制される瞬間は、カウント方向に応じて自動的に調整されることを示しています。

- アップカウント (DIR ビット = 0) の場合、エンコーダの状態が「11」 (ChA = 1, ChB = 1) のとき、カウンタは 0 にセットされます。
- ダウンカウント (DIR ビット = 1) の場合、「11」状態を終了するとき、カウンタは TIMx_ARR にセットされます。

インデックス検出イベント時に割込みを発行できます。

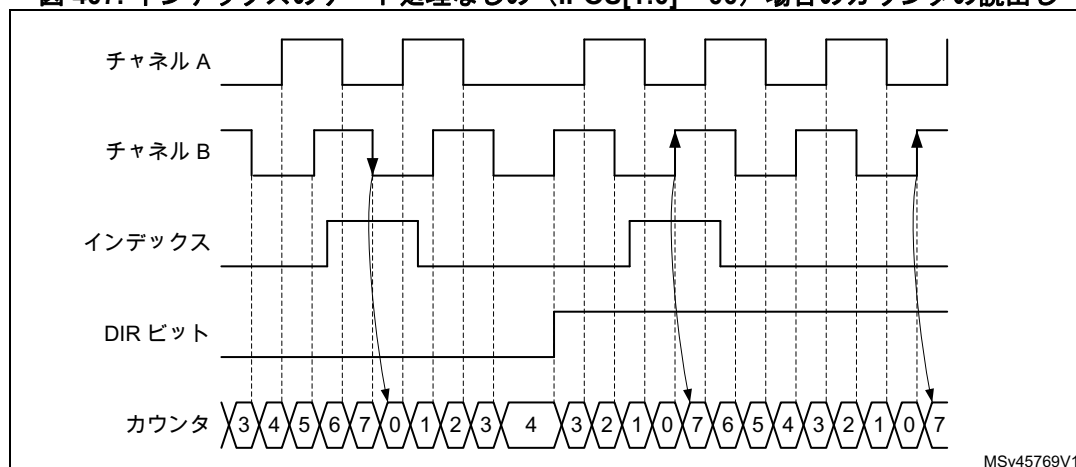
矢印は、どの遷移でインデックスイベント割込みが生成されるかを示しています。

図 406. インデックスがチャンネル A でゲート処理された (IPOS[1:0] = 11) 場合のカウンタの読み出し



以下の図 407. に、ゲート処理なしモードの場合の波形と対応する値を示します。矢印は、どの遷移でインデックスイベントが生成されるかを示しています。

図 407. インデックスのゲート処理なしの (IPOS[1:0] = 00) 場合のカウンタの読出し



以下の図 408. は、さまざまなパルスアライメントシナリオでの「A & B でゲート処理」モードの処理方法を示しています。矢印は、どの遷移でインデックスイベントが生成されるかを示しています。

図 408. インデックスがチャンネル A および B でゲート処理された場合のカウンタの読出し

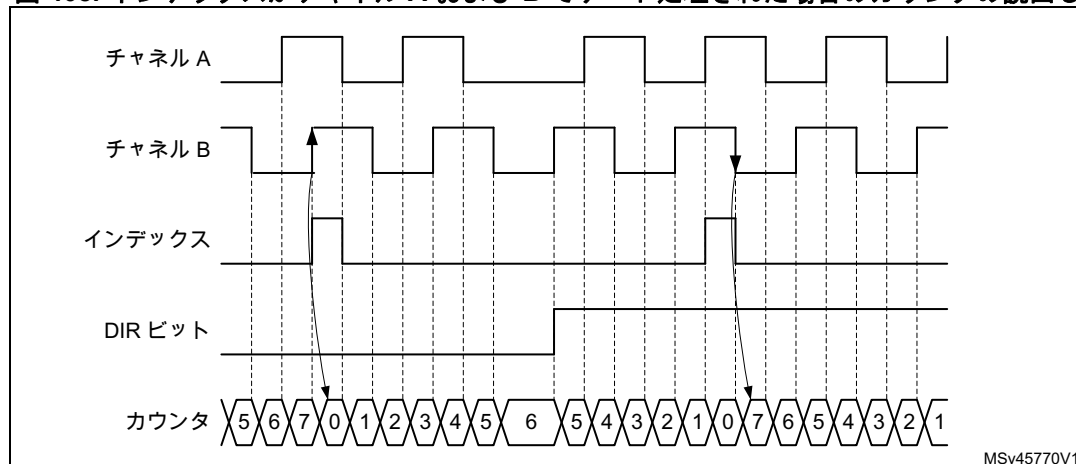


図 409 と 図 410 には、後続のインデックスパルスが、エンコーダクロック周期の 4 分の 1 よりも狭いことがあるケースが詳しく示されています。

図 409. 狭いインデックスパルスの場合のエンコーダモードの動作 (IPOS[1:0] = 11)

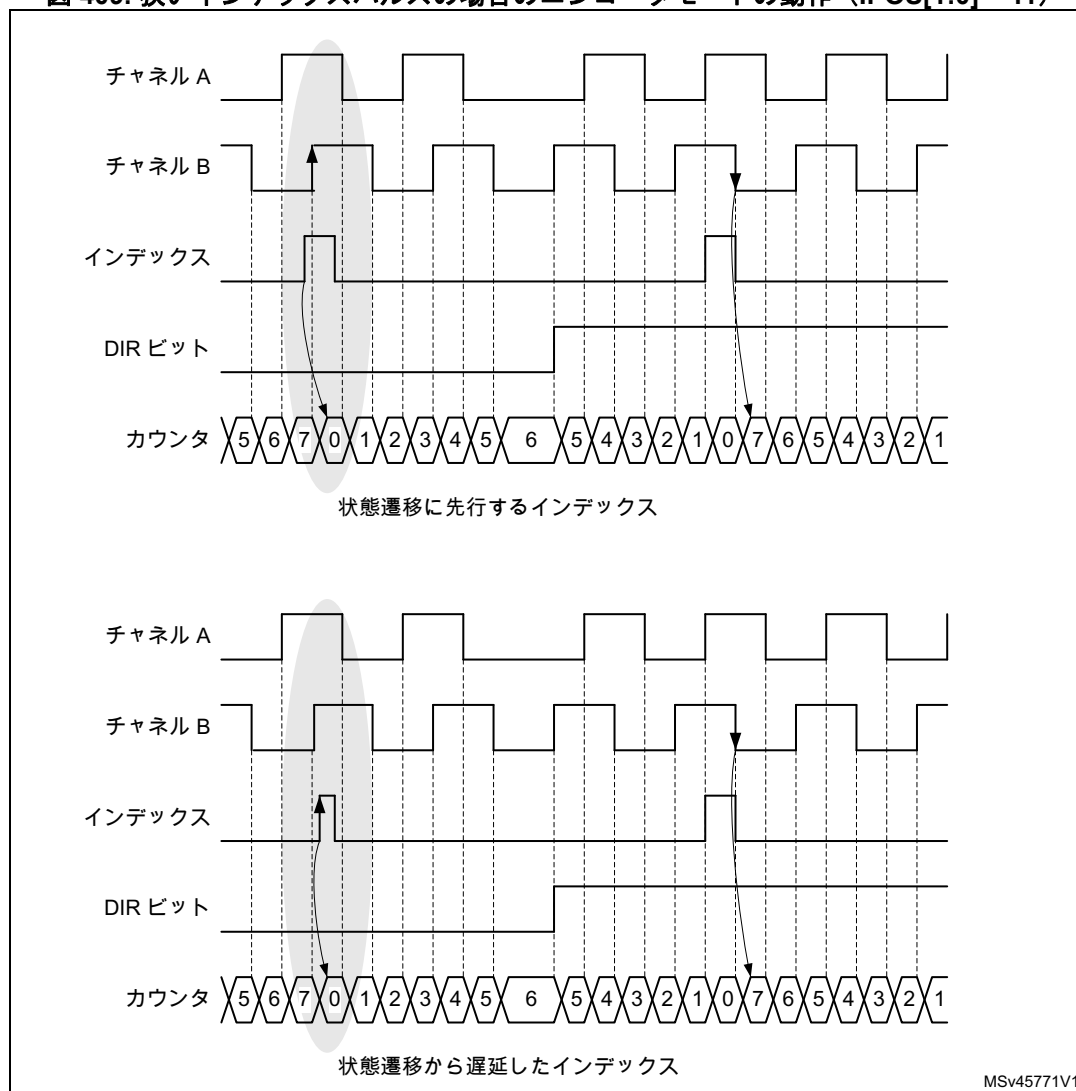
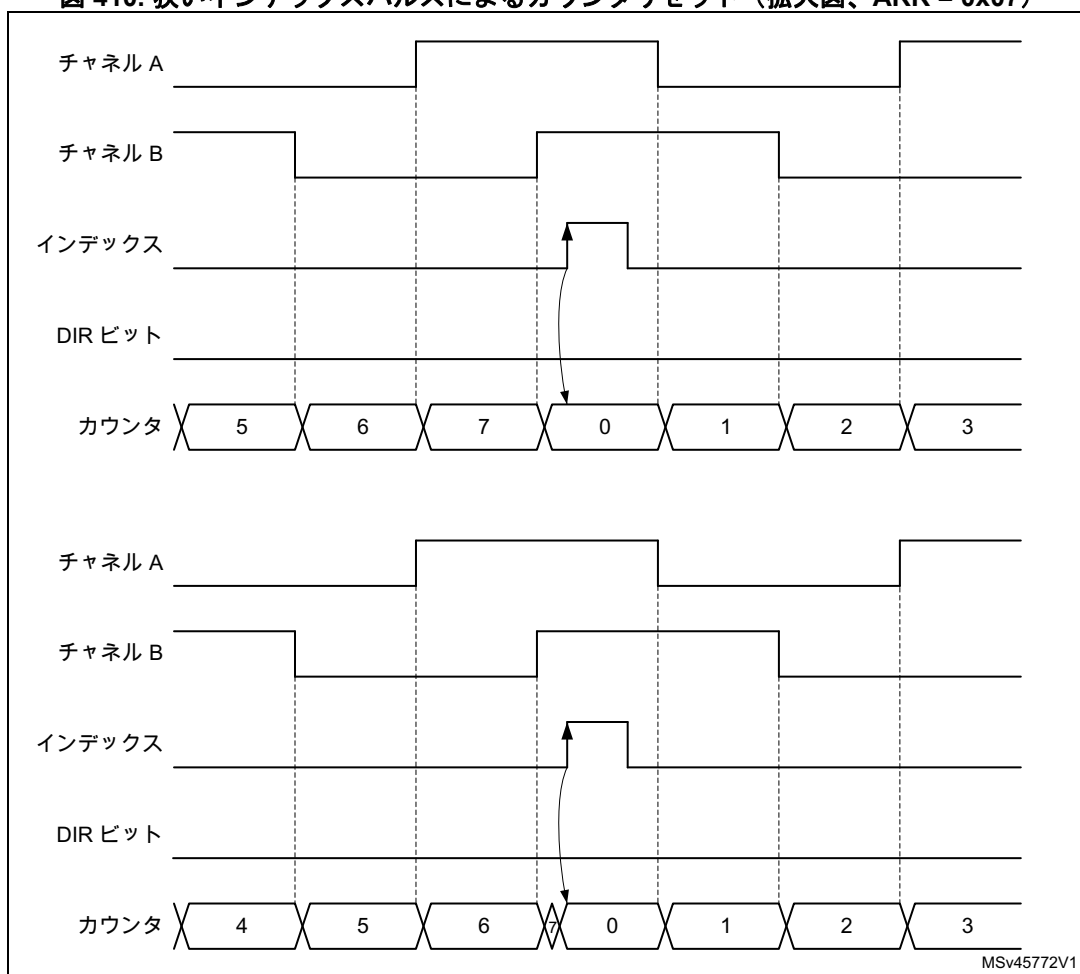
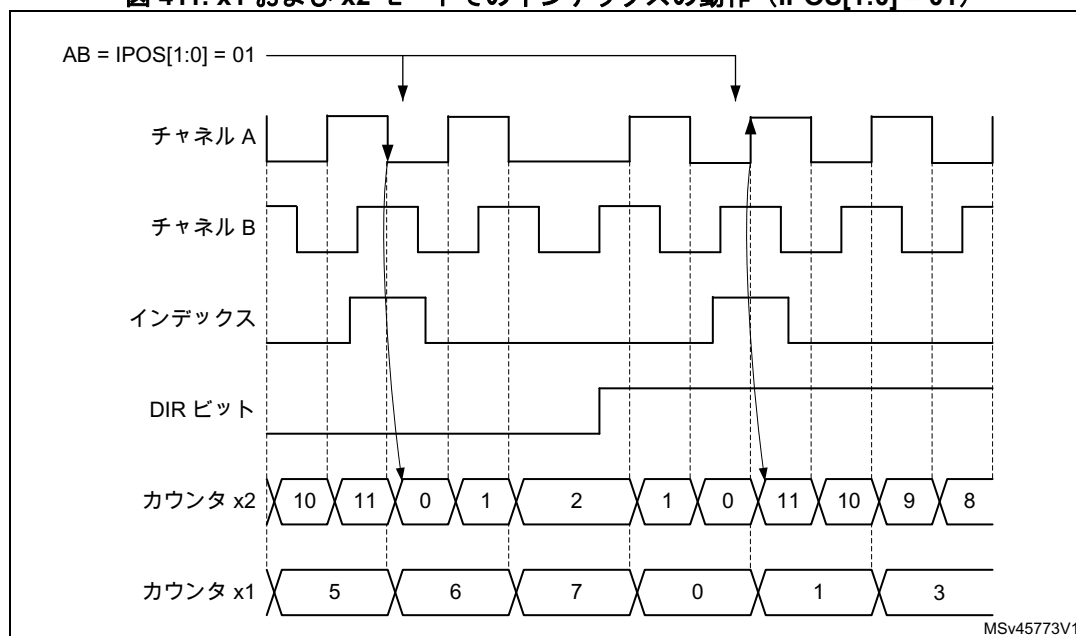


図 410. 狭いインデックスパルスによるカウンタリセット (拡大図、ARR = 0x07)



下記の図 411 は、x1 および x2 モードでインデックスがどのように管理されるかを示します。

図 411. x1 および x2 モードでのインデックスの動作 (IPOS[1:0] = 01)

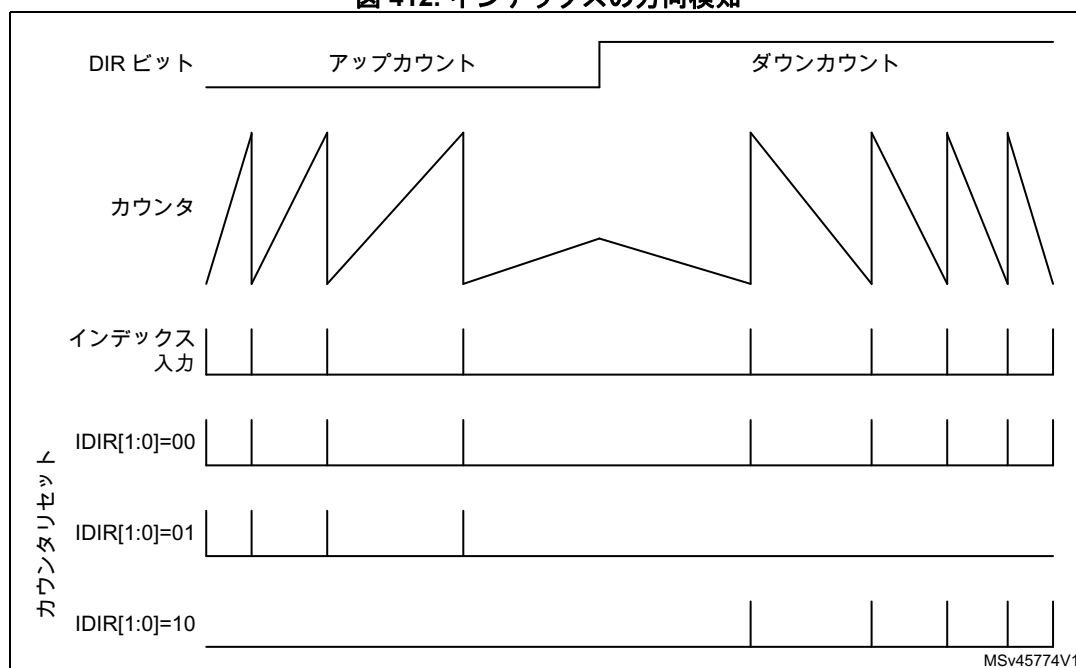


インデックスの方向検知

TIMx_ECR レジスタの IDIR[1:0] ビットフィールドにより、選択したカウント方向でのみインデックスをアクティブにできます。

以下の図 412 は、IDIR[1:0] 値に応じた、インデックスとカウンタのリセットイベントの関係を示しています。

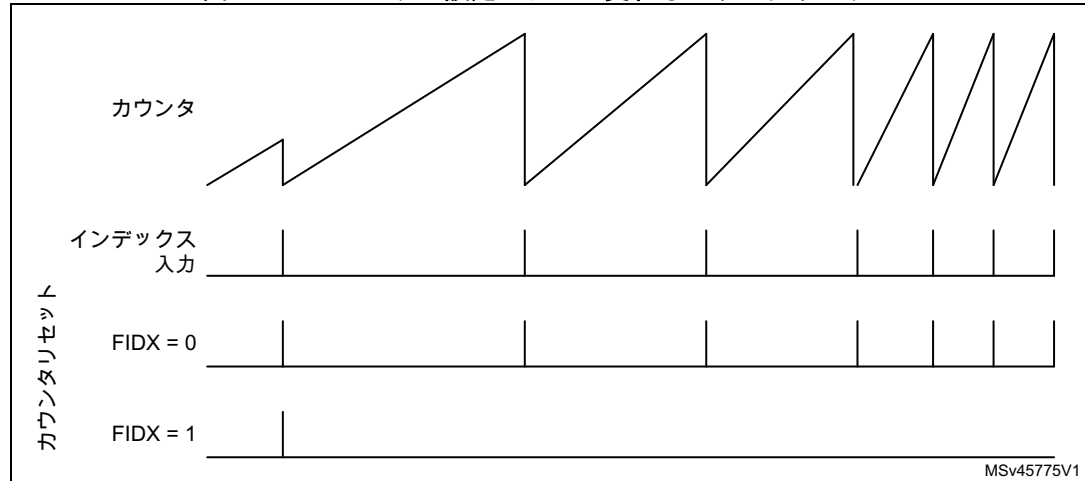
図 412. インデックスの方向検知



特殊な最初のインデックスイベント管理

TIMx_ECR レジスタの FIDX ビットによって、以下の図 413 に示すように、インデックスを 1 回だけ取得できます。最初のインデックスが到着すると、その後のインデックスは無視されます。必要に応じて、FIDX ビットに 0 を書き込み、再度 1 にセットすることにより、回路を再設定できます。

図 413. FIDX ビット設定によって変わるカウンタリセット



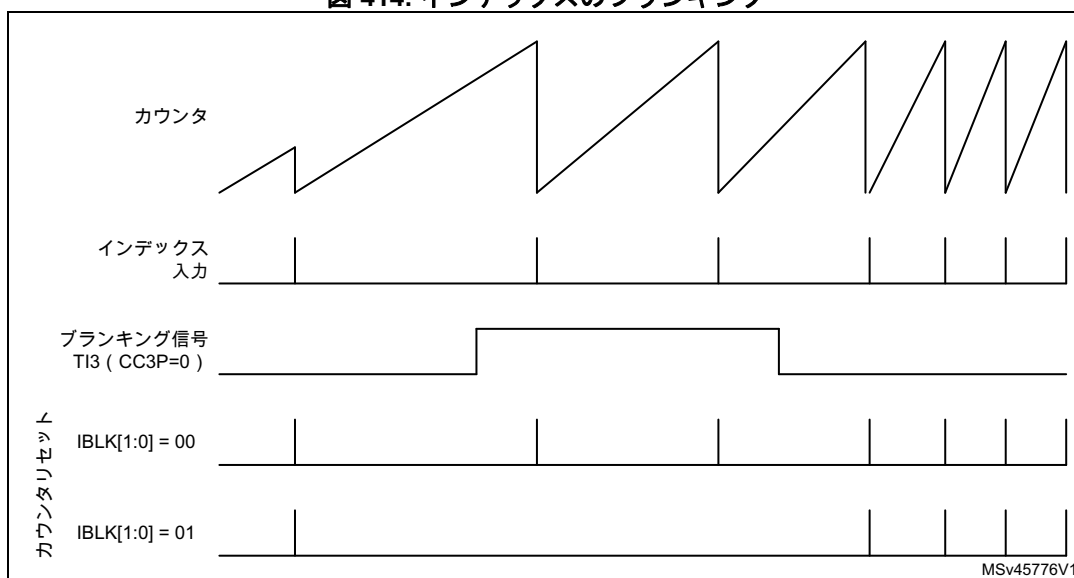
インデックスのブランキング

インデックスイベントは、tim_ti3 または tim_ti4 の入力を使用してブランキングすることができます。ブランキング期間中は、以下の図 414 に示すように、インデックスイベントがカウンタをリセットすることはありません。

このモードは TIMx_ECR レジスタの IBLK[1:0] ビットフィールドを使用して、以下のように有効にされます。

- IBLK[1:0] = 00 : インデックス信号は常にアクティブです。
- IBLK[1:0] = 01 : インデックス信号は tim_ti3 入力時にブランキングされます。
- IBLK[1:0] = 10 : インデックス信号は tim_ti4 入力時にブランキングされます。

図 414. インデックスのブランキング



非直交モードでのインデックス管理

以下の図 415 および図 416 では、SMS[3:0] ビットフィールドが 1010、1011、1100、1101 に等しい場合に、方向クロックモードおよびクロックプラス方向モードでインデックスがどのように管理されるかを詳しく説明します。

これらのモードの両方で、インデックスの検出は次のように IPOS[0] ビットで設定されます。

- IPOS[0] = 0 : インデックスはクロックのローレベルで検出されます。
- IPOS[0] = 1 : インデックスはクロックのハイレベルで検出されます。

IPOS[1] ビットは意味がありません。

図 415. クロックプラス方向モードでのインデックスの動作 (IPOS[0] = 1)

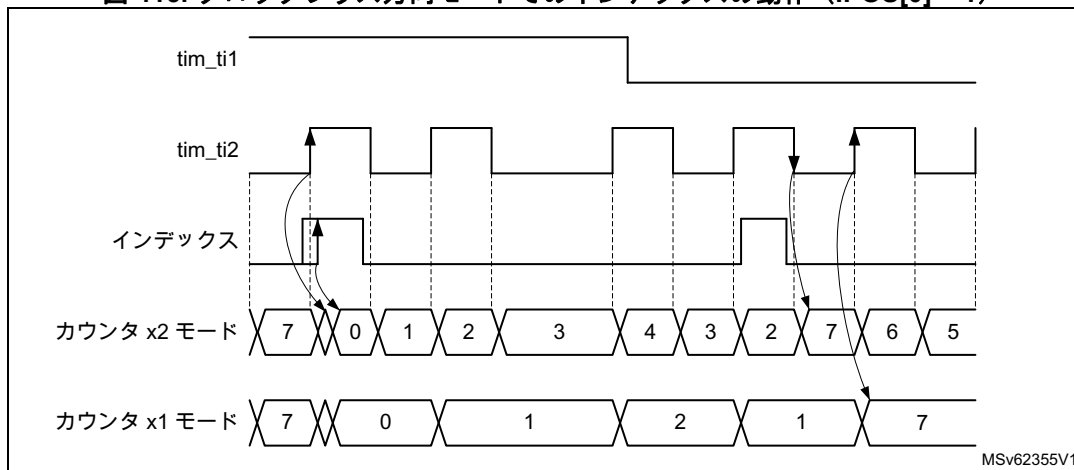
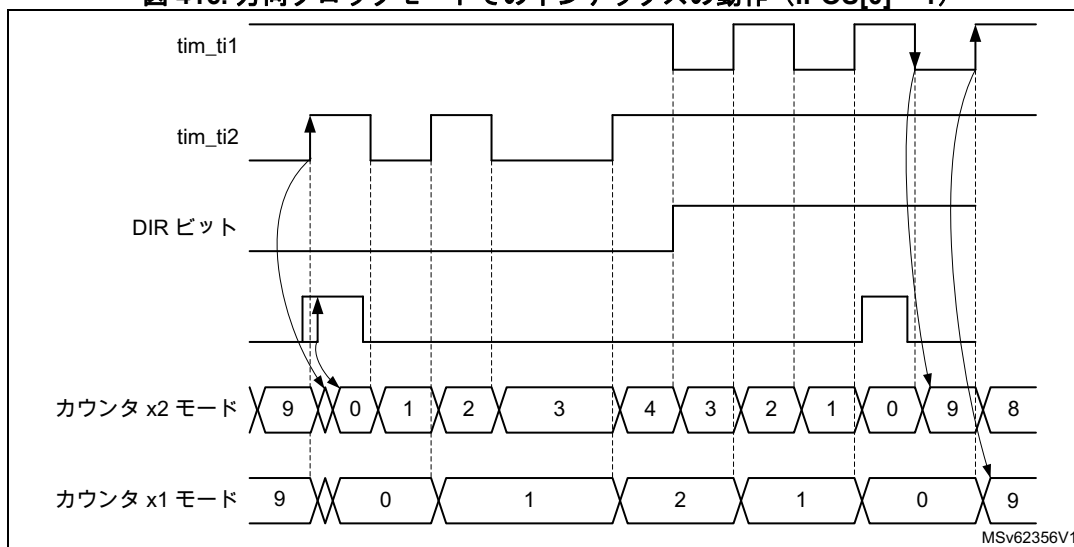


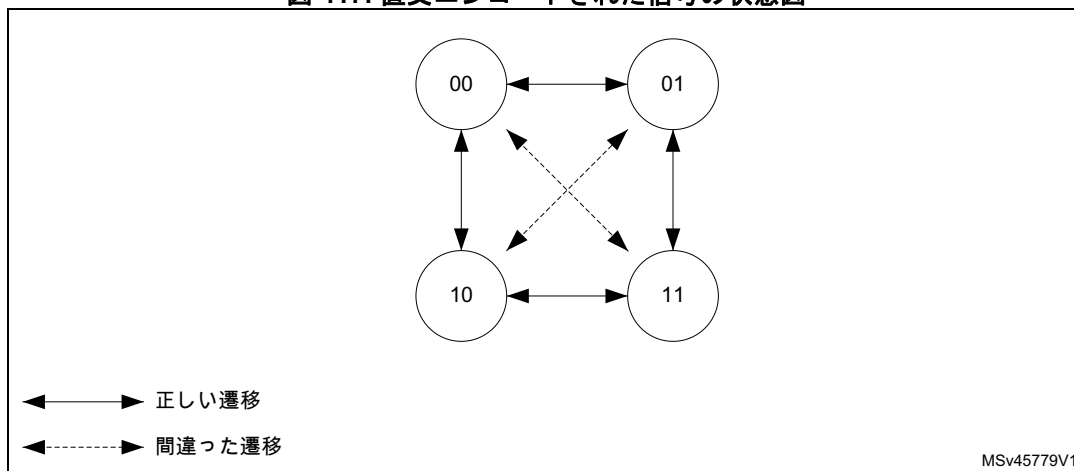
図 416. 方向クロックモードでのインデックスの動作 (IPOS[0] = 1)



エンコーダのエラー管理

2つの直交信号が使用できるエンコーダ構成の場合、遷移エラーを検出できます。2つの入力の読み値は、以下の図 417 で、状態図として表すことができる2ビットのグレイコードに対応しています。一度に1つのビットだけが変化することが予測されています。間違った遷移があると、TIMx_SR ステータスレジスタの TERRF 割込みフラグがセットされます。TIMx_DIER レジスタの TERRIE ビットがセットされていると、遷移エラー割込みが生成されます。

図 417. 直交エンコードされた信号の状態図



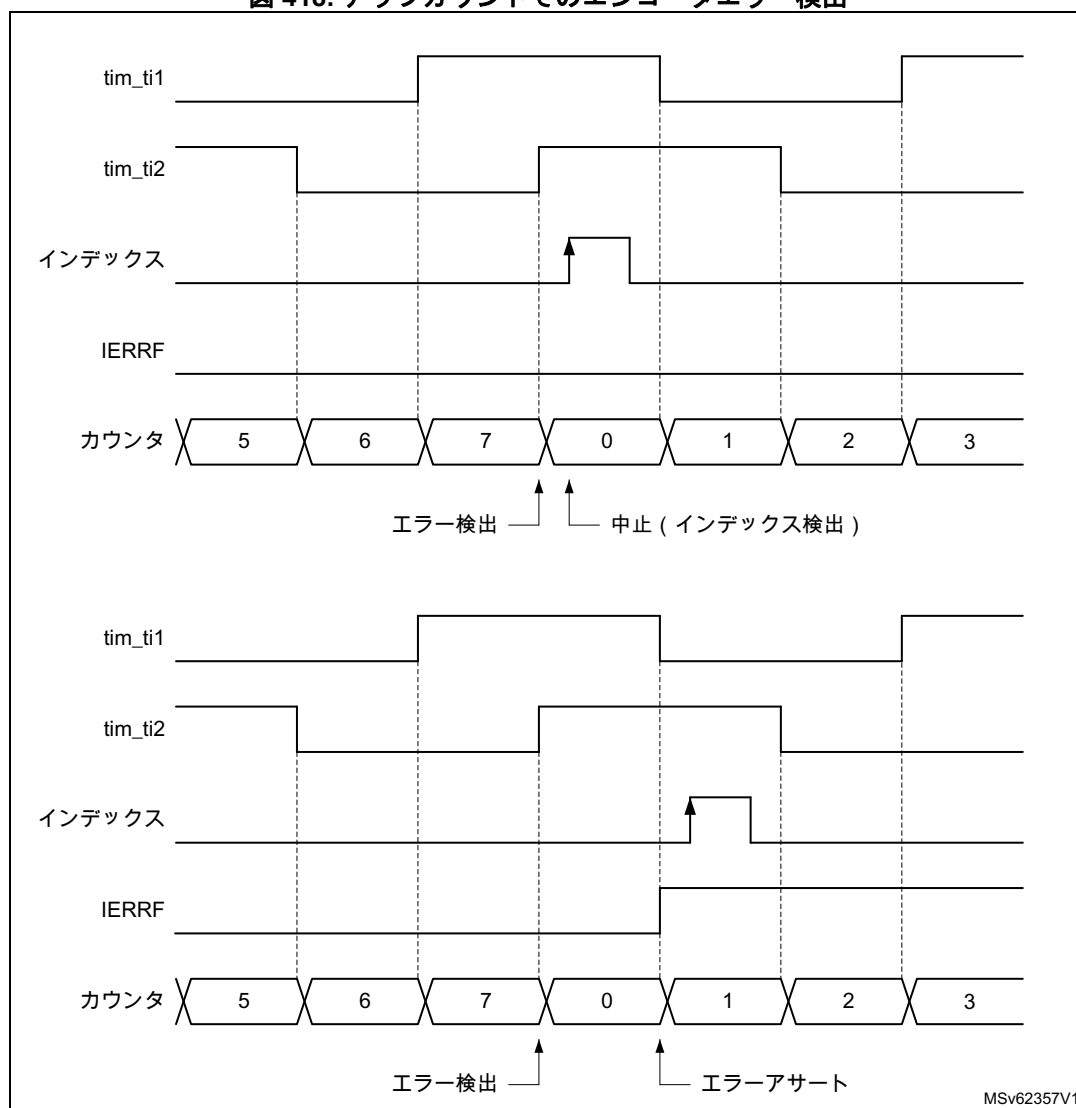
インデックス信号を持つエンコーダの場合、1回転あたりのパルスが過剰になる異常動作を検出できます。1回転あたり N 個のパルスを持つエンコーダでは、1回転あたり 4xN カウントが提供されます。インデックス信号によって、4xN クロック周期ごとにカウンタがリセットされます。

カウンタ値がインデックスイベントなしに TIMx_ARR から 0 にインクリメントされるか、0 から TIMxARR 値にデクリメントされる場合、これはインデックス位置エラーとして報告されます。

オーバーフロー閾値は、TIMx_ARR レジスタを使ってプログラムします。1000 行のエンコーダでは、カウンタ値が 0~3999 になります (4x 読み取りモードの場合)。オーバーフロー検出閾値は、TIMx_ARR = 3999 + 1 = 4000 を設定してプログラムする必要があります。

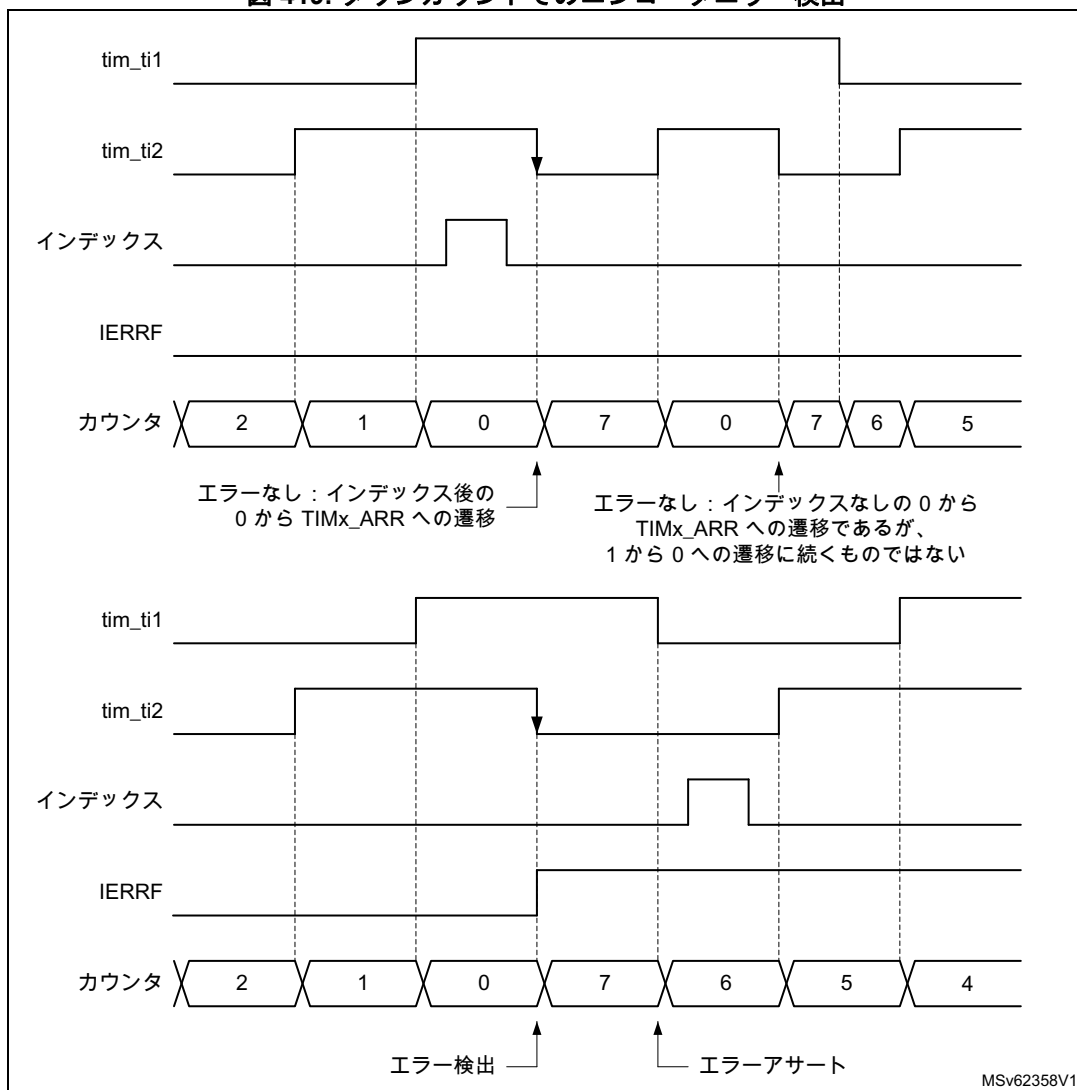
アップカウントでは、エラーアサーションは 0 から 1 への遷移まで遅延されます。これは、以下の図 418 に示すように、A および B ゲート処理モードでの狭いインデックスパルスへの対処です。

図 418. アップカウントでのエンコーダエラー検出



ダウンカウントモードでは、検出は 1 から 0 への事前遷移によって条件付けられます。これは、以下の図 419 に示すように、A および B ゲート処理モードでの狭いインデックスパルスに対処し、インデックス検出直後にエンコーダが TIMx_ARR と 0 の間でディザリングする場合の誤ったエラー検出を回避するためです。

図 419. ダウンカウントでのエンコーダエラー検出



インデックスエラーによって、TIMx_SR ステータスレジスタの IERRF 割込みフラグがセットされます。TIMx_DIER レジスタの IERRIE ビットがセットされていると、インデックスエラー割込みが生成されます。

エンコーダの機能的割込み

エンコーダモードでは、以下の割込みも使用できます。

- 方向変更: エンコーダモードでのカウント方向の変更によって、TIMx_CR1 レジスタの DIR ビットが反転します。方向変更によって、TIMx_SR ステータスレジスタの DIRF 割込みフラグがセットされます。TIMx_DIER レジスタの DIRIE ビットがセットされていると、方向変更割込みが生成されます。

- インデックスイベント: インデックスイベントによって、TIMx_SR ステータスレジスタの IDXF 割込みフラグがセットされます。TIMx_DIER レジスタの IDXIE ビットがセットされていると、インデックス割込みが生成されます。

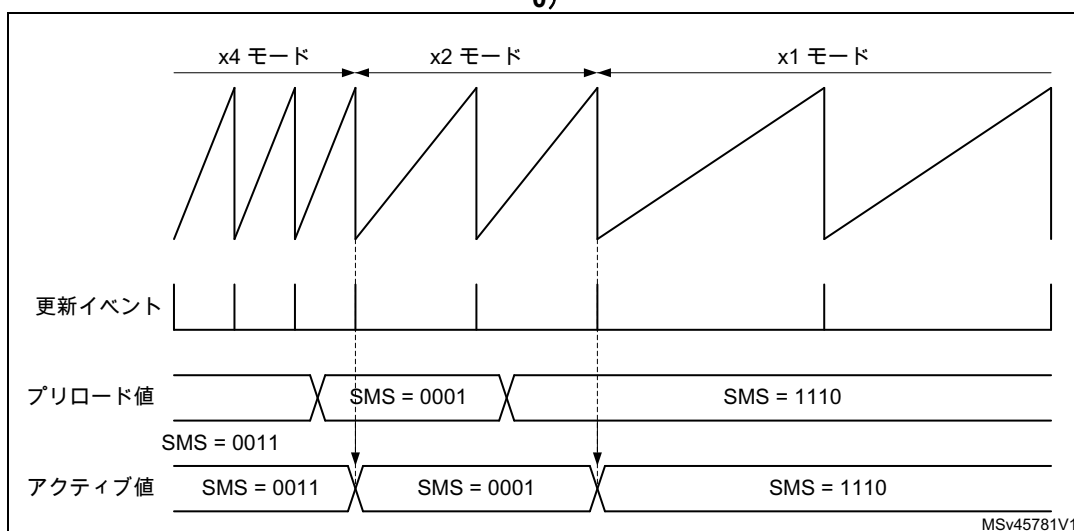
実行時エンコーダモード更新のためのスレーブモード選択プリロード

実行中に、あるエンコーダモードを別のモードに切り替える必要があることがあります。以下の図 420 に示すように、これは通常、高速時に更新割込みレートを低下させるために、x4 モードから x2 に、x1 に切り替えることにより、実行されます。

この目的のために、SMS[3:0] ビットをプリロードすることができます。これは、TIMx_SMCR レジスタの SMSPE イネーブルビットをセットすることで有効になります。SMS[3:0] プリロードからアクティブ値へ転送するトリガは、TIMx_SMCR レジスタの SMSPS ビットで選択できます。

- SMSPS = 0: 転送は、アップカウント時にカウンタがオーバーフローするとき、およびダウンカウント時にアンダーフローするときが発生する更新イベント (UEV) によってトリガされます。
- SMSPS = 1: 転送は、インデックスイベントによってトリガされます。

図 420. 更新によって転送されるプリロードを使ったエンコーダのモード変更 (SMSPS = 0)



エンコーダクロック出力

エンコーダモードの動作原理は、十分な数のクロックエッジ取得と正確な測定を行うために比較的長い積分時間を必要とするため、低速においては、高分解能の速度測定には完全には適していません。

低速でのより良いソリューションは、エッジ間のクロック周期測定を行うことです。これは、スレーブタイマを使用して達成できます。タイマによって、tim_trgo 出力にエンコーダクロック情報を出力できます。その後、スレーブタイマはエンコーダクロックエッジごとに周期測定を実行し、速度情報を提供できます。

このモードは TIMx_CR2 レジスタの MMS[3:0] ビットフィールドを 1000 に設定することによって有効になります。これは、以下の SMS[3:0] 値に対して有効です。0001、0010、0011、1010、1011、1100、1101、1110、1111。その他の SMS[3:0] コードは許されず、予想外の動作につながる可能性があります。

38.3.26 方向ビットの出力

tim_oc3n および tim_oc4 出力信号に、タイマから方向信号を出力できます (TIMx_CR1 レジスタの DIR ビットのコピー)。これを達成するには、TIMx_CCMR2 レジスタの OC3M[3:0] または OC4M[3:0] ビットフィールドを 1011 にセットします。

この機能を使用すると、エンコーダモードでカウント方向 (または回転方向) を監視すること、またはセンタアライン PWM モードでアップ/ダウンフェーズを示す信号を得ることができます。

38.3.27 UIF ビットの再配置

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特殊なケースでは、バックグラウンドタスク (カウンタの読出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

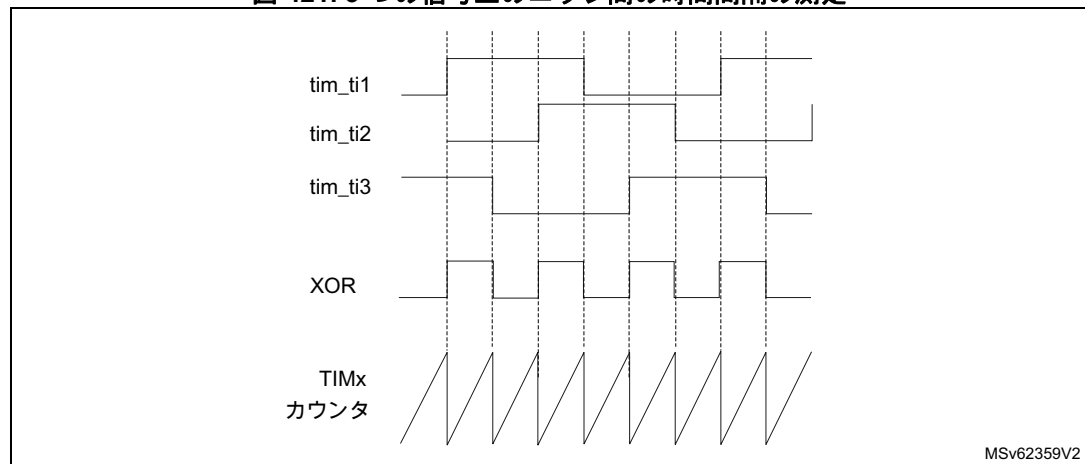
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

38.3.28 タイマ入力 XOR 機能

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン tim_ti1、tim_ti2、および tim_ti3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の図 421 に示すように、2 つの入力信号上のエッジ間の間隔を測定するのに便利です。

図 421. 3 つの信号上のエッジ間の時間間隔の測定



38.3.29 ホールセンサとのインタフェース

これは、高機能制御タイマを使用して PWM 信号を生成し、モータと図 422 で「インタフェースタイマ」と記されている別のタイマ TIMx を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて tim_ti1 入力チャンネル (TIMx_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (tim_ti1、tim_ti2、および tim_ti3) をキャプチャします。

スレーブモードコントローラはリセットモードに設定され、スレーブ入力は tim_ti1f_ed です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャンネル 1 がキャプチャモードで設定され、キャプチャ信号は tim_trc です (図 364: 1434 ページのキャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ) を参照)。キャプチャされた値は、入力の 2 回の変化の間の経過時間に対応し、モータの速度情報を与えます。

「インタフェースタイマ」を出力モードで使用して、(COM イベントをトリガすることによって) 高機能制御タイマのチャンネルの設定を変更するパルスを生成できます。高機能制御タイマは、モータを駆動する PWM 信号を生成するために使用されます。このためには、プログラミングした遅延の後に正パルスが生成されるように (出力比較モードまたは PWM モードで) インタフェースタイマチャンネルをプログラミングする必要があります。このパルスは、tim_trgo 出力を通じて高機能制御タイマに送信されます。

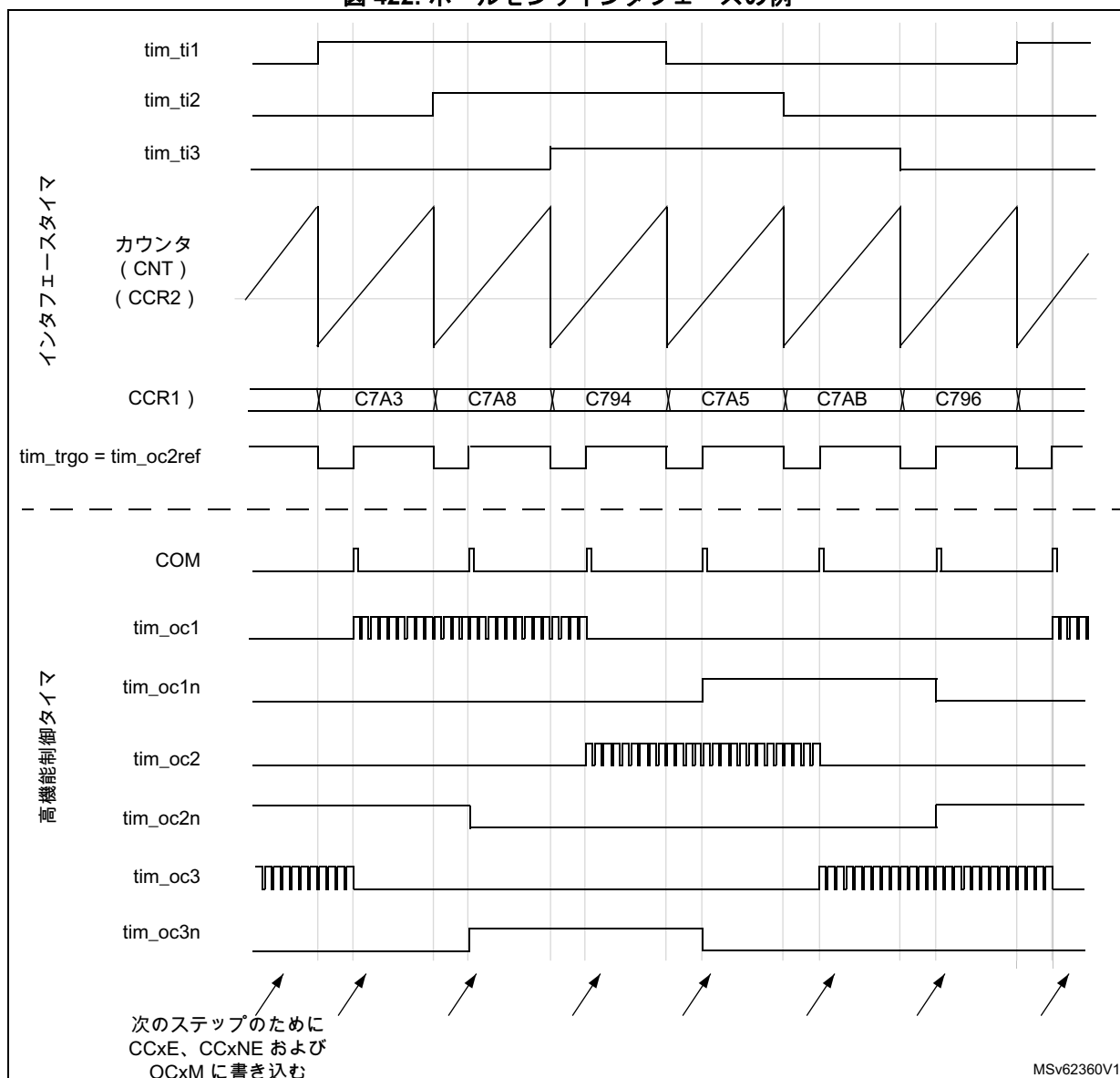
例: TIMx タイマの 1 つに接続されているホール入力が増加するたびに、プログラミングした遅延の後に高機能制御タイマの PWM 設定を変更するとします。

- 3 つのタイマ入力を tim_ti1 入力チャンネルに OR 接続します。このためには、TIMx_CR2 レジスタの TI1S ビットに“1”を書き込みます。
- タイムベースをプログラムします。このためには、TIMx_ARR に最大値を書き込み、tim_ti1 の変化でカウンタがクリアされるようにします。最大カウンタ時間がセンサの 2 回の変化の間の時間より長くなるように、プリスケールを設定します。
- チャンネル 1 をキャプチャモード (tim_trc 選択) にプログラムします。すなわち、TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャンネル 2 を PWM 2 モードにプログラミングし、希望の遅延を指定します。このためには、TIMx_CCMR1 レジスタの OC2M ビットに“111”を、CC2S ビットに“00”を書き込みます。
- tim_trgo 上のトリガ出力として tim_oc2ref を選択します。このためには、TIMx_CR2 レジスタの MMS ビットに“101”を書き込みます。

高機能制御タイマで、トリガ入力として適切な tim_itrx 入力を選択する必要があり、タイマが PWM 信号を生成するようにプログラミングし、キャプチャ/比較制御信号がプリロードされ (TIMx_CR2 レジスタの CCPC=1)、COM イベントがトリガ入力によって制御されなければなりません (TIMx_CR2 レジスタの CCUS=1)。PWM 制御ビット (CCxE、OCxM) は、COM イベント後に次のステップのために書き込まれます (これは、tim_oc2ref の立上がりエッジによって生成される割り込みサブルーチンで行うことができます)。

図 422 に、この例を示します。

図 422. ホールセンサインタフェースの例



38.3.30 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。詳細については、[セクション 39.4.23 : タイマの同期](#)を参照してください。これらのタイマは、いくつかのモードで同期させることができます。すなわち、リセットモード、ゲートモード、トリガモード、リセット+トリガモード、およびゲート+リセットモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

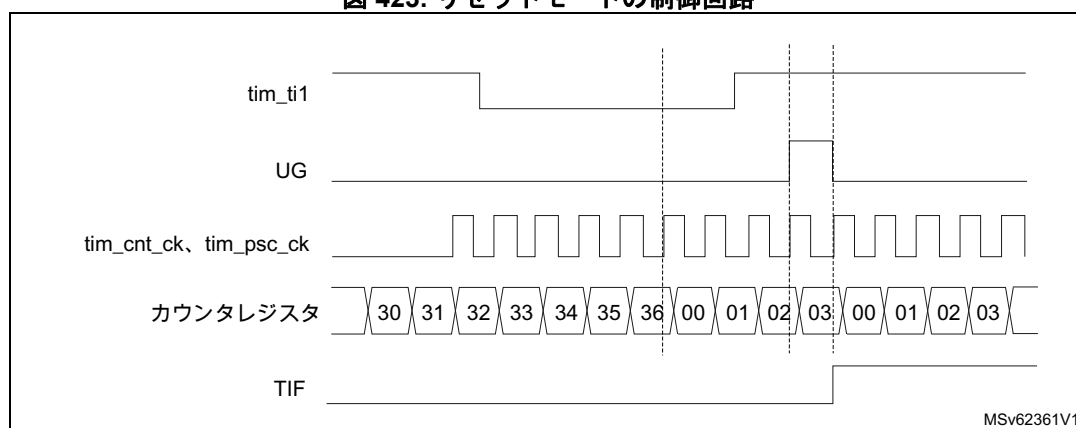
次の例では、tim_ti1 入力の立上がりエッジに応じて、アップカウンタがクリアされます。

- tim_ti1 の立上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないなので、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01）。TIMx_CCER レジスタに CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします（そして、立上がりエッジのみを検出します）。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、tim_ti1 の立上がりエッジまで通常の動作を行います。tim_ti1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ（TIMx_SR レジスタの TIF ビット）、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます（TIMx_DIER レジスタの TIE および TDE ビットに依存）。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。tim_ti1 の立上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_ti1 入力の同期回路によるものです。

図 423. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

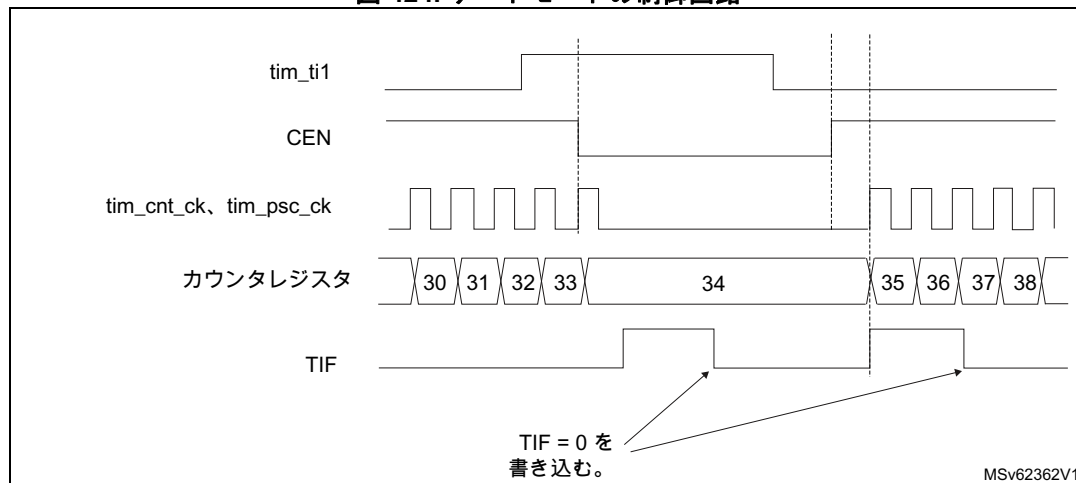
次の例では、アップカウンタは tim_ti1 入力が高レベルのときだけカウントします。

- tim_ti1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないなので、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01 ビット）。TIMx_CCER レジスタで CC1P=1 と CC1NP=0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします（ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

カウンタは、tim_ti1 がローになると内部クロックでカウントを開始して、tim_ti1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

tim_ti1 の立上がりエッジから実際にカウンタが停止するまでの遅延は、tim_ti1 入力の再同期回路によるものです。

図 424. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

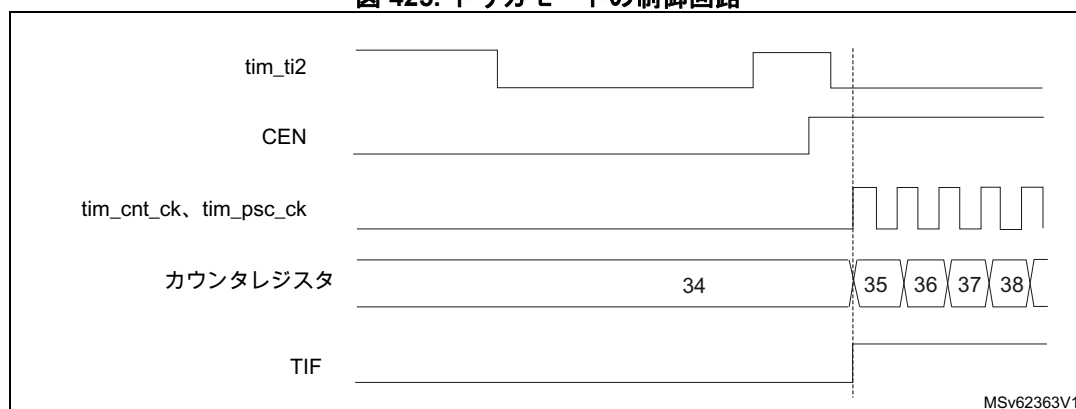
次の例では、アップカウンタは、tim_ti2 入力の立上がりエッジに応じて開始します。

- tim_ti2 の立上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないなので、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=00110 を書き込むことによって、入力ソースとして tim_ti2 を選択します。

tim_ti2 で立上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際にカウンタが開始するまでの遅延は、tim_ti2 入力の再同期回路によるものです。

図 425. トリガモードの制御回路



スレーブモード：リセットモードとトリガモードの組み合わせ

この場合、選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

スレーブモード：ゲートモードとリセットモードの組み合わせ

カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

このモードは、範囲外の PWM 信号 (最大期待値を超えるデューティサイクル) を検出するために使用されます。

スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、tim_etr_in 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、またはトリガモード)。TIMx_SMCR レジスタの TS ビットを通じて tim_trgi として tim_etr_in を選択しないようにしてください。

次の例では、アップカウンタは、tim_ti1 の立上がりエッジが発生すると、tim_etr_in 信号の立上がりエッジのたびにインクリメントされます。

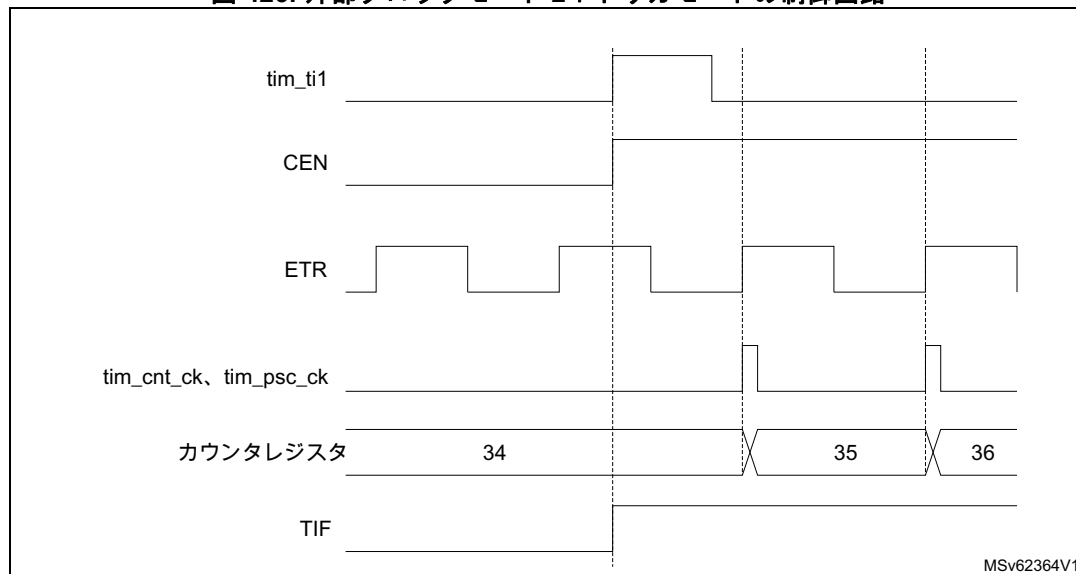
1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000 : フィルタなし
 - ETPS=00 : プリスケール無効
 - ETP=0 : tim_etr_in の立上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。
2. TI1 の立上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケールはトリガには使用されないなので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします (そして、立上がりエッジのみを検出します)。

3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。

tim_ti1 の立上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、その後 tim_etr_in の立上がりエッジでカウントします。

tim_etr_in 信号の立上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_etrp 入力の再同期回路によるものです。

図 426. 外部クロックモード 2+トリガモードの制御回路



注： tim_trgo または tim_trgo2 の信号を受信するスレーブペリフェラル（タイマ、ADC など）のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間はクロック周波数（プリスケアラ）を動作中に変更しないでください。

38.3.31 ADC トリガ

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイベントを生成できます。以下のような内部エッジ検出から発行されたパルスを生成することもできます。

- OC4ref の立上がりおよび立下がりエッジ
- OC5ref の立上がりエッジまたは OC6ref の立下がりエッジ

トリガは、ADC にリダイレクトされる tim_trgo2 内部ラインで発行されます。使用できるイベントは全部で 16 個あり、TIMx_CR2 レジスタの MMS2[3:0] ビットを使用して選択できます。

3 相モータ駆動のアプリケーションの例については、[1452 ページの図 379](#) を参照してください。

注： tim_trgo または tim_trgo2 の信号を受信するスレーブペリフェラル（タイマ、ADC など）のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間はクロック周波数（プリスケアラ）を動作中に変更しないでください。

注： ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

38.3.32 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト（バースト）を開始します。TIMx_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx_DMAR アドレスに対して読出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数（ハーフワード単位またはバイト単位）を示します。

TIMx_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し／書き込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx_CR1

00001 : TIMx_CR2

00010 : TIMx_SMCR

TIMx_DCR レジスタの DBSS[3:0] ビットでは、DMA バースト転送をトリガする割り込みソースを定義します（詳細は[セクション 38.6.29 : TIMx DMA 制御レジスタ \(TIMx_DCR\) \(x = 1, 8\)](#)を参照)。

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 回転送、DBA = 0xE、および DBSS = 1。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタの UDE ビットをセット)。
4. TIMx を有効にします。
5. DMA チャンネルを有効にします。

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 でなければなりません。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

注： null 値を予約済みレジスタに書き込むことができます。

38.3.33 TIM1/TIM8DMA リクエスト

以下の表に示すように、TIM1/TIM8 は DMA リクエストを生成できます。

表 392. DMA リクエスト

DMA リクエスト信号	DMA リクエストあり	有効制御ビット
tim_upd_dma	更新	UDE
tim_cc1_dma	キャプチャ/比較 1	CC1DE
tim_cc2_dma	キャプチャ/比較 2	CC2DE
tim_cc3_dma	キャプチャ/比較 3	CC3DE
tim_cc4_dma	キャプチャ/比較 4	CC4DE
tim_com_dma	転流 (COM)	COMDE
tim_trg_dma	トリガ	TDE

38.3.34 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M33 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止できます。

デバッグモードでの動作は、デバッグサポート (DBG) モジュールでタイマごとに専用の設定ビットを使用してプログラムできます。

安全のため、カウンタが停止すると、出力は無効になります (MOE ビットのリセット時と同じ)。通常強制的にハイインピーダンスにするために、出力を強制的にインアクティブ状態にするか (OSSI ビット = 1)、GPIO コントローラで制御することができます (OSSI ビット = 0)。

詳細については、デバッグサポート (DBG) のセクションを参照してください。

38.4 TIM1/TIM8 低電力モード

表 393. 低消費電力モードが TIM1/TIM8 に与える影響

モード	説明
SLEEP	影響はなく、ペリフェラルはアクティブです。割込みによって、デバイスは SLEEP モードから復帰できます。
STOP	タイマの動作は停止され、レジスタの内容は保持されます。割込みは生成できません。
STANDBY	タイマはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

38.5 TIM1/TIM8 割込み

表 394 に示すように、TIM1/TIM8 は複数の割込みを生成できます。

表 394. 割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベント フラグ	有効制御 ビット	割込みのクリア方法	SLEEP モードの 終了	STOP および STANDBY モードの終了
TIM_UP	更新	UIF	UIE	UIF に 0 を書き込む	可	不可
TIM_CC	キャプチャ/比較 1	CC1IF	CC1IE	CC1IF に 0 を書き込む	可	不可
	キャプチャ/比較 2	CC2IF	CC2IE	CC2IF に 0 を書き込む	可	不可
	キャプチャ/比較 3	CC3IF	CC3IE	CC3IF に 0 を書き込む	可	不可
	キャプチャ/比較 4	CC4IF	CC4IE	CC4IF に 0 を書き込む	可	不可
TIM_TRG_COM	転流 (COM)	COMIF	COMIE	COMIF に 0 を書き込む	可	不可
	トリガ	TIF	TIE	TIF に 0 を書き込む	可	不可
TIM_DIR_IDX	インデックス	IDXF	IDXIE	IDXF に 0 を書き込む	可	不可
	方向	DIRF	DIRIE	DIRF に 0 を書き込む	可	不可
TIM_BRK	ブレーク	BIF	BIE	BIF に 0 を書き込む	可	不可
	Break2	B2IF		B2IF に 0 を書き込む	可	不可
	システムブレーク	SBIF		SBIF に 0 を書き込む	可	不可
TIM_IERR	インデックスエラー	IERRF	IERRIE	IERRF に 0 を書き込む	可	不可
TIM_TER	遷移エラー	TERRF	TERRIE	TERRF に 0 を書き込む	可	不可

38.6 TIM1/TIM8 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2](#) を参照してください。

38.6.1 TIMx 制御レジスタ 1 (TIMx_CR1) (x = 1、8)

アドレスオフセット : 0x000

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デッドタイムジェネレータとデジタルフィルタ (tim_etr_in、tim_tix) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

00 : $t_{DTS} = t_{tim_ker_ck}$

01 : $t_{DTS} = 2 * t_{tim_ker_ck}$

10 : $t_{DTS} = 4 * t_{tim_ker_ck}$

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]** : センターアラインモード選択

00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01 : センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10 : センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントアップしているときのみセットされます。

11 : センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注 : カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

- 0 : カウンタはアップカウンタとして使用されます。
- 1 : カウンタはダウンカウンタとして使用されます。

注 : このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

- 1 : カウンタオーバーフロー/アンダーフローのみが更新割込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

- 1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

注 : 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

38.6.2 TIMx 制御レジスタ 2 (TIMx_CR2) (x = 1, 8)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MMS[3]	Res.	MMS2 [3:0]				Res.	OIS6	Res.	OIS5
						rw		rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **MMS2[3:0]** : マスタモード選択 2

これらのビットにより、選択される同期 (tim_trgo2) について、ADC に情報を送信できるようになります。組み合わせは、次のとおりです。

0000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (tim_trgo2) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、tim_trgo2 信号は実際のリセットより遅延します。

0001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (tim_trgo2) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理積 (AND) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、tim_trgo2 には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

0010 : **更新** - 更新イベントがトリガ出力 (tim_trgo2) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

0011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (tim_trgo2)。

0100 : **比較** - tim_oc1refc 信号がトリガ出力 (tim_trgo2) として使用されます。

0101 : **比較** - tim_oc2refc 信号がトリガ出力 (tim_trgo2) として使用されます。

0110 : **比較** - tim_oc3refc 信号がトリガ出力 (tim_trgo2) として使用されます。

0111 : **比較** - tim_oc4refc 信号がトリガ出力 (tim_trgo2) として使用されます。

1000 : **比較** - tim_oc5refc 信号がトリガ出力 (tim_trgo2) として使用されます。

1001 : **比較** - tim_oc6refc 信号がトリガ出力 (tim_trgo2) として使用されます。

1010 : **パルス比較** - tim_oc4refc の立上がりエッジまたは立下がりエッジによって、tim_trgo2 にパルスが生成されます。

1011 : **パルス比較** - tim_oc6refc の立上がりエッジまたは立下がりエッジによって、tim_trgo2 にパルスが生成されます。

1100 : **パルス比較** - tim_oc4refc または tim_oc6refc の立上がりエッジによって、tim_trgo2 にパルスが生成されます。

1101 : **パルス比較** - tim_oc4refc の立上がりエッジまたは tim_oc6refc の立下がりエッジによって、tim_trgo2 にパルスが生成されます。

1110 : **パルス比較** - tim_oc5refc または tim_oc6refc の立上がりエッジによって、tim_trgo2 にパルスが生成されます。

1111 : **パルス比較** - tim_oc5refc の立上がりエッジまたは tim_oc6refc の立下がりエッジによって、tim_trgo2 にパルスが生成されます。

注 : スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

ビット 19 予約済みであり、リセット値に保持する必要があります。

- ビット 18 **OIS6** : 出力アイドル状態 6 (tim_oc6 出力)
OIS1 ビットの説明を参照してください。
- ビット 17 予約済みであり、リセット値に保持する必要があります。
- ビット 16 **OIS5** : 出力アイドル状態 5 (tim_oc5 出力)
OIS1 ビットの説明を参照してください。
- ビット 15 **OIS4N** : 出力アイドル状態 4 (tim_oc4n 出力)
OIS1N ビットの説明を参照してください。
- ビット 14 **OIS4** : 出力アイドル状態 4 (tim_oc4 出力)
OIS1 ビットの説明を参照してください。
- ビット 13 **OIS3N** : 出力アイドル状態 3 (tim_oc3n 出力)
OIS1N ビットの説明を参照してください。
- ビット 12 **OIS3** : 出力アイドル状態 3 (tim_oc3n 出力)
OIS1 ビットの説明を参照してください。
- ビット 11 **OIS2N** : 出力アイドル状態 2 (tim_oc2n 出力)
OIS1N ビットの説明を参照してください。
- ビット 10 **OIS2** : 出力アイドル状態 2 (tim_oc2 出力)
OIS1 ビットの説明を参照してください。
- ビット 9 **OIS1N** : 出力アイドル状態 1 (tim_oc1n 出力)
0 : MOE=0 のとき、デッドタイム後に tim_oc1n=0
1 : MOE=0 のとき、デッドタイム後に tim_oc1n=1
注 : このビットは、**LOCK** レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの **LOCK** ビット)。
- ビット 8 **OIS1** : 出力アイドル状態 1 (tim_oc1 出力)
0 : MOE=0 のとき、tim_oc1=0 (デッドタイム後に)
1 : MOE=0 のとき、tim_oc1=1 (デッドタイム後に)
注 : このビットは、**LOCK** レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの **LOCK** ビット)。
- ビット 7 **TI1S** : tim_ti1 選択
0 : tim_ti1_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます。
1 : tim_ti1_in[15:0]、tim_ti2_in[15:0]、および tim_ti3_in[15:0] マルチプレクサ出力が XOR 演算されて tim_ti1 入力に接続されます。

ビット 25、6:4 **MMS[3:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択します (tim_trgo)。組み合わせは、次のとおりです。

0000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (tim_trgo) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、tim_trgo 信号は実際のリセットより遅延します。

0001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (tim_trgo) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理積 (AND) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、tim_trgo には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

0010 : **更新** - 更新イベントがトリガ出力 (tim_trgo) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

0011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (tim_trgo)。

0100 : **比較** - tim_oc1refc 信号がトリガ出力 (tim_trgo) として使用されます。

0101 : **比較** - tim_oc2refc 信号がトリガ出力 (tim_trgo) として使用されます。

0110 : **比較** - tim_oc3refc 信号がトリガ出力 (tim_trgo) として使用されます。

0111 : **比較** - tim_oc4refc 信号がトリガ出力 (tim_trgo) として使用されます。

1000 : **エンコーダクロック出力** - エンコーダクロック信号がトリガ出力 (tim_trgo) として使用されます。このコードは次の SMS[3:0] 値に対して有効です。0001、0010、0011、1010、1011、1100、1101、1110、1111。その他の SMS[3:0] コードは許されず、予想外の動作につながる可能性があります。

その他のコードは予約済み

注 : スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ/比較制御ビットがプリロードされているときに (CCPC=1)、COMG ビットをセットすることによって、または tim_trgi の立上がりエッジで更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書込みの後、これらのビットは、転流イベント (COM) が発生したときのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは tim_trgi で立上がりエッジが検出されたとき)。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

38.6.3 TIMx スレーブモード制御レジスタ (TIMx_SMCR) (x = 1、8)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SMSPS	SMSPE	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
						rw	rw			rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **SMSPS** : SMS プリロードソース

このビットは、プリロードからアクティブへの SMS[3:0] ビットフィールドの転送をトリガするイベントのいずれかを選択します。

0 : 転送は、タイマの更新イベントによってトリガされます。

1 : 転送は、インデックスイベントによってトリガされます。

ビット 24 **SMSPE** : SMS プリロードイネーブル

このビットは、SMS[3:0] ビットフィールドをプリロードするかどうかを選択します。

0 : SMS[3:0] ビットフィールドはプリロードされません。

1 : SMS[3:0] プリロードは有効です。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **TS[4:3]** : トリガ選択 - ビット 4:3

TS[2:0] 説明を参照 - ビット 6:4

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **ETP** : 外部トリガ極性

このビットは、tim_etr_in と tim_etr_in のいずれがトリガ動作に使用されるかを選択します。

0 : tim_etr_in は反転されず、ハイレベルまたは立上がりエッジでアクティブです。

1 : tim_etr_in は反転され、ローレベルまたは立下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0 : 外部クロックモード 2 は無効です。

1 : 外部クロックモード 2 は有効です。カウンタは、tim_etrf 信号のアクティブエッジによってクロック供給されます。

注 : ECE ビットをセットすることは、tim_trgi が tim_etrf に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=00111)。

外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、tim_trgi を tim_etrf に接続することはできません (TS ビットが 00111 でないことが必要)。

外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は tim_etrf です。

ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 `tim_etrp` の周波数は、TIMxCLK 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、`tim_etrp` 周波数を低減できます。これは、高速な外部クロックを `tim_etr_in` に入力するときに役立ちます。

00 : プリスケアラオフ

01 : `tim_etr_in` 周波数は 2 分周されます。

10 : `tim_etr_in` 周波数は 4 分周されます。

11 : `tim_etr_in` 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、`tim_etrp` 信号をサンプルする周波数と、`tim_etrp` に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはペントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (`tim_trgi`) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (`tim_trgo` を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS[2:0]** : トリガ選択

このビットフィールドは TS[4:3] ビットと結合されます。

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

00000 : 内部トリガ 0 (tim_itr0)
00001 : 内部トリガ 1 (tim_itr1)
00010 : 内部トリガ 2 (tim_itr2)
00011 : 内部トリガ 3 (tim_itr3)
00100 : tim_ti1 エッジ検出回路 (tim_ti1f_ed)
00101 : フィルタタイマ入力 1 (tim_ti1fp1)
00110 : フィルタタイマ入力 2 (tim_ti2fp2)
00111 : 外部トリガ入力 (tim_etr1)
01000 : 内部トリガ 4 (tim_itr4)
01001 : 内部トリガ 5 (tim_itr5)
01010 : 内部トリガ 6 (tim_itr6)
01011 : 内部トリガ 7 (tim_itr7)
01100 : 内部トリガ 8 (tim_itr8)
01101 : 内部トリガ 9 (tim_itr9)
01110 : 内部トリガ 10 (tim_itr10)
01111 : 内部トリガ 11 (tim_itr11)
10000 : 内部トリガ 12 (tim_itr12)
10001 : 内部トリガ 13 (tim_itr13)
10010 : 内部トリガ 14 (tim_itr14)
10011 : 内部トリガ 15 (tim_itr15)

その他 : 予約済み

各タイマでの tim_itrx の詳細については、表 381 : 内部トリガ接続を参照してください。

注 : 遷移時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき（たとえば、SMS = 000 のとき）にのみ変更しなければなりません。

ビット 3 **OCCS** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

0 : tim_ocref_clr_int は tim_ocref_clr 入力に接続されます。

1 : tim_ocref_clr_int は tim_etr1 に接続されます。

ビット 16、2:0 **SMS[3:0]** : スレーブモード選択

外部信号が選択されると、トリガ信号 (tim_trgi) のアクティブエッジが、外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : 直交エンコーダモード 1、x2 モード - カウンタは、tim_ti2fp2 のレベルに応じて、tim_ti1fp1 のエッジでカウントアップ/ダウンします。

0010 : 直交エンコーダモード 2、x2 モード - カウンタは、tim_ti1fp1 のレベルに応じて、tim_ti2fp2 のエッジでカウントアップ/ダウンします。

0011 : 直交エンコーダモード 3、x4 モード - カウンタは、他の入力のレベルに応じて、tim_ti1fp1 と tim_ti2fp2 の両方のエッジでカウントアップ/ダウンします。

0100 : リセットモード - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ tim_trgi の立上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (tim_trgi) の立上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1001 : ゲートモードとトリガモードの組み合わせ - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

1010 : エンコーダモード : クロックプラス方向、x2 モード。

1011 : エンコーダモード : クロックプラス方向、x1 モード。tim_ti2fp2 エッジ検出は CC2P によって設定されます。

1100 : エンコーダモード : 方向クロック、x2 モード。

1101 : エンコーダモード : 方向クロック、x1 モード。tim_ti1fp1 と tim_ti2fp2 エッジ検出は CC1P と CC2P によって設定されます。

1110 : 直交エンコーダモード : x1 モード、tim_ti1fp1 エッジのみでカウント、エッジ検出は CC1P によって設定されます。

1111 : 直交エンコーダモード : x1 モード、tim_ti2fp2 エッジのみでカウント、エッジ検出は CC2P によって設定されます。

注 : トリガ入力として tim_ti1f_ed が選択されている場合 (TS=00100)、ゲートモードを使用することはできません。tim_ti1f_ed は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注 : tim_trgo または tim_trgo2 の信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

38.6.4 TIMx DMA／割込み有効レジスタ (TIMx_DIER) (x = 1、8)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR IE	IERRIE	DIRIE	IDXIE	Res.	Res.	Res.	Res.
								rW	rW	rW	rW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TERRIE** : 遷移エラー割込みイネーブル

0 : 遷移エラー割込みは無効です。

1 : 遷移エラー割込みは有効です。

ビット 22 **IERRIE** : インデックスエラー割込みイネーブル

0 : インデックスエラー割込みは無効です。

1 : インデックスエラー割込みは有効です。

ビット 21 **DIRIE** : 方向変更割込みイネーブル

0 : 方向変更割込みは無効です。

1 : 方向変更割込みは有効です。

ビット 20 **IDXIE** : インデックス割込みイネーブル

0 : インデックス割込みは無効です。

1 : インデックス変更割込みは有効です。

ビット 19:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

0 : COM DMA リクエストは無効です。

1 : COM DMA リクエストは有効です。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

- ビット 7 **BIE** : ブレーク割込みイネーブル
 0 : ブレーク割込みは無効です。
 1 : ブレーク割込みは有効です。
- ビット 6 **TIE** : トリガ割込みイネーブル
 0 : トリガ割込みは無効です。
 1 : トリガ割込みは有効です。
- ビット 5 **COMIE** : COM 割込み有効
 0 : COM 割込み無効
 1 : COM 割込み有効
- ビット 4 **CC4IE** : キャプチャ/比較 4 割込みイネーブル
 0 : CC4 割込みは無効です。
 1 : CC4 割込みは有効です。
- ビット 3 **CC3IE** : キャプチャ/比較 3 割込みイネーブル
 0 : CC3 割込みは無効です。
 1 : CC3 割込みは有効です。
- ビット 2 **CC2IE** : キャプチャ/比較 2 割込みイネーブル
 0 : CC2 割込みは無効です。
 1 : CC2 割込みは有効です。
- ビット 1 **CC1IE** : キャプチャ/比較 1 割込みイネーブル
 0 : CC1 割込みは無効です。
 1 : CC1 割込みは有効です。
- ビット 0 **UIE** : 更新割込みイネーブル
 0 : 更新割込みは無効です。
 1 : 更新割込みは有効です。

38.6.5 TIMx ステータスレジスタ (TIMx_SR) (x = 1、8)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERRF	IERRF	DIRF	IDXF	Res.	Res.	CC6IF	CC5IF
								rc_w0	rc_w0	rc_w0	rc_w0			rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TERRF** : 遷移エラー割込みフラグ

このフラグは、エンコーダモードで遷移エラーが検出されたときに、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : エンコーダ遷移エラーは検出されていません。

1 : エンコーダ遷移エラーが検出されています。

ビット 22 **IERRF** : インデックスエラー割込みフラグ

このフラグは、インデックスエラーが検出されたときに、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : インデックスエラーは検出されていません。

1 : インデックスエラーが検出されています。

ビット 21 **DIRF** : 方向変更割込みフラグ

このフラグは、エンコーダモードで方向が変わった (TIMx_CR の DIR ビット値が変わった) ときに、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : 方向の変更なし
1 : 方向の変更

ビット 20 **IDXF** : インデックス割込みフラグ

このフラグは、インデックスイベントが検出されたときに、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : インデックスイベントは発生していません。
1 : インデックスイベントが発生しています。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC6IF** : 比較 6 割込みフラグ

CC1IF の説明を参照してください。

注： チャンネル 6 は出力としてのみ設定できます。

ビット 16 **CC5IF** : 比較 5 割込みフラグ

CC1IF の説明を参照してください。

注： チャンネル 5 は出力としてのみ設定できます。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **SBIF** : システムブレーク割込みフラグ

このフラグは、システムブレーク入力がアクティブになると、ハードウェアによってセットされます。システムブレーク入力 that アクティブでない場合、ソフトウェアによってクリアできます。

PWM 動作をリスタートするには、このフラグをリセットする必要があります。

0 : ブレークイベントは発生していません。
1 : システムブレーク入力 that アクティブレベルが検出されました。TIMx_DIER レジスタの BIE=1 の場合、割込みが生成されます。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 **B2IF** : ブレーク 2 割込みフラグ

このフラグは、ブレーク 2 入力 that アクティブになると、ハードウェアによってセットされます。ブレーク 2 入力 that アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。
1 : ブレーク 2 入力 that アクティブレベルが検出されました。TIMx_DIER レジスタの BIE=1 の場合、割込みが生成されます。

ビット 7 BIF : ブレーク割込みフラグ

このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力アクティブレベルが検出されました。TIMx_DIER レジスタの BIE=1 の場合、割込みが生成されます。

ビット 6 TIF : トリガ割込みフラグ

このフラグは、TRG トリガイベント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、tim_trgi 入力アクティブエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

0 : トリガイベントは発生していません。

1 : トリガ割込みが保留中です。

ビット 5 COMIF : COM 割込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます（キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき）。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割込みがペンディング中です。

ビット 4 CC4IF : キャプチャ/比較 4 割込みフラグ

CC1IF の説明を参照してください。

ビット 3 CC3IF : キャプチャ/比較 3 割込みフラグ

CC1IF の説明を参照してください。

ビット 2 CC2IF : キャプチャ/比較 2 割込みフラグ

CC1IF の説明を参照してください。

ビット 1 CC1IF : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって（入力キャプチャモードまたは出力比較モード）、または TIMx_CCR1 レジスタを読み出すことによって（入力キャプチャモードのみ）クリアされます。

0 : 比較一致/入力キャプチャは発生していません。

1 : 比較一致または入力キャプチャが発生しました。

チャネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた（TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ検知に従って、IC1 でエッジが検出された）ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフローまたはアンダーフロー（繰り返しカウンタ=0 の場合の更新）、および TIMx_CR1 レジスタの UDIS=0 の場合。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき（[セクション 38.6.3: TIMx スレーブモード制御レジスタ \(TIMx_SMCR\) \(x = 1、8\)](#) を参照）。

38.6.6 TIMx イベント生成レジスタ (TIMx_EGR) (x = 1、8)

アドレスオフセット : 0x014

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **B2G** : ブレーク 2 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレーク 2 イベントが生成されます。MOE ビットがクリアされ、B2IF フラグがセットされます。有効な場合は、関連する割込みが発生します。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ/比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCxE、CCxNE、OCxM ビットの更新 (CCPC ビットが設定されている場合)。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4 **CC4G** : キャプチャ/比較 4 生成

CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ／比較 3 生成

CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ／比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウンタアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウンタダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

38.6.7 TIMx キャプチャ／比較モードレジスタ 1 (TIMx_CCMR1) (x = 1, 8)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード (このセクション) または出力比較モード (次のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます (たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F [3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

入力キャプチャモード :

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F[3:0]** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャネルは出力として設定されます。

01 : CC2 チャネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : CC2 チャネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC2S ビットは、チャネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$, $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$, $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$, $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$, $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$, $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$, $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$, $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$, $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$, $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$, $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$, $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$, $N = 8$

ビット 3:2 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャネルは出力として設定されます。

01 : CC1 チャネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC1S ビットは、チャネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

38.6.8 TIMx キャプチャ／比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 1、8)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M [2:0]			OC2 PE	OC2 FE	CC2S [1:0]		OC1 CE	OC1M [2:0]			OC1 PE	OC1 FE	CC1S [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード :

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 24、14:12 **OC2M[3:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : tim_oc1ref は tim_ocref_clr_int 信号の影響を受けません。

1 : tim_oc1ref は tim_ocref_clr_int 信号のハイレベルが検出されるとクリアされます (tim_ocref_clr 入力または tim_etrif 入力)。

ビット 16、6:4 OC1M[3:0] : 出力比較 1 モード

これらのビットは、tim_oc1 および tim_oc1n が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 および tim_oc1n のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。このモードは、タイマをソフトウェアのタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードが有効になると、出力は停止状態になる前の状態（アクティブまたはインアクティブ）を維持します。

0001 : 一致時にチャンネル 1 をアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ / 比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 1 をインアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ / 比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制非アクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (tim_oc1ref="0") に、そうでない場合はアクティブ (tim_oc1ref="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 OR です。

1101 : 組み合わせ PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 AND です。

1110 : 非対称 PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は、カウンタがカウントアップするときに tim_oc1ref を出力し、カウントダウンするときに tim_oc2ref を出力します。

1111 : 非対称 PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に動作します。tim_oc1refc は、カウンタがカウントアップするときに tim_oc1ref を出力し、カウントダウンするときに tim_oc2ref を出力します。

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

注 : PWM モードでは、比較結果が変化したとき、出力比較モードが「停止」モードから「PWM」モードに切り替わったとき、および出力比較モードが「強制アクティブ/インアクティブ」モードから「PWM」モードに切り替わったときに、OCREF のレベルが変化します。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとります。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注： これらのビットは、**LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。**

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、トリガイイベントとタイマ出力の遷移の間の待ち時間を短縮します。開始トリガの後、できるだけ早く出力パルスを開始するには、1 パルスモード (TIMx_CR1 レジスタの OPM ビットのセット) で使用する必要があります。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

38.6.9 TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2) (x = 1、8)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード (このセクション) または出力比較モード (次のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます (たとえば、チャンネル 3 は入力キャプチャモードで、チャンネル 4 は出力比較モード)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F [3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F [3:0]				IC3PSC [1:0]		CC3S [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC4F[3:0]** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC[1:0]** : 入力キャプチャ 4 プリスケアラ

ビット 9:8 **CC4S[1:0]** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti4 に配置されます。

10 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti3 に配置されます。

11 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC4S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC3F[3:0]** : 入力キャプチャ 3 フィルタビット 3:2 **IC3PSC[1:0]** : 入力キャプチャ 3 プリスケアラビット 1:0 **CC3S[1:0]** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti3 に配置されます。

10 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti4 に配置されます。

11 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC3S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

38.6.10 TIMx キャプチャ／比較モードレジスタ 2 [オルタネート] (TIMx_CCMR2) (x = 1、8)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（たとえば、チャンネル 3 は入力キャプチャモードで、チャンネル 4 は出力比較モード）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							r/w								r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M [2:0]			OC4 PE	OC4 FE	CC4S [1:0]		OC3 CE	OC3M [2:0]			OC3 PE	OC3 FE	CC3S [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 24、14:12 **OC4M[3:0]** : 出力比較 4 モード

OC3M[3:0] ビットの説明を参照してください。

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S[1:0]** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti4 に配置されます。

10 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti3 に配置されます。

11 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 16、6:4 OC3M[3:0] : 出力比較 3 モード

これらのビットは、tim_oc3 および tim_oc3n が導き出される出力基準信号 tim_oc3ref の動作を定義します。tim_oc3ref はアクティブハイですが、tim_oc3 および tim_oc3n のアクティブレベルは CC3P および CC3NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx_CCR3 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

0001 : 一致時にチャンネル 3 をアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 3 (TIMx_CCR3) と一致すると、tim_oc3ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 3 をインアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 3 (TIMx_CCR3) と一致すると、tim_oc3ref 信号は強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR3 のとき、tim_oc3ref は反転します。

0100 : 強制非アクティブレベル - tim_oc3ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc3ref は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 3 は、TIMx_CNT < TIMx_CCR3 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 3 は、TIMx_CNT > TIMx_CCR3 の場合はインアクティブ (tim_oc3ref="0") に、そうでない場合はアクティブ (tim_oc3ref="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 3 は、TIMx_CNT < TIMx_CCR3 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 3 は、TIMx_CNT > TIMx_CCR3 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 比較によるパルス : TIMx_ECR の PWPRSC[2:0] と PW[7:0] ビットフィールドのプログラミングに従って、CCR3 一致イベント時に、パルスが tim_oc3ref に生成されます。

1011 : 方向の出力。tim_oc3ref 信号は DIR ビットのコピーによって上書きされます。

1100 : 組み合わせ PWM モード 1 - tim_oc3ref は、PWM モード 1 と同様に挙動します。tim_oc3refc は tim_oc3ref と tim_oc4ref の論理 OR です。

1101 : 組み合わせ PWM モード 2 - tim_oc3ref は、PWM モード 2 と同様に動作します。tim_oc3refc は tim_oc3ref と tim_oc4ref の論理 AND です。

1110 : 非対称 PWM モード 1 - tim_oc3ref は、PWM モード 1 と同様に挙動します。tim_oc3refc は、カウンタがカウントアップするときに tim_oc3ref を出力し、カウントダウンするときに tim_oc4ref を出力します。

1111 : 非対称 PWM モード 2 - tim_oc3ref は、PWM モード 2 と同様に動作します。tim_oc3refc は、カウンタがカウントアップするときに tim_oc3ref を出力し、カウントダウンするときに tim_oc4ref を出力します。

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

注 : PWM モードでは、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、OC3M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとります。

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S[1:0]** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti3 に配置されます。

10 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti4 に配置されます。

11 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

38.6.11 TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 1, 8)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rW	rW			rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **CC6P** : キャプチャ/比較 6 出力極性

CC1P の説明を参照してください。

ビット 20 **CC6E** : キャプチャ/比較 6 出力イネーブル

CC1E の説明を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC5P** : キャプチャ/比較 5 出力極性

CC1P の説明を参照してください。

ビット 16 **CC5E** : キャプチャ/比較 5 出力イネーブル

CC1E の説明を参照してください。

ビット 15 **CC4NP** : キャプチャ/比較 4 相補出力極性

CC1NP の説明を参照してください。

ビット 14 **CC4NE** : キャプチャ/比較 4 相補出力イネーブル

CC1NE の説明を参照してください。

ビット 13 **CC4P** : キャプチャ/比較 4 出力極性

CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ/比較 4 出力イネーブル

CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ/比較 3 相補出力極性

CC1NP の説明を参照してください。

ビット 10 **CC3NE** : キャプチャ/比較 3 相補出力イネーブル

CC1NE の説明を参照してください。

ビット 9 **CC3P** : キャプチャ / 比較 3 出力極性
CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ / 比較 3 出力イネーブル
CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ / 比較 2 相補出力極性
CC1NP の説明を参照してください。

ビット 6 **CC2NE** : キャプチャ / 比較 2 相補出力イネーブル
CC1NE の説明を参照してください。

ビット 5 **CC2P** : キャプチャ / 比較 2 出力極性
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ / 比較 2 出力イネーブル
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ / 比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : tim_oc1n はアクティブハイです。

1 : tim_oc1n はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、tim_ti1fp1 と tim_ti2fp1 の極性を定義するために CC1P と組み合わせて使用されます。
CC1P の説明を参照してください。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力イネーブル

0 : オフ - tim_oc1n はアクティブではありません。tim_oc1n のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - tim_oc1n 信号は、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

0 : OC1 のアクティブハイ (出力モード) / エッジ感度を選択 (入力モード、下記を参照)

1 : OC1 のアクティブロー (出力モード) / エッジ感度を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立上がりエッジ。この回路は TIxFP1 の立上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立下がりエッジ。この回路は TIxFP1 の立下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されます (ゲートモードまたはエンコーダモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ/この回路は TIxFP1 の立上がりエッジと立下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 **CC1E** : キャプチャ / 比較 1 出力イネーブル

0 : キャプチャモードは無効/OC1 は非アクティブ (下記を参照)。

1 : キャプチャモードは有効/OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが出力として設定されている場合、OC1 レベルは、CC1E ビットの状態にかかわらず、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1NE ビットに依存します。詳細については、表 395 を参照してください。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

表 395. ブレーク機能を持つ相補 tim_ocx および tim_ocxn チャンネルの出力制御ビット

制御ビット					出力状態 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	tim_ocx 出力状態	tim_ocxn 出力状態
1	X	X	0	0	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocx=0、tim_ocxn=0	
		0	0	1	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocx=0	tim_ocxref + 極性 tim_ocxn = tim_ocxref xor CCxNP
		0	1	0	tim_ocxref + 極性 tim_ocx=tim_ocxref xor CCxP	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocxn=0
		X	1	1	OCREF + 極性 + デッドタイム	OCREF に対する相補（OCREF ではなく）+ 極性 + デッドタイム
		1	0	1	オフ状態（インアクティブ状態で出力有効） tim_ocx=CCxP	tim_ocxref + 極性 tim_ocxn = tim_ocxref xor CCxNP
		1	1	0	tim_ocxref + 極性 tim_ocx=tim_ocxref xor CCxP	オフ状態（インアクティブ状態で出力有効） tim_ocxn=CCxNP
0	0	X	X	X	出力無効（タイマによって駆動されない：ハイインピーダンス）。	
	1		0	0		
			0	1		
			1	0		
				1	1	クロックが存在する場合、(tim_brk がトリガされた場合にのみ有効)：アクティブな状態で OISx と OISxN が OCX と tim_ocxn にそれぞれ対応しないとみなされる場合（ハーフブリッジ設定でスイッチを駆動した場合に短絡の原因となる）、デッドタイム後に tim_ocx=OISx および tim_ocxn=OISxN となります。 注： tim_brk2 は OSSI = OSSR = 1 の場合にのみ使用できます。

1. チャンネルの両方の出力が使用されないとき (GPIO が制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP はクリアされたままではなりません。

注 : 相補 tim_ocx および tim_ocxn チャンネルに接続されている外部入出力ピンの状態は、tim_ocx および tim_ocxn チャンネルの状態と、GPIO レジスタに依存します。

38.6.12 TIMx カウンタ (TIMx_CNT) (x = 1、8)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー。TIMxCR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分は得られません。

38.6.13 TIMx プリスケアラ (TIMx_PSC) (x = 1、8)

アドレスオフセット : 0x028

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 ($f_{tim_cnt_ck}$) は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

38.6.14 TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 1、8)

アドレスオフセット : 0x02C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 38.3.3 : 1414 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、自動再ロード値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

38.6.15 TIMx 繰返しカウンタレジスタ (TIMx_RCR) (x = 1、8)

アドレスオフセット : 0x030

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REP[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **REP[15:0]** : 繰返しカウンタの再ロード値

このビットフィールドによって、プリロードレジスタが有効なときの比較レジスタの更新レート (プリロードレジスタからアクティブレジスタへの周期的な転送) を定義します。また、更新割込み生成の頻度も定義します (この割込みが有効な場合)。

繰返しダウンカウンタがゼロに達すると、更新イベントが生成され、REP 値からカウントをリスタートします。繰返しカウンタは、繰返し更新イベント UEV でのみ、REP 値で再ロードされるので、TIMx_RCR レジスタへの書込みは、次の繰返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) は次のことを意味します。

–エッジアラインモードでは、PWM 周期の数

–センターアラインモードでは、PWM の 1/2 周期の数

38.6.16 TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 1、8)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR1[19:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 : CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 : CR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] にキャプチャ値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] にキャプチャ値を保持します。CCR1[3:0] ビットはリセットされています。

38.6.17 TIMx キャプチャ／比較レジスタ 2 (TIMx_CCR2) (x = 1、8)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR2[19:0] : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合: CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc2 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] に比較値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] に整数部を保持します。CCR2[3:0] ビットフィールドにはディザリング部分が含まれます。

CC2 チャンネルが入力として設定されている場合: CCR2 は、最後の入力キャプチャ 2 イベント (tim_ic2) によって転送されたカウンタ値です。TIMx_CCR2 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] にキャプチャ値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] にキャプチャ値を保持します。CCR2[3:0] ビットはリセットされています。

38.6.18 TIMx キャプチャ／比較レジスタ 3 (TIMx_CCR3) (x = 1、8)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR3[19:0]** : キャプチャ / 比較値

CC3 チャンネルが出力として設定されている場合 : CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc3 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR3[15:0] に比較値を保持します。CCR3[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[19:4] に整数部を保持します。CCR3[3:0] ビットフィールドにはディザリング部分が含まれます。

CC3 チャンネルが入力として設定されている場合 : CCR3 は、最後の入力キャプチャ 3 イベント (tim_ic3) によって転送されたカウンタ値です。TIMx_CCR3 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR3[15:0] にキャプチャ値を保持します。CCR3[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[19:4] にキャプチャを保持します。CCR3[3:0] ビットはリセットされています。

38.6.19 TIMx キャプチャ／比較レジスタ 4 (TIMx_CCR4) (x = 1、8)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR4[19:0] : キャプチャ / 比較値

CC4 チャンネルが出力として設定されている場合 : CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、tim_oc4 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR4[15:0] に比較値を保持します。CCR4[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[19:4] に整数部を保持します。CCR4[3:0] ビットフィールドにはディザリング部分が含まれます。

CC4 チャンネルが入力として設定されている場合 : CCR4 は、最後の入力キャプチャ 4 イベント (tim_ic4) によって転送されたカウンタ値です。TIMx_CCR4 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR4[15:0] にキャプチャ値を保持します。CCR4[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[19:4] にキャプチャを保持します。CCR4[3:0] ビットはリセットされています。

38.6.20 TIMx ブレークおよびデッドタイムレジスタ (TIMx_BDTR) (x = 1, 8)

アドレスオフセット : 0x044

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F [3:0]				BKF[3:0]			
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

注： ビット BKBID/BK2BID/BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **BK2BID** : ブレーク 2 双方向
BKBID の説明を参照してください

ビット 28 **BKBID** : ブレーク双方向
0 : ブレーク入力 tim_brk は入力モードです。
1 : ブレーク入力 tim_brk は双方向モードです。
双方向モード (BKBID ビットが 1 にセット) では、ブレーク入力が入力モードとオープンドレイン出力モード両方で設定されます。アクティブなブレークイベントで、ブレーク入力の低ロジックレベルをアサートし、外部デバイスに対する内部ブレークイベントを表します。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注： このビットへの書込み操作では、書込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 27 **BK2DSRM** : ブレーク 2 解除
BKDSRM の説明を参照してください

ビット 26 **BKDSRM** : ブレーク解除
0 : ブレーク入力 tim_brk が設定されます。
1 : ブレーク入力 tim_brk は解除されます。
このビットは、アクティブなブレークソースがない場合、ハードウェアによってクリアされます。
BKDSRM ビットは、双方向出力制御 (ハイインピーダンス状態でのオープンドレイン出力) を解放するためにソフトウェアでセットしてから、ハードウェアによってリセットされ、障害状態がなくなったことを示すまでポーリングする必要があります

注： このビットへの書込み操作では、書込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 25 **BK2P** : ブレーク 2 極性
0 : ブレーク入力 tim_brk2 はアクティブラローです。
1 : ブレーク入力 tim_brk2 はアクティブハイです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注： このビットへの書込み操作では、書込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 24 **BK2E** : ブレーク 2 イネーブル

このビットは、完全なブレーク 2 保護を有効にします (図 384 : ブレークおよびブレーク 2 回路の概要のように bk_acth および BKIN ソースにそれぞれ接続されたすべてのソースを含む)。

0 : ブレーク 2 機能は無効です。

1 : ブレーク 2 機能は有効です。

注 : **BRKIN2** は **OSSR = OSS1 = 1** の場合にのみ使用してください。

注 : このビットは、**LOCK** レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの **LOCK** ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 23:20 **BK2F[3:0]** : ブレーク 2 フィルタ

このビットフィールドは、tim_brk2 入力をサンプリングする周波数と、tim_brk2 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、tim_brk2 は非同期として動作します。

0001 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 2

0010 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 4

0011 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 8

0100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 6

0101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 8

0110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 6

0111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 8

1000 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N = 6

1001 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N = 8

1010 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 5

1011 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 6

1100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 8

1101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 5

1110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 6

1111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 8

注 : このビットは、**LOCK** レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの **LOCK** ビット)。

ビット 19:16 **BKF[3:0]** : ブレークフィルタ

このビットフィールドは、tim_brk 入力をサンプリングする周波数と、tim_brk に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、tim_brk は非同期で動作します。

0001 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 2

0010 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 4

0011 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、N = 8

0100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 6

0101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 8

0110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 6

0111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 8

1000 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N = 6

1001 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N = 8

1010 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 5

1011 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 6

1100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N = 8

1101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 5

1110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 6

1111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N = 8

注： このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 15 **MOE** : メイン出力イネーブル

このビットは、ブレーク入力の 1 つ (tim_brk または tim_brk2) がアクティブになると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : ブレーク 2 イベントへの対応。OC および OCN 出力は無効です。

ブレークイベントへの対応、または MOE が 0 に書き込まれた場合 : OC および OCN 出力が無効か、OSSI ビットによって強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 38.6.11 : TIMx キャプチャ/比較有効レジスタ \(TIMx_CCER\) \(x = 1、8\)](#))。

ビット 14 **AOE** : 自動出力イネーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 tim_brk および tim_brk2 のいずれもがアクティブでない場合)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 **BKP** : ブレーク極性

0 : ブレーク入力 tim_brk はアクティブローです。

1 : ブレーク入力 tim_brk はアクティブハイです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注： このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 **BKE** : ブレークイネーブル

このビットは、完全なブレーク保護を有効にします (図 384 : ブレークおよびブレーク 2 回路の概要のように bk_acth および BKIN ソースにそれぞれ接続されたすべてのソースを含む)。

0 : ブレーク機能は無効です。

1 : ブレーク機能は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 **OSSR** : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 38.6.11 : TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 1, 8))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御される)。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 **OSSI** : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルでのブレークイベントやソフトウェアの書き込みによって使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 38.6.11 : TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 1, 8))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。

1 : インアクティブのとき、OC/OCN 出力はまず強制的にインアクティブレベルにされ、次にデッドタイム後に強制的にアイドルレベルにされます。タイマは出力の制御を保持します。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 9:8 **LOCK[1:0]** : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ — どのビットも書き込み保護されません。

01 : LOCK レベル 1 — TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKBID/BK2BID/BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 — LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込めなくなります。

11 : LOCK レベル 3 — LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMRx レジスタの OCxM および OCxPE ビット) が書き込めなくなります。

注 : LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで停止されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$ 。

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$ 。

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$ 。

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$ 。

例 : $T_{DTS}=125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

38.6.21 TIMx キャプチャ/比較レジスタ 5 (TIMx_CCR5) (x = 1、8)

アドレスオフセット : 0x048

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[19:16]			
r/w	r/w	r/w										r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **GC5C3** : チャネル 5 およびチャネル 3 のグループ化

チャネル 3 出力におけるひずみ :

0 : tim_oc5ref の tim_oc3refc への影響はありません。

1 : tim_oc3refc は、tim_oc3ref と tim_oc5ref の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR2 でプリロード機能を選択している場合)。

注 : このひずみを組み合わせ PWM 信号に適用することもできます。

ビット 30 **GC5C2** : チャネル 5 およびチャネル 2 のグループ化

チャネル 2 出力におけるひずみ :

0 : tim_oc5ref の tim_oc2refc への影響はありません。

1 : tim_oc2refc は、tim_oc2ref と tim_oc5ref の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。

注 : このひずみを組み合わせ PWM 信号に適用することもできます。

ビット 29 **GC5C1** : チャネル 5 およびチャネル 1 のグループ化

チャネル 1 出力におけるひずみ :

0 : oc5ref の oc1refc への影響はありません。

1 : oc1refc は、oc1ref と oc5ref の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。

注 : このひずみを組み合わせ PWM 信号に適用することもできます。

ビット 28:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR5[19:0]** : キャプチャ／比較 5 値

CCR5 は、実際のキャプチャ／比較 5 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR3 レジスタの OC5PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 5 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc5 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR5[15:0] に比較値を保持します。CCR5[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR5[19:4] に整数部を保持します。CCR5[3:0] ビットフィールドにはディザリング部分が含まれます。

38.6.22 TIMx キャプチャ／比較レジスタ 6 (TIMx_CCR6) (x = 1, 8)

アドレスオフセット : 0x04C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR6[19:0]** : キャプチャ／比較 6 値

CCR6 は、実際のキャプチャ／比較 6 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR3 レジスタの OC6PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 6 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc6 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR6[15:0] に比較値を保持します。CCR6[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR6[19:4] に整数部を保持します。CCR6[3:0] ビットフィールドにはディザリング部分が含まれます。

38.6.23 TIMx キャプチャ/比較モードレジスタ 3 (TIMx_CCMR3) (x = 1、8)

アドレスオフセット : 0x050

リセット値 : 0x0000 0000

上記の CCMR1 レジスタの説明を参照してください。チャンネル 5 および 6 は出力としてのみ設定できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M [2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC6CE** : 出力比較 6 クリアイネーブル

ビット 24、14:12 **OC6M[3:0]** : 出力比較 6 モード

ビット 11 **OC6PE** : 出力比較 6 プリロードイネーブル

ビット 10 **OC6FE** : 出力比較 6 高速イネーブル

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OC5CE** : 出力比較 5 クリアイネーブル

ビット 16、6:4 **OC5M[3:0]** : 出力比較 5 モード

ビット 3 **OC5PE** : 出力比較 5 プリロードイネーブル

ビット 2 **OC5FE** : 出力比較 5 高速イネーブル

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

38.6.24 TIMx タイマデッドタイムレジスタ 2 (TIMx_DTR2) (x = 1、8)

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGF[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DTPE** : デッドタイムプリロードイネーブル

0 : デッドタイム値はプリロードされません。

1 : デッドタイム値プリロードは有効です。

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 16 **DTAE** : デッドタイム非対称イネーブル

0 : 立上がりおよび立下がりエッジでのデッドタイムは同一で、DTG[7:0] レジスタで定義されます。

1 : 立上がりエッジのデッドタイムは DTG[7:0] レジスタで定義され、立下がりエッジのデッドタイムは DTGF[7:0] ビットで定義されます。

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DTGF[7:0]** : デッドタイム立下がりエッジジェネレータのセットアップ

このビットフィールドでは、立下がりエッジにおいて、相補出力の間に挿入されるデッドタイムの長さを指定します。

DTGF[7:5]=0xx => DTF=DTGF[7:0]x t_{dtg} 、ここで $t_{dtg}=t_{DTS}$ 。DTGF[7:5]=10x => DTF=(64+DTGF[5:0])x t_{dtg} 、ここで $T_{dtg}=2xt_{DTS}$ 。DTGF[7:5]=110 => DTF=(32+DTGF[4:0])x t_{dtg} 、ここで $T_{dtg}=8xt_{DTS}$ 。DTGF[7:5]=111 => DTF=(32+DTGF[4:0])x t_{dtg} 、ここで $T_{dtg}=16xt_{DTS}$ 。例 : $T_{DTS}=125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)32 μs から 63 μs (1 μs 単位)64 μs から 126 μs (2 μs 単位)

注： このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

38.6.25 TIMx タイマエンコーダ制御レジスタ (TIMx_ECR) (x = 1、8)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PWPRSC[2:0]			PW[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPOS[1:0]		FIDX	IBLK[1:0]		IDIR[1:0]		IE
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **PWPRSC[2:0]** : パルス幅プリスケアラ

このビットフィールドは、パルス発生回路用のクロックプリスケアラを次のように設定します。

$$t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$$

ビット 23:16 **PW[7:0]** : パルス幅

このビットフィールドは、パルスの長さを次のように定義します。

$$t_{PW} = PW[7:0] \times t_{PWG}$$

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **IPOS[1:0]** : インデックス位置決め

直交エンコーダモード (SMS[3:0] = 0001、0010、0011、1110、1111) で、このビットは、インデックスイベントがカウンタをリセットする AB 入力の設定を示します。

00 : AB = 00 のときにインデックスがカウンタをリセットします。

01 : AB = 01 のときにインデックスがカウンタをリセットします。

10 : AB = 10 のときにインデックスがカウンタをリセットします。

11 : AB = 11 のときにインデックスがカウンタをリセットします。

方向クロックモードまたはクロックプラス方向モード (SMS[3:0] = 1010、1011、1100、1101) で、これらのビットは、インデックスイベントがカウンタをリセットするレベルを示します。双方向クロックモードでは、これは両方のクロック入力に適用されます。

x0 : クロックが 0 のときにインデックスがカウンタをリセットします。

x1 : クロックが 1 のときにインデックスがカウンタをリセットします。

注 : IPOS[1] ビットは意味がありません。

ビット 5 **FIDX** : 最初のインデックス

このビットは、最初のインデックスのみが考慮されるかどうかを示します。

0 : インデックスは常にアクティブです。

1 : 最初のインデックスのみがカウンタをリセットします。

ビット 4:3 **IBLK[1:0]** : インデックスのブランキング

このビットは、インデックスイベントが tim_ti3 または tim_ti4 の入力によって調整されるかどうかを示します。

00 : インデックスは常にアクティブです。

01 : インデックスは、CC3P ビットフィールドに従って、tim_ti3 入力がアクティブなときに無効になります。

10 : インデックスは、CC4P ビットフィールドに従って、tim_ti4 入力がアクティブなときに無効になります。

11 : 予約済み

ビット 2:1 **IDIR[1:0]** : インデックスの方向

このビットは、どちらの方向のときにインデックスイベントがカウンタをリセットするかを示します。

00 : 方向に関係なく、インデックスがカウンタをリセットします。

01 : アップカウントのときのみ、インデックスがカウンタをリセットします。

10 : ダウンカウントのときのみ、インデックスがカウンタをリセットします。

11 : 予約済み

ビット 0 **IE**インデックスイネーブル

このビットは、インデックスイベントがカウンタをリセットするかどうかを示します。

0 : インデックスは無効です。

1 : インデックスは有効です

38.6.26 TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 1、8)

アドレスオフセット : 0x05C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL [3:0]				Res.	Res.	Res.	Res.	TI3SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **TI4SEL[3:0]** : tim_ti4[15:0] 入力を選択します。

0000 : tim_ti4_in0 : TIMx_CH4

0001 : tim_ti4_in1

.....

1111 : tim_ti4_in15

相互接続一覧については、[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **TI3SEL[3:0]** : tim_ti3[15:0] 入力を選択します。

0000 : tim_ti3_in0 : TIMx_CH2

0001 : tim_ti3_in1

.....

1111 : tim_ti3_in15

相互接続一覧については、[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TI2SEL[3:0]** : tim_ti2[15:0] 入力を選択します。

0000 : tim_ti2_in0 : TIMx_CH2

0001 : tim_ti2_in1

.....

1111 : tim_ti2_in15

相互接続一覧については、[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1[15:0] 入力を選択します。

0000 : tim_ti1_in0 : TIMx_CH1

0001 : tim_ti1_in1

.....

1111 : tim_ti1_in15

相互接続一覧については、[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

38.6.27 TIMx オルタネート機能オプションレジスタ 1 (TIMx_AF1) (x = 1、8)

アドレスオフセット : 0x060

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		BK CMP4P	BK CMP3P	BK CMP2P	BK CMP1P	BKINP	BK CMP8E	BK CMP7E	BK CMP6E	BK CMP5E	BK CMP4E	BK CMP3E	BK CMP2E	BK CMP1E	BKINE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:14 **ETRSEL[3:0]** : etr_in ソース選択

これらのビットは、etr_in 入力ソースを選択します。

0000 : tim_etr0 : TIMx_ETR 入力

0001 : tim_etr1

.....

1111 : tim_etr15

製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

注： これらのビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 **BKCMP4P** : tim_brk_cmp4 入力極性

このビットは、tim_brk_cmp4 入力の感度を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp4 入力の極性は反転されません (BKP = 0 の場合はアクティブロー、BKP = 1 の場合はアクティブハイ)。

1 : tim_brk_cmp4 入力の極性を反転させます (BKP = 0 の場合はアクティブハイ、BKP = 1 の場合はアクティブロー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 12 **BKCMP3P** : tim_brk_cmp3 入力極性

このビットは、tim_brk_cmp3 入力の感度を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp3 入力の極性は反転されません (BKP = 0 の場合はアクティブロー、BKP = 1 の場合はアクティブハイ)。

1 : tim_brk_cmp3 入力の極性を反転させます (BKP = 0 の場合はアクティブハイ、BKP = 1 の場合はアクティブロー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 11 **BKCMP2P** : tim_brk_cmp2 入力極性

このビットは、tim_brk_cmp2 入力の感度を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp2 入力の極性は反転されません (BKP = 0 の場合はアクティブロー、BKP = 1 の場合はアクティブハイ)。

1 : tim_brk_cmp2 入力の極性を反転させます (BKP = 0 の場合はアクティブハイ、BKP = 1 の場合はアクティブロー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 10 **BKCMP1P** : tim_brk_cmp1 入力極性

このビットは、tim_brk_cmp1 入力の感度を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp1 入力の極性は反転されません (BKP = 0 の場合はアクティブロー、BKP = 1 の場合はアクティブハイ)。

1 : tim_brk_cmp1 入力の極性を反転させます (BKP = 0 の場合はアクティブハイ、BKP = 1 の場合はアクティブロー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 9 **BKINP** : TIMx_BKIN 入力の極性

このビットは、TIMx_BKIN オルタネート機能入力の感度を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : TIMx_BKIN 入力の極性は反転されません (BKP = 0 の場合はアクティブロー、BKP = 1 の場合はアクティブハイ)。

1 : TIMx_BKIN 入力の極性を反転させます (BKP = 0 の場合はアクティブハイ、BKP = 1 の場合はアクティブロー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 8 **BKCMP8E** : tim_brk_cmp8 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp8 を有効化します。tim_brk_cmp8 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp8 入力は無効です。

1 : tim_brk_cmp8 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 7 **BKCMP7E** : tim_brk_cmp7 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp7 を有効化します。tim_brk_cmp7 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp7 入力は無効です。

1 : tim_brk_cmp7 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 6 **BKCMP6E** : tim_brk_cmp6 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp6 を有効化します。tim_brk_cmp6 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp6 入力は無効です。

1 : tim_brk_cmp6 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 5 **BKCMP5E** : tim_brk_cmp5 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp5 を有効化します。tim_brk_cmp5 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp5 入力は無効です。

1 : tim_brk_cmp5 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 4 **BKCMP4E** : tim_brk_cmp4 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp4 を有効化します。tim_brk_cmp4 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp4 入力は無効です。

1 : tim_brk_cmp4 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 3 **BKCMP3E** : tim_brk_cmp3 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp3 を有効化します。tim_brk_cmp3 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp3 入力は無効です。

1 : tim_brk_cmp3 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 2 **BKCMP2E** : tim_brk_cmp2 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp2 を有効化します。tim_brk_cmp2 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp2 入力は無効です。

1 : tim_brk_cmp2 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 1 **BKCMP1E** : tim_brk_cmp1 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp1 を有効化します。tim_brk_cmp1 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp1 入力は無効です。

1 : tim_brk_cmp1 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 0 **BKINE** : TIMx_BKIN 入力イネーブル

このビットは、タイマの tim_brk 入力に対して TIMx_BKIN オルタネート機能入力を有効化します。TIMx_BKIN 入力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : TIMx_BKIN 入力は無効です。

1 : TIMx_BKIN 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注： 製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

38.6.28 TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 1、8)

アドレスオフセット : 0x064

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]		
													r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	BK2CM P4P	BK2CM P3P	BK2CM P2P	BK2CM P1P	BK2IN P	BK2CMP8 E	BK2CM P7E	BK2CM P6E	BK2CM P5E	BK2CM P4E	BK2CMP3 E	BK2CMP2 E	BK2CMP 1E	BK2INE
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **OCRSEL[2:0]** : ocref_clr ソース選択

これらのビットは、ocref_clr 入力ソースを選択します。

000 : tim_ocref_clr0

001 : tim_ocref_clr1

.....

111 : tim_ocref_clr7

製品固有の情報については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。**注：** これらのビットは、**LOCK レベル 1** がプログラムされている場合 (TIMx_BDTR レジスタの **LOCK ビット**)、**変更できません**。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **BK2CMP4P** : tim_brk2_cmp4 入力極性

このビットは、tim_brk2_cmp4 入力の検知を選択します。BK2P 極性ビットとともにプログラムする必要があります。

0 : tim_brk2_cmp4 入力の極性は反転されません (BK2P = 0 の場合はアクティブラー、BK2P = 1 の場合はアクティブハイ)。

1 : tim_brk2_cmp4 入力の極性を反転させます (BK2P = 0 の場合はアクティブハイ、BK2P = 1 の場合はアクティブラー)。

注： このビットは、**LOCK レベル 1** がプログラムされている場合 (TIMx_BDTR レジスタの **LOCK ビット**)、**変更できません**。ビット 12 **BK2CMP3P** : tim_brk2_cmp3 入力極性

このビットは、tim_brk2_cmp3 入力の検知を選択します。BK2P 極性ビットとともにプログラムする必要があります。

0 : tim_brk2_cmp3 入力の極性は反転されません (BK2P = 0 の場合はアクティブラー、BK2P = 1 の場合はアクティブハイ)。

1 : tim_brk2_cmp3 入力の極性を反転させます (BK2P = 0 の場合はアクティブハイ、BK2P = 1 の場合はアクティブラー)。

注： このビットは、**LOCK レベル 1** がプログラムされている場合 (TIMx_BDTR レジスタの **LOCK ビット**)、**変更できません**。

ビット 11 BK2CMP2P : tim_brk2_cmp2 入力極性

このビットは、tim_brk2_cmp2 入力の検知を選択します。BK2P 極性ビットとともにプログラムする必要があります。

0 : tim_brk2_cmp2 入力の極性は反転されません (BK2P = 0 の場合はアクティブラー、BK2P = 1 の場合はアクティブハイ)。

1 : tim_brk2_cmp2 入力の極性を反転させます (BK2P = 0 の場合はアクティブハイ、BK2P = 1 の場合はアクティブラー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 10 BK2CMP1P : tim_brk2_cmp1 入力極性

このビットは、tim_brk2_cmp1 入力の検知を選択します。BK2P 極性ビットとともにプログラムする必要があります。

0 : tim_brk2_cmp1 入力の極性は反転されません (BK2P = 0 の場合はアクティブラー、BK2P = 1 の場合はアクティブハイ)。

1 : tim_brk2_cmp1 入力の極性を反転させます (BK2P = 0 の場合はアクティブハイ、BK2P = 1 の場合はアクティブラー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 9 BK2INP : TIMx_BKIN2 入力の極性

このビットは、TIMx_BKIN2 オルタネート機能入力の検知を選択します。BK2P 極性ビットとともにプログラムする必要があります。

0 : TIMx_BKIN2 入力の極性は反転されません (BK2P = 0 の場合はアクティブラー、BK2P = 1 の場合はアクティブハイ)。

1 : TIMx_BKIN2 入力の極性を反転させます (BK2P = 0 の場合はアクティブハイ、BK2P = 1 の場合はアクティブラー)。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 8 BK2CMP8E : tim_brk2_cmp8 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp8 を有効化します。tim_brk2_cmp8 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp8 入力は無効です。

1 : tim_brk2_cmp8 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 7 BK2CMP7E : tim_brk2_cmp7 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp7 を有効化します。tim_brk2_cmp7 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp7 入力は無効です。

1 : tim_brk2_cmp7 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 6 BK2CMP6E : tim_brk2_cmp6 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp6 を有効化します。tim_brk2_cmp6 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp6 入力は無効です。

1 : tim_brk2_cmp6 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 5 **BK2CMP5E** : tim_brk2_cmp5 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp5 を有効化します。tim_brk2_cmp5 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp5 入力は無効です。

1 : tim_brk2_cmp5 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 4 **BK2CMP4E** : tim_brk2_cmp4 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp4 を有効化します。tim_brk2_cmp4 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp4 入力は無効です。

1 : tim_brk2_cmp4 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 3 **BK2CMP3E** : tim_brk2_cmp3 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp3 を有効化します。tim_brk2_cmp3 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp3 入力は無効です。

1 : tim_brk2_cmp3 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 2 **BK2CMP2E** : tim_brk2_cmp2 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp2 を有効化します。tim_brk2_cmp2 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp2 入力は無効です。

1 : tim_brk2_cmp2 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 1 **BK2CMP1E** : tim_brk2_cmp1 イネーブル

このビットは、タイマの tim_brk2 入力に対して tim_brk2_cmp1 を有効化します。tim_brk2_cmp1 出力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : tim_brk2_cmp1 入力は無効です。

1 : tim_brk2_cmp1 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 0 **BK2INE** : TIMx_BKIN2 入力イネーブル

このビットは、タイマの tim_brk2 入力に対して TIMx_BKIN2 オルタネート機能入力を有効化します。TIMx_BKIN2 入力は、ほかの tim_brk2 ソースとの「論理和」がとられます。

0 : TIMx_BKIN2 入力は無効です。

1 : TIMx_BKIN2 入力は有効です。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

注： 製品固有の実装については[セクション 38.3.2 : TIM1/TIM8 ピンおよび内部信号](#)を参照してください。

38.6.29 TIMx DMA 制御レジスタ (TIMx_DCR) (x = 1, 8)

アドレスオフセット : 0x3DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **DBSS[3:0]** : DMA バーストソース選択

このビットフィールドは、DMA バースト転送をトリガする割込みソースを定義します (タイマは、TIMx_DMAR アドレスに対して読出しまたは書き込みアクセスが行われるときにバースト転送を認識します)。

0000 : 予約済み

0001 : 更新

0010 : CC1

0011 : CC2

0100 : CC3

0101 : CC4

0110 : COM

0111 : トリガ

その他 : 予約済み

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、転送数など DMA 転送長 (タイマは、TIMx_DMAR アドレスに対して読出しまたは書き込みアクセスが行われるときにバースト転送を認識します) を指定します。転送は、ハーフワードまたはバイトです (以下の例を参照)。

00000 : 1 回転送

00001 : 2 回転送

00010 : 3 回転送

.....

11010 : 26 回転送

例 : 次の転送を考えます : DBL = 7 バイトかつ DBA = TIM2_CR1。

–DBL = 7 バイトおよび DBA = TIM2_CR1 が転送するバイトのアドレスを表す場合、転送のアドレスは次の式で与えられます。

(TIMx_CR1 アドレス) + DBA + (DMA インデックス)、ここで DMA インデックス = DBL

この例では、(TIMx_CR1 アドレス) + DBA に 7 バイトが追加され、データのコピー元/コピー先アドレスが与えられます。この場合、転送は、以下のアドレスから始めて、7 つのレジスタに対して行われます。(TIMx_CR1 アドレス) + DBA

DMA データサイズの設定に応じて、いくつかのケースが想定されます。

–DMA データサイズをハーフワードで設定した場合、7 つのレジスタにそれぞれ 16 ビットのデータが転送されます。

–DMA データサイズをバイトで設定した場合も、データは 7 つのレジスタに転送されます (最初のレジスタには最初の MSB バイトが含まれ、2 番目のレジスタには最初の LSB バイトが含まれるなど、以降同様)。タイマへの転送で、DMA によって転送されるデータサイズを指定する必要があります。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し/書込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 :TIMx_CR1

00001 :TIMx_CR2

00010 :TIMx_SMCR

.....

38.6.30 完全転送の TIMx DMA アドレス (TIMx_DMAR) (x = 1、8)

アドレスオフセット : 0x3E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **DMAB[31:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読出しまたは書込み動作は、(TIMx_CR1 アドレス) + (DBA + DMA インデックス) x 4 のアドレスにあるレジスタへのアクセスとなります。

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスはDMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL はTIMx_DCR 内で設定)。

38.6.31 TIMx レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 396. TIMxレジスタマップとリセット値

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DITHEN	UIFREMAP	Res.	CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN		
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	
0x004	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	MMS[3]	Res.	MMS2 [3:0]				Res.	OIS6	Res.	OIS5	OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TIS1	MMS [2:0]		CCDS	CCUS	Res.	CCPC	
	リセット値							0		0	0	0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x008	TIMx_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	SMSPS	SMSPE	Res.	Res.	TS [4:3]	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]	ETF[3:0]			MSM	TS[2:0]		OCCS	SMS[2:0]					
	リセット値							0	0			0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERRIE	IERRIE	DIRIE	IDXIE	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	リセット値									0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERRF	IERRF	DIRF	IDXF	Res.	Res.	Res.	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF	
	リセット値									0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	リセット値																								0	0	0	0	0	0	0	0	0
0x018	TIMx_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]			IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]			IC1PSC [1:0]		CC1S [1:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]			
	リセット値								0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	TIMx_CCMR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]			
	リセット値								0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIMx_CCMR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F [3:0]			IC4PSC [1:0]		CC4S [1:0]		IC3F [3:0]			IC3PSC [1:0]		CC3S [1:0]			
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x020	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	リセット値											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x024	TIMx_CNT	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 396. TIMxレジスタマップとリセット値 (続き)

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x02C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																			
	リセット値													0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x030	TIMx_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1 [19:0]															
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	TIMx_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2 [19:0]															
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	TIMx_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3 [19:0]															
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	TIMx_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4 [19:0]															
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	TIMx_BDTR	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F [3:0]				BKF[3:0]			MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]									
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	TIMx_CCR5	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5 [19:0]																			
	リセット値	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	TIMx_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6 [19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	TIMx_CCMR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]		OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]		OC5PE	OC5FE	Res.	Res.		
	リセット値								0								0	0	0	0	0	0	0			0	0	0	0	0	0		

表 396. TIMxレジスタマップとリセット値 (続き)

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
0x054	TIMx_DTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGTF[7:0]													
	リセット値															0	0										0	0	0	0	0	0	0	0						
0x058	TIMx_ECR	Res.	Res.	Res.	Res.	Res.	PWPR SC[2:0]			PW[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPOS[1:0]		FIDX	IBLK [1:0]	IDIR [1:0]	IE								
	リセット値						0	0	0	0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0						
0x05C	TIMx_TISEL	Res.	Res.	Res.	Res.	TI4SEL [3:0]				Res.	Res.	Res.	Res.	TI3SEL [3:0]				Res.	Res.	Res.	TI2SEL [3:0]					Res.	Res.	Res.	TI1SEL [3:0]											
	リセット値					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0							
0x060	TIMx_AF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]			BKCMP4P	BKCMP3P	BKCMP2P	BKCMP1P	BKINP	BKCMP8E	BKCMP7E	BKCMP6E	BKCMP5E	BKCMP4E	BKCMP3E	BKCMP2E	BKCMP1E	BKINE								
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1							
0x064	TIMx_AF2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]			Res.	Res.	BK2CMP4P	BK2CMP3P	BK2CMP2P	BK2CMP1P	BK2INP	BK2CMP8E	BK2CMP7E	BK2CMP6E	BK2CMP5E	BK2CMP4E	BK2CMP3E	BK2CMP2E	BK2CMP1E	BK2INE							
	リセット値														0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	1							
0x068.. 0x3D8	予約済み	Res.																																						
0x3DC	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]												
	リセット値													0	0	0	0				0	0	0	0	0				0	0	0	0	0							
0x3E0	TIMx_DMAR	DMAB[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

レジスタ境界アドレスについては [110ページのセクション 2.3](#) を参照してください。

39 汎用タイマ (TIM2/TIM3/TIM4/TIM5)

39.1 TIM2/TIM3/TIM4/TIM5 の概要

この汎用タイマは、プログラマブルなプリスケアラによって駆動される 16 ビットまたは 32 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較と PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 39.4.23 : タイマの同期](#)に示すように、相互に同期させることができます。

39.2 TIM2/TIM3/TIM4/TIM5 の主な特徴

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビットまたは 32 ビットのアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65535 の間で分周する 16 ビットプログラム可能プリスケアラ。
- 次の機能を持つ、最大 4 つの独立チャネル：
 - 入力キャプチャ
 - 出力比較
 - PWM 生成 (エッジアラインモードとセンターアラインモード)
 - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み/DMA 生成：
 - 更新：カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - 入力キャプチャ
 - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

39.3 TIM2/TIM3/TIM4/TIM5 の実装

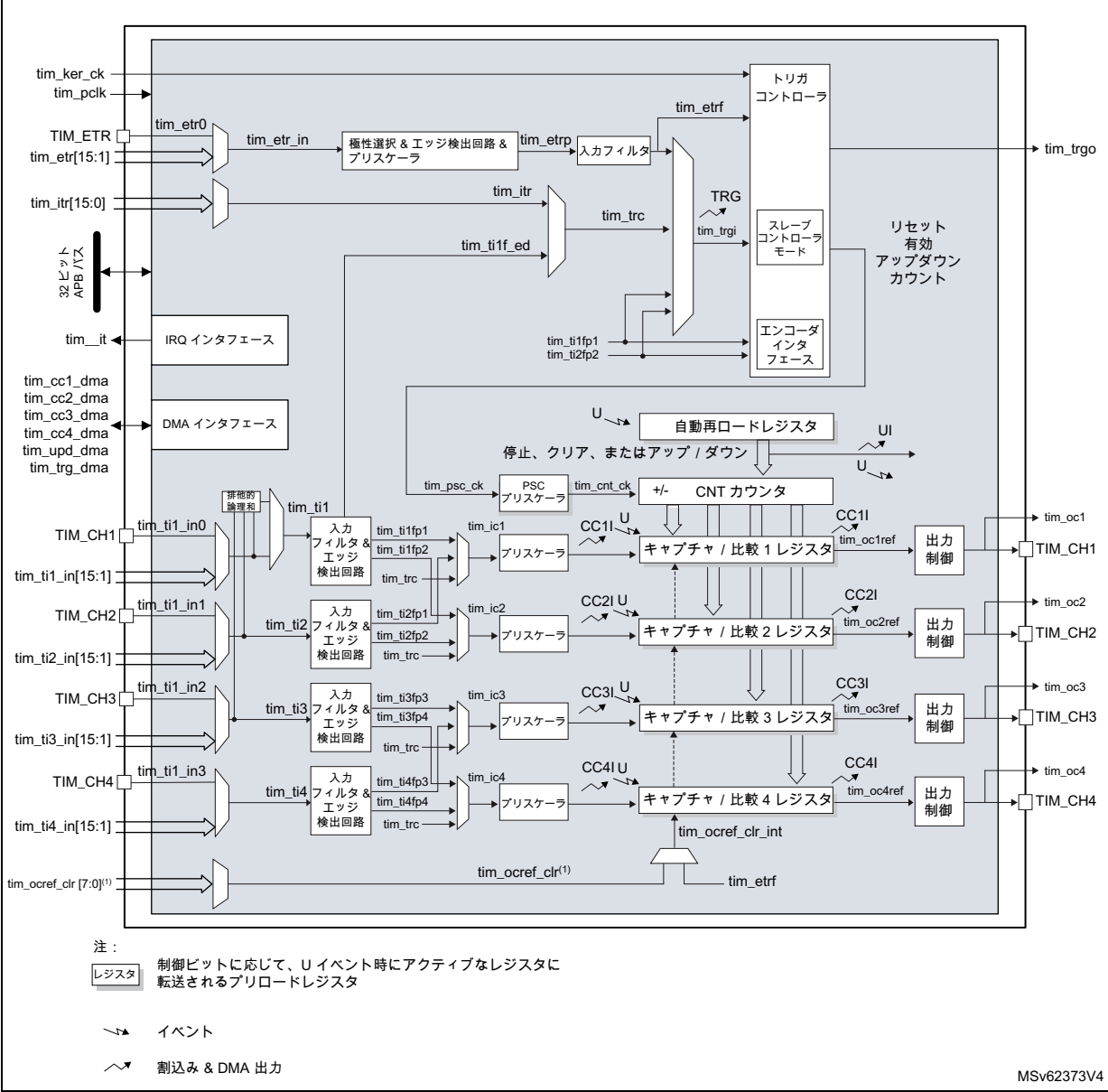
表 397. STM32H563/H573および STM32H562汎用タイマ

タイマインスタンス	TIM2	TIM3	TIM4	TIM5
分解能	32 ビット	16 ビット	16 ビット	32 ビット
OCREF クリア選択 ソース	可能 tim_etrf 予約済み	可能 tim_etrf 予約済み	可能 tim_etrf 予約済み	可能 tim_etrf 予約済み

39.4 TIM2/TIM3/TIM4/TIM5機能詳細

39.4.1 ブロック図

図 427. 汎用タイマのブロック図



1. この機能は一部のタイマでは利用できません (セクション 39.3 : TIM2/TIM3/TIM4/TIM5 の実装を参照)。

39.4.2 TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号

このセクションの表 398 および表 399 は、TIM の入力および出力の一覧です。

表 398. TIM の入出力ピン

ピン名	信号タイプ	説明
TIM_CH1 TIM_CH2 TIM_CH3 TIM_CH4	入力／出力	タイマ多目的チャンネル。 各チャンネルは、キャプチャ、比較、または PWM で使用できます。 TIM_CH1 と TIM_CH2 は、外部クロック (tim_ker_ck クロックの 1/4 未満)、外部トリガ、および直交エンコーダの入力としても使用できます。 TIM_CH1、TIM_CH2、および TIM_CH3 は、デジタルホール効果センサとのインタフェースに使用できます。
TIM_ETR	入力	外部トリガ入力。この入力 は、外部トリガまたは外部クロックソースとして使用できます。tim_etr_in プリスケーラが使用されている場合、この入力は tim_ker_ck よりも高い周波数のクロックを受信できます。

表 399. TIM 内部入力／出力信号

内部信号名	信号タイプ	説明
tim_ti1_in [15:0] tim_ti2_in [15:0] tim_ti3_in [15:0] tim_ti4_in [15:0]	入力	内部タイマ入力バス。tim_ti1_in[15:0] と tim_ti2_in[15:0] の入力は、キャプチャ用に、または外部クロック (tim_ker_ck クロックの 1/4 未満) として、および直交エンコーダの信号に使用できます。
tim_etr[15:0]	入力	外部トリガ内部入力バス。これらの入力は、トリガ、外部クロックとして、またはハードウェアのサイクルごとのパルス幅制御に使用できます。tim_etr_in プリスケーラが使用されている場合、これらの入力は tim_ker_ck よりも高い周波数のクロックを受信できます。
tim_itr[15:0]	入力	内部トリガ入力バス。これらの入力は、スレーブモードコントローラ用に、または入力クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_trgo	出力	内部トリガ出力。このトリガは他のオンチップペリフェラルをトリガできます。
tim_ocref_clr[7:0]	入力	タイマの tim_ocref_clr 入力バス。これらの入力は、通常ハードウェアのサイクルごとのパルス幅制御のために、tim_ocxref 信号をクリアするために使用できます。
tim_pclk	入力	タイマ APB クロック。
tim_ker_ck	入力	タイマカーネルクロック

表 399. TIM 内部入力／出力信号 (続き)

内部信号名	信号タイプ	説明
tim_it	出力	キャプチャ／比較、更新、およびブレークトリガリクエストを取りまとめるグローバルタイマ割込み。
tim_cc1_dma tim_cc2_dma tim_cc3_dma tim_cc4_dma	出力	タイマキャプチャ／比較 1～4 DMA リクエスト。
tim_upd_dma	出力	タイマ更新 DMA リクエスト。
tim_trg_dma	出力	タイマトリガ DMA リクエスト。

表 400、表 401、表 402、および表 403 に、tim_ti[4:1] 入力マルチプレクサに接続されるソースを列挙します。

表 400. tim_ti1 入力マルチプレクサへの相互接続

tim_ti1 入力	ソース			
	TIM2	TIM3	TIM4	TIM5
tim_ti1_in0	TIM2_CH1	TIM3_CH1	TIM4_CH1	TIM5_CH1
tim_ti1_in1	eth_ptp_pps_o	eth_ptp_pps_o	予約済み	予約済み
tim_ti1_in[15:2]	予約済み			

表 401. tim_ti2 入力マルチプレクサへの相互接続

tim_ti2 入力	ソース			
	TIM2	TIM3	TIM4	TIM5
tim_ti2_in0	TIM2_CH2	TIM3_CH2	TIM4_CH2	TIM5_CH2
tim_ti2_in [15:1]	予約済み			

表 402. tim_ti3 入力マルチプレクサへの相互接続

tim_ti3 入力	ソース			
	TIM2	TIM3	TIM4	TIM5
tim_ti3_in0	TIM2_CH3	TIM3_CH3	TIM4_CH3	TIM5_CH3
tim_ti3_in [15:1]	予約済み			

表 403. tim_ti4 入力マルチプレクサへの相互接続

tim_ti4 入力	ソース			
	TIM2	TIM3	TIM4	TIM5
tim_ti4_in0	TIM2_CH4	TIM3_CH4	TIM4_CH4	TIM5_CH4
tim_ti4_in [15:1]	予約済み			

表 404 に、tim_etr 入力マルチプレクサに接続される内部ソースを列挙します。

表 404. TIMx 内部トリガ接続

TIMx	TIM2	TIM3	TIM4	TIM5
tim_itr0	tim1_trgo	tim1_trgo	tim1_trgo	tim1_trgo
tim_itr1	予約済み	tim2_trgo	tim2_trgo	tim2_trgo
tim_itr2	tim3_trgo	予約済み	tim3_trgo	tim3_trgo
tim_itr3	tim4_trgo	tim4_trgo	予約済み	tim4_trgo
tim_itr4	tim5_trgo	tim5_trgo	tim5_trgo	予約済み
tim_itr5	tim8_trgo	tim8_trgo	tim8_trgo	tim8_trgo
tim_itr6	tim12_trgo	tim12_trgo	tim12_trgo	tim12_trgo
tim_itr7	tim13_oc1	tim13_oc1	tim13_oc1	tim13_oc1
tim_itr8	tim14_oc1	tim14_oc1	tim14_oc1	tim14_oc1
tim_itr9	tim15_trgo	tim15_trgo	tim15_trgo	tim15_trgo
tim_itr10	tim16_oc1	tim16_oc1	tim16_oc1	tim16_oc1
tim_itr11	tim17_oc1	tim17_oc1	tim17_oc1	tim17_oc1
tim_itr12	USBSOF	予約済み	予約済み	USBSOF
tim_itr[15:13]	予約済み			

表 405 に、tim_etr 入力マルチプレクサに接続される内部ソースを列挙します。

表 405. tim_etr 入力マルチプレクサへの相互接続

タイマ外部トリガ入力信号	タイマ外部トリガ信号割り当て			
	TIM2	TIM3	TIM4	TIM5
tim_etr0	TIM2_ETR	TIM3_ETR	TIM4_ETR	TIM5_ETR
tim_etr1	予約済み	予約済み	予約済み	sai2_fs_a
tim_etr2				sai2_fs_b
tim_etr3	RCC_LSE	予約済み	予約済み	予約済み
tim_etr4	sai1_fs_a			
tim_etr5	sai1_fs_b			
tim_etr6	予約済み			
tim_etr7				
tim_etr8		TIM2_ETR	TIM2_ETR	TIM2_ETR
tim_etr9	TIM3_ETR	予約済み	TIM3_ETR	TIM3_ETR
tim_etr10	TIM4_ETR	TIM4_ETR	予約済み	TIM4_ETR
tim_etr11	TIM5_ETR	TIM5_ETR	TIM5_ETR	予約済み
tim_etr12	予約済み	予約済み	予約済み	予約済み
tim_etr13				
tim_etr14	eth_ptp_pps_o	eth_ptp_pps_o		
tim_etr15	予約済み			

39.4.3 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC) :
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ有効な、プリスケアラ出力 tim_cnt_ck から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

実際のカウンタイネーブル信号 CNT_EN は、CEN の 1 クロックサイクル後にセットされます。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビット/32 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 428 と 図 429 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 428. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

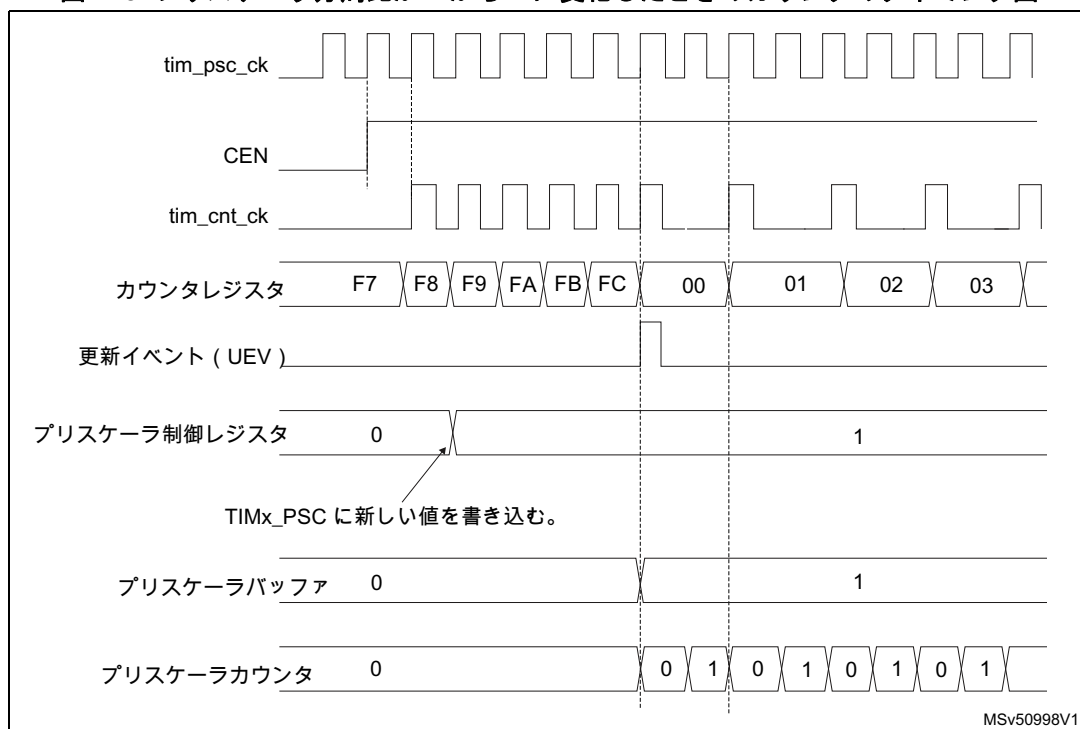
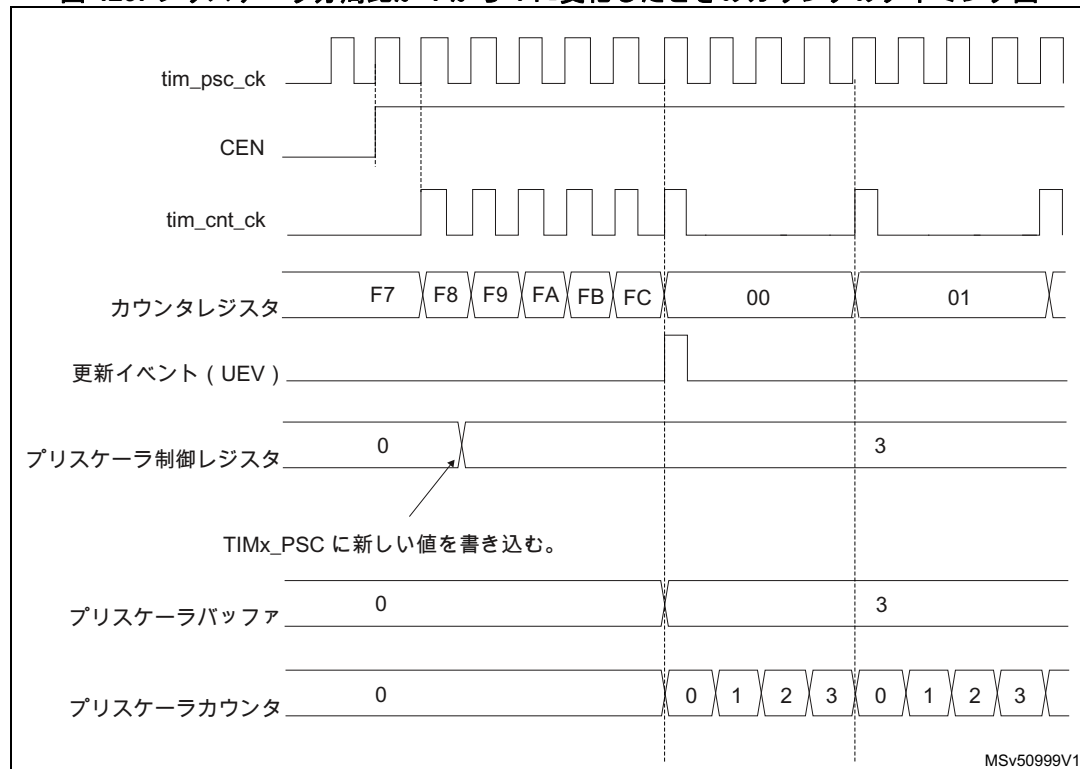


図 429. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



39.4.4 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることで生成できます。

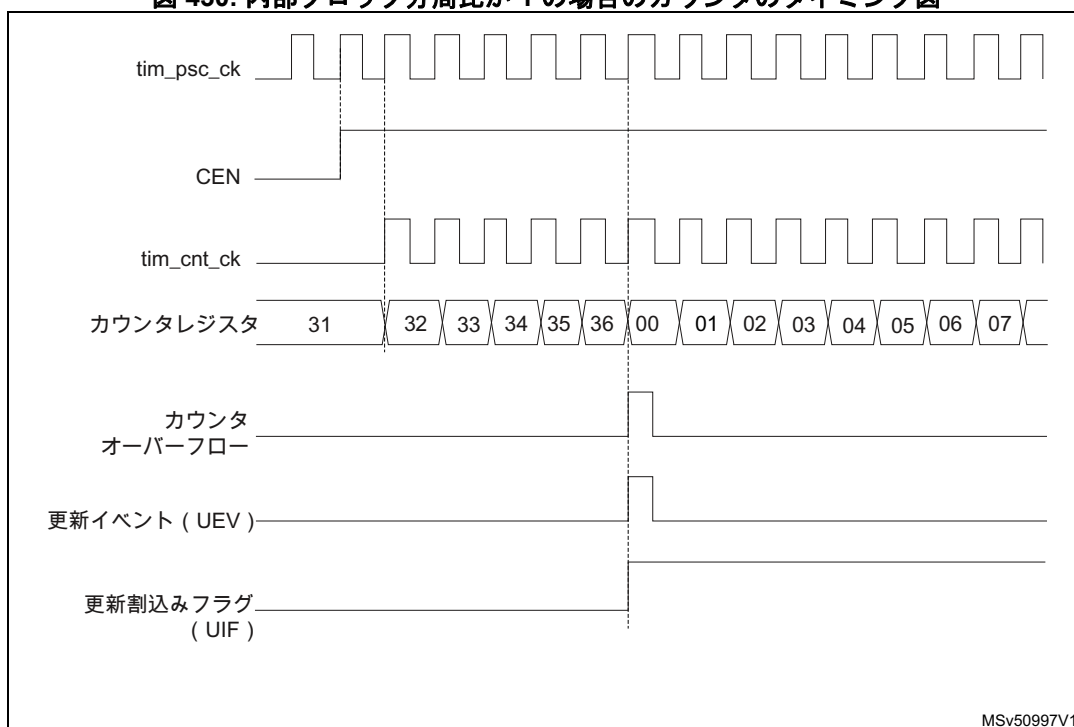
UEV イベントは、TIMx_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 430. 内部クロック分周比が 1 の場合のカウンタのタイミング図



MSv50997V1

図 431. 内部クロック分周比が 2 の場合のカウンタのタイミング図

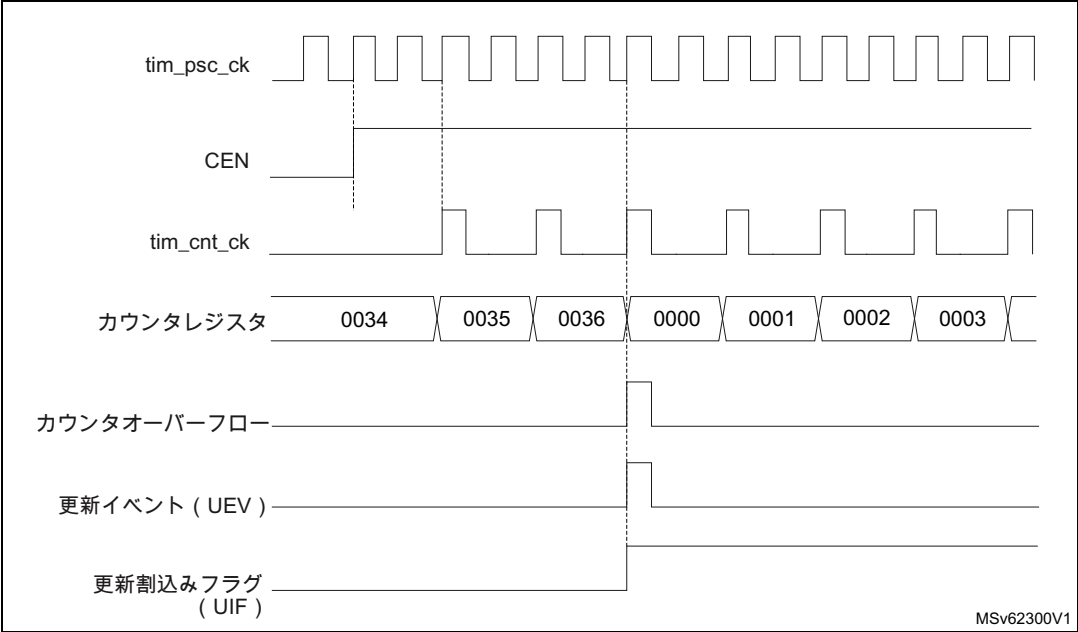


図 432. 内部クロック分周比が 4 の場合のカウンタのタイミング図

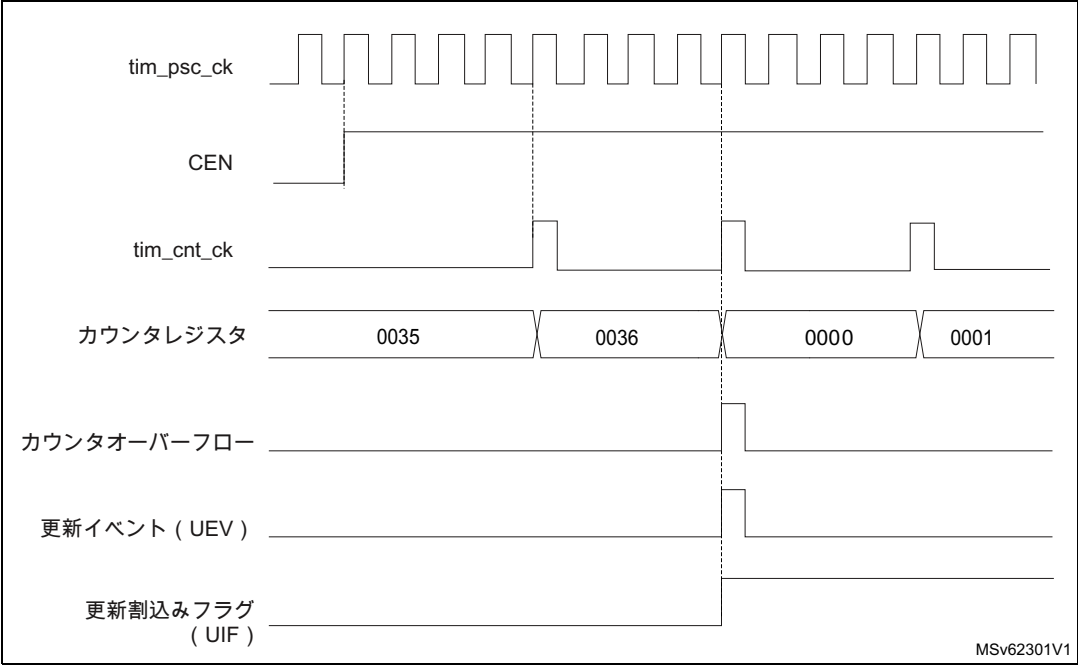


図 433. 内部クロック分周比が N の場合のカウンタのタイミング図

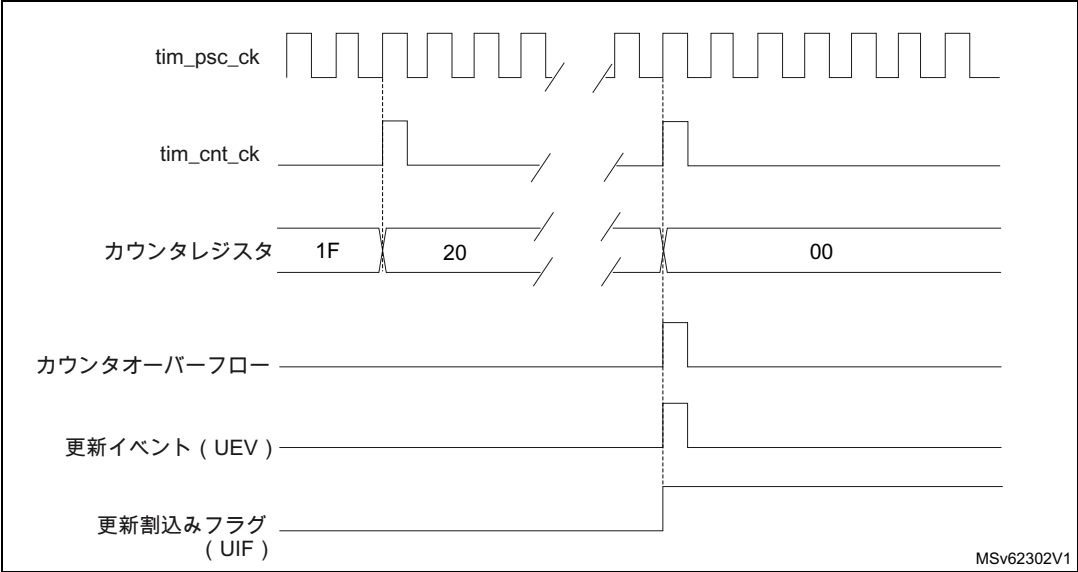


図 434. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

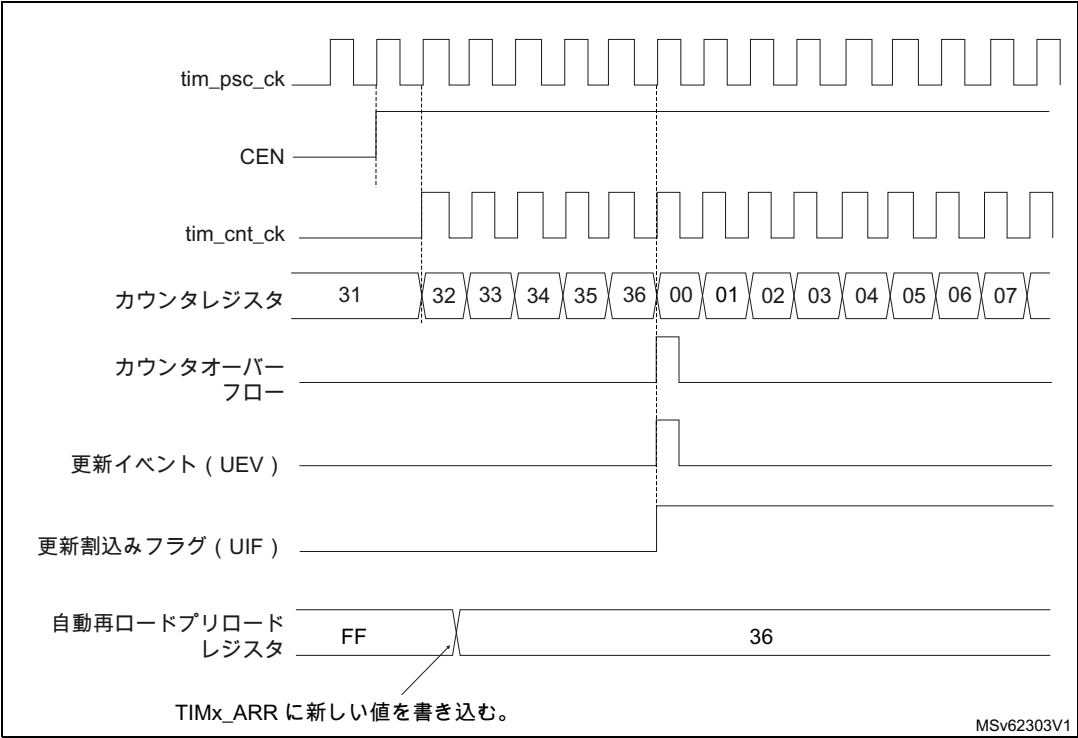
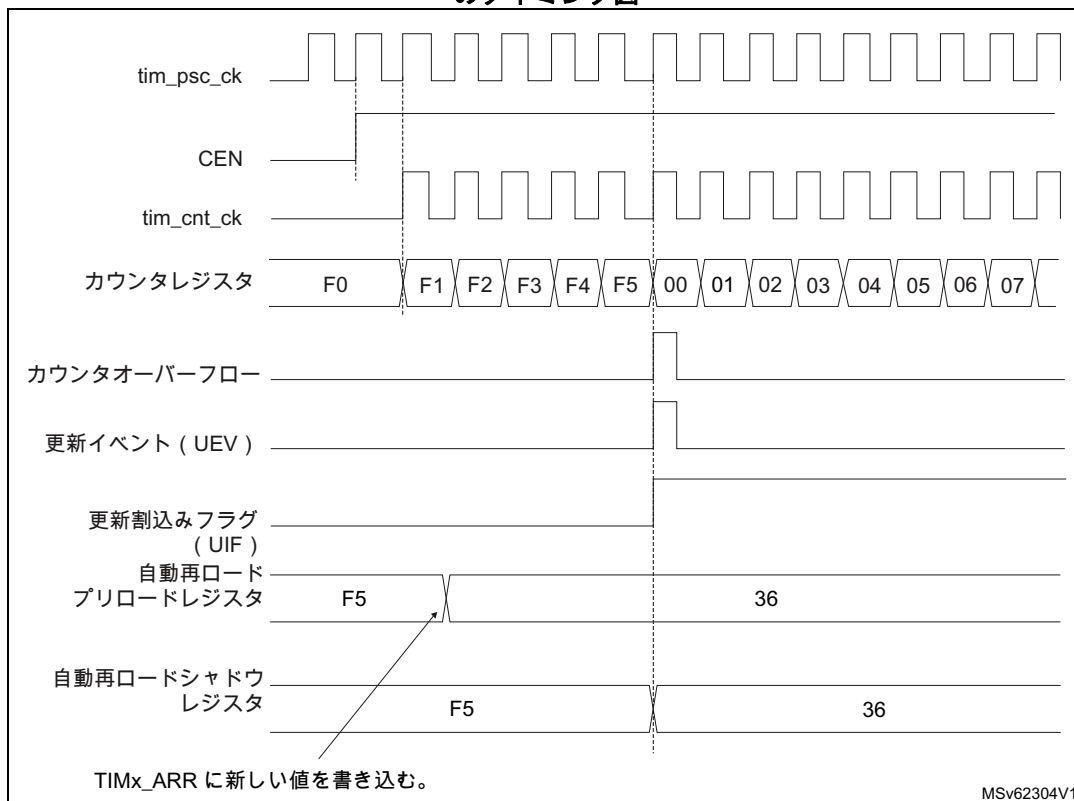


図 435. ARPE=1 (TIMx_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

更新イベントは、カウンタアンダーフローごとに、または、(ソフトウェアで、または、スレープモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることにより生成できます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 436. 内部クロック分周比が 1 の場合のカウンタのタイミング図

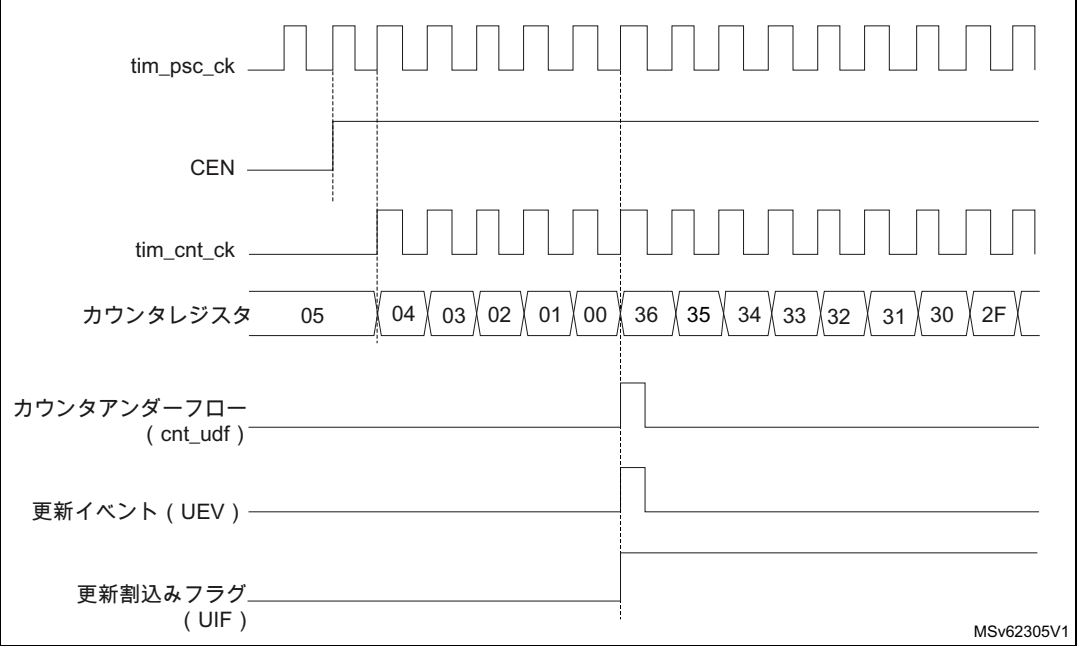


図 437. 内部クロック分周比が 2 の場合のカウンタのタイミング図

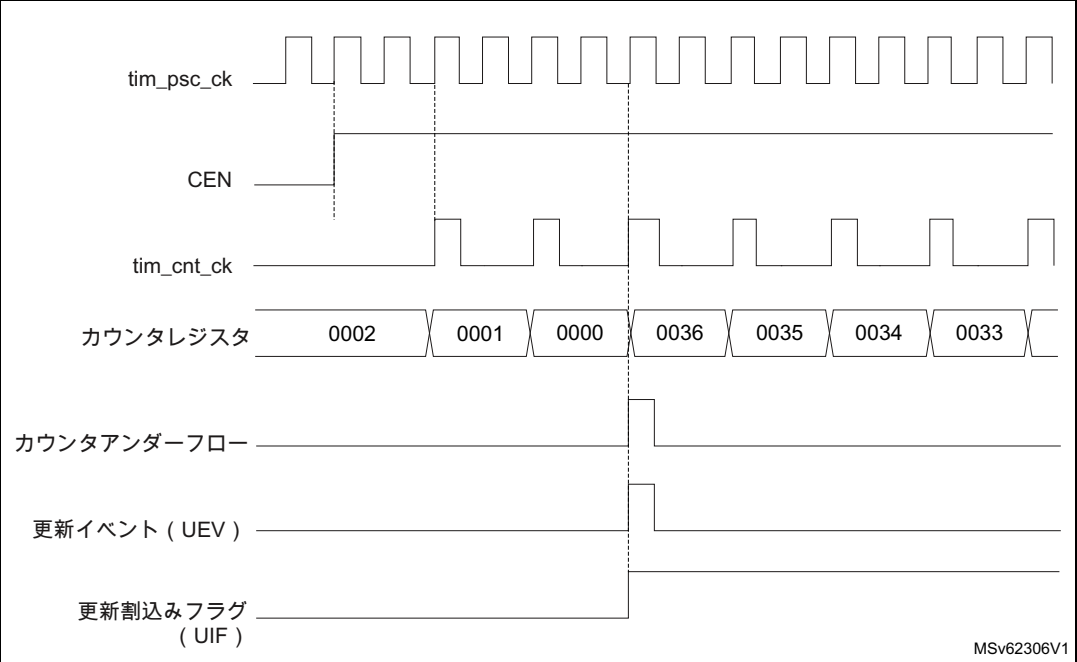


図 438. 内部クロック分周比が 4 の場合のカウンタのタイミング図

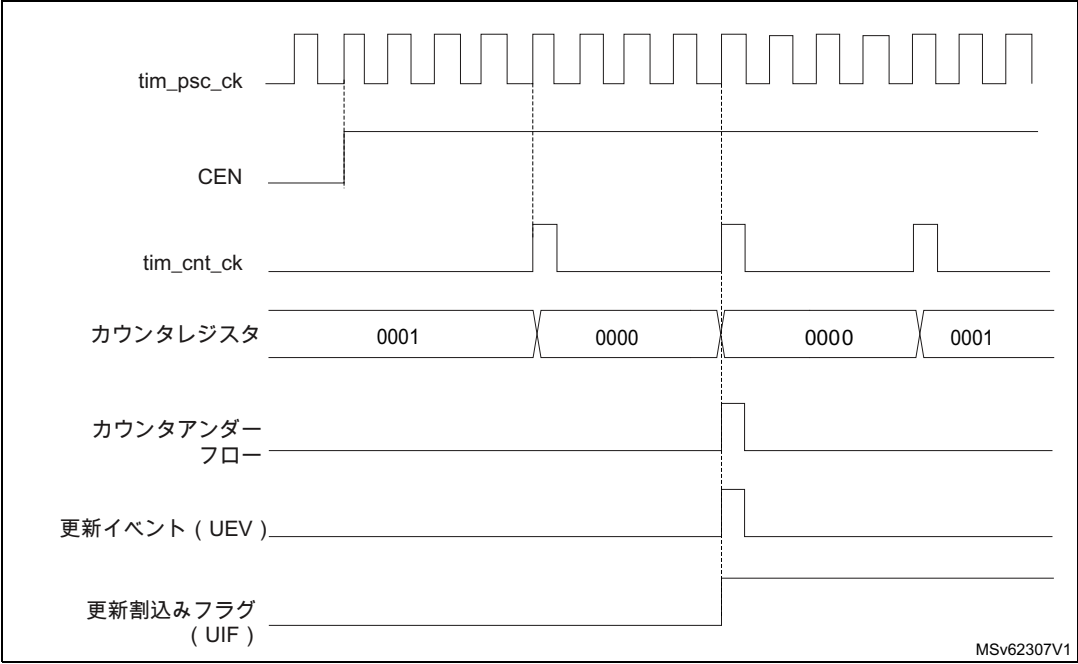


図 439. 内部クロック分周比が N の場合のカウンタのタイミング図

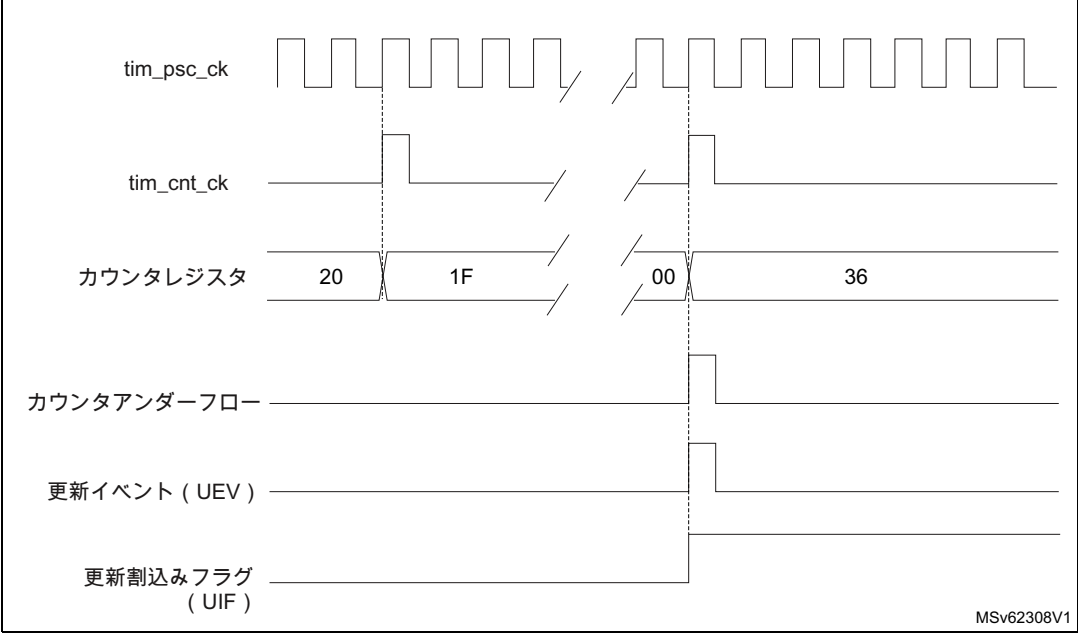
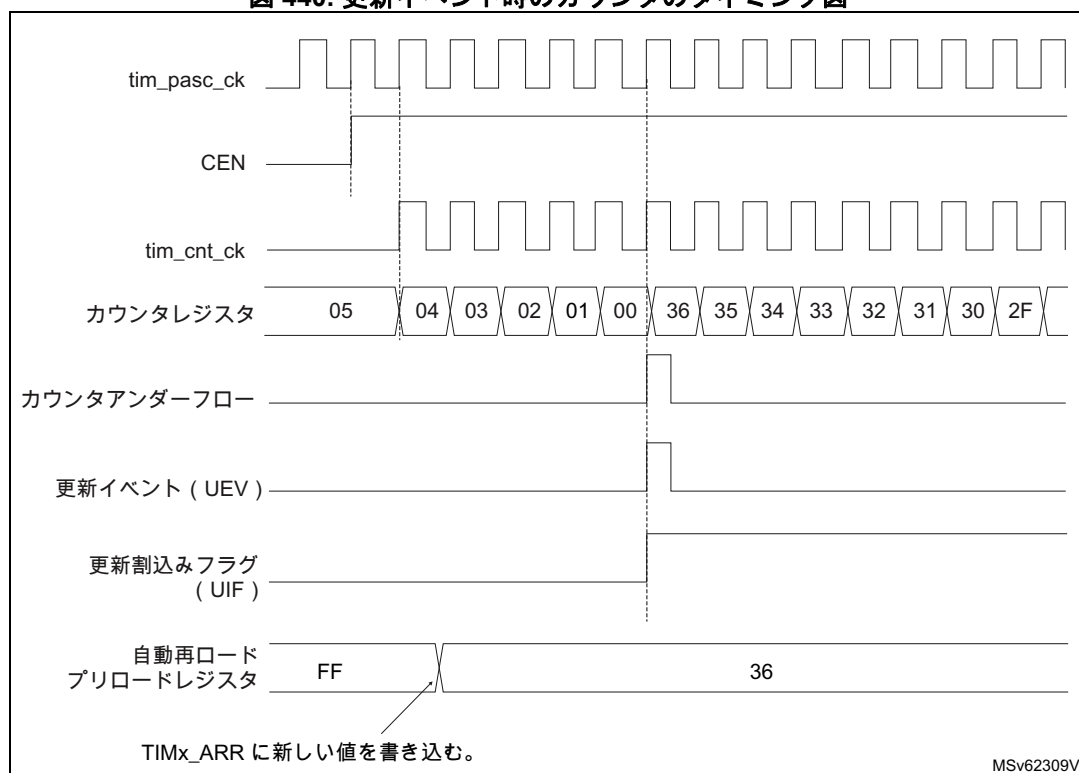


図 440. 更新イベント時のカウンタのタイミング図



センターアラインモード（アップ/ダウンカウント）

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割込みフラグは、カウンタがカウントダウンするとき（センターアラインモード 1、CMS=01）、カウンタがカウントアップするとき（センターアラインモード 2、CMS=10）、またはカウンタがカウントアップしてカウントダウンするとき（センターアラインモード 3、CMS=11）にセットされます。

このモードでは、方向ビット (TIMx_CR1 レジスタの DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、（ソフトウェアで、またはスレーブモードコントローラを使用して）TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

さらに、TIMx_CR1 レジスタの URS ビット（更新リクエスト選択）がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません（した

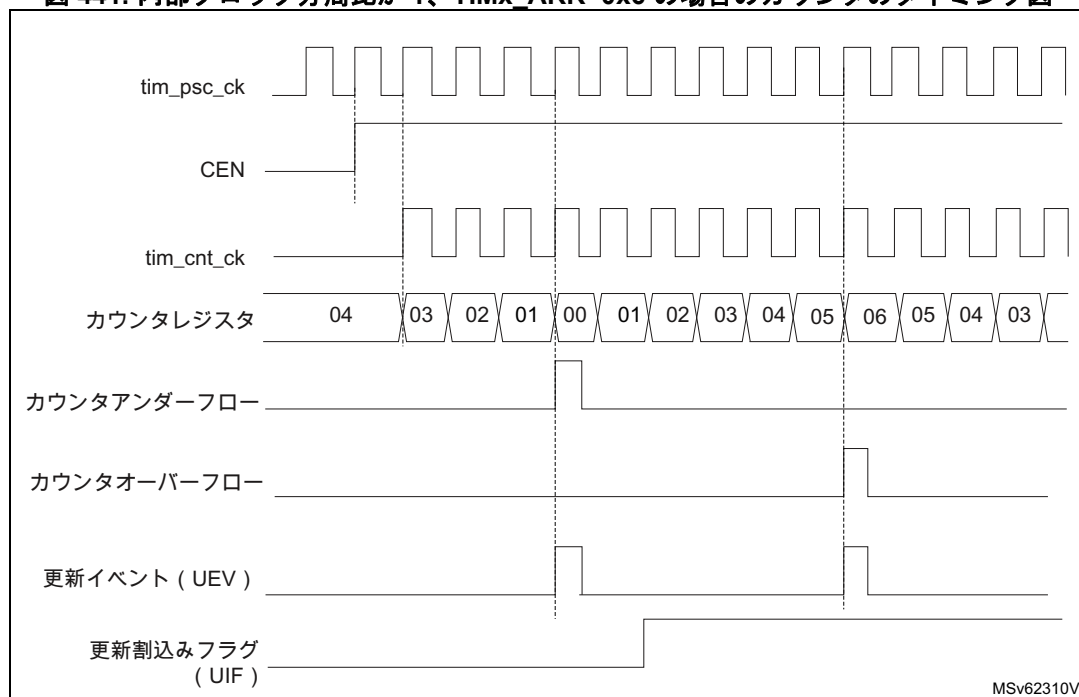
がって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 441. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 39.5.1 : 1625 ページのTIMx 制御レジスタ 1 \(TIMx_CR1\) \(x = 2~5\)](#) を参照)。

図 442. 内部クロック分周比が 2 の場合のカウンタのタイミング図

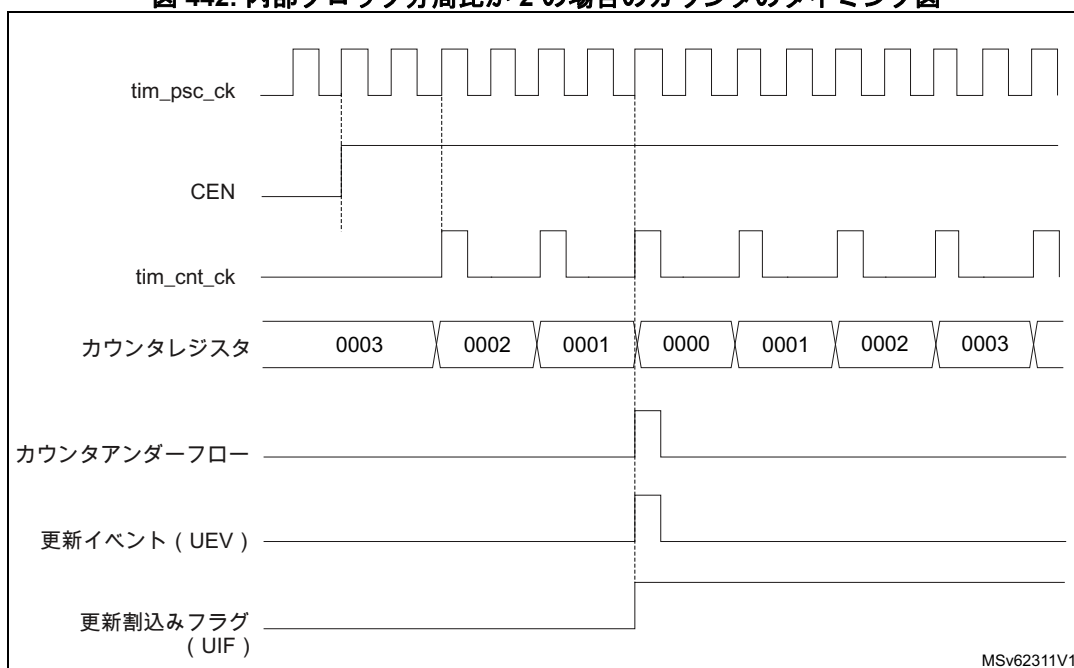
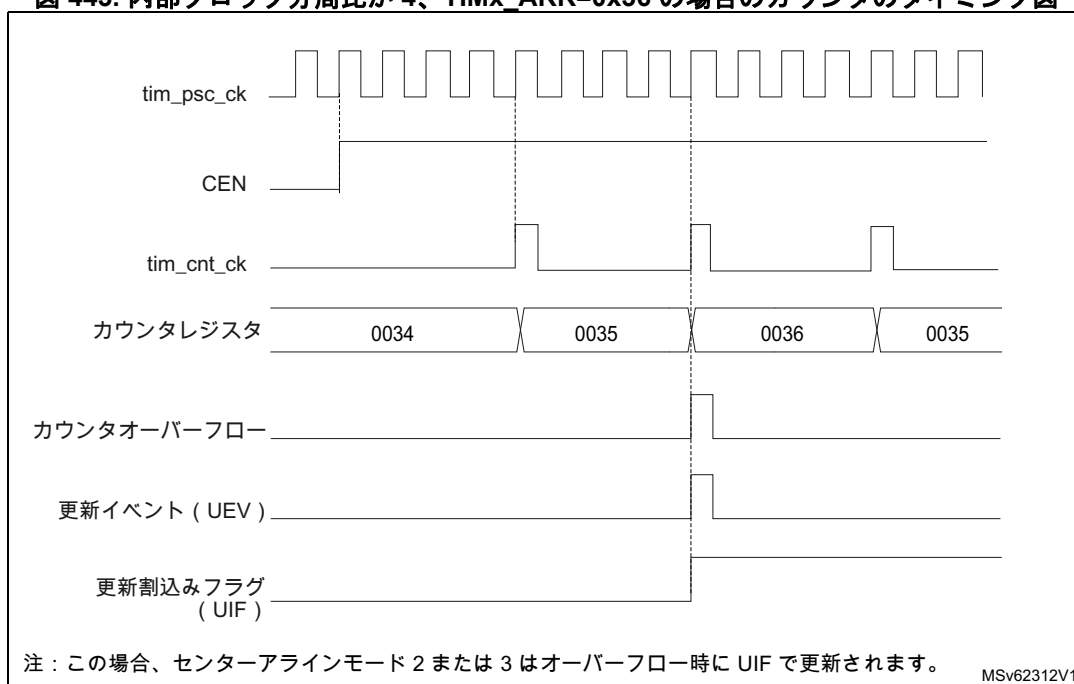


図 443. 内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 444. 内部クロック分周比が N の場合のカウンタのタイミング図

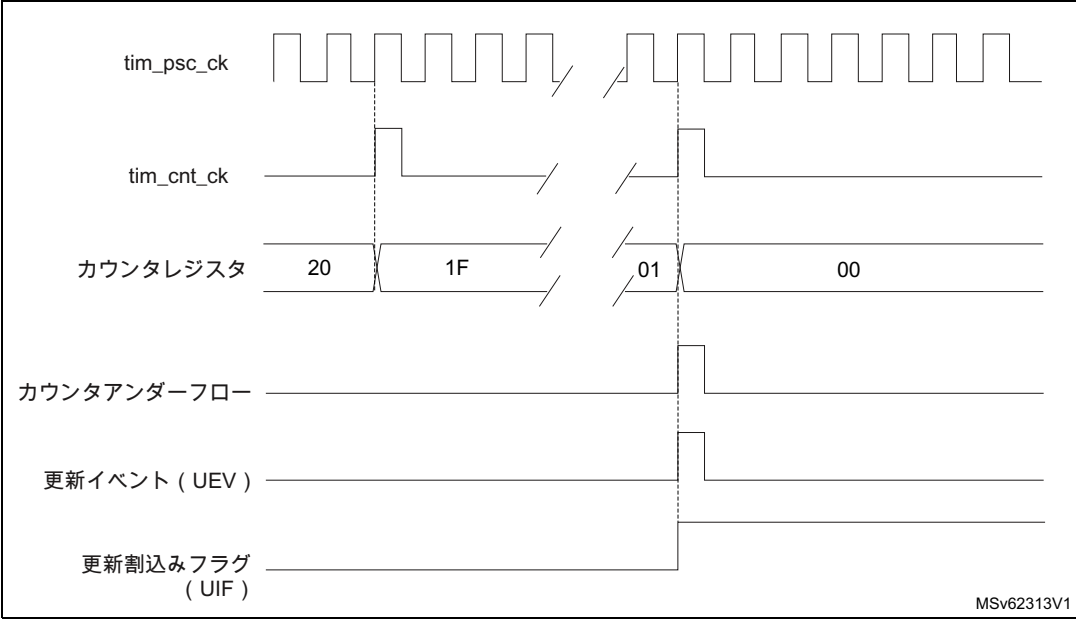


図 445. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図

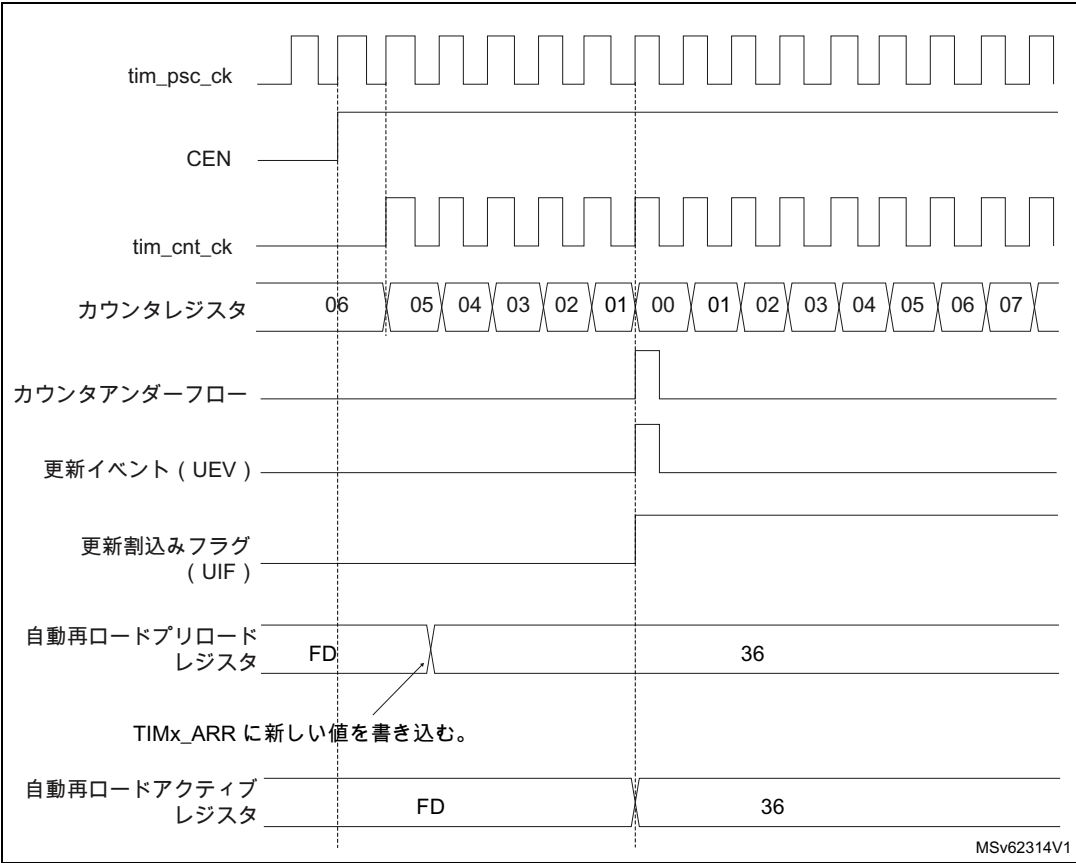
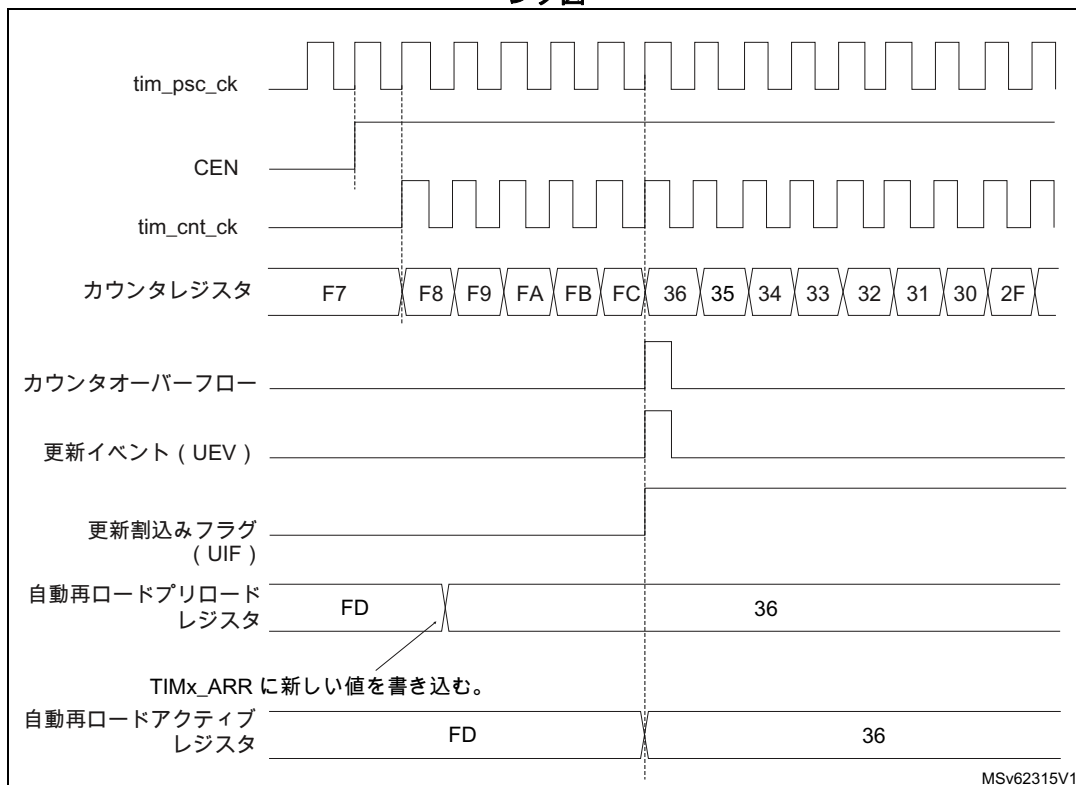


図 446. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



39.4.5 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

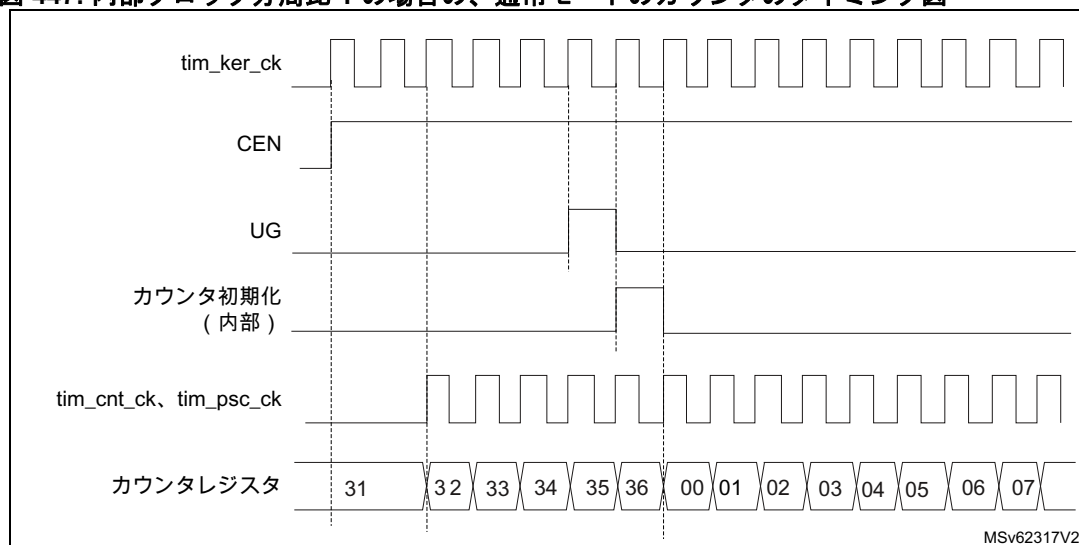
- 内部クロック (tim_ker_ck)
- 外部クロックモード 1 : 外部入力ピン (tim_ti1 または tim_ti2)
- 外部クロックモード 2 : 外部トリガ入力 (tim_etr_in)
- 内部トリガ入力 (tim_itr) : あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 1 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[1617ページのタイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

内部クロックソース (tim_ker_ck)

スレーブモードコントローラが無効の場合 (TIMx_SMCR レジスタの SMS=000)、CEN、DIR ビット (TIMx_CR1 レジスタ) と UG ビット (TIMx_EGR レジスタ) が実際の制御ビットであり、ソフトウェアでのみ変更できます (自動的にクリアされたままの UG ビットを除きます)。CEN ビットに 1 が書き込まれるとすぐに、プリスケアラには内部クロック tim_ker_ck が供給されます。

図 447 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

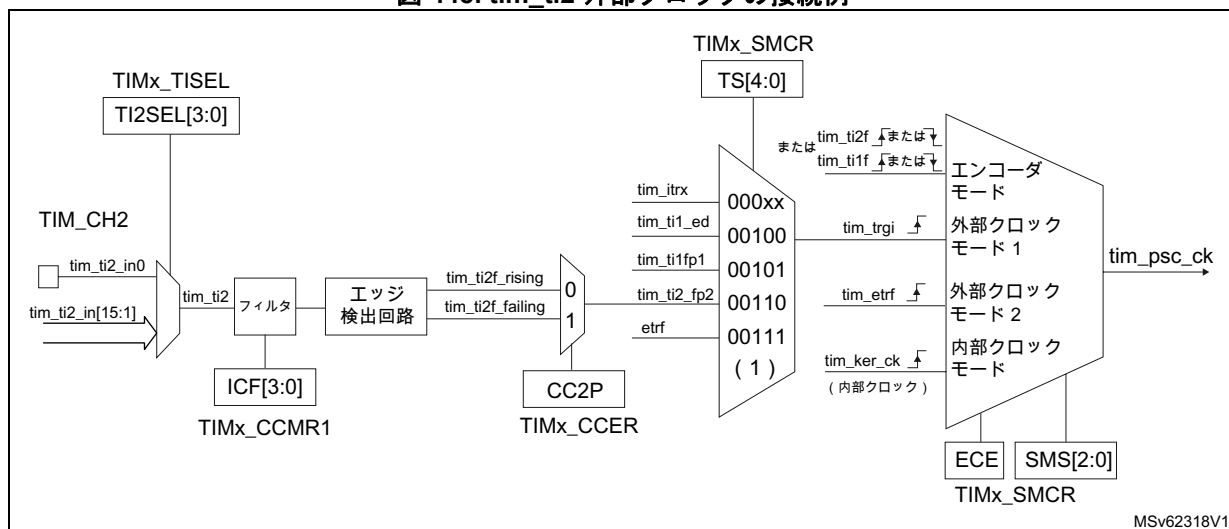
図 447. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 448. tim_ti2 外部クロックの接続例



1. 01000 から 11111 の範囲のコード : tim_itr[15:0]。

たとえば、tim_ti2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

たとえば、tim_ti2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:0] ソース（内部または外部）を選択します。
2. TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、チャンネル 2 が tim_ti2 入力の立ち上がりエッジを検出するように設定します。

3. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。

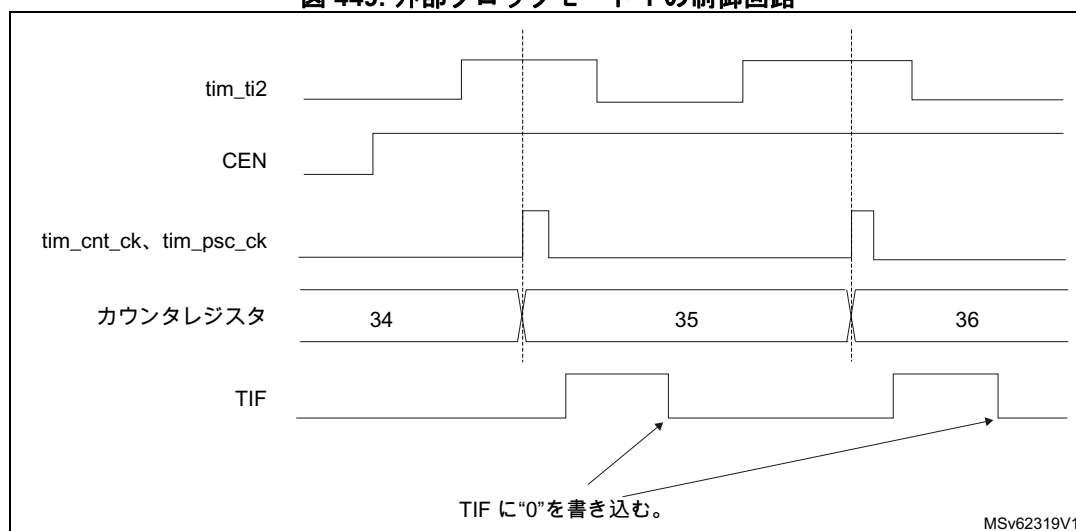
注： キャプチャプリスケアラはトリガには使用されないで、設定は不要です。

4. CC2P=0、CC2NP=0、および CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
5. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
6. TIMx_SMCR レジスタに TS=00110 を書き込むことによって、入力ソースとして tim_ti2 を選択します。
7. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

tim_ti2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

tim_ti2 の立ち上がりエッジから実際のカウンタクロックまでの間には、tim_ti2 入力の再同期回路による遅延があります。

図 449. 外部クロックモード 1 の制御回路



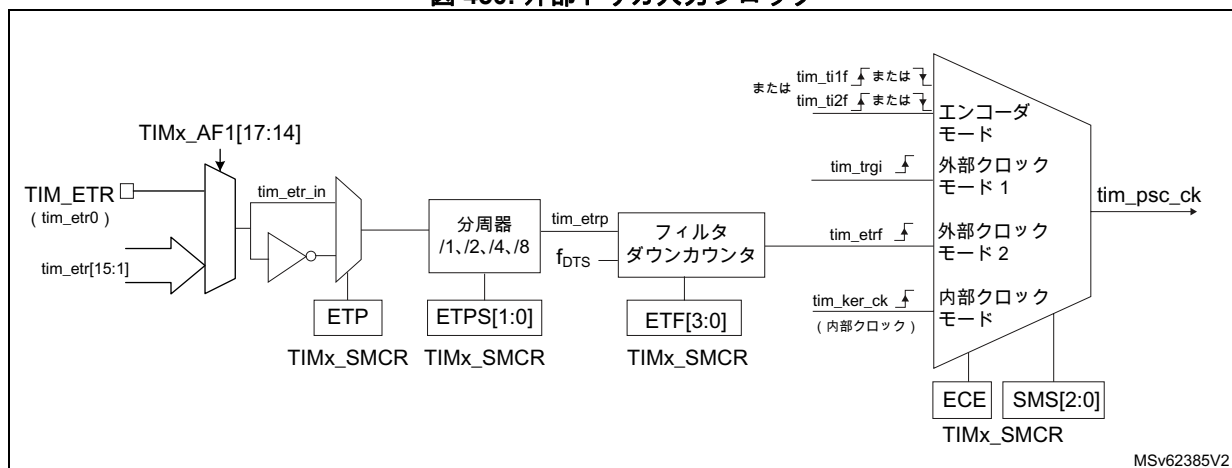
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 tim_etr_in の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 450 に、外部トリガ入力ブロックの概要を示します。

図 450. 外部トリガ入力ブロック



MSv62385V2

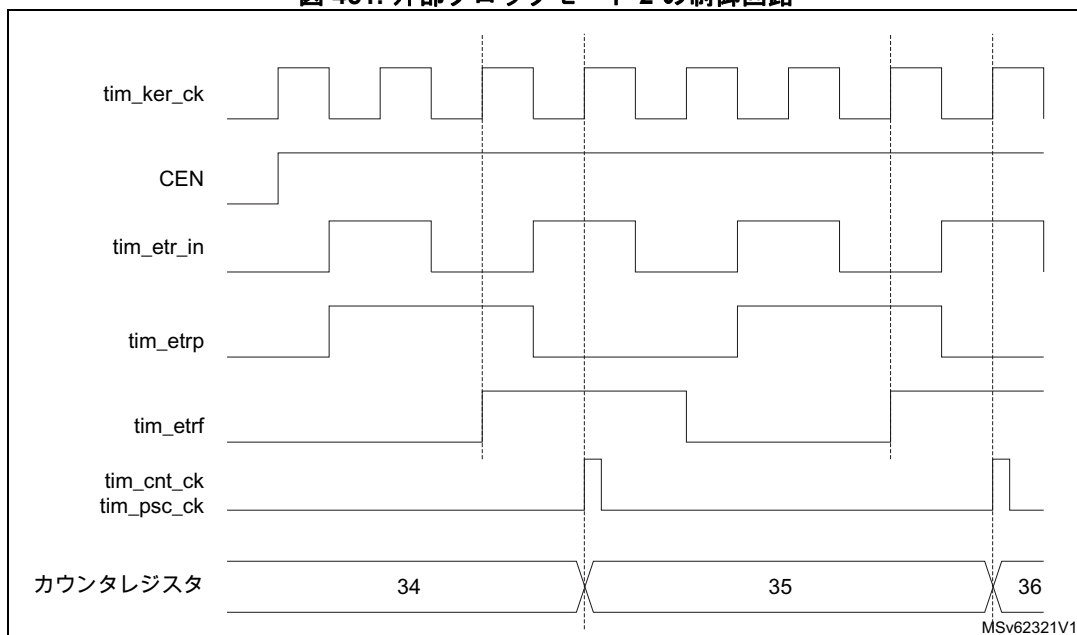
たとえば、tim_etr_in の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. TIMx_AF1 レジスタの ETRSEL[3:0] ビットで、適切な tim_etr_in ソース（内部または外部）を選択します。
2. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
3. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
4. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、tim_etr_in の立ち上がりエッジ検出を選択します。
5. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の tim_etr_in 立ち上がりエッジごとに 1 回カウントします。

tim_etr_in の立ち上がりエッジから実際のカウンタクロックまでの間に、tim_etrp 信号の再同期回路による遅延があります。結果として、カウンタで正しくキャプチャできる最大周波数は、最大で TIMxCLK 周波数の 1/4 です。ETRP 信号が高速の場合、ユーザは適切な ETPS プリスケアラ設定によって外部信号の分周比を適用する必要があります。

図 451. 外部クロックモード 2 の制御回路



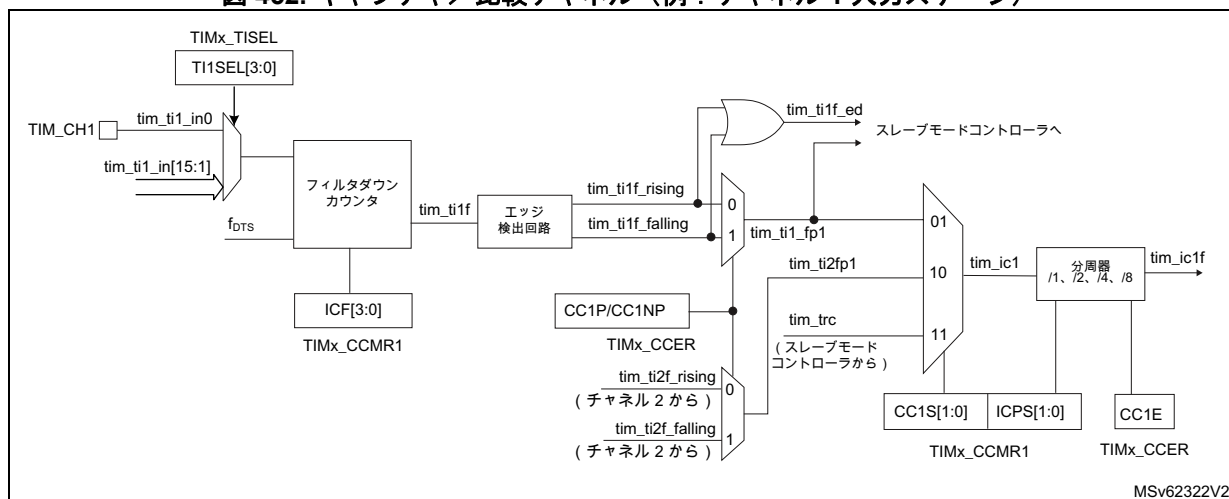
39.4.6 キャプチャ／比較チャンネル

各キャプチャ／比較チャンネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクス、プリスケータ）、および出カステージ（コンパレータと出力制御）から構成されています。

次の図に、キャプチャ／比較チャンネルの概要を示します。

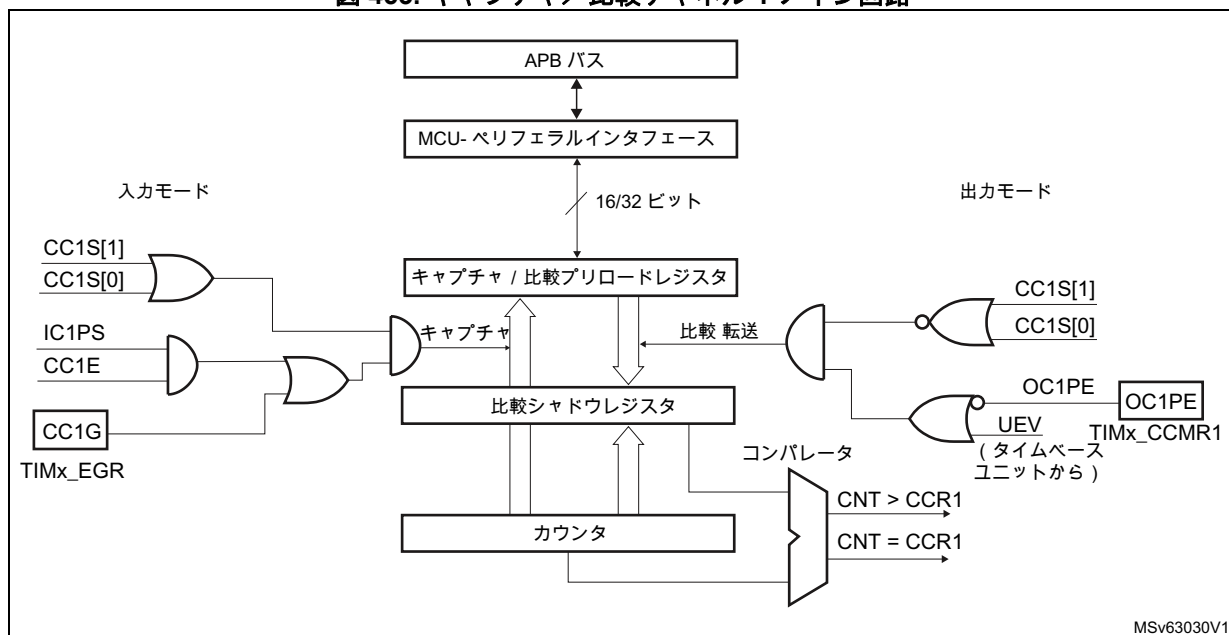
入カステージは、対応する `tim_tix` 入力をサンプリングして、フィルタリングを行った信号 `tim_tixf` を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号（`tim_tixfpy`）を生成します。この信号はプリスケータを通じて、キャプチャレジスタ（`ICxPS`）に渡されます。

図 452. キャプチャ／比較チャンネル（例：チャンネル 1 入カステージ）



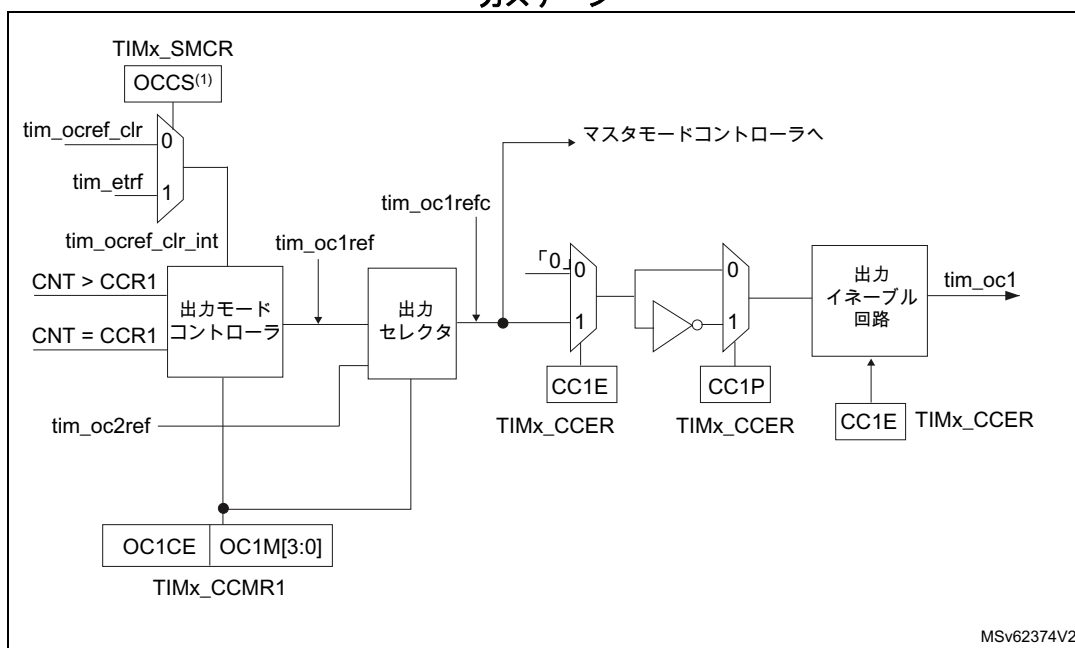
出力ステージは、その後、基準として使用される中間波形 `tim_ocxref` (アクティブハイ) を生成します。信号の極性は最終出力に影響を与えます。

図 453. キャプチャ／比較チャンネル 1 メイン回路



MSv63030V1

図 454. キャプチャ／比較チャンネル (チャンネル 1、同じくチャンネル 2、3、および 4) の出力ステージ



MSv62374V2

1. いくつかのインスタンスのみで使用できます。使用できない場合は、`tim_etrf` は直接 `tim_ocref_clr_int` に接続されます。

キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

39.4.7 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、0 を書き込むとクリアされます。

次の例は、tim_ti1 入力が増加したときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. TIMx_TISEL レジスタの TI1SEL[3:0] ビットで、適切な tim_tix_in[15:0] ソース (内部または外部) を選択します。
2. アクティブ入力を選択します。TIMx_CCR1 は tim_ti1 入力とリンクされていなければならないので、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
3. タイマに接続される信号に関して、必要な入力フィルタ時間をプログラムします (入力が増加したとき、TIMx_CCMRx レジスタの ICxF ビットによって)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、tim_ti1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
4. tim_ti1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P、CC1NP、および CC1NP ビットに“000”を書き込みます (この場合、立ち上がりエッジの選択)。
5. 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いたいので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに 00 を書き込みます)。
6. TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
7. 必要な場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE ビットをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注： IC 割込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

39.4.8 PWM 入力モード

このモードは、単一の tim_tix 入力に接続された PWM 信号の周期とデューティサイクルの両方を測定するために使用されます。

- TIMx_CCR1 レジスタは周期値（2 つの連続した立ち上がりエッジ間の間隔）を保持します。
- TIM_CCR2 レジスタはパルス幅（2 つの連続した立ち上がり立ち上がりエッジ間の間隔）を保持します。

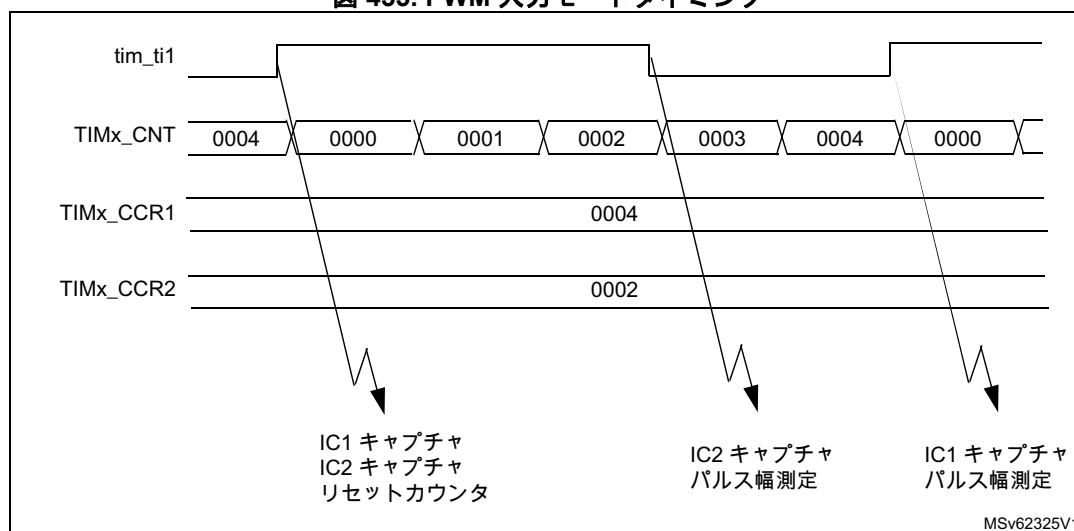
このモードは、入力キャプチャモードの特殊ケースです。設定手順は似ていますが、以下の違いがあります。

- 2 つの ICx 信号が同じ tim_tix 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

tim_ti1 に適用される PWM 信号の周期とパルス幅は、以下の手順を使用して測定できます。

1. TIMx_TISEL レジスタの TI1SEL[3:0] ビットで、適切な tim_tix_in[15:0] ソース（内部または外部）を選択します。
2. TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込むことによって（tim_ti1 を選択）、TIMx_CCR1のアクティブ入力を選択します。
3. CC1P ビットと CC1NP ビットに“0”を書き込むことによって（立ち上がりエッジでアクティブ）、tim_ti1fp1 のアクティブ極性を選択します（TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用します）。
4. TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって（tim_ti1 を選択）、TIMx_CCR2のアクティブ入力を選択します。
5. CC2P ビットに“1”を、CC2NP ビットに“0”を書き込むことによって（立ち下がりエッジでアクティブ）、tim_ti1fp2 のアクティブ極性を選択します（TIMx_CCR2 のキャプチャに使用されます）。
6. TIMx_SMCR レジスタの TS ビットに 00101 を書き込むことによって（tim_ti1fp1 を選択）、有効なトリガ入力を選択します。
7. TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
8. TIMx_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

図 455. PWM 入力モードタイミング



1. `tim_ti1fp1` と `tim_ti2fp2` のみがスレーブモードコントローラに接続されているので、PWM 入力モードは `TIMx_CH1/TIMx_CH2` 信号でのみ使用できます。

39.4.9 強制出力モード

このモード (`TIMx_CCMRx` レジスタの `CCxS` ビット = 00) では、各出力比較信号 (`tim_ocxref`、そして `tim_ocx`) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (`tim_ocxref/tim_ocx`) を強制的にアクティブレベルにするには、対応する `TIMx_CCMRx` レジスタの `OCxM` ビットに 101 を書き込む必要があります。これにより、`tim_ocxref` は強制的にハイレベルになり (`tim_ocxref` は常にアクティブハイ)、`tim_ocx` は `CCxP` 極性ビットと逆の値になります。

例: `CCxP=0` (`tim_ocx` アクティブハイ) => `tim_ocx` は強制的にハイレベルになります。

`TIMx_CCMRx` レジスタの `OCxM` ビットに 100 を書き込むことによって、`tim_ocxref` 信号を強制的にローにできます。

いずれにしても、`TIMx_CCRx` シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割込みや DMA リクエストを送信できます。これについては、出力比較モードのセクションで説明します。

39.4.10 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (`TIMx_CCMRx` レジスタの `OCxM` ビット) と出力極性 (`TIMx_CCER` レジスタの `CCxP` ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (`OCxM=000`)、アクティブにセットされるか (`OCxM=001`)、非アクティブにセットされるか (`OCxM=010`)、または反転されます (`OCxM=011`)。
- 割込みステータスレジスタのフラグをセットします (`TIMx_SR` レジスタの `CCxIF` ビット)。
- 対応する割込みマスク (`TIMx_DIER` レジスタの `CCXIE` ビット) がセットされている場合は、割込みを生成します。

- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

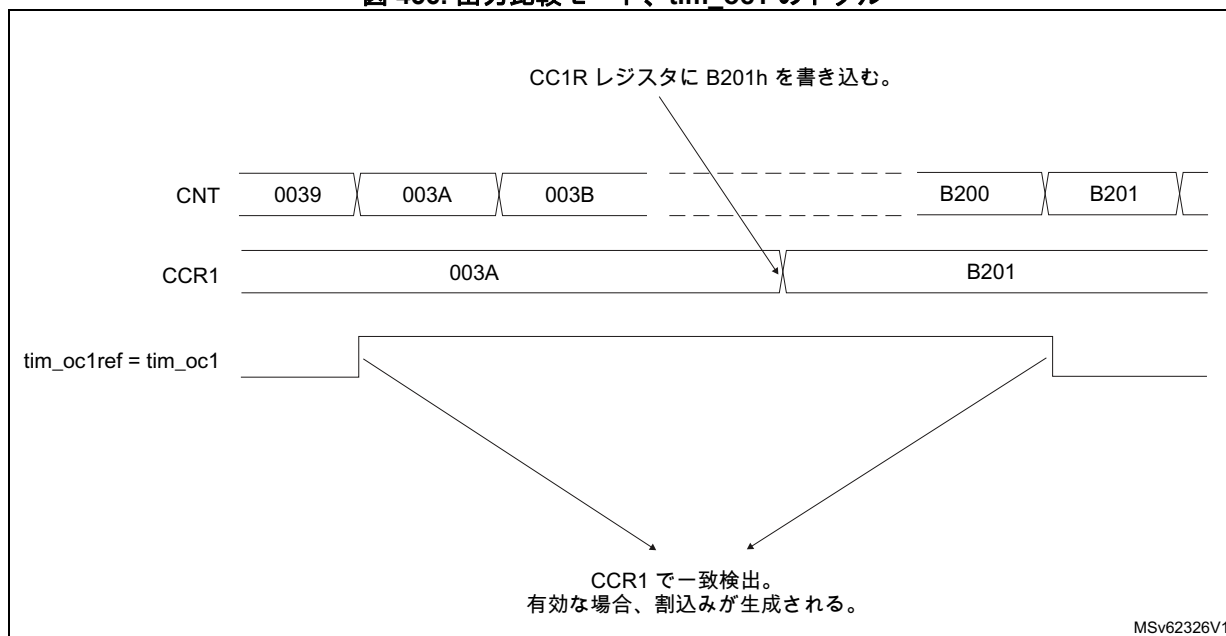
出力比較モードでは、更新イベント UEV は tim_ocxref および tim_ocx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順

- カウンタクロックを選択します (内部、外部、プリスケアラ)。
- TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
- 割込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
- 出力モードを選択します。例：
 - CNT と CCRx が一致したときに tim_ocx 出力ピンを反転するには、OCxM ビットに 0011 を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
- TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアで TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を 図 456 に示します。

図 456. 出力比較モード、tim_oc1 のトグル



39.4.11 PWM モード

パルス幅変調 (PWM) モードは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで、信号を生成するために使用されます。

PWM モードは、個々のチャネル (tim_ocx 出力ごとに PWM 1 波形) で、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) や“111” (PWM モード 2) を書き込むことで、独自に選択できます。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります、また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

tim_ocx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。これは、アクティブハイまたはアクティブローとしてプログラムできます。tim_ocx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx \leq TIMx_CNT または TIMx_CNT \leq TIMx_CCRx かどうか判断されます (カウントの方向によります)。tim_ocref_clr は、tim_etr_in または tim_ocref_clr 信号を介した外部イベントによってクリアできません。この場合、tim_ocref_clr 信号は、以下の場合にのみ、アサートされます。

- 比較一致イベントの後。
- 出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

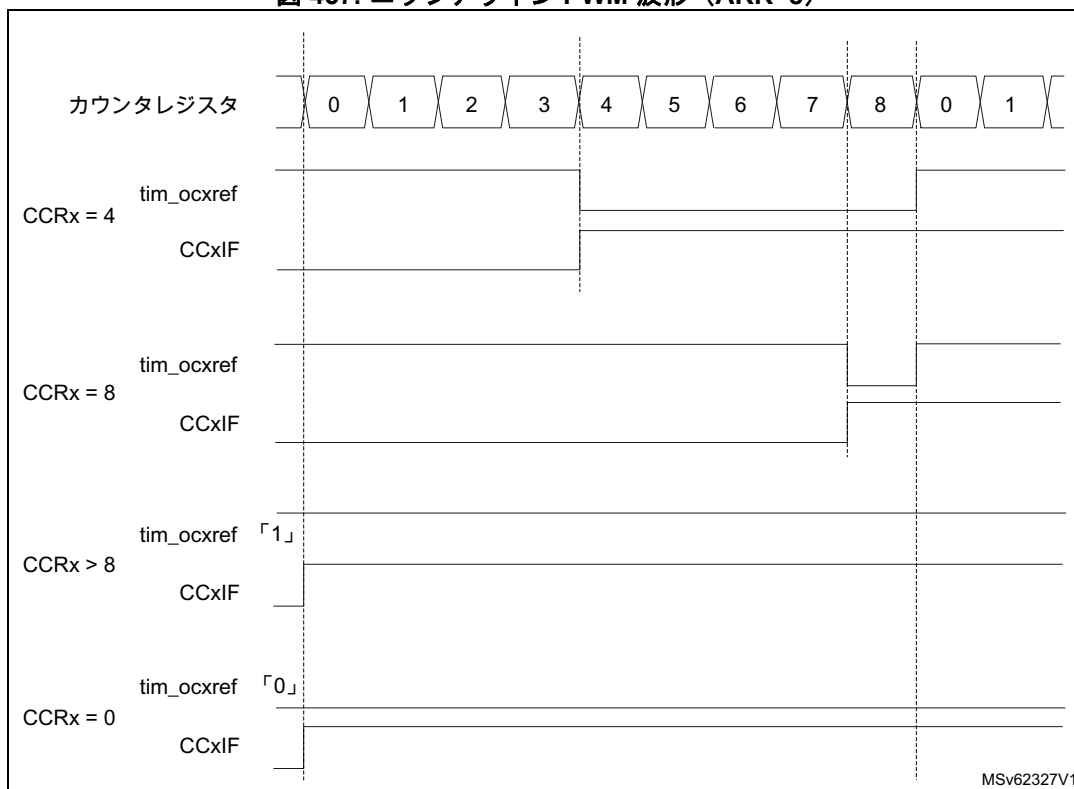
PWM エッジアラインモード

アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[1558 ページのアップカウントモード](#)を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 tim_ocxref は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、tim_ocxref は“1”に保持されます。比較値が 0 の場合、tim_ocxref は“0”に保持されます。[図 457](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 457. エッジアライン PWM 波形 (ARR=8)



ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[1561 ページのダウンカウントモード](#)を参照してください。

PWM モード 1 では、基準信号 tim_ocxref は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、tim_ocxref は 100% に保持されます。このモードでは、PWM 信号を生成することはできません。

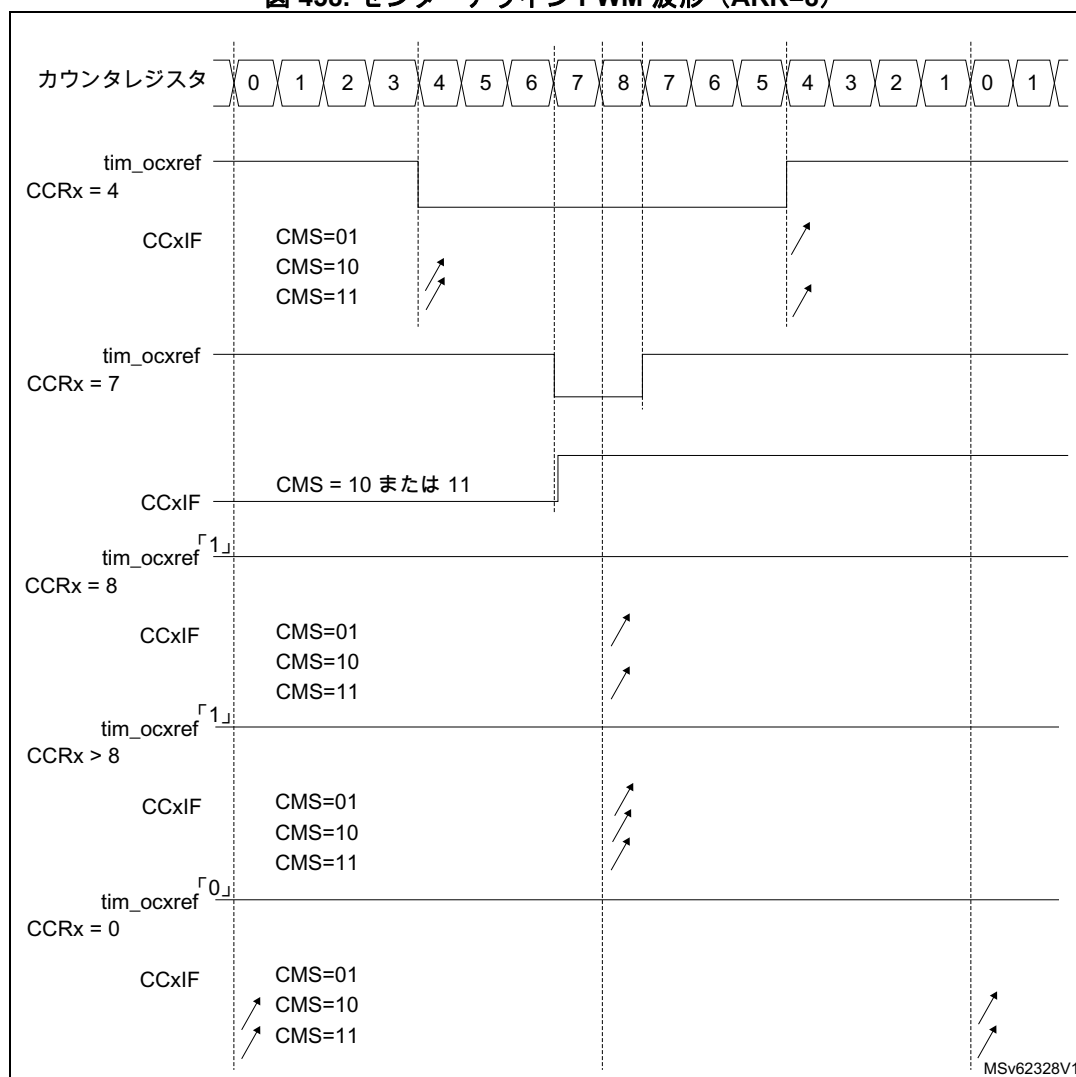
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての設定は、tim_ocxref/tim_ocx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[1564 ページのセンターアラインモード（アップ/ダウンカウント）](#)を参照してください。

図 458 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 458. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

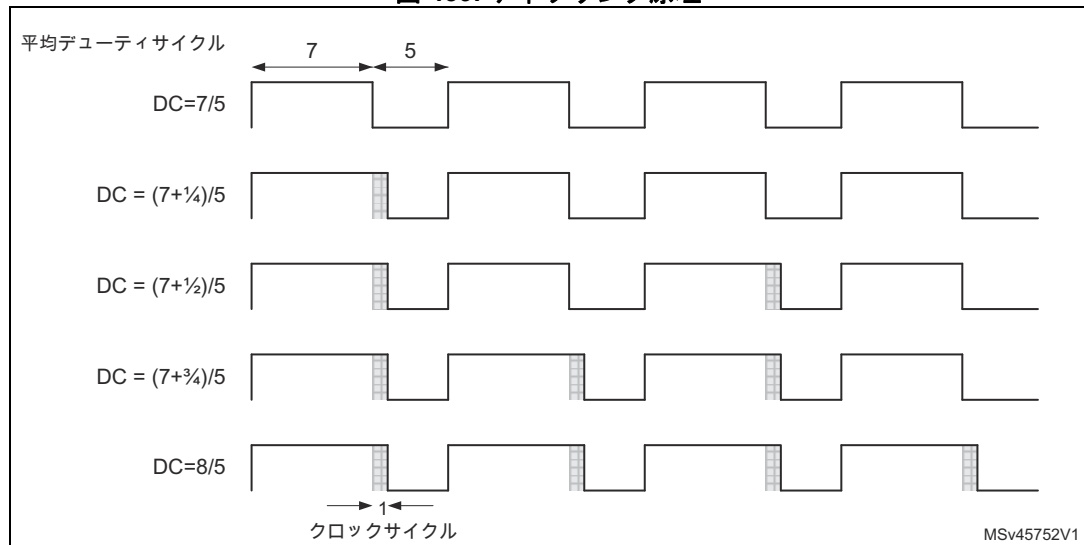
- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - 自動再ロード値より大きい値がカウンタに書き込まれた場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

ディザリングモード

PWM モードの有効分解能は、TIMx_CR1 レジスタの DITHEN ビットを使用して、ディザリングモードを有効にすることで増加できます。これは、CCR（デューティサイクル分解能の増加）と ARR（PWM 周波数分解能の増加）の両方に適用されます。

動作原理は、事前に定義されたパターンで、16 個の連続した PWM 周期にわたって実際の CCR（または ARR）値をわずかに変更する（1 タイマクロック周期を追加する、またはしない）ことです。これにより、平均のデューティサイクルまたは PWM 周期を考えると、16 倍の分解能の向上が可能になります。下記の図 459 は、4 つの連続する PWM サイクルに適用されたディザリング原理を示します。

図 459. ディザリング原理



ディザリングモードが有効になっている場合、レジスタのコーディングは以下のように変更されます（たとえば、図 460 を参照）。

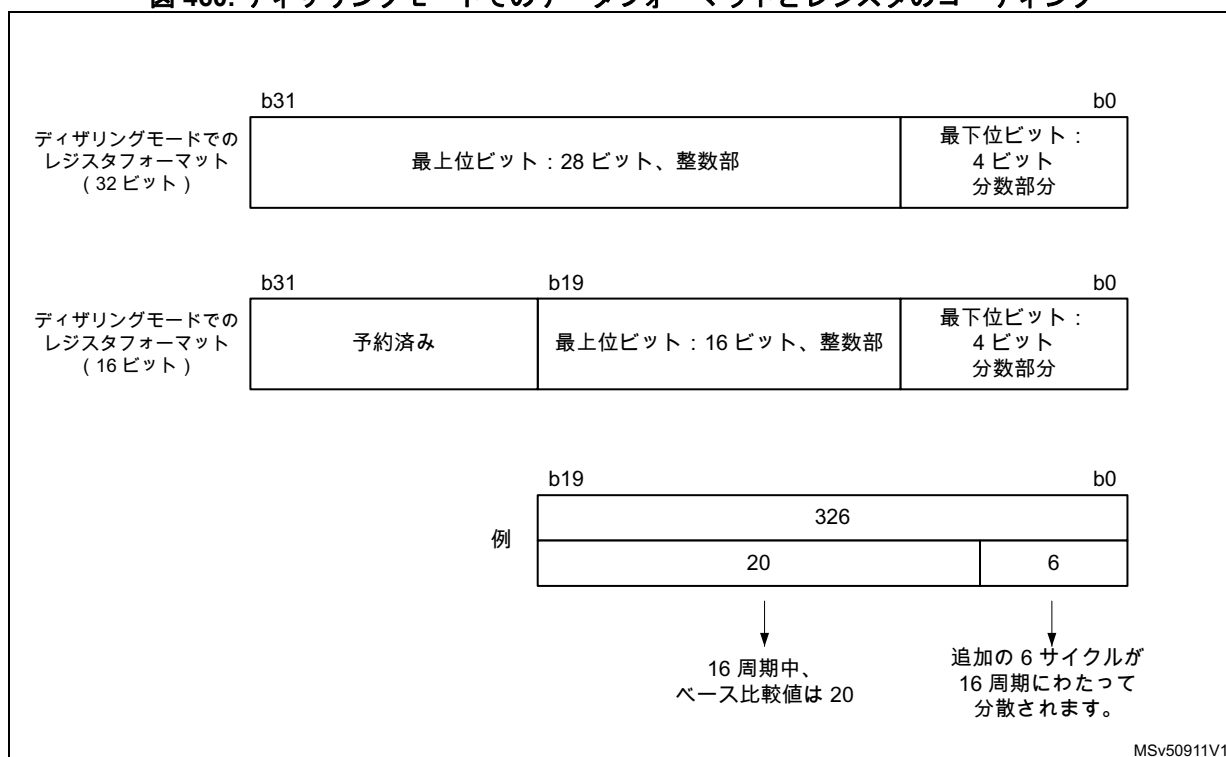
- 下位 4 ビットは、拡張分解能部分（分数部分）用です。
- 上位ビットは、4 桁だけ左にシフトされ、ベース値用のコーディングになります。16 ビットモードでは、16 ビットのフォーマットが維持されます。

注：

DITHEN ビットをリセットするときは、次の手順に従う必要があります。

1. CEN および ARPE ビットをリセットする必要があります。
2. DITHEN ビットをリセットする必要があります。
3. CCIF フラグをクリアする必要があります。
4. CEN ビットをセットすることができます（最終的に ARPE = 1 とともに）。

図 460. ディザリングモードでのデータフォーマットとレジスタのコーディング



最小周波数は次の計算式で与えられます。

$$\text{分解能} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

$$\text{ディザリングモード無効時 : } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

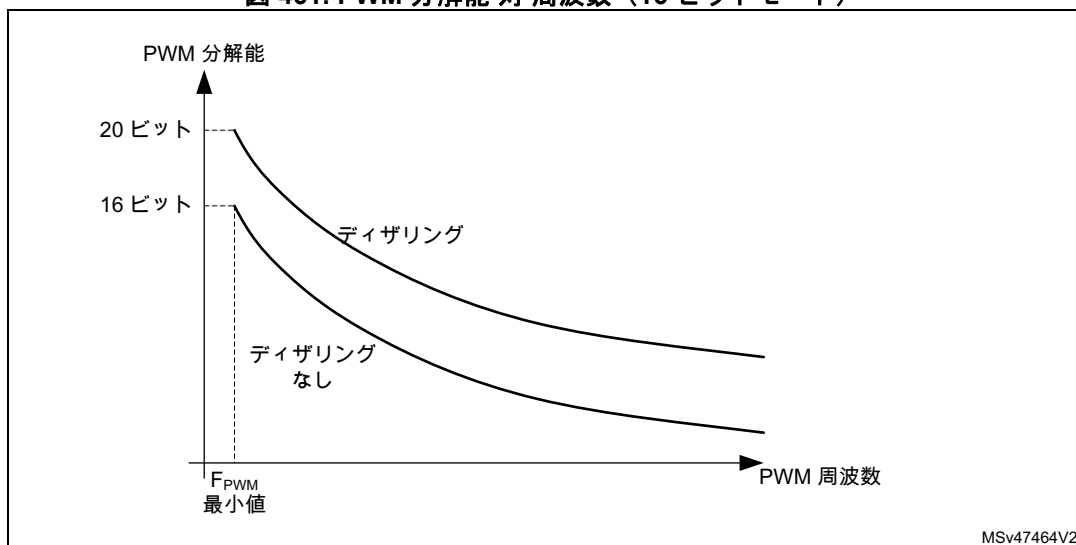
$$\text{ディザリングモード (16 ビットタイマ) : } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

$$\text{ディザリングモード (32 ビットタイマ) : } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{268435454 + \frac{15}{16}}$$

注 : 16 ビットタイマの場合、TIMx_ARR および TIMx_CCRy の最大値は、ディザリングモードでは 0xFFFFEF に制限されています (整数部では 65534、ディザリング部分では 15 に対応します)。
32 ビットタイマの場合、TIMx_ARR および TIMx_CCRy の最大値は、ディザリングモードでは 0xFFFFFEEF に制限されています (整数部では 268435454、ディザリング部分では 15 に対応します)。

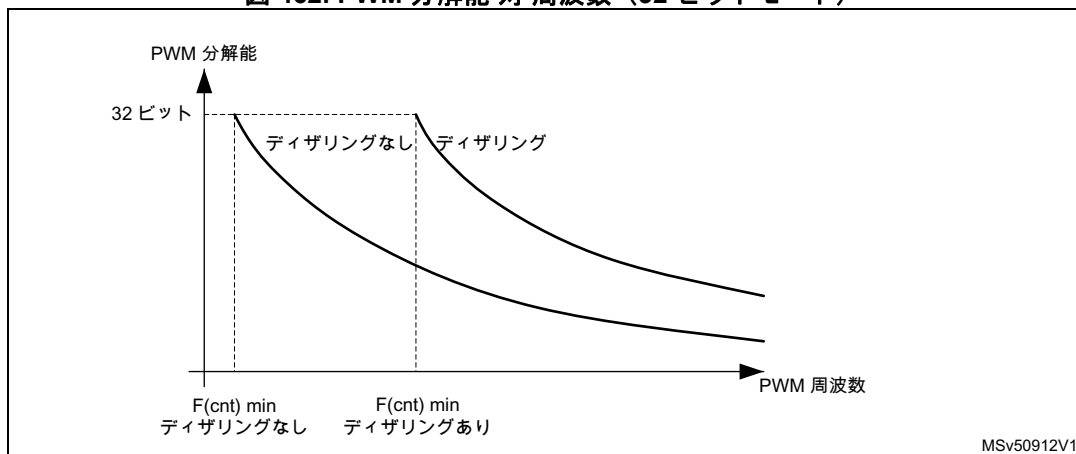
下記の図 461 と図 462 に示すように、ディザリングモードは PWM 分解能を増加するために使用されます。

図 461. PWM 分解能 対 周波数 (16 ビットモード)



MSv47464V2

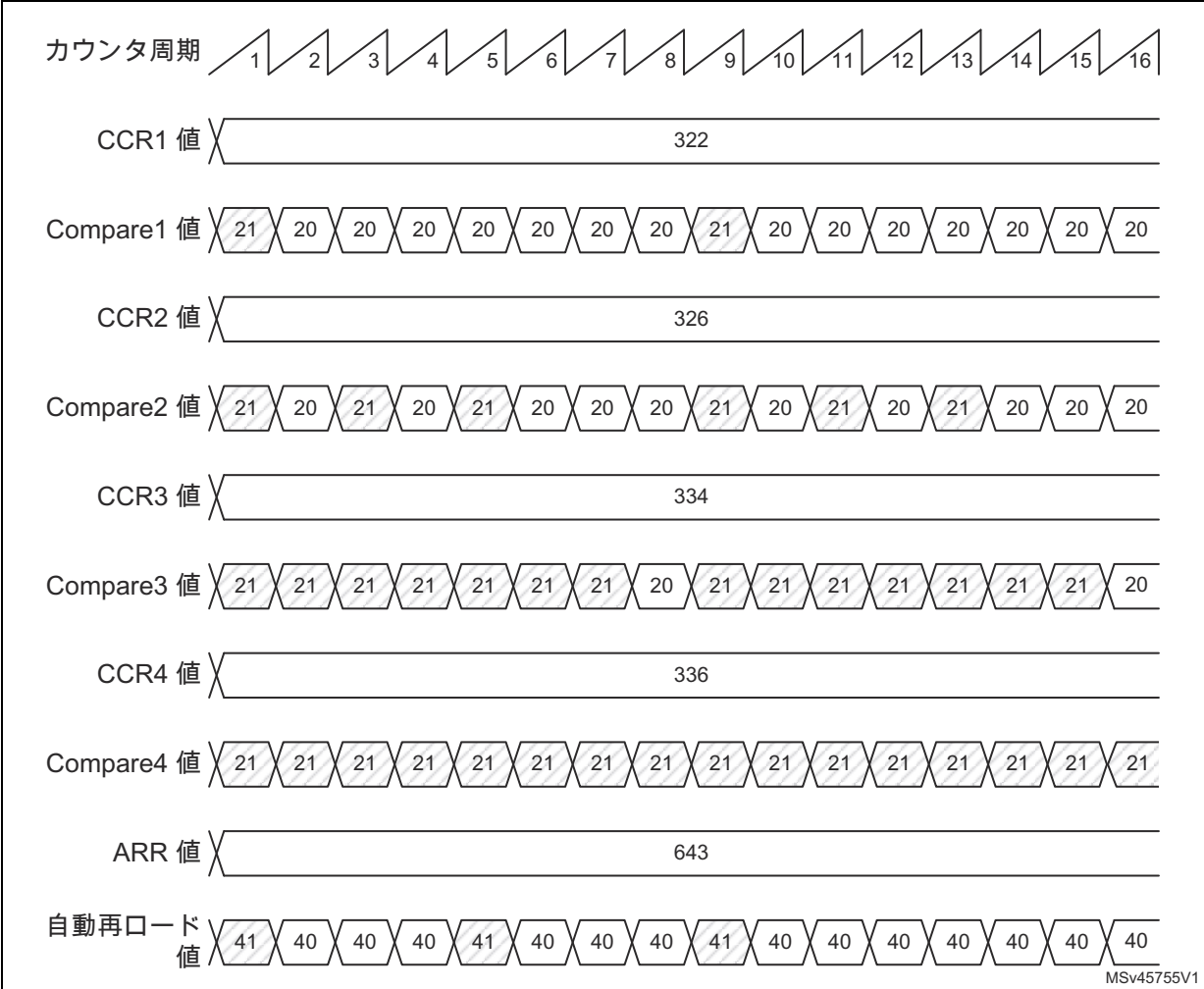
図 462. PWM 分解能 対 周波数 (32 ビットモード)



MSv50912V1

下記の図 463 で説明するように、デューティサイクルや周期の変更は、連続した 16 周期にわたって広がります。

図 463. PWM ディザリングのパターン



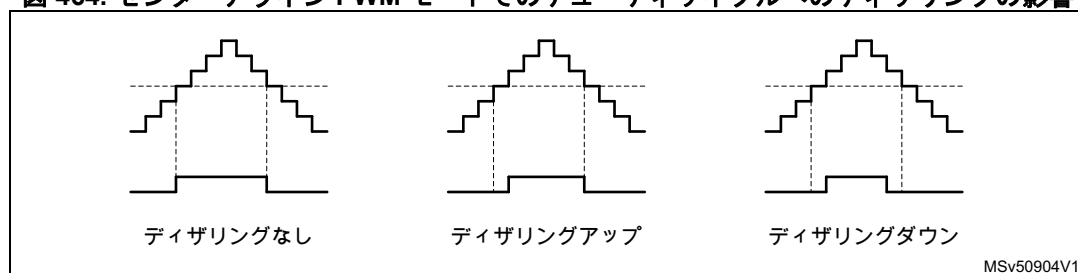
自動再ロード値と比較値の増分は、下記の表 406 に記載された特定のパターンに従って分布します。ディザリングシーケンスは、増分が可能な限り均等に分散され、全体のリップルが最小になるように行われます。

表 406. CCR および ARR レジスタの変更のディザリングパターン

LSB 値	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

ディザリングモードはセンターアライン PWM モード (TIMx_CR1 レジスタの CMS ビットが“00”に等しくないとき) でも使用できます。この場合、ディザリングパターンは、以下の図 464 に示すように、アップおよびダウンカウントフェーズを考慮して、8 つの連続した PWM 周期にわたって適用されます。

図 464. センターアライン PWM モードでのデューティサイクルへのディザリングの影響



下記の表 407 は、センタアライン PWM モードでどのようにディザリングパターンが追加されるかを示しています。

表 407. センタアライン PWM モードでの CCR レジスタ変更のディザリングパターン

LSB 値	PWM 周期															
	1		2		3		4		5		6		7		8	
	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn	アップ	Dn
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

39.4.12 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって 2 つのセンタアライン PWM 信号の生成を可能にします。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx_CCRx レジスタペアで決定されます。1 つ目のレジスタがアップカウント時の PWM を制御し、2 つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- tim_oc1refc (または tim_oc2refc) は、TIMx_CCR1 および TIMx_CCR2 によって制御されます。
- tim_oc3refc (または tim_oc4refc) は、TIMx_CCR3 および TIMx_CCR4 によって制御されます。

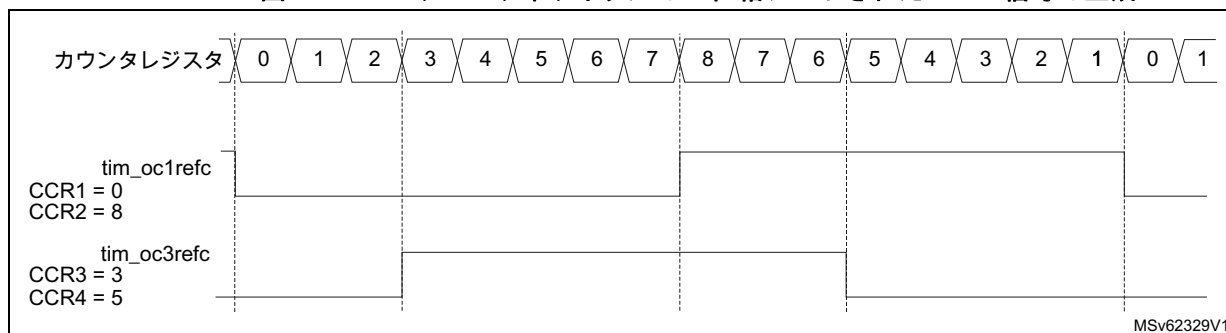
非対称 PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2 チャンネルごとに選択できます (CCR レジスタペアごとに 1 つの tim_ocx 出力)。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

特定のチャンネルが非対称の PWM チャンネルとして使用されると、その 2 次チャンネルも使用できます。たとえば、tim_oc1refc 信号がチャンネル 1 (非対称 PWM モード 1) に生成されると、チャンネル 2 の tim_oc2refc 信号、または非対称 PWM モード 2 の結果として得られる tim_oc2refc 信号を出力できます。

図 465 は、非対称 PWM モードを使用して生成される信号の例を表します（チャンネル 1 から 4 は非対称 PWM モード 2 として設定されます）。

図 465. 50% デューティサイクルの 2 位相シフトされた PWM 信号の生成



39.4.13 組合せ PWM モード

組み合わせ PWM モードでは、2 つのエッジラインまたはセンターライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2 つの TIMx_CCRx レジスタで決定されます。結果として得られる信号 tim_ocxrefc は、2 つの PWM 基準信号の OR または AND による論理結合から成ります。

– tim_oc1refc（または tim_oc2refc）は、TIMx_CCR1 および TIMx_CCR2 によって制御されます。

– tim_oc3refc（または tim_oc4refc）は、TIMx_CCR3 および TIMx_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1100”（組み合わせ PWM モード 1）または“1101”（組み合わせ PWM モード 2）を書き込むことによって、2 チャンネルごとに選択できます（CCR レジスタペアごとに 1 つの tim_ocx 出力）。

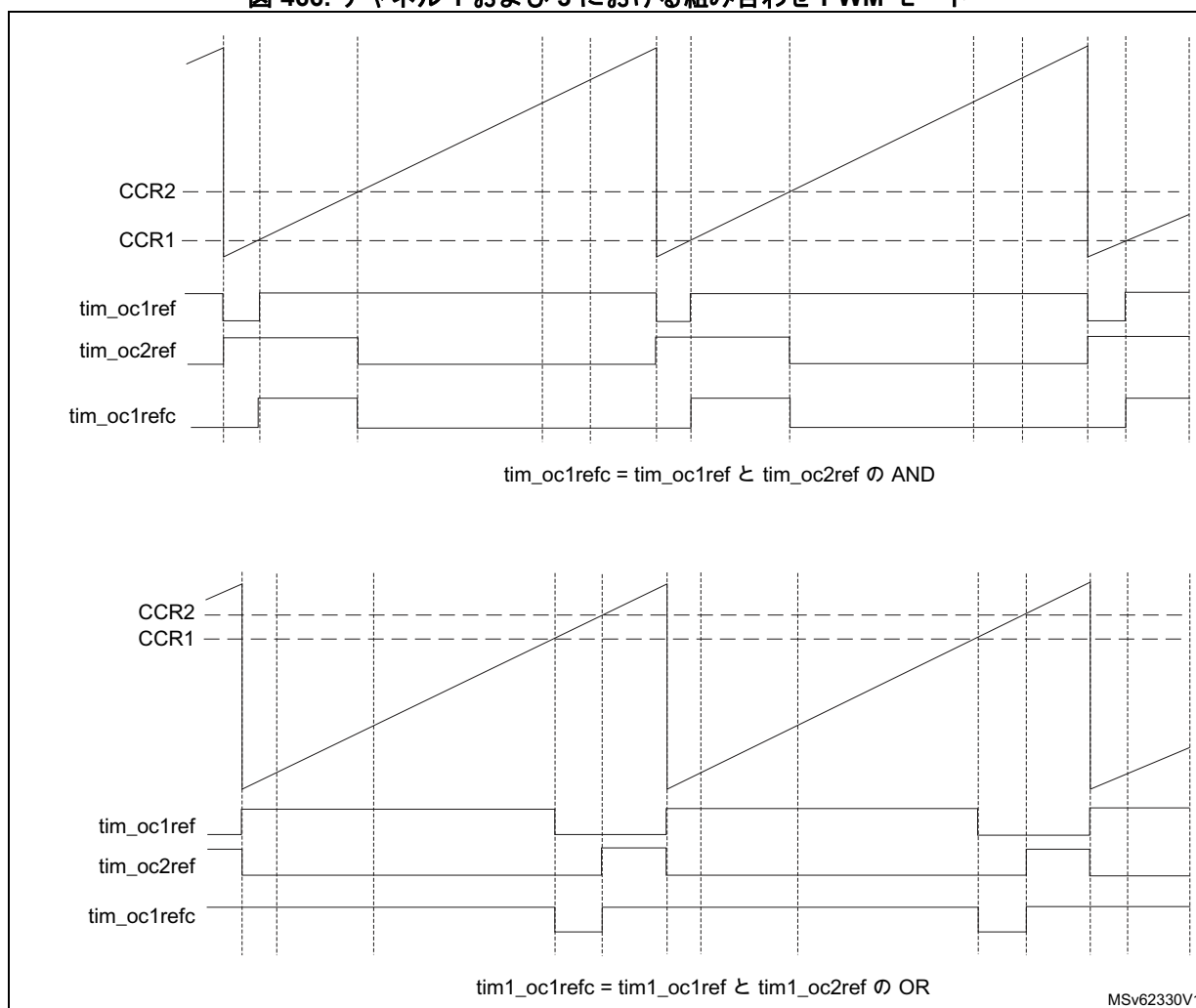
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、2 次チャンネルを反対の PWM モードに設定する必要があります（たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします）。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

図 466 は、次の設定で取得可能な組み合わせ PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合
- チャンネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 4 が PWM モード 1 で設定されている場合

図 466. チャンネル 1 および 3 における組み合わせ PWM モード



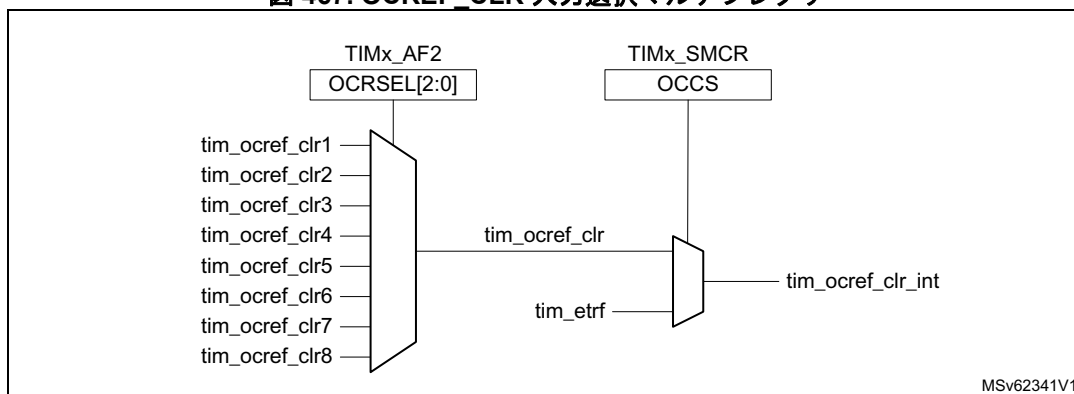
39.4.14 外部イベントによる tim_ocxref 信号のクリア

特定のチャンネルの tim_ocxref 信号は tim_ocref_clr_int 入力にハイレベルを適用するとクリアされます（対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする）。tim_ocxref は、次の PWM サイクルでアクティブ状態に移るまで、ローのままになります。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

tim_ocref_clr_int のソースは、OCREF クリア選択機能の実装に依存します。[セクション 39.3 : TIM2/TIM3/TIM4/TIM5 の実装](#) を参照してください。

OCREF クリア選択機能が実装されている場合は、tim_ocref_clr_int は、TIMx_SMCR レジスタで OCCS ビットを設定することで、tim_ocref_clr 入力と tim_etr_in 入力（フィルタ後の tim_etr_in）の間で選択できます。tim_ocref_clr 入力は、以下の [図 467](#) に示すように、TIMx_AF2 レジスタの OCRSEL[2:0] ビットフィールドを使用して複数の tim_ocref_clr[7:0] 入力から選択できます。

図 467. OCREF_CLR 入力選択マルチプレクサ



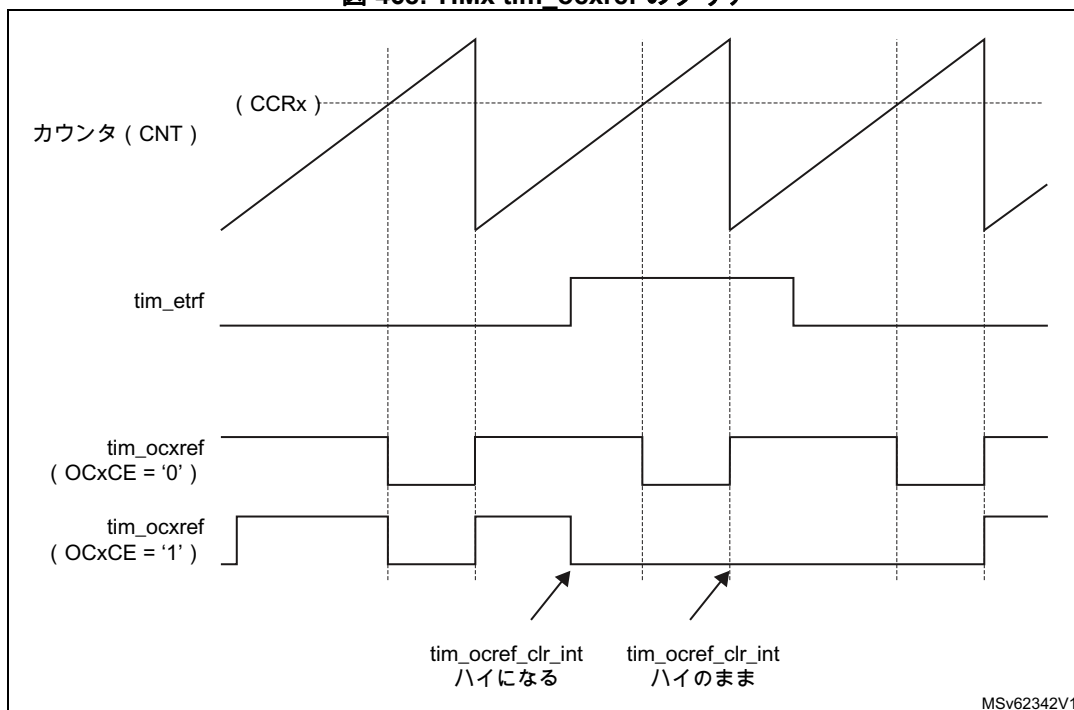
OCREF クリア選択機能が実装されていない場合は、tim_ocref_clr_int 入力は直接 tim_etrif 入力に接続されます。

たとえば、tim_ocref_clr_int 信号をコンパレータの出力に接続して、電流処理に使用することができます。この場合、tim_etr_in は次のように設定する必要があります。

1. 外部トリガプリスケラをオフ状態に保持する必要があります。すなわち、TIMx_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。

図 468 に、OCxCE イネーブルビットの両方の値について、tim_etrif 入力がハイになったときの tim_ocxref 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 468. TIMx tim_ocxref のクリア



注： 100% デューティサイクルの PWM の場合 (CCR_x > ARR の場合)、次のカウンタオーバーフローで tim_ocxref が再度有効になります。

39.4.15 ワンパルスモード

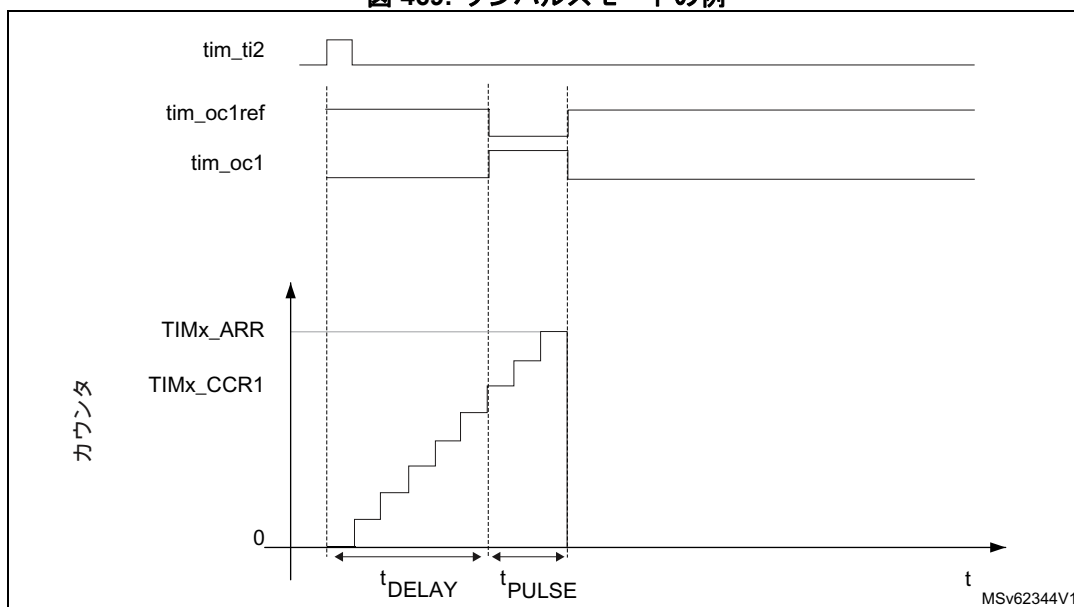
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIM_x_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- $CNT < CCR_x \leq ARR$ (特に、 $0 < CCR_x$)

図 469. ワンパルスモードの例



たとえば、tim_ti2 入力ピンで立ち上がりエッジが検出されたときに、tim_oc1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

tim_ti2fp2 をトリガ 1 として使用します。

1. TIM_x_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:0] ソース (内部または外部) を選択します。
2. TIM_x_CCMR1 レジスタに CC2S=01 を書き込むことによって、tim_ti2fp2 を tim_ti2 に配置します。
3. tim_ti2fp2 は、立ち上がりエッジを検出して、TIM_x_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込みます。
4. tim_ti2fp2 をスレーブモードコントローラのトリガ (tim_trgi) として構成するために、TIM_x_SMCR レジスタの TS=“00110”を書き込みます。
5. tim_ti2fp2 を使用してカウンタを開始するために、TIM_x_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタに OC1M=111 を書き込むことによって、PWM モード 2 を有効にする必要があります。オプションで、TIMx_CCMR1 レジスタに OC1PE=1 を書き込み、TIMx_CR1 レジスタの ARPE ビットを書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込む必要があります。次に、UG ビットをセットすることによって更新を生成し、tim_ti2 で外部トリガイイベントを待つ必要があります。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローにする必要があります。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊ケース : tim_ocx 高速イネーブル :

ワンパルスモードでは、tim_tix 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、tim_ocxref (および tim_ocx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

39.4.16 再トリガ可能なワンパルスモード

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 39.4.15](#)で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

タイマはスレーブモードである必要があり、このときビットは TIMx_SMCR レジスタで SMS[3:0] = 「1000」 (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が「1000」または「1001」にセットされています。

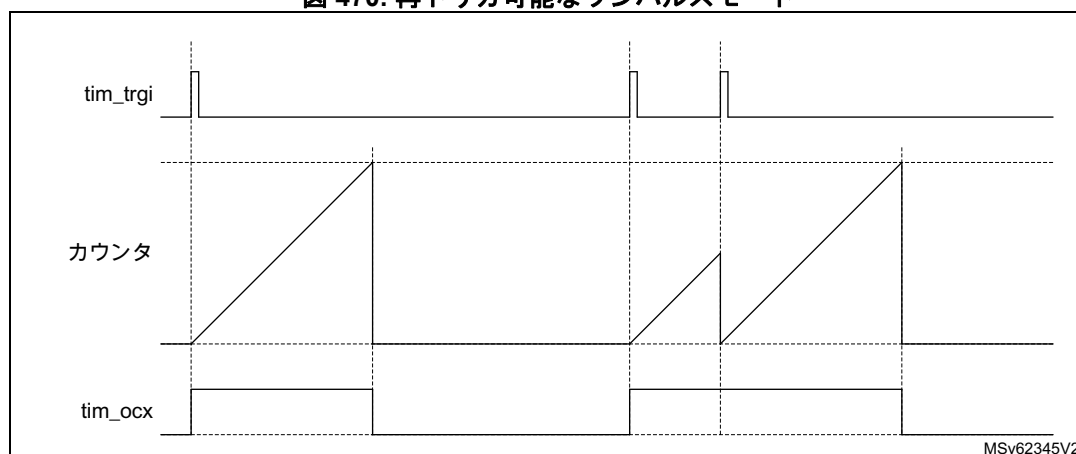
タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

注 : 再トリガ可能なワンパルスモードでは、CCxIF フラグは意味を持ちません。

OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

このモードをセンターアライン PWM モードと組み合わせて使用することはできません。TIMx_CR1 では、CMS[1:0] = 00 にする必要があります。

図 470. 再トリガ可能なワンパルスモード

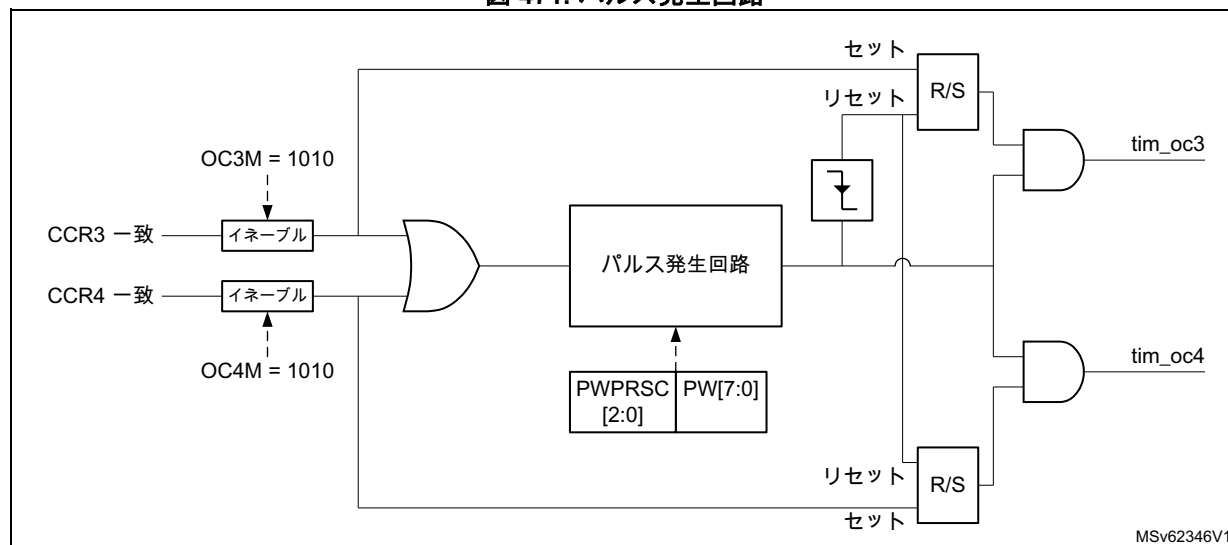


39.4.17 比較モードでのパルス

比較一致イベント時にパルスを生成できます。デバッグまたは同期のために、カウンタ値が特定の比較値と等しいときに、プログラム可能なパルス幅を持つ信号が生成されます。

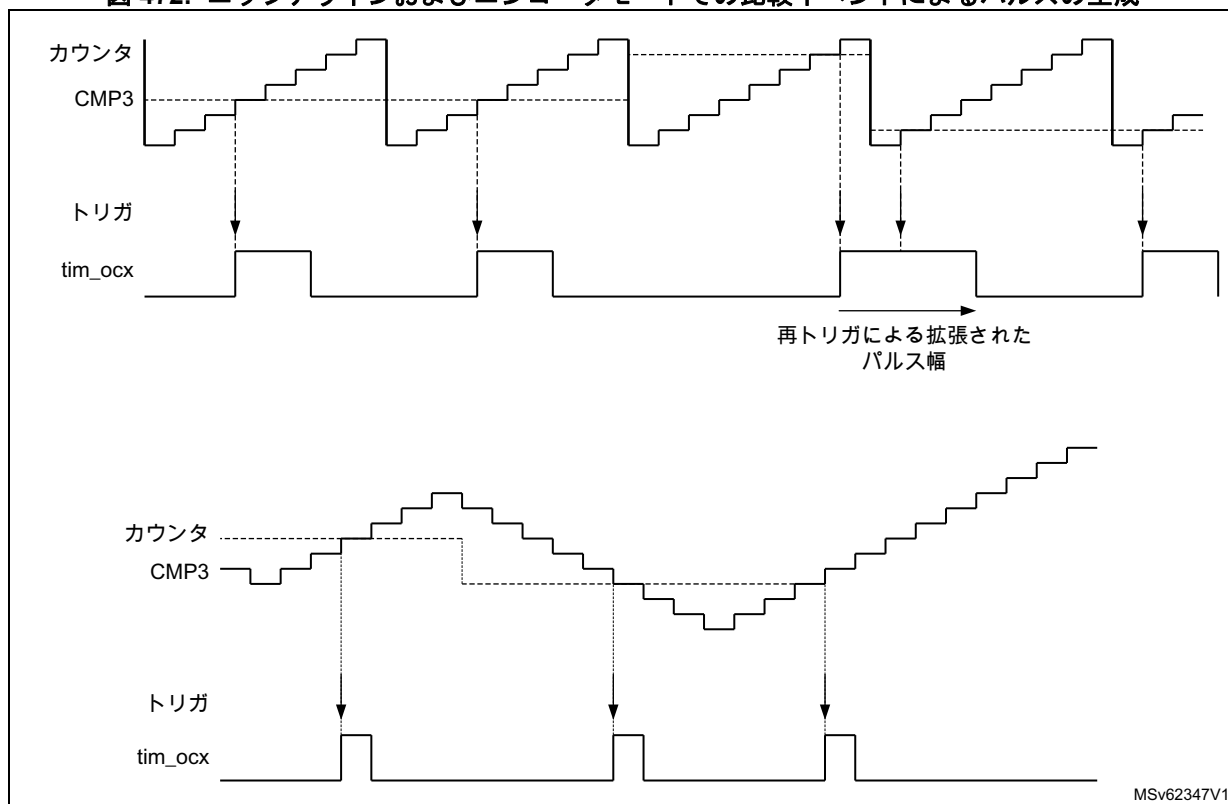
このモードは、エンコーダモードを含むすべてのスレーブモードの選択で、エッジアラインおよびセンタアラインのカウンタモードで使用できます。これは、チャンネル 3 とチャンネル 4 のみで使用できます。以下の図 471 に示すように、パルス発生回路はただ 1 つであり、2 つのチャンネルで共有されます。

図 471. パルス発生回路



下記の図 472 は、エッジアラインおよびエンコーダ動作モードでパルスがどのように生成されるかを示しています。

図 472. エッジアラインおよびエンコーダモードでの比較イベントによるパルスの生成



この出力比較モードは、TIMx_CCMR2 レジスタの OC3M[3:0] および OC4M[3:0] ビットフィールドを使用して選択されます。

パルス幅は、次のように、PWPRSC[2:0] ビットに従ってプリスケールされた特定のクロックを使用し、レジスタの PW[7:0] ビットフィールドを使用してプログラムします。

$$t_{PW} = PW[7:0] \times t_{PWG}$$

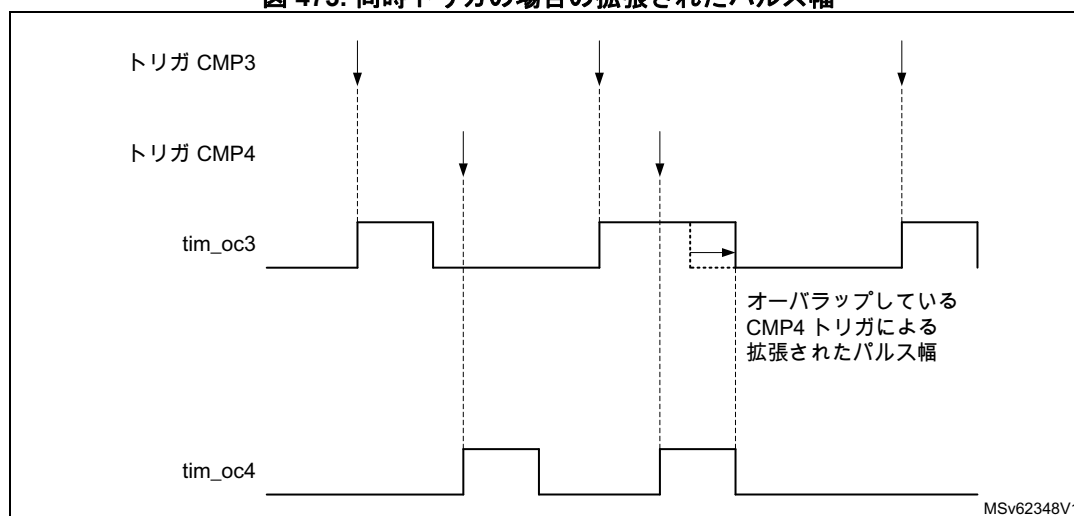
$$\text{ここで、} t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$$

上記は、プリスケラの値に応じた分解能と最大値を示します。

パルスは再トリガ可能です。パルスが進行中に新しいトリガが入ると、パルスは拡張されます。

注： 2つのチャンネルが同時に有効になっている場合、1つのチャンネルのトリガが並行出力上に生成されたパルスとオーバーラップしない限り、パルスは独立して発行されます。反対に、2つのトリガがオーバーラップしている場合、最初に到着したトリガに関連するパルス幅は（再トリガのために）拡張されますが、後に到着したトリガのパルス幅は正しくなります（以下の図 473 に示すとおり）。

図 473. 同時トリガの場合の拡張されたパルス幅



39.4.18 エンコーダインタフェースモード

直交エンコーダ

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが tim_ti1 エッジのみをカウントしている場合は SMS=0001 を、tim_ti2 エッジのみをカウントしている場合は SMS=0010 を、tim_ti1 と tim_ti2 の両方のエッジをカウントしている場合は SMS=0011 を書き込みます。

tim_ti1 と tim_ti2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。CC1NP と CC2NP はクリア状態に維持する必要があります。必要なときには、入力フィルタもプログラミングできます。

2つの入力 tim_ti1 と tim_ti2 は、インクリメンタルエンコーダとのインタフェースに使用されます。表 408 を参照してください。カウンタのクロックは、tim_ti1fp1 または tim_ti2fp2 (入力フィルタおよび極性選択後は tim_ti1 および tim_ti2。フィルタされず、反転されない場合は tim_ti1fp1=tim_ti1、フィルタされず、反転されない場合は tim_ti2fp2=tim_ti2) の有効な変化によって駆動されます。ただし、カウンタが有効なことが前提となります (TIMx_CR1 レジスタの CEN ビットが“1”)。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが tim_ti1 のみ、tim_ti2 のみ、または tim_ti1 と tim_ti2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (tim_ti1 または tim_ti2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (tim_ti1 と tim_ti2 は同時に切り替わらないと想定しています)。

表 408. カウント方向とエンコーダ信号 (CC1P = CC2P = 0)

アクティブエッジ	SMS[3:0]	他方の信号のレベル (tim_ti2 に対する tim_ti1fp1、tim_ti1 に対する tim_ti2fp2)	tim_ti1fp1 信号		tim_ti2fp2 信号	
			立ち上がり	立ち下がり	立ち上がり	立ち下がり
tim_ti1 のみカウント x1 モード	1110	高	ダウン	アップ	カウントなし	カウントなし
		低	カウントなし	カウントなし	カウントなし	カウントなし
tim_ti2 のみカウント x1 モード	1111	高	カウントなし	カウントなし	アップ	ダウン
		低	カウントなし	カウントなし	カウントなし	カウントなし
tim_ti1 のみカウント x2 モード	0001	高	ダウン	アップ	カウントなし	カウントなし
		低	アップ	ダウン	カウントなし	ダウン
tim_ti2 のみカウント x2 モード	0010	高	カウントなし	カウントなし	アップ	ダウン
		低	カウントなし	カウントなし	ダウン	アップ
tim_ti1 と tim_ti2 の 両方をカウント x4 モード	0011	高	ダウン	アップ	アップ	ダウン
		低	アップ	ダウン	ダウン	アップ

直交エンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部トリガ入力に接続して、カウンタのリセットをトリガできます。

図 474 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S = 01 (TIMx_CCMR1 レジスタ、tim_ti1fp1 は tim_ti1 に配置)
- CC2S = 01 (TIMx_CCMR2 レジスタ、tim_ti2fp2 は tim_ti2 に配置)
- CC1P および CC1NP = 0 (TIMx_CCER レジスタ、tim_ti1fp1 非反転、tim_ti1fp1=tim_ti1)
- CC2P および CC2NP = 0 (TIMx_CCER レジスタ、tim_ti2fp2 非反転、tim_ti2fp2=tim_ti2)
- SMS=0011 (TIMx_SMCR レジスタ、両方の入力立ち上がり立ち下がり両エッジでアクティブ)
- CEN=1 (TIMx_CR1 レジスタ、カウンタ有効)

図 474. エンコーダインタフェースモードにおけるカウンタの動作例

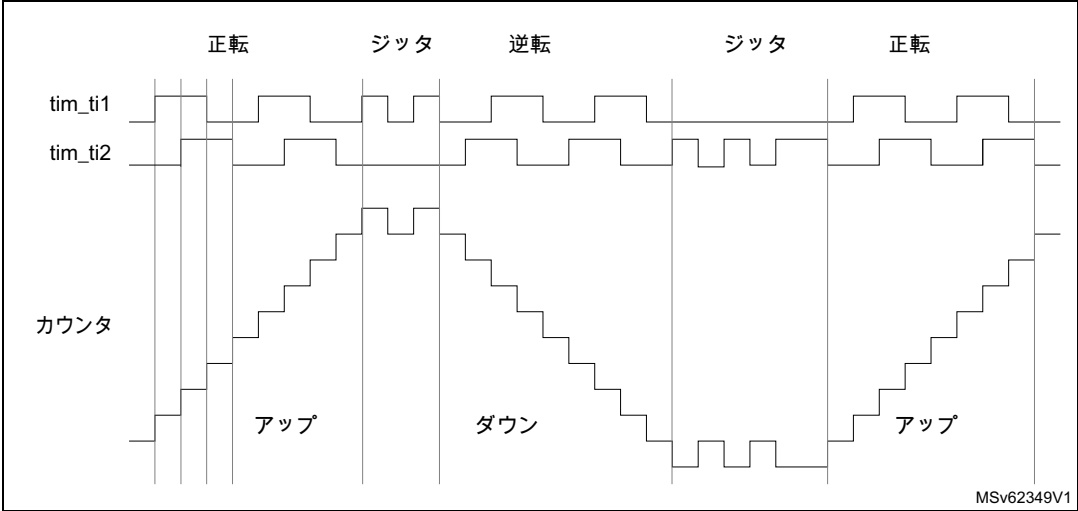
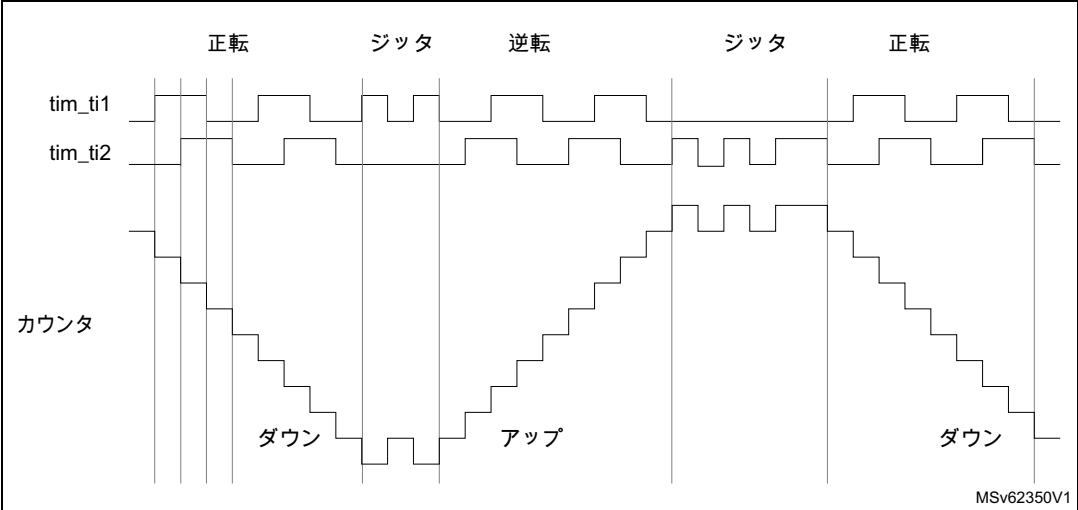


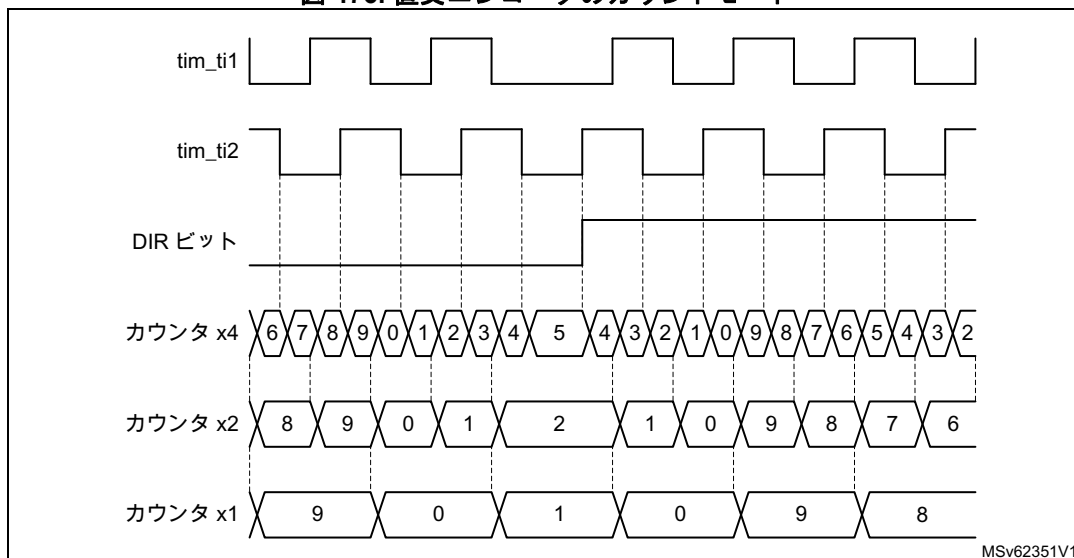
図 475 に、`tim_ti1fp1` の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、`CC1P=1`）。

図 475. `tim_ti1fp1` の極性を反転したエンコーダインタフェースモードの例



以下の図 476 は、さまざまなカウントモードでの速度反転中のタイマカウンタ値を示しています。

図 476. 直交エンコーダのカウントモード



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、DMA リクエストを通じて値を読み出すことも可能です。

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。バックグラウンドタスク（カウンタの読み出し）と割り込み（更新割り込み）との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読み出しアクセス時に UIFCPY フラグによって上書きされます（カウンタの最上位ビットには書き込みモード時のみアクセス可能）。

クロックプラス方向エンコーダモード

直交エンコーダモードのほか、タイマは他のタイプのエンコーダもサポートしています。

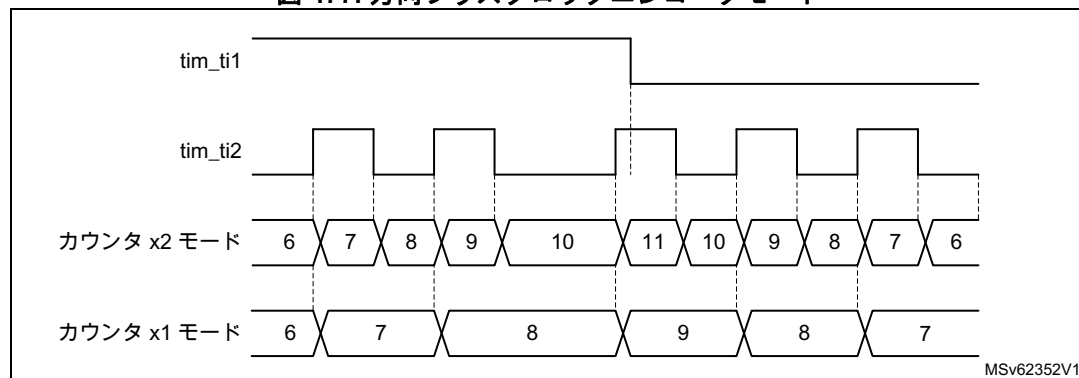
図 477 に示す「クロックプラス方向」モードでは、`tim_ti2` の単一ラインでクロックが提供され、`tim_ti1` 入力を使用して方向が強制されます。

このモードは TIMx_SMCR レジスタの SMS[3:0] ビットフィールドによって、以下のように有効にされます。

- 1010 : X2 モード。カウンタはクロックの立ち上がり立ち下がりの両エッジで更新されます。
- 1011 : X1 モード。カウンタは 次の CC2P ビット値に従って単一のクロックエッジで更新されます。CC2P = 0 は立ち上がりエッジ検出に対応し、CC2P = 1 は立ち下がりエッジ検出に対応します。

tim_ti1 の方向信号の極性は、次の CC1P ビットで設定されます。0 は正の極性 (tim_ti1 がハイのときアップカウントで、tim_ti1 がローのときダウンカウント) に対応し、CC1P = 1 は負の極性 (tim_ti1 がローのときアップカウント) に対応します。

図 477. 方向プラスクロックエンコーダモード



方向クロックエンコーダモード

図 478 の「方向クロック」モードでは、クロックは 2 本のラインで提供され、1 本のアップカウントクロックラインと 1 本のダウンカウントクロックラインを持つというように、方向に応じて一度に 1 つが提供されます。

このモードは TIMx_SMCR レジスタの SMS[3:0] ビットフィールドによって、以下のように有効にされます。

- 1100 : x2 モード。カウンタは、2 つのクロックラインのいずれかの立ち上がりと立ち下りの両エッジで更新されます。CC1P および CC2P ビットは、クロックアイドル状態用のコーディングです。CCxP = 0 はハイレベルアイドル状態に対応し (下記図 478 を参照)、CCxP = 1 はローレベルアイドル状態に対応します (下記図 479 を参照)。
- 1101 : x1 モード。カウンタは、CC1P および CC2P ビット値に従って、単一のクロックエッジで更新されます。CCxP = 0 は立ち下り検出とハイレベルアイドル状態に対応し (下記図 478 を参照)、CCxP = 1 は立ち上がり検出とローレベルアイドル状態に対応します (下記図 479 を参照)。

図 478. 方向クロックエンコーダモード (CC1P = CC2P = 0)

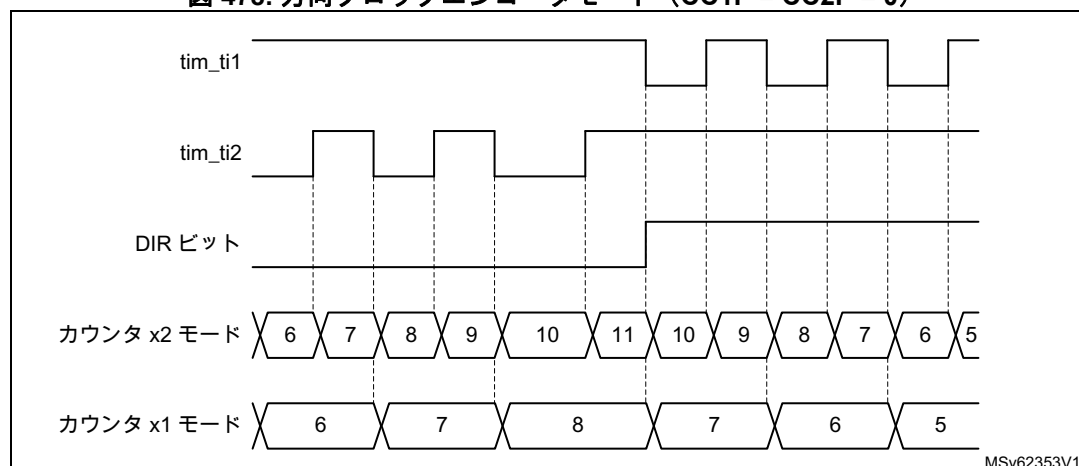
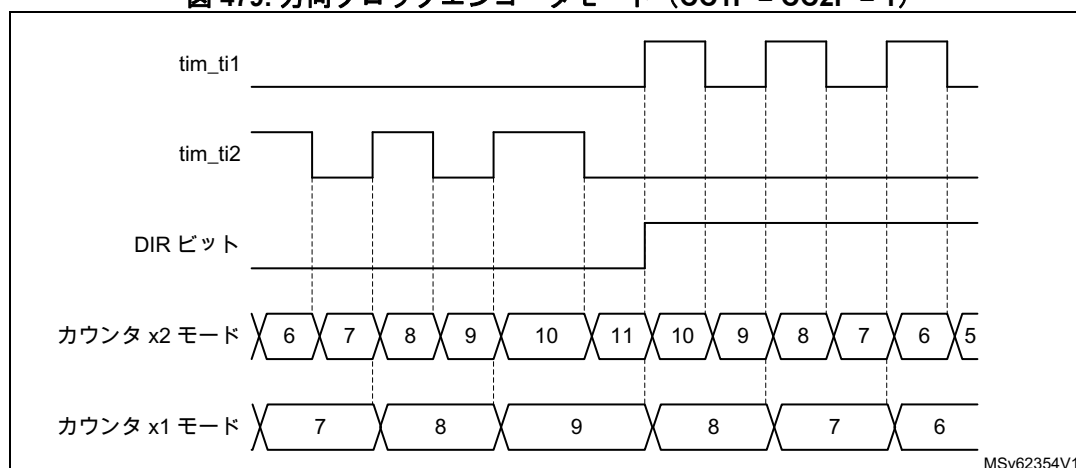


図 479. 方向クロックエンコーダモード (CC1P = CC2P = 1)



以下の表 409 に、任意の入力遷移に対して方向クロックモードがどのように動作するかを詳しく説明します。

表 409. カウント方向 対 エンコーダ信号および極性設定

方向クロックモード	SMS[3:0]	他方の信号のレベル (tim_ti2 に対する tim_ti1fp1、 tim_ti1 に対する tim_ti2fp2)	tim_ti1fp1 信号		tim_ti2fp2 信号	
			立ち上がり	立ち下がり	立ち上がり	立ち下がり
x2 モード CCxP=0	1100	高	ダウン	ダウン	アップ	アップ
		低	カウントなし	カウントなし	カウントなし	カウントなし
x2 モード CCxP=1	1100	高	カウントなし	カウントなし	カウントなし	カウントなし
		低	ダウン	ダウン	アップ	アップ
x1 モード CCxP=0	1101	高	カウントなし	ダウン	カウントなし	アップ
		低	カウントなし	カウントなし	カウントなし	カウントなし
x1 モード CCxP=1	1101	高	カウントなし	カウントなし	カウントなし	カウントなし
		低	ダウン	カウントなし	アップ	カウントなし

インデックス入力

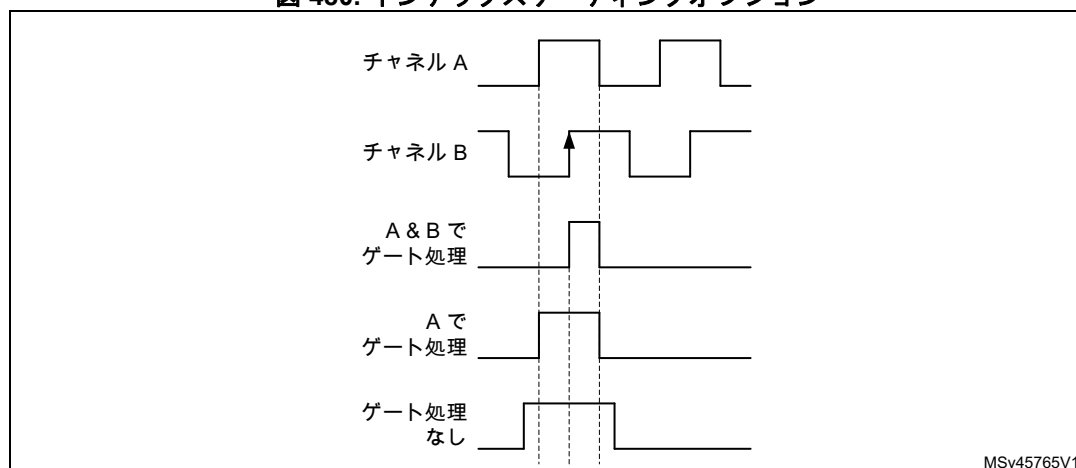
カウンタは、絶対基準位置を示すエンコーダからのインデックス信号によってリセットできます。インデックス信号は、tim_etr_in 入力に接続する必要があります。これは、デジタル入力フィルタを使用してフィルタリングできます。

インデックス機能は、TIMx_ECR レジスタの IE ビットで有効にされます。IE ビットは、SMS[3:0] ビットフィールドが次の値であるときに、エンコーダモードでのみセットする必要があります。0001、0010、011、1010、1011、1100、1101、1110、1111。

以下の図 480 に示すように、インデックスパルス調整のいくつかのオプションを備えた市販のエンコーダが提案されています。

- A および B でゲート処理：パルス幅は 1 つのチャンネル周期の 1/4 であり、A および B の両方のエッジにそろえられます。
- A でゲート処理（または B でゲート処理）：パルス幅は 1 つのチャンネル周期の 1/2 であり、チャンネル A（または チャンネル B）の 2 つのエッジにそろえられます。
- ゲート処理なし：パルス幅は最大 1 チャンネル周期であり、どのエッジにもそろえられません。

図 480. インデックスゲーティングオプション

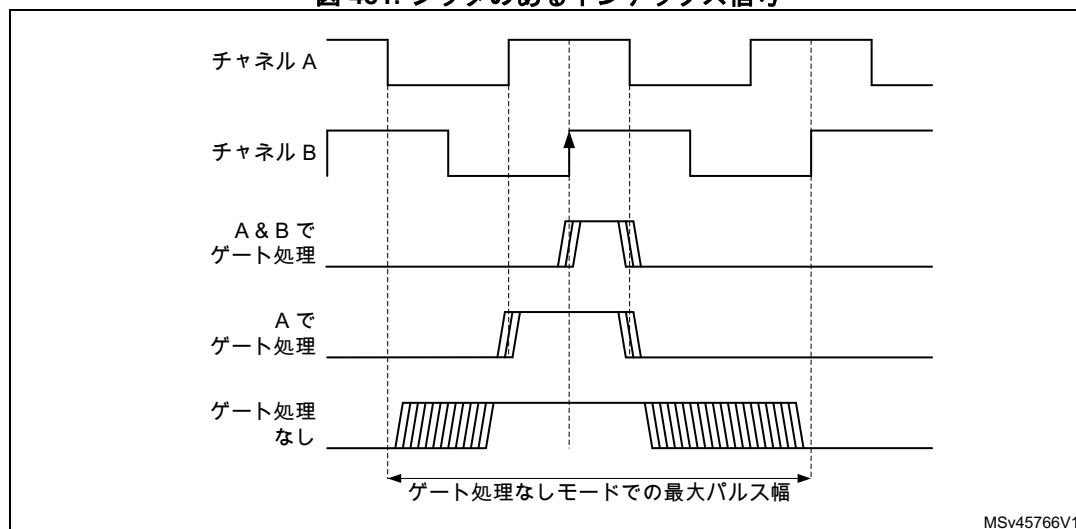


MSv45765V1

以下の図 481 に示すように、ゲーティングモードに関係なく、回路はインデックス信号のジッタを許容します。

ゲート処理なしモードでは、信号はエンコーダの 2 周期より厳密に小さくしなければなりません。パルス幅がエンコーダの 2 周期以上の場合、カウンタは複数回リセットされます。

図 481. ジッタのあるインデックス信号



MSv45766V1

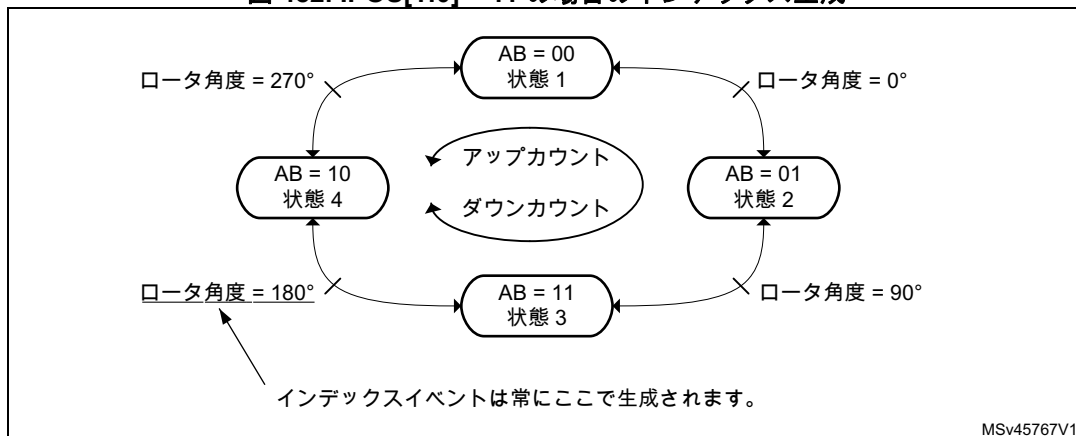
タイマは、特定のプログラミングを必要とせずに、3 つのゲーティングオプションを同様にサポートします。TIMx_ECR レジスタの IPOS[1:0] ビットフィールドを使用して、インデックスを同期する必要があるエンコーダ状態（チャンネル A とチャンネル B の状態の組み合わせ）を定義するだけです。

インデックス検出イベントは、速度の反転中に対称動作を確保するために、カウント方向に応じて異なる動作をします。

- カウンタはアップカウント中 (DIR ビット = 0) に、リセットされます。
- カウンタはダウンカウント時に、TIMx_ARR に設定されます。

これにより、カウント方向に関係なく、まったく同じ機械的角度位置でインデックスを生成できます。以下の図 482 は、単純な例 (機械的回転ごとに 4 つのエッジを提供するエンコーダ) について、生成されるインデックスの位置を示しています。

図 482. IPOS[1:0] = 11 の場合のインデックス生成



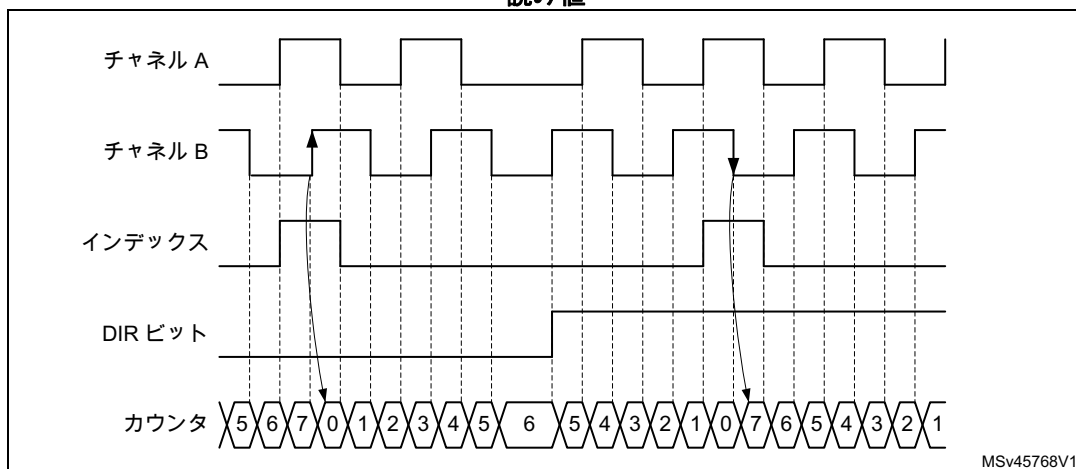
以下の図 483 に、IPOS[1:0] = 11 の場合の波形と対応する値を示します。カウンタの値が強制される瞬間は、カウント方向に応じて自動的に調整されることを示しています。

- アップカウント (DIR ビット = 0) の場合、エンコーダの状態が「11」 (ChA = 1, ChB = 1) のとき、カウンタは 0 にセットされます。
- ダウンカウント (DIR ビット = 1) の場合、「11」状態を終了するとき、カウンタは TIMx_ARR にセットされます。

インデックス検出イベント時に割込みを発行できます。

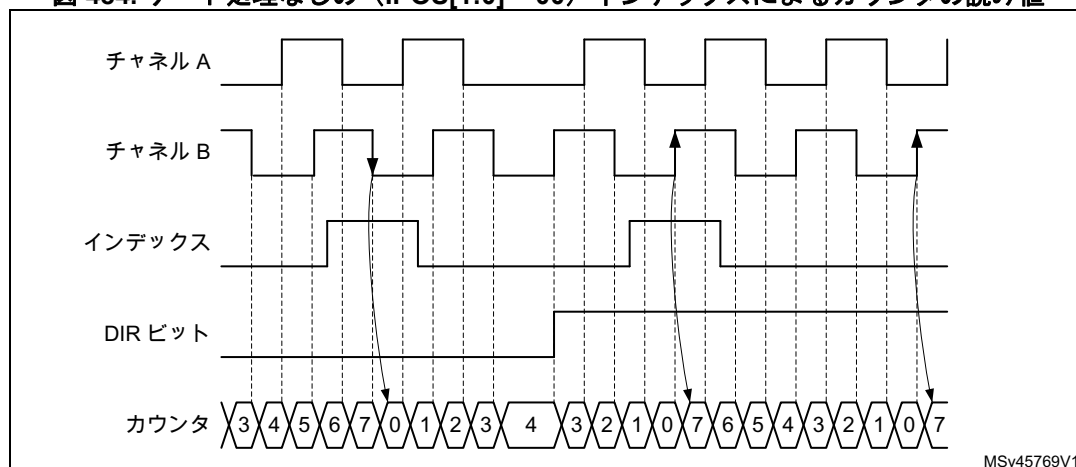
矢印は、どの遷移でインデックスイベント割込みが生成されるかを示しています。

図 483. チャンネル A でゲート処理された (IPOS[1:0] = 11) インデックスによるカウンタの読み値



以下の図 484 に、ゲート処理なしモードの場合の波形と対応する値を示します。矢印は、どの遷移でインデックスイベントが生成されるかを示しています。

図 484. ゲート処理なしの (IPOS[1:0] = 00) インデックスによるカウンタの読み値



以下の図 485 は、さまざまなパルスアライメントシナリオでの「A & B でゲート処理」モードの処理方法を示しています。矢印は、どの遷移でインデックスイベントが生成されるかを示しています。

図 485. チャンネル A および B でゲート処理されたインデックスによるカウンタの読み値

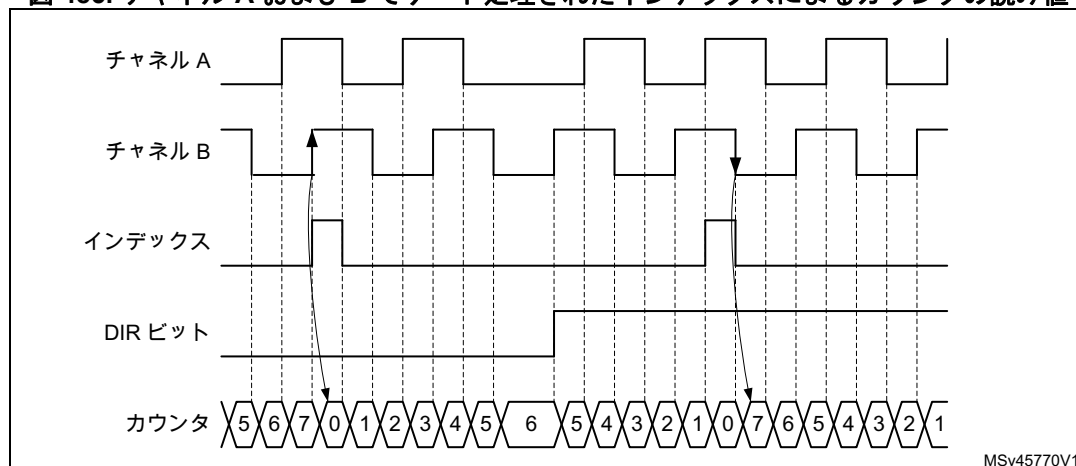


図 486 と 図 487 には、後続のインデックスパルスが、エンコーダクロック周期の 4 分の 1 よりも狭いことがあるケースが詳しく示されています。

図 486. 狭いインデックスパルスの場合のエンコーダモードの動作 (IPOS[1:0] = 11)

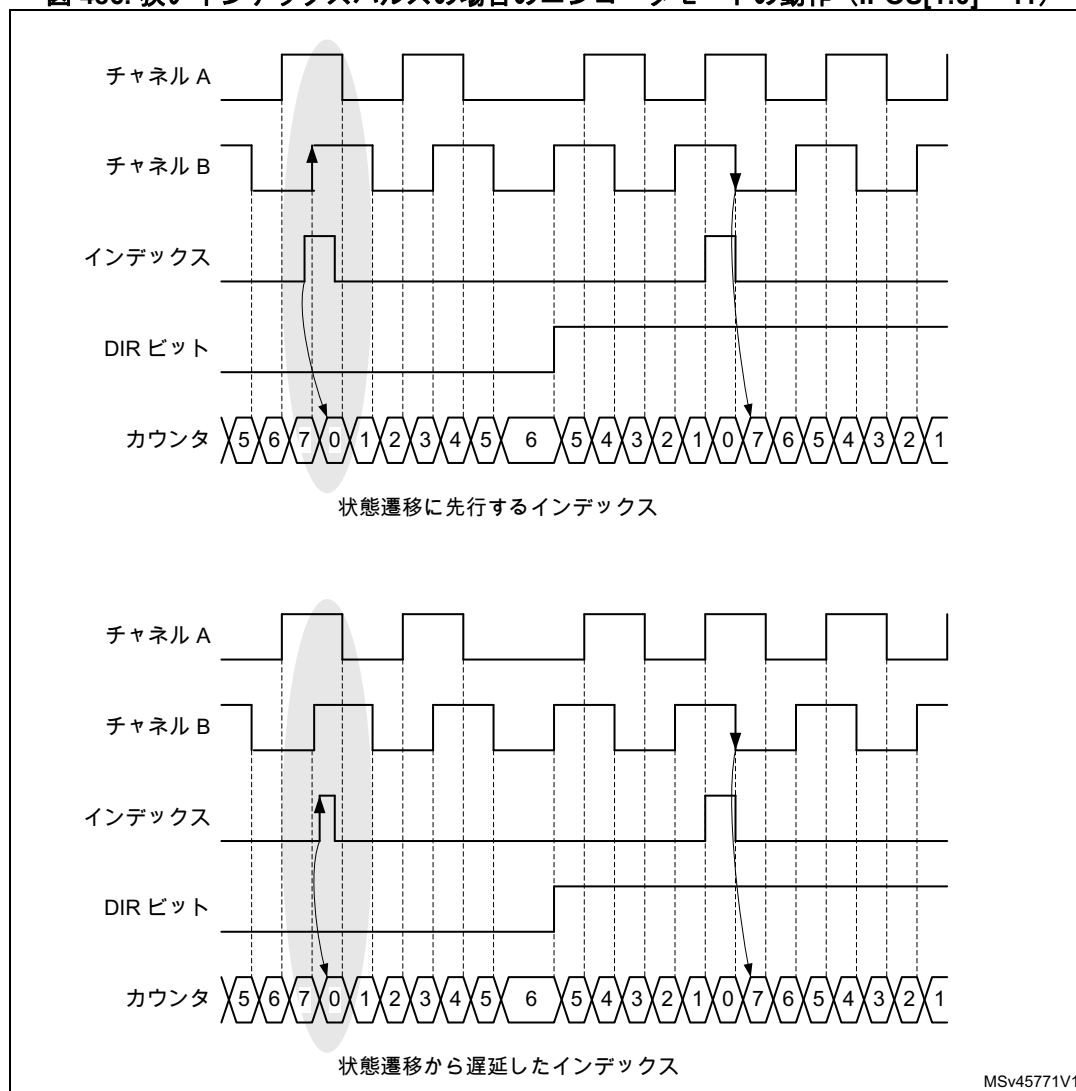
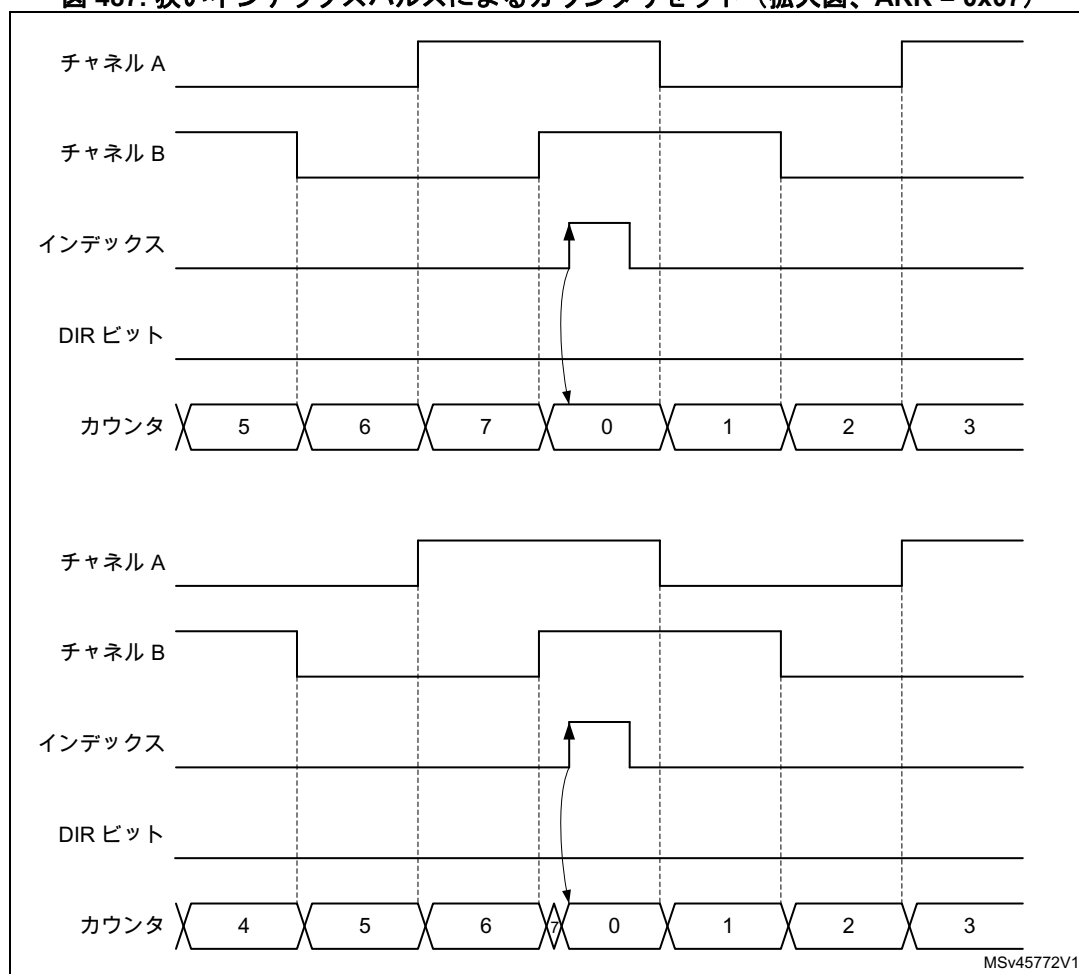
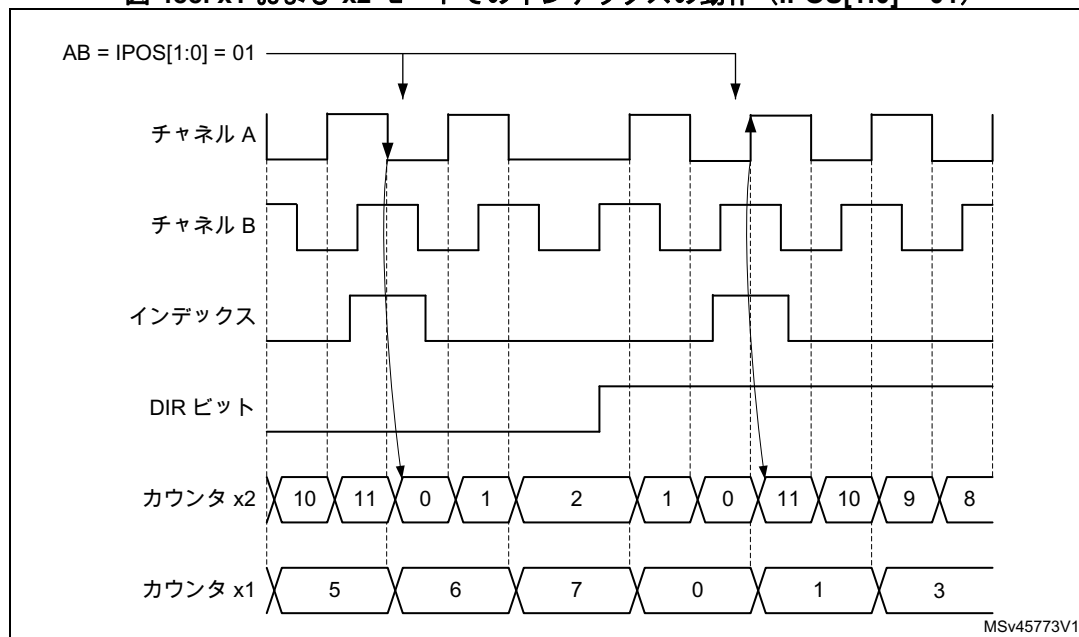


図 487. 狭いインデックスパルスによるカウンタリセット (拡大図、ARR = 0x07)



下記の図 488 は、x1 および x2 モードでインデックスがどのように管理されるかを示します。

図 488. x1 および x2 モードでのインデックスの動作 (IPOS[1:0] = 01)

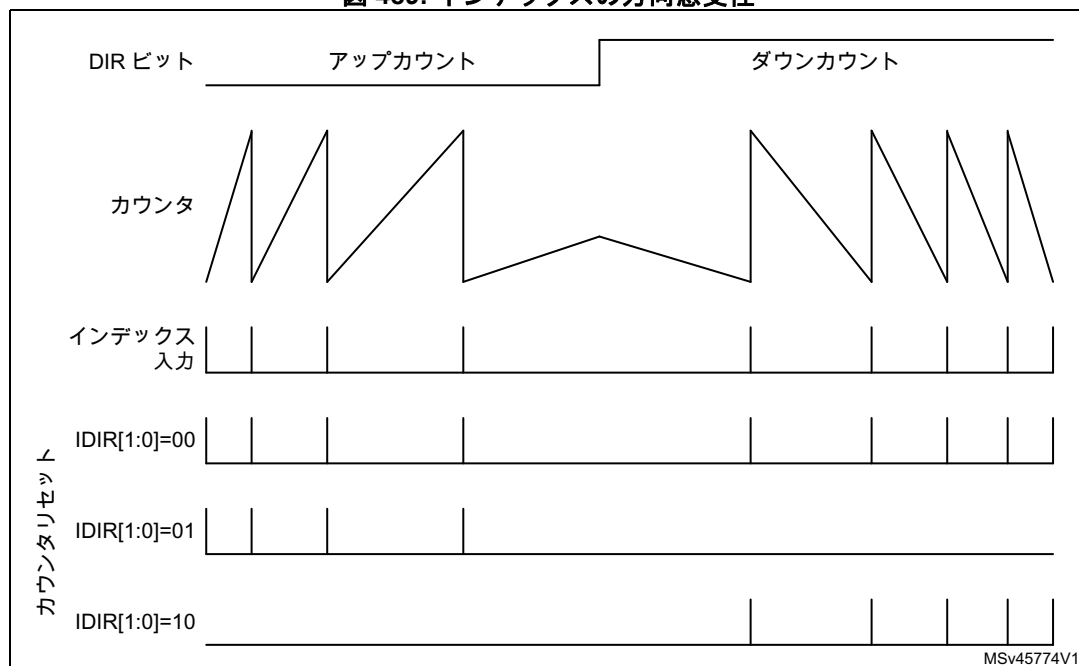


インデックスの方向感受性

TIMx_ECR レジスタの IDIR[1:0] ビットフィールドにより、選択したカウント方向でのみインデックスをアクティブにできます。

以下の図 489 は、IDIR[1:0] 値に応じた、インデックスとカウンタのリセットイベントの関係を示しています。

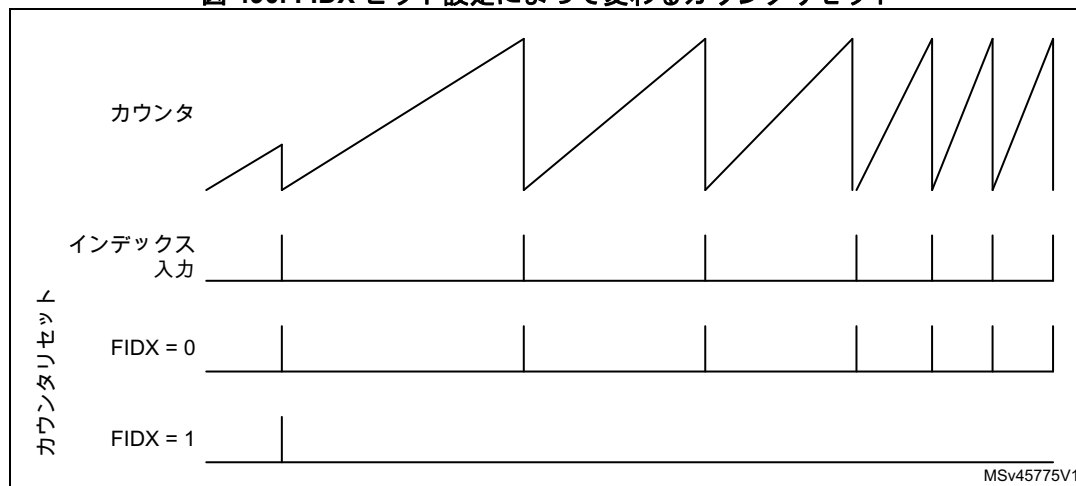
図 489. インデックスの方向感受性



特殊な最初のインデックスイベント管理

TIMx_ECR レジスタの FIDX ビットによって、以下の図 490 に示すように、インデックスを 1 回だけ取得できます。最初のインデックスが到着すると、その後のインデックスは無視されます。必要に応じて、FIDX ビットに 0 を書き込み、再度 1 にセットすることにより、回路を再設定できます。

図 490. FIDX ビット設定によって変わるカウンタリセット



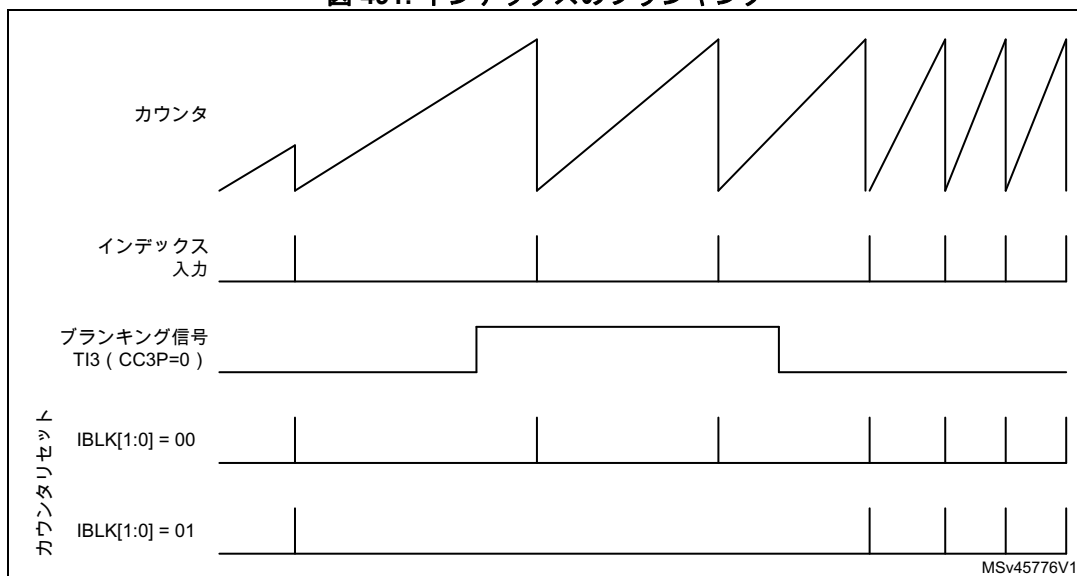
インデックスのブランキング

インデックスイベントは tim_ti3 または tim_ti4 入力を使用して、ブランキングすることができます。ブランキングウィンドウでは、インデックスイベントは以下の図 491 に示すようにカウンタのリセットを停止します。

このモードは TIMx_ECR レジスタの IBLK[1:0] ビットフィールドを使用して、以下のように有効にされます。

- IBLK[1:0] = 00 : インデックス信号は常にアクティブです。
- IBLK[1:0] = 01 : インデックス信号は tim_ti3 入力でブランキングします。
- IBLK[1:0] = 10 : インデックス信号は tim_ti4 入力でブランキングします。

図 491. インデックスのブランキング



非直交モードでのインデックス管理

以下の図 492 および図 493 では、SMS[3:0] ビットフィールドが 1010、1011、1100、1101 に等しい場合に、方向クロックモードおよびクロックプラス方向モードでインデックスがどのように管理されるかを詳しく説明します。

これらのモードの両方で、インデックスの検出は次のように IPOS[0] ビットで設定されます。

- IPOS[0] = 0 : インデックスはクロックのローレベルで検出されます。
- IPOS[0] = 1 : インデックスはクロックのハイレベルで検出されます。

IPOS[1] ビットは意味がありません。

図 492. クロックプラス方向モードでのインデックスの動作 (IPOS[0] = 1)

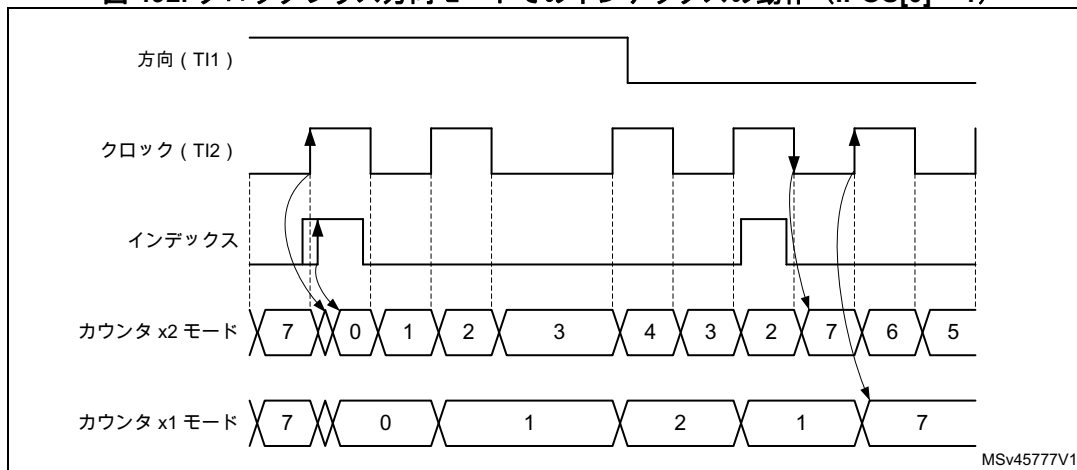
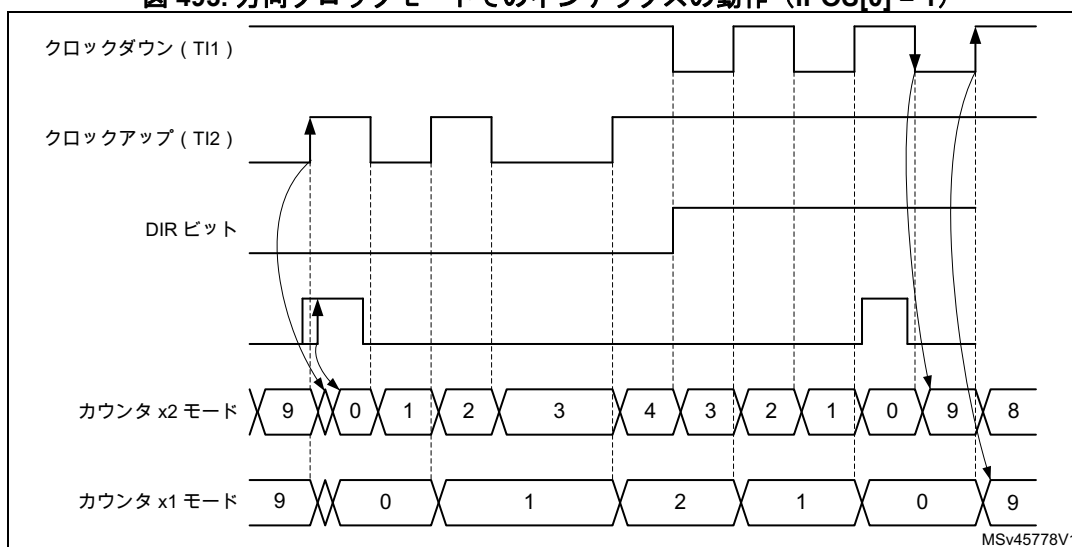


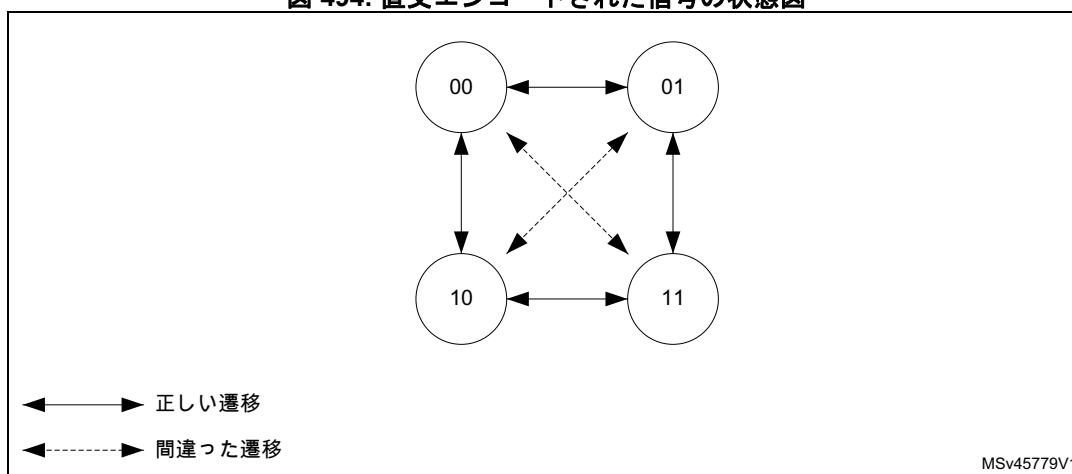
図 493. 方向クロックモードでのインデックスの動作 (IPOS[0] = 1)



エンコーダのエラー管理

2つの直交信号が使用できるエンコーダ構成の場合、遷移エラーを検出できます。2つの入力の読み値は、以下の図 494 で、状態図として表すことができる2ビットのグレイコードに対応しています。一度に1つのビットだけが変化することが予測されています。間違った遷移があると、TIMx_SR ステータスレジスタの TERRF 割込みフラグがセットされます。TIMx_DIER レジスタの TERRIE ビットがセットされていると、遷移エラー割込みが生成されます。

図 494. 直交エンコードされた信号の状態図



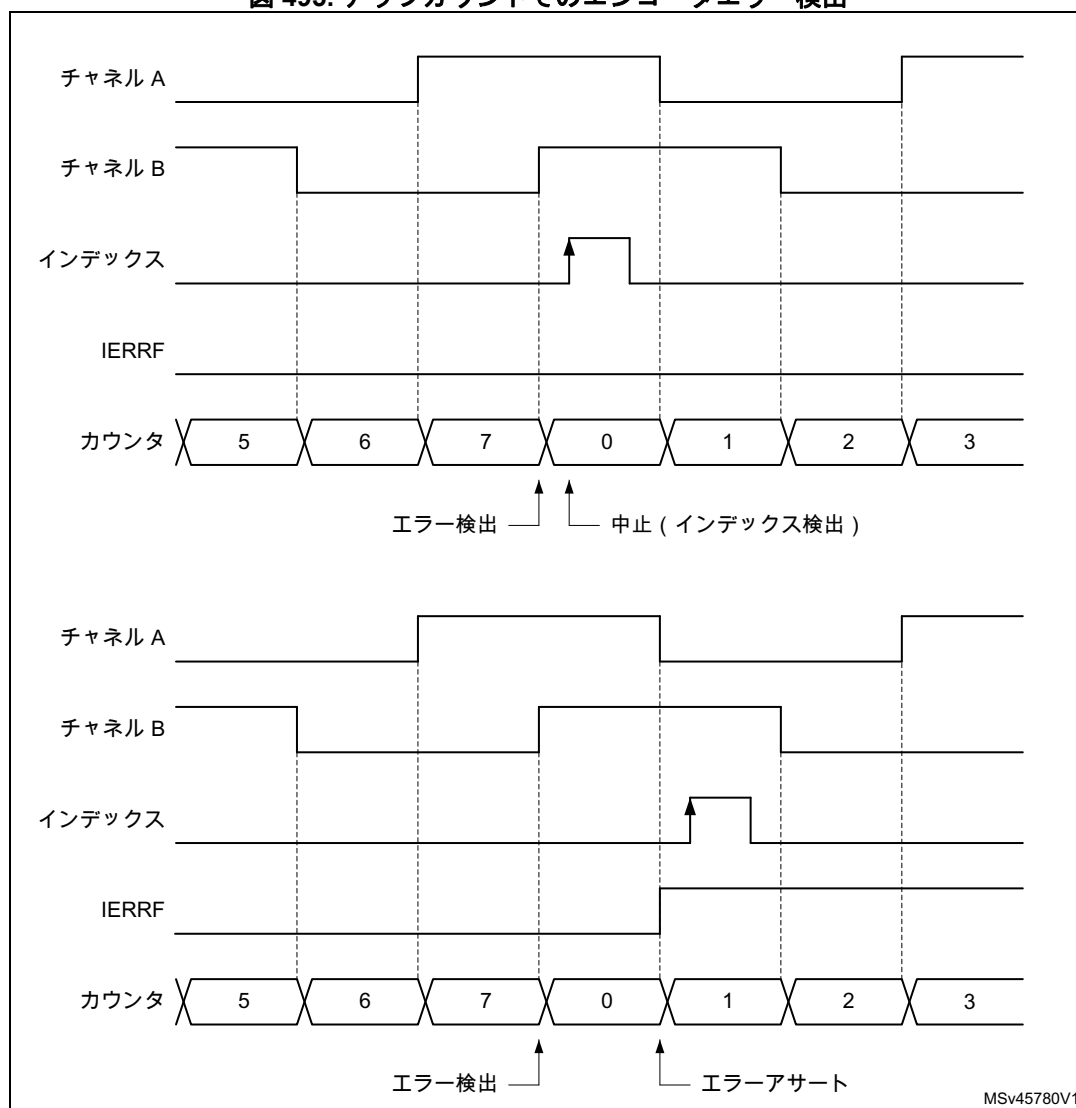
インデックス信号を持つエンコーダの場合、1回転あたりのパルスが過剰になる異常動作を検出できます。1回転あたり N 個のパルスを持つエンコーダでは、1回転あたり 4xN カウントが提供されます。インデックス信号によって、4xN クロック周期ごとにカウンタがリセットされます。

カウンタ値がインデックスイベントなしに TIMx_ARR から 0 にインクリメントされるか、0 から TIMxARR 値にデクリメントされる場合、これはインデックス位置エラーとして報告されます。

オーバーフロー閾値は、TIMx_ARR レジスタを使ってプログラムします。1000 行のエンコーダでは、カウンタ値が 0~3999 になります (4x 読み取りモードの場合)。オーバーフロー検出閾値は、TIMx_ARR = 3999 + 1 = 4000 を設定してプログラムする必要があります。

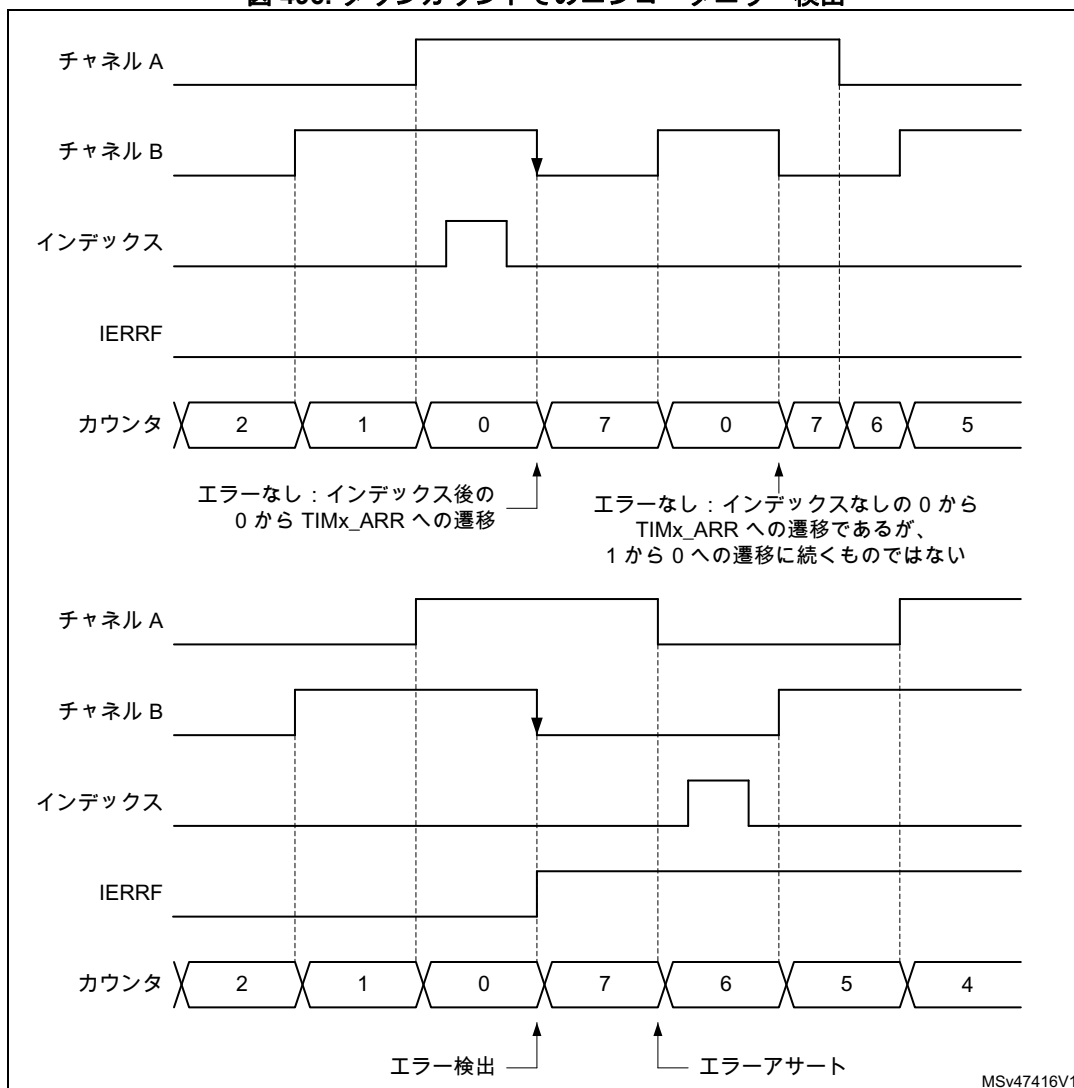
アップカウントでは、エラーアサーションは 0 から 1 への遷移まで遅延されます。これは、以下の図 495 に示すように、A および B ゲート処理モードでの狭いインデックスパルスへの対処です。

図 495. アップカウントでのエンコーダエラー検出



ダウンカウントモードでは、検出は 1 から 0 への事前遷移によって条件付けられます。これは、以下の図 496 に示すように、A および B ゲート処理モードでの狭いインデックスパルスに対処し、インデックス検出直後にエンコーダが TIMx_ARR と 0 の間でディザリングする場合の誤ったエラー検出を回避するためです。

図 496. ダウンカウントでのエンコーダエラー検出



インデックスエラーによって、TIMx_SR ステータスレジスタの IERRF 割込みフラグがセットされます。TIMx_DIER レジスタの IERRIE ビットがセットされていると、インデックスエラー割込みが生成されます。

エンコーダの機能的割込み

エンコーダモードでは、以下の割込みも使用できます。

- 方向変更: エンコーダモードでのカウント方向の変更によって、TIMx_CR1 レジスタの DIR ビットが反転します。方向変更によって、TIMx_SR ステータスレジスタの DIRF 割込みフラグがセットされます。TIMx_DIER レジスタの DIRIE ビットがセットされていると、方向変更割込みが生成されます。

- インデックスイベント: インデックスイベントによって、TIMx_SR ステータスレジスタの IDXF 割込みフラグがセットされます。TIMx_DIER レジスタの IDXIE ビットがセットされていると、インデックス割込みが生成されます。

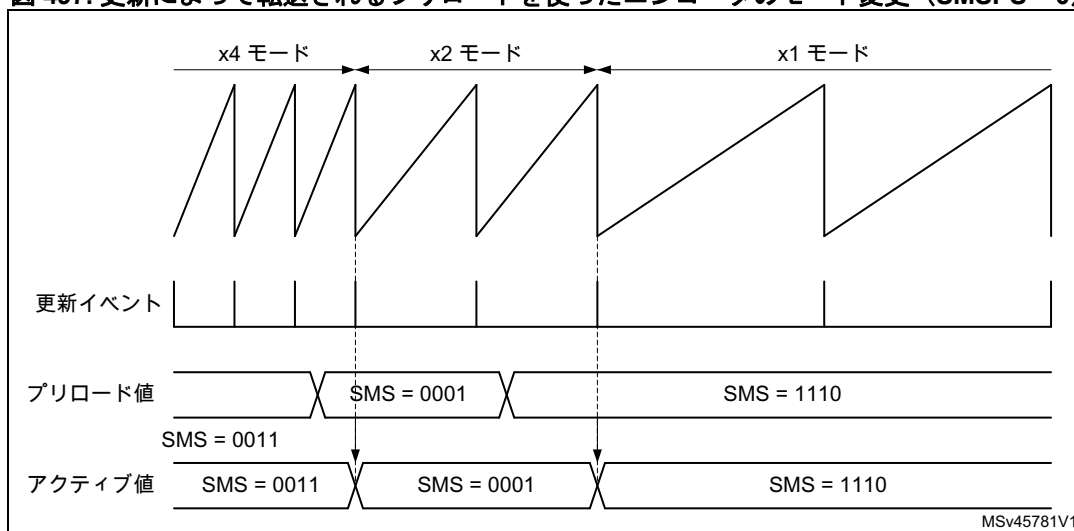
実行時エンコーダモード更新のためのスレーブモード選択プリロード

実行中に、あるエンコーダモードを別のモードに切り替える必要があることがあります。以下の図 497 に示すように、これは通常、高速時に更新割込みレートを低下させるために、x4 モードから x2 に、x1 に切り替えることにより、実行されます。

この目的のために、SMS[3:0] ビットをプリロードすることができます。これは、TIMx_SMCR レジスタの SMSPE イネーブルビットをセットすることで有効になります。SMS[3:0] プリロードからアクティブ値へ転送するトリガは、TIMx_SMCR レジスタの SMSPS ビットで選択できます。

- SMSPS = 0: 転送は、アップカウント時にカウンタがオーバーフローするとき、およびダウンカウント時にアンダーフローするときが発生する更新イベント (UEV) によってトリガされます。
- SMSPS = 1: 転送は、インデックスイベントによってトリガされます。

図 497. 更新によって転送されるプリロードを使ったエンコーダのモード変更 (SMSPS = 0)



エンコーダクロック出力

エンコーダモードの動作原理は、十分な数のクロックエッジ取得と正確な測定を行うために比較的長い積分時間を必要とするため、低速においては、高分解能の速度測定には完全には適していません。

低速でのより良いソリューションは、エッジ間のクロック周期測定を行うことです。これは、スレーブタイマを使用して達成できます。タイマによって、tim_trgo 出力にエンコーダクロック情報を出力できます。その後、スレーブタイマはエンコーダクロックエッジごとに周期測定を実行し、速度情報を提供できます。

このモードは TIMx_CR2 レジスタの MMS[3:0] ビットフィールドを 1000 に設定することによって有効になります。これは、以下の SMS[3:0] 値に対して有効です。0001、0010、0011、1010、1011、1100、1101、1110、1111。その他の SMS[3:0] コードは許されず、予想外の動作につながる可能性があります。

39.4.19 方向ビットの出力

tim_oc3 および tim_oc4 出力信号に、タイマから方向信号を出力できます (TIMx_CR1 レジスタの DIR ビットのコピー)。これを達成するには、TIMx_CCMR2 レジスタの OC3M[3:0] または OC4M[3:0] ビットフィールドを 1011 にセットします。

この機能を使用すると、エンコーダモードでカウント方向（または回転方向）を監視すること、またはセンタアライン PWM モードでアップ/ダウンフェーズを示す信号を得ることができます。

39.4.20 UIF ビットの再配置

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割込みフラグ (UIF) を連続コピーします。これは、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取るために使用されます。バックグラウンドタスク（カウンタの読出し）と割込み（更新割込み）との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読出しアクセス時に UIFCPY フラグによって上書きされます（カウンタの最上位ビットには書き込みモード時のみアクセス可能）。

39.4.21 タイマ入力 XOR 機能

TIM1xx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン tim_ti1、tim_ti2、および tim_ti3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

この機能をホールセンサのインタフェースに使用した例を[セクション 38.3.29: 1488 ページのホールセンサとのインタフェース](#)に示します。

39.4.22 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。すなわち、リセットモード、ゲートモード、トリガモード、リセット + トリガモード、およびゲート + リセットモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

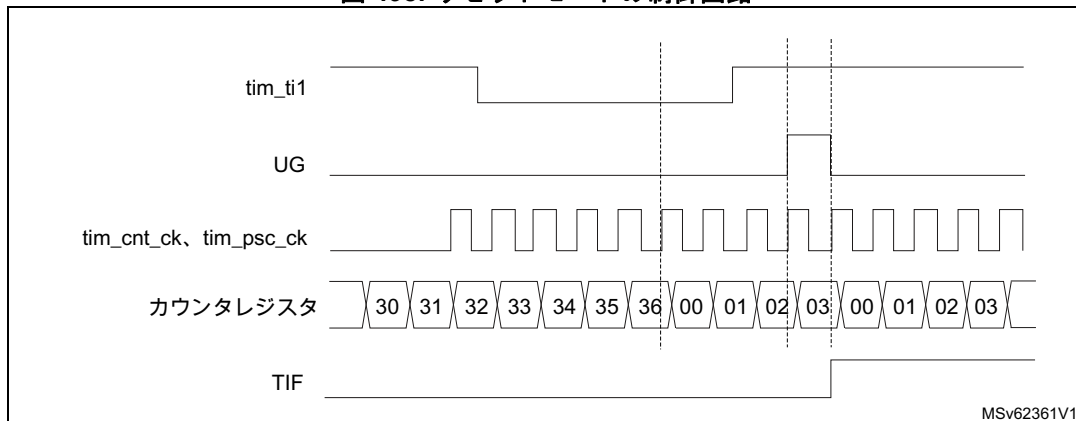
次の例では、tim_ti1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. tim_ti1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
2. TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。
3. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、tim_ti1 の立ち上がりエッジまで通常の動作を行います。tim_ti1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビットに依存)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。tim_ti1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_ti1 入力の同期回路によるものです。

図 498. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

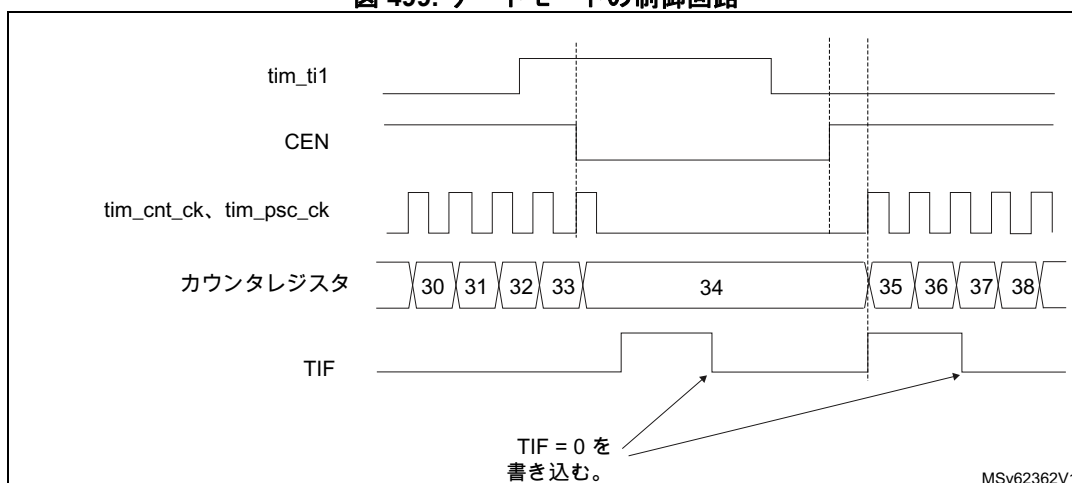
次の例では、アップカウンタは tim_ti1 入力が高レベルのときだけカウントします。

1. tim_ti1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01 ビット)。TIMx_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
2. TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。
3. TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、tim_ti1 がローになると内部クロックでカウントを開始して、tim_ti1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

tim_ti1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、tim_ti1 入力の再同期回路によるものです。

図 499. ゲートモードの制御回路



注：ゲートモードはエッジではなくレベルに対して動作するため、 $CCxP=CCxNP=1$ の設定（立ち上がりと立ち下がり両エッジの検出）はゲートモードでは意味がありません。

スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

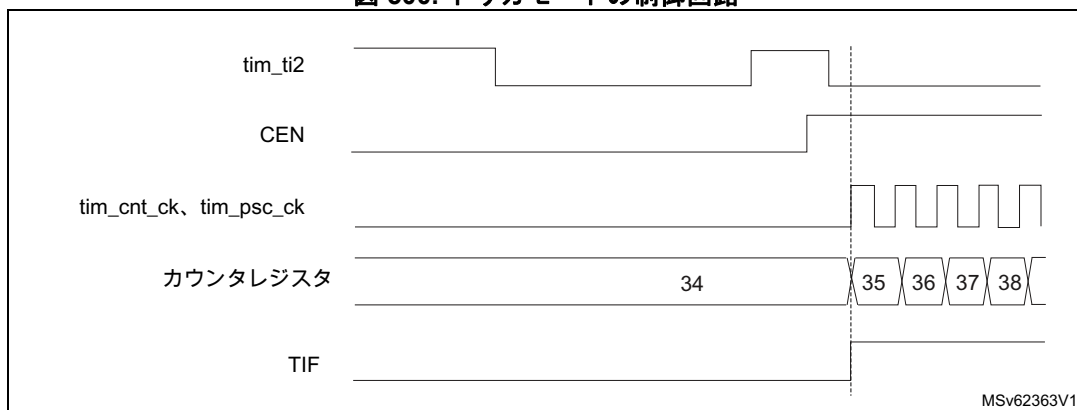
次の例では、アップカウンタは、tim_ti2 入力の立ち上がりエッジに応じて開始します。

1. tim_ti2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC2S=01）。TIMx_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
2. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=00110 を書き込むことによって、入力ソースとして tim_ti2 を選択します。

tim_ti2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

tim_ti2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、tim_ti2 入力の再同期回路によるものです。

図 500. トリガモードの制御回路



実行時エンコーダモード更新のためのスレーブモード選択プリロード

SMS[3:0] ビットをプリロードすることができます。これは、TIMx_SMCR レジスタの SMSPE イネーブルビットをセットすることで有効になります。SMS[3:0] プリロードからアクティブ値へ転送するトリガは、カウンタがオーバーフローするときに発生する更新イベント (UEV) です。

スレーブモード - リセットモードとトリガモードの組み合わせ

この場合、選択されたトリガ入力 (tim_trgi) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

スレーブモード - ゲートモードとリセットモードの組み合わせ

カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

このモードは、範囲外の PWM 信号 (最大期待値を超えるデューティサイクル) を検出するために使用されます。

スレーブモード - 外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、tim_etr_in 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、およびトリガモードで動作している場合)。TIMx_SMCR レジスタの TS ビットを通じて tim_trgi として tim_etr_in を選択しないようにしてください。

次の例では、アップカウンタは、tim_ti1 の立ち上がりエッジが発生すると、tim_etr_in 信号の立ち上がりエッジのたびにインクリメントされます。

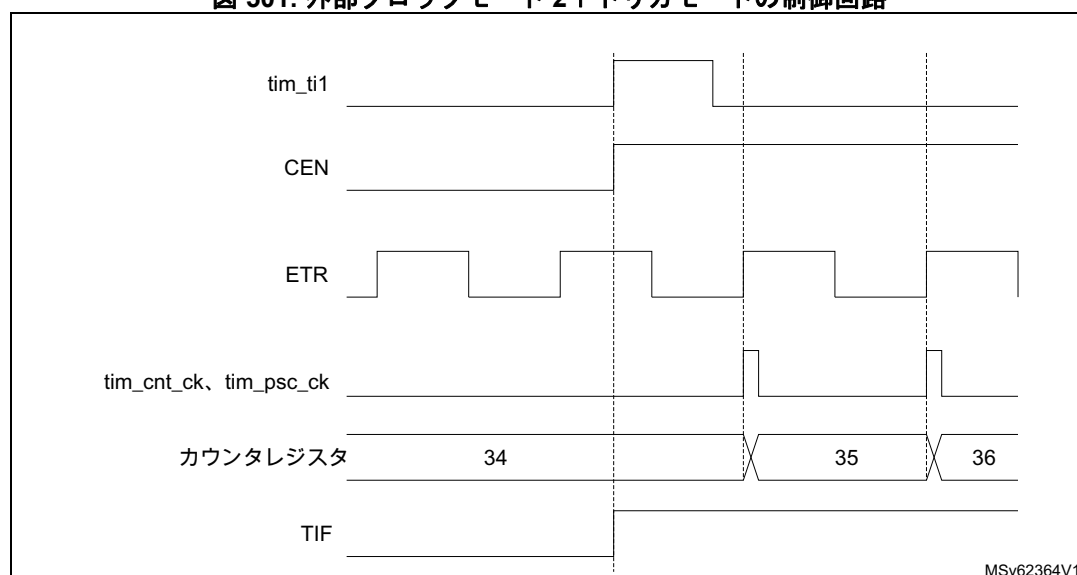
1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000 : フィルタなし
 - ETPS=00 : プリスケール無効
 - ETP=0 : tim_etr_in の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。

2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケラはトリガには使用されないので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。

tim_ti1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、その後 tim_etr_in の立ち上がりエッジでカウントします。

tim_etr_in 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_etrp 入力の再同期回路によるものです。

図 501. 外部クロックモード 2+トリガモードの制御回路



MSv62364V1

39.4.23 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。マスタモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

図 502 と 図 503 に、マスタ/スレーブタイマー接続の例を示します。

図 502. マスタ/スレーブタイマの例

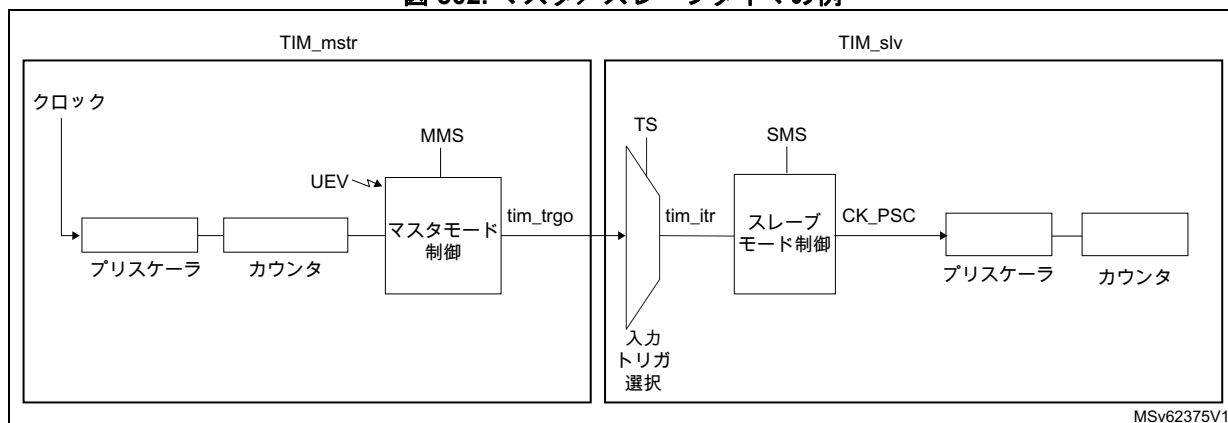
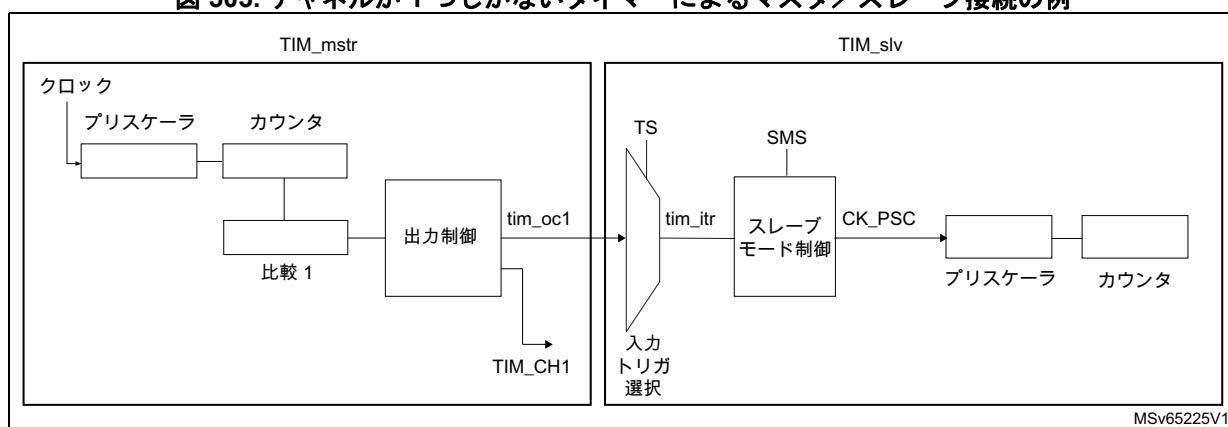


図 503. チャンネルが1つしかないタイマーによるマスタ/スレーブ接続の例



注： チャンネルが1つしかないタイマー（図 503 を参照）には、マスタモードはありません。ただし、tim_oc1 出力信号をスレーブタイマのトリガとして使用できます（[セクション 39.4.2: TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)の TIMx 内部トリガ接続の表を参照）。

tim_oc1 信号パルス幅は、スレーブタイマがトリガを検出するように、転送先のタイマの少なくとも 2 クロックサイクルになるようにプログラムする必要があります。

たとえば、転送先のタイマ tim_ker_ck のクロックが、転送元のタイマより 4 倍遅い場合、OC1 パルス幅は 8 クロックサイクルである必要があります。

タイマを別のタイマのプリスケアラとして使用する

たとえば、TIM_mstr が TIM_slv のプリスケアラとして動作するように設定できます。図 502 を参照してください。このためには、次の操作を行います。

1. 更新イベント UEV ごとに定期的なトリガ信号を出力するように、TIM_mstr をマスタモードに設定します。TIM_mstr_CR2 レジスタの MMS=010 を書き込んだ場合、更新イベントが生成されるたびに、tim_trgo で立ち上がりエッジが出力されます。
2. TIM_mstr の tim_trgo 出力を TIM_slv に接続するためには、TIM_slv を、ITR2 を内部トリガとして使用するスレーブモードに設定する必要があります。これは、TIM_slv_SMCR レジスタの TS ビットを通して選択されます（TS=00010 を書き込みます）。
3. 次に、スレーブモードコントローラを外部クロックモード 1 に設定しなければなりません（TIM_slv_SMCR レジスタで SMS=111 を書き込みます）。これにより TIM_slv は、TIM_mstr の定期的なトリガ信号の立ち上がりエッジ（TIM_mstr カウンタのオーバーフローに対応しています）をクロックとして動作します。

- 最後に、それぞれの CEN ビット (TIMx_CR1 レジスタ) をセットすることによって、両方のタイマを有効にする必要があります。

注： TIM_mstr のトリガ出力として tim_ocx が選択された場合 (MMS=1xx)、その立ち上がりエッジが TIM_slv カウンタのクロックとして使用されます。

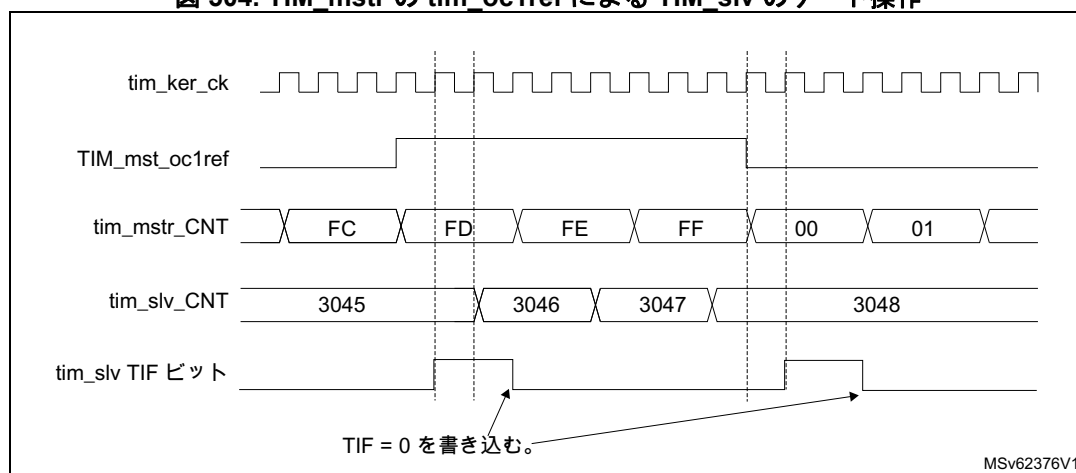
タイマを使用して別のタイマを有効にする

この例では、TIM_slv の有効化を、TIM_mstr の出力比較 1 で制御します。接続については、[図 502](#) を参照してください。TIM_slv は、分周された内部クロックでカウントします (TIM_mstr の tim_oc1ref がハイのときのみ)。両方のカウンタクロック周波数は、tim_ker_ck をプリスケアラで 3 分周したものです ($f_{tim_cnt_ck} = f_{tim_ker_ck}/3$)。

- TIM_mstr をマスタモードに設定して、その出力比較 1 基準 (tim_oc1ref) 信号をトリガ出力として送信します (TIM_mstr_CR2 レジスタの MMS=100)。
- TIM_mstr の tim_oc1ref 波形を設定します (TIM_mstr_CCMR1 レジスタ)。
- TIM_slv が TIM_mstr から入力トリガを受け取るように設定します (TIM_slv_SMCR レジスタの TS=00010)。
- TIM_slv をゲートモードに設定します (TIM_slv_SMCR レジスタの SMS=101)。
- CEN ビット (TIM_slv_CR1 レジスタ) に“1”を書き込んで、TIM_slv を有効にします。
- CEN ビット (TIM_mstr_CR1 レジスタ) に“1”を書き込んで、TIM_mstr を開始します。

注： スレーブタイマのカウンタクロックはマスタタイマのカウンタクロックと同期しないので、このモードは TIM_slv カウンタのイネーブル信号にのみ影響を与えます。

図 504. TIM_mstr の tim_oc1ref による TIM_slv のゲート操作



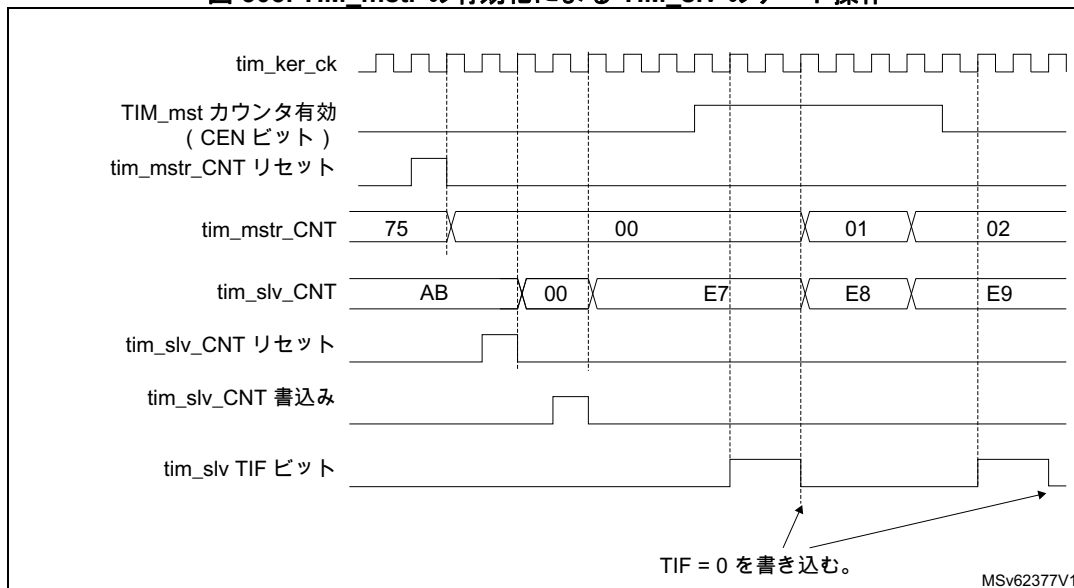
MSv62376V1

[図 504](#) の例では、TIM_slv のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。TIM_mstr を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。その後、任意の値をタイマカウンタに書き込むことができます。TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では ([図 505](#) を参照)、TIM_mstr と TIM_slv を同期させます。TIM_mstr はマスタであり、0 からスタートします。TIM_slv はスレーブであり、0xE7 から開始します。プリスケアラの分周比は両方のタイマで同じです。TIM_mstr_CR1 レジスタの CEN ビットに 0 を書き込むことによって TIM_mstr を無効にすると、TIM_slv は停止します。

1. TIM_mstr をマスタモードに設定して、その出力比較 1 基準 (tim_oc1ref) 信号をトリガ出力として送信します (TIM_mstr_CR2 レジスタの MMS=100)。
2. TIM_mstr の tim_oc1ref 波形を設定します (TIM_mstr_CCMR1 レジスタ)。
3. TIM_slv が TIM_mstr から入カトリガを受け取るように設定します (TIM_slv_SMCR レジスタの TS=00010)。
4. TIM_slv をゲートモードに設定します (TIM_slv_SMCR レジスタの SMS=101)。
5. UG ビット (TIM_mstr_EGR レジスタ) に“1”を書き込むことによって、TIM_mstr をリセットします。
6. UG ビット (TIM_slv_EGR レジスタ) に“1”を書き込むことによって、TIM_slv をリセットします。
7. TIM_slv カウンタ (TIM_slv_CNT) に“0xE7”を書き込んで、TIM_slv を 0xE7 に初期化します。
8. CEN ビット (TIM_slv_CR1 レジスタ) に“1”を書き込んで、TIM_slv を有効にします。
9. CEN ビット (TIM_mstr_CR1 レジスタ) に“1”を書き込んで、TIM_mstr を開始します。
10. CEN ビット (TIM_mstr_CR1 レジスタ) に“0”を書き込んで、TIM_mstr を停止します。

図 505. TIM_mstr の有効化による TIM_slv のゲート操作

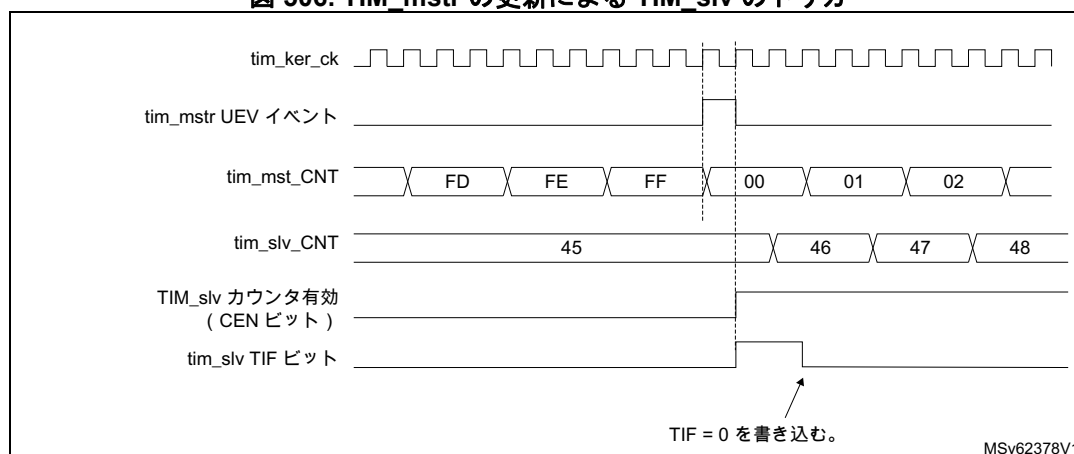


タイマを使用して別のタイマを開始する

この例では、TIM_mstr の更新イベントによって TIM_slv の有効化を設定します。接続については、[図 502](#) を参照してください。TIM_mstr によって更新イベントが生成されると、TIM_slv は、分周された内部クロックで現在値 (ゼロである必要はありません) からカウントを開始します。TIM_slv がトリガ信号を受信すると、その CEN ビットが自動的にセットされ、カウンタは TIM_slv_CR1 レジスタの CEN ビットに 0 が書き込まれるまでカウントします。両方のカウンタクロック周波数は、tim_ker_ck をプリスケアラで 3 分周したものです ($f_{tim_cnt_ck} = f_{tim_ker_ck}/3$)。

1. TIM_mstr をマスタモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM_mstr_CR2 レジスタの MMS=010)。
2. TIM_mstr の周期を設定します (TIM_mstr_ARR レジスタ)。
3. TIM_slv が TIM_mstr から入カトリガを受け取るように設定します (TIM_slv_SMCR レジスタの TS=00010)。
4. TIM_slv をトリガモードに設定します (TIM_slv_SMCR レジスタの SMS=110)。
5. CEN ビット (TIM_mstr_CR1 レジスタ) に“1”を書き込んで、TIM_mstr を開始します。

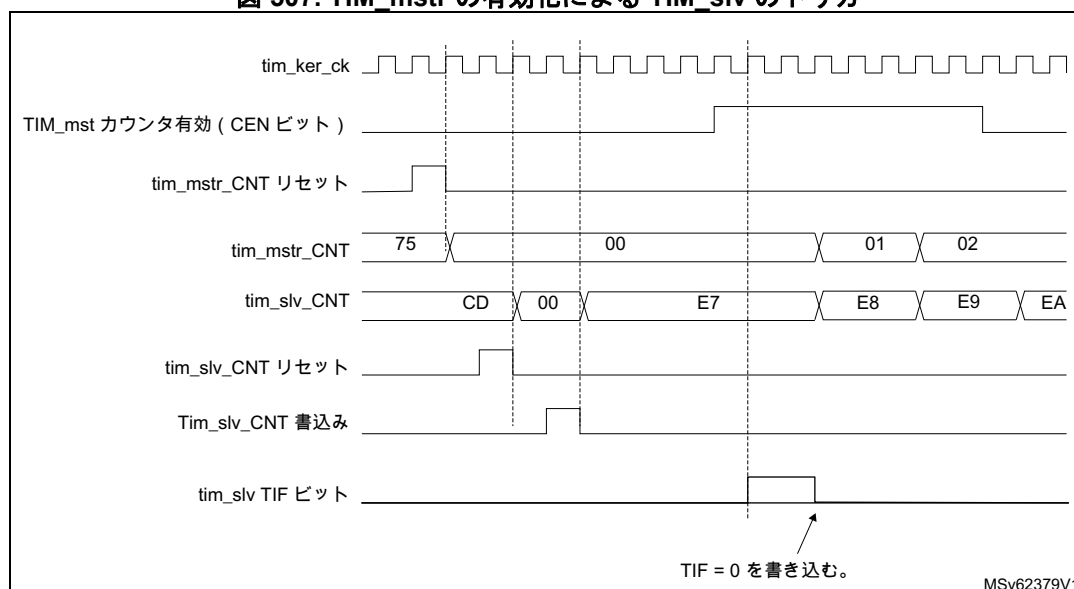
図 506. TIM_mstr の更新による TIM_slv のトリガ



MSv62378V1

前の例と同じように、カウントを開始する前に両方のカウンタを初期化できます。図 507 は図 506 と同じ設定ですが、ゲートモードではなくトリガモードでの動作を示します (TIM_slv_SMCR レジスタの SMS=110)。

図 507. TIM_mstr の有効化による TIM_slv のトリガ



MSv62379V1

外部トリガに対応して 2 つのタイマを同期して開始します。

この例では、tim_ti1 入力の立ち上がりエッジで TIM_mstr を有効にし、TIM_mstr が有効になると、TIM_slv を有効にします。接続については、図 502 を参照してください。カウンタの同時性を確保するため、TIM_mstr はマスタ/スレーブモードに設定する必要があります (tim_ti1 に対してはスレーブ、TIM_slv に対してはマスタ)。

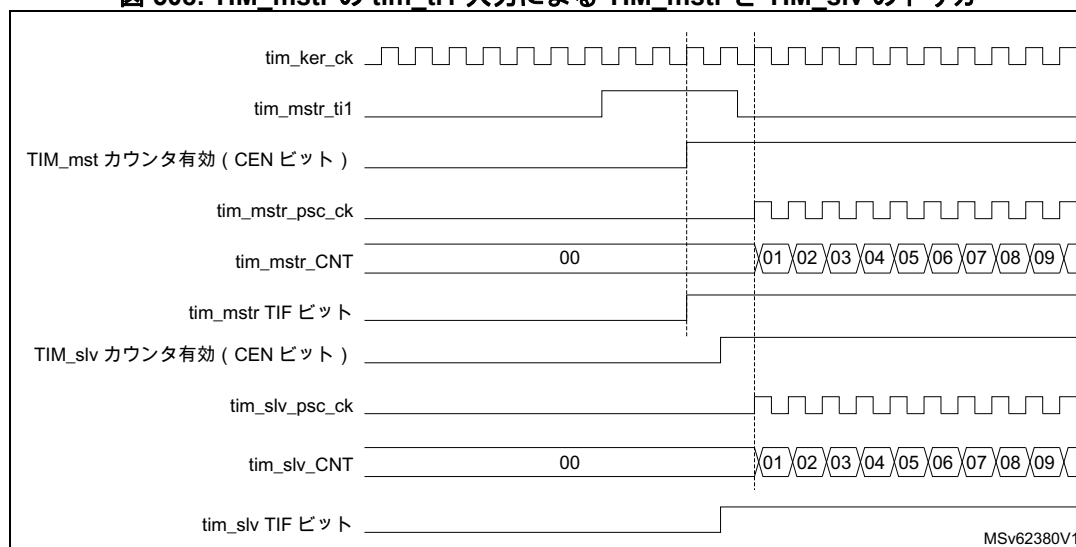
1. TIM_mstr をマスタモードに設定して、その有効化をトリガ出力として送信します (TIM_mstr_CR2 レジスタの MMS=001)。
2. TIM_mstr をスレーブモードに設定して、tim_ti1 から入力トリガを受け取るようにします (TIM_mstr_SMCR レジスタの TS=00100)。
3. TIM_mstr をトリガモードに設定します (TIM_mstr_SMCR レジスタの SMS=110)。

4. MSM=1 (TIM_mstr_SMCR レジスタ) を書き込むことによって、TIM_mstr をマスタ/スレーブモードに設定します。
5. TIM_slv が TIM_mstr から入力トリガを受け取るように設定します (TIM_slv_SMCR レジスタの TS=00000)。
6. TIM_slv をトリガモードに設定します (TIM_slv_SMCR レジスタの SMS=110)。

tim_ti1 (TIM_mstr) で立ち上がりエッジが発生すると、両方のカウンタが同期して内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

注： この例では、両方のタイマが開始前に初期化されます (それぞれの UG ビットをセットすることによって)。両方のカウンタは 0 からスタートしますが、カウンタレジスタ (TIMx_CNT) のいずれかに書き込むことによって、両者間に容易にオフセットを挿入できます。マスタ/スレーブモードでは、TIM_mstr の CNT_EN と CK_PSC の間に遅延が挿入されます。

図 508. TIM_mstr の tim_ti1 入力による TIM_mstr と TIM_slv のトリガ



注： tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

39.4.24 ADC トリガ

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイイベントを生成できます。

注： tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

39.4.25 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト（バースト）を開始します。TIMx_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx_DMAR アドレスに対して読出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数（ハーフワード単位またはバイト単位）を示します。

TIMx_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し／書き込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx_CR1

00001 : TIMx_CR2

00010 : TIMx_SMCR

TIMx_DCR レジスタの DBSS[3:0] ビットは、DMA バースト転送をトリガする割り込みソースを定義します（詳細は[セクション 39.5.29 : TIMx DMA 制御レジスタ \(TIMx_DCR\) \(x = 2~5\)](#)を参照)。

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 回転送、DBA = 0xE および DBSS = 1。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタのUDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注：

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 である必要があります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

注： null 値を予約済みレジスタに書き込むことができます。

39.4.26 TIM2/TIM3/TIM4/TIM5 DMA リクエスト

表 410 に示すように、TIM2/TIM3/TIM4/TIM5 は DMA リクエストを生成できます。

表 410. DMA リクエスト

DMA リクエスト信号	DMA リクエストあり	有効制御ビット
tim_upd_dma	更新	UDE
tim_cc1_dma	キャプチャ／比較 1	CC1DE
tim_cc2_dma	キャプチャ／比較 2	CC2DE
tim_cc3_dma	キャプチャ／比較 3	CC3DE
tim_cc4_dma	キャプチャ／比較 4	CC4DE
tim_trg_dma	トリガ	TDE

注：一部のタイマの DMA リクエストを、DMA コントローラに接続してはなりません。詳細については、DMA のセクションを参照してください。

39.4.27 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M33 コアは停止状態)、TIMx カウンタは、通常どおりに動作を続けるか、または停止できます。

デバッグモードでの動作は、デバッグサポート (DBG) モジュールでタイマごとに専用の設定ビットを使用してプログラムできます。

詳細については、デバッグサポート (DBG) のセクションを参照してください。

39.4.28 TIM2/TIM3/TIM4/TIM5 低電力モード

表 411. 低消費電力モードが TIM2/TIM3/TIM4/TIM5 に与える影響

モード	説明
SLEEP	影響はなく、ペリフェラルはアクティブです。割込みによって、デバイスは SLEEP モードから復帰できます。
STOP	タイマの動作は停止され、レジスタの内容は保持されます。割込みは生成できません。
STANDBY	タイマはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

39.4.29 TIM2/TIM3/TIM4/TIM5 割込み

表 412 に示すように、TIM2/TIM3/TIM4/TIM5 は複数の割込みを生成できます。

表 412. 割込みリクエスト

項目 (割込みの 略称)	割込みイベント	イベント フラグ	有効制御 ビット	割込みの クリア方法	SLEEP モード の終了	STOP および STANDBY モードの 終了
TIM_UP	更新	UIF	UIE	UIF に 0 を 書き込む	可能	不可
TIM_CC	キャプチャ/比較 1	CC1IF	CC1IE	CC1IF に 0 を 書き込む	可能	不可
	キャプチャ/比較 2	CC2IF	CC2IE	CC2IF に 0 を 書き込む	可能	不可
	キャプチャ/比較 3	CC3IF	CC3IE	CC3IF に 0 を 書き込む	可能	不可
	キャプチャ/比較 4	CC4IF	CC4IE	CC4IF に 0 を 書き込む	可能	不可
TIM_TRG	トリガ	TIF	TIE	TIF に 0 を 書き込む	可能	不可
TIM_DIR _IDX	インデックス	IDXF	IDXIE	IDXF に 0 を 書き込む	可能	不可
	方向	DIRF	DIRIE	DIRF に 0 を 書き込む	可能	不可
TIM_IERR	インデックスエラー	IERRF	IERRIE	IERRF に 0 を 書き込む	可能	不可
TIM_TER	遷移エラー	TERRF	TERRIE	TERRF に 0 を 書き込む	可能	不可

39.5 TIM2/TIM3/TIM4/TIM5 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

39.5.1 TIMx 制御レジスタ 1 (TIMx_CR1) (x = 2~5)

アドレスオフセット : 0x000

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
			r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デジタルフィルタ (tim_etr_in、tim_tix) によって使用されるサンプリングクロックとの間の分周比を示します。

00 : $t_{DTS} = t_{tim_ker_ck}$

01 : $t_{DTS} = 2 \times t_{tim_ker_ck}$

10 : $t_{DTS} = 4 \times t_{tim_ker_ck}$

11 : 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]** : センターアラインモード選択

00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01 : センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10 : センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントアップしているときのみセットされます。

11 : センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注 : カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

- 0 : カウンタはアップカウンタとして使用されます。
- 1 : カウンタはダウンカウンタとして使用されます。

注 : このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。
0 : 次のイベントのいずれかが更新割込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCRx) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

注 : 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

39.5.2 TIMx 制御レジスタ 2 (TIMx_CR2) (x = 2~5)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								rw	rw	rw	rw	rw			

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 24:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S** : tim_ti1 選択

0 : tim_ti1_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます。

1 : tim_ti1_in[15:0]、tim_ti2_in[15:0]、および tim_ti3_in[15:0] マルチプレクサ出力が XOR 演算されて tim_ti1 入力に接続されます。 [セクション 38.3.29 : 1488 ページのホールセンサとのインタフェース](#) も参照してください。

ビット 25、6、5、4 **MMS[3:0]** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (tim_trgo) を選択するために使用します。組み合わせは、次のとおりです。

0000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (tim_trgo) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、tim_trgo 信号は実際のリセットより遅延します。

0001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (tim_trgo) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (AND) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、tim_trgo には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

0010 : **更新** - 更新イベントがトリガ出力 (tim_trgo) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

0011 : **パルス比較** - キャプチャまたは比較一致が発生すると (tim_trgo)、CC1IF フラグがセットされるとき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します

0100 : **比較** - tim_oc1refc 信号がトリガ出力 (tim_trgo) として使用されます。

0101 : **比較** - tim_oc2refc 信号がトリガ出力 (tim_trgo) として使用されます。

0110 : **比較** - tim_oc3refc 信号がトリガ出力 (tim_trgo) として使用されます。

0111 : **比較** - tim_oc4refc 信号がトリガ出力 (tim_trgo) として使用されます。

1000 : **エンコーダクロック出力** - エンコーダクロック信号がトリガ出力 (tim_trgo) として使用されます。このコードは次の SMS[3:0] 値に対して有効です。0001、0010、0011、1010、1011、1100、1101、1110、1111。その他の SMS[3:0] コードは許されず、予想外の動作につながる可能性があります。

その他 : 予約済み

注 : スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

39.5.3 TIMx スレーブモード制御レジスタ (TIMx_SMCR) (x = 2~5)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	SMSPS	SMSPE	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
						rw	rw			rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **SMSPS** : SMS プリロードソース

このビットは、プリロードからアクティブへの SMS[3:0] ビットフィールドの転送をトリガするイベントのいずれかを選択します。

0 : 転送は、タイマの更新イベントによってトリガされます。

1 : 転送は、インデックスイベントによってトリガされます。

ビット 24 **SMSPE** : SMS プリロードイネーブル

このビットは、SMS[3:0] ビットフィールドをプリロードするかどうかを選択します。

0 : SMS[3:0] ビットフィールドはプリロードされません。

1 : SMS[3:0] プリロードは有効です。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **ETP** : 外部トリガ極性

このビットは、tim_etr_in と tim_etr_in のいずれがトリガ動作に使用されるかを選択します。

0 : tim_etr_in は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。

1 : tim_etr_in は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0 : 外部クロックモード 2 は無効です。

1 : 外部クロックモード 2 は有効です。カウンタは、tim_etrif 信号のアクティブエッジによってクロック供給されます。

注 : **ECE** ビットをセットすることは、tim_trgi が tim_etrif に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=00111)。

外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、tim_trgi を tim_etrif に接続することはできません (TS ビットが 00111 でないことが必要)。

外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は tim_etrif です。

ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 tim_etrp の周波数は、tim_ker_ck 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、tim_etrp 周波数を低減できます。これは、高速な外部クロックを tim_etr_in に入力するときに役立ちます。

00 : プリスケアラオフ

01 : tim_etrp 周波数は 2 分周されます。

10 : tim_etrp 周波数は 4 分周されます。

11 : tim_etrp 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、tim_etrp 信号をサンプルする周波数と、tim_etrp に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (tim_trgi) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (tim_trgo を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 21、20、6、5、**TS[4:0]** : トリガ選択

4 このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

00000 : 内部トリガ 0 (tim_itr0)

00001 : 内部トリガ 1 (tim_itr1)

00010 : 内部トリガ 2 (tim_itr2)

00011 : 内部トリガ 3 (tim_itr3)

00100 : tim_ti1 エッジ検出回路 (tim_ti1f_ed)

00101 : フィルタタイマ入力 1 (tim_ti1fp1)

00110 : フィルタタイマ入力 2 (tim_ti2fp2)

00111 : 外部トリガ入力 (tim_etrp)

01000 : 内部トリガ 4 (tim_itr4)

01001 : 内部トリガ 5 (tim_itr5)

01010 : 内部トリガ 6 (tim_itr6)

01011 : 内部トリガ 7 (tim_itr7)

01100 : 内部トリガ 8 (tim_itr8)

01101 : 内部トリガ 9 (tim_itr9)

01110 : 内部トリガ 10 (tim_itr10)

01111 : 内部トリガ 11 (tim_itr11)

10000 : 内部トリガ 12 (tim_itr12)

10001 : 内部トリガ 13 (tim_itr13)

10010 : 内部トリガ 14 (tim_itr14)

10011 : 内部トリガ 15 (tim_itr15)

その他 : 予約済み

製品固有の実装の詳細は [セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#) を参照してください。

注 : 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS = 000 のときなど) にのみ変更しなければなりません。

ビット 3 **OCCS** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

0 : tim_ocref_clr_int は tim_ocref_clr 入力に接続されます。

1 : tim_ocref_clr_int は tim_etr1 に接続されます。

注 : OCREF クリア選択機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に "0" に設定されます。セクション 39.3 : TIM2/TIM3/TIM4/TIM5 の実装を参照してください。

ビット 16、2、1、0 **SMS[3:0]** : スレーブモード選択

外部信号が選択されると、トリガ信号 (tim_trgi) のアクティブエッジが、外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : エンコーダモード 1 - カウンタは、tim_ti2fp2 のレベルに応じて、tim_ti1fp1 のエッジでカウントアップ/ダウンします。

0010 : エンコーダモード 2 - カウンタは、tim_ti1fp1 のレベルに応じて、tim_ti2fp2 のエッジでカウントアップ/ダウンします。

0011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、tim_ti1fp1 と tim_ti2fp2 の両方のエッジでカウントアップ/ダウンします。

0100 : リセットモード - 選択されたトリガ入力 (tim_trgi) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ tim_trgi の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (tim_trgi) の立ち上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (tim_trgi) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1001 : ゲートモードとリセットモードの組み合わせ - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

1010 : エンコーダモード : クロックプラス方向、x2 モード。

1011 : エンコーダモード : クロックプラス方向、x1 モード。tim_ti2fp2 エッジ検出は CC2P によって設定されます。

1100 : エンコーダモード : 方向クロック、x2 モード。

1101 : エンコーダモード : 方向クロック、x1 モード。tim_ti1fp1 と tim_ti2fp2 エッジ検出は CC1P と CC2P によって設定されます。

1110 : 直交エンコーダモード : x1 モード、tim_ti1fp1 エッジのみでカウント、エッジ検出は CC1P によって設定されます。

1111 : 直交エンコーダモード : x1 モード、tim_ti2fp2 エッジのみでカウント、エッジ検出は CC2P によって設定されます。

注 : トリガ入力として tim_ti1f_ed が選択されている場合 (TS=00100)、ゲートモードを使用することはできません。tim_ti1f_ed は tim_ti1f の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注 : tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

39.5.4 TIMx DMA／割込み有効レジスタ (TIMx_DIER) (x = 2~5)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR IE	IERR IE	DIRIE	IDXIE	Res.	Res.	Res.	Res.
								rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TERRIE** : 遷移エラー割込みイネーブル

0 : 遷移エラー割込みは無効です。

1 : 遷移エラー割込みは有効です。

ビット 22 **IERRIE** : インデックスエラー割込みイネーブル

0 : インデックスエラー割込みは無効です。

1 : インデックスエラー割込みは有効です。

ビット 21 **DIRIE** : 方向変更割込みイネーブル

0 : 方向変更割込みは無効です。

1 : 方向変更割込みは有効です。

ビット 20 **IDXIE** : インデックス割込みイネーブル

0 : インデックス割込みは無効です。

1 : インデックス割込みは有効です。

ビット 19:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割込みイネーブル

- 0 : トリガ割込みは無効です。
- 1 : トリガ割込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IE** : キャプチャ／比較 4 割込みイネーブル

- 0 : CC4 割込みは無効です。
- 1 : CC4 割込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割込みイネーブル

- 0 : CC3 割込みは無効です。
- 1 : CC3 割込みは有効です。

ビット 2 **CC2IE** : キャプチャ／比較 2 割込みイネーブル

- 0 : CC2 割込みは無効です。
- 1 : CC2 割込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割込みイネーブル

- 0 : CC1 割込みは無効です。
- 1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル

- 0 : 更新割込みは無効です。
- 1 : 更新割込みは有効です。

39.5.5 TIMx ステータスレジスタ (TIMx_SR) (x = 2~5)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERRF	IERRF	DIRF	IDXF	Res.	Res.	Res.	Res.
								rc_w0	rc_w0	rc_w0	rc_w0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TERRF** : 遷移エラー割込みフラグ

このフラグは、エンコーダモードで遷移エラーが検出されたときに、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

- 0 : エンコーダ遷移エラーは検出されていません。
- 1 : エンコーダ遷移エラーが検出されています。

ビット 22 **IERRF** : インデックスエラー割込みフラグ

このフラグは、インデックスエラーが検出されたときに、ハードウェアによってセットされます。ソフトウェアによって

“0”を書き込むことによってクリアされます。

- 0 : インデックスエラーは検出されていません。
- 1 : インデックスエラーが検出されています。

ビット 21 DIRF : 方向変更割込みフラグ

このフラグは、エンコーダモードで方向が変わった (TIMx_CR の DIR ビット値が変わった) ときに、ハードウェアによってセットされます。

これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : 方向の変更なし

1 : 方向の変更

ビット 20 IDXF : インデックス割込みフラグ

このフラグは、インデックスイベントが検出されたときに、ハードウェアによってセットされます。ソフトウェアによって

“0”を書き込むことによってクリアされます。

0 : インデックスイベントは発生していません。

1 : インデックスイベントが発生しています。

ビット 19:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 CC4OF : キャプチャ/比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 CC3OF : キャプチャ/比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 CC2OF : キャプチャ/比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 CC1OF : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 TIF : トリガ割込みフラグ

このフラグは、TRG トリガイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、tim_trgi 入力でアクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

0 : トリガイベントは発生していません。

1 : トリガ割込みが保留中です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 CC4IF : キャプチャ/比較 4 割込みフラグ

CC1IF の説明を参照してください。

ビット 3 CC3IF : キャプチャ/比較 3 割込みフラグ

CC1IF の説明を参照してください。

ビット 2 CC2IF : キャプチャ/比較 2 割込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって（入力キャプチャモードまたは出力比較モード）、または TIMx_CCR1 レジスタを読み出すことによって（入力キャプチャモードのみ）クリアされます。

0 : 比較一致/入力キャプチャは発生していません。

1 : 比較一致または入力キャプチャが発生しました。

チャンネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャンネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた（TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ感度に従って、IC1 でエッジが検出された）ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

オーバーフローまたはアンダーフロー時と、TIMx_CR1 レジスタで UDIS=0 の場合。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき（同期制御レジスタの説明を参照）。

39.5.6 TIMx イベント生成レジスタ (TIMx_EGR) (x = 2~5)

アドレスオフセット : 0x014

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4G** : キャプチャ/比較 4 生成

CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ/比較 3 生成

CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ／比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

39.5.7 TIMx キャプチャ／比較モードレジスタ 1 (TIMx_CCMR1) (x = 2~5)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード (このセクション) または出力比較モード (次のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます (例 : 入力キャプチャモードでのチャンネル 1 と出力比較モードでのチャンネル 2)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F [3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F[3:0]** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 3:2 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E=0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

39.5.8 TIMx キャプチャ／比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 2~5)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（例：入力キャプチャモードでのチャンネル 1 と出力比較モードでのチャンネル 2）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 24、14:12 **OC2M[3:0]** : 出力比較 2 モード

ビット 6:4 の OC1M 説明を参照

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : tim_oc1ref は tim_ocref_clr_int 入力の影響を受けません。

1 : tim_oc1ref は tim_ocref_clr_int 入力のハイレベルが検出されるとクリアされます。

ビット 16、6:4 OC1M[3:0] : 出力比較 1 モード

これらのビットは、tim_oc1 が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 のアクティブレベルは CC1P ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。このモードは、タイマーをソフトウェアのタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードを有効にすると、出力は停止状態に入る前の状態（アクティブまたは非アクティブ）を保持します。

0001 : 一致時にチャンネル 1 をアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 1 をインアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制非アクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (tim_oc1ref=0) に、そうでない場合はアクティブ (tim_oc1ref=1) になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 OR です。

1101 : 組み合わせ PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 AND です。

1110 : 非対称 PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は、カウンタがカウントアップするときに tim_oc1ref を出力し、カウントダウンするときに tim_oc2ref を出力します。

1111 : 非対称 PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に挙動します。tim_oc1refc は、カウンタがカウントアップするときに tim_oc1ref を出力し、カウントダウンするときに tim_oc2ref を出力します。

注 : PWM モードでは、比較結果が変化したとき、出力比較モードが「停止」モードから「PWM」モードに変更されたとき、および出力比較モードが「強制アクティブ/非アクティブ」モードから「PWM」モードに変更されたときに、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われず。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、トリガイベントとタイマ出力の遷移の間の待ち時間を短縮します。開始トリガの後、できるだけ早く出力パルスを開始するには、1 パルスモード (TIMx_CR1 レジスタの OPM ビットのセット) で使用する必要があります。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

39.5.9 **TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2) (x = 2~5)**

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード (このセクション) または出力比較モード (次のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます (例 : 入力キャプチャモードでのチャンネル 1 と出力比較モードでのチャンネル 2)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F [3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F [3:0]				IC3PSC [1:0]		CC3S [1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC4F[3:0]** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC[1:0]** : 入力キャプチャ 4 プリスケアラ

ビット 9:8 **CC4S[1:0]** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti4 に配置されます。

10 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti3 に配置されます。

11 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC4S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC3F[3:0]** : 入力キャプチャ 3 フィルタビット 3:2 **IC3PSC[1:0]** : 入力キャプチャ 3 プリスケアラビット 1:0 **CC3S[1:0]** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti3 に配置されます。

10 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti4 に配置されます。

11 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC3S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

39.5.10 TIMx キャプチャ／比較モードレジスタ 2 [オルタネート] (TIMx_CCMR2) (x = 2~5)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（例：入力キャプチャモードでのチャンネル 1 と出力比較モードでのチャンネル 2）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 24、14:12 **OC4M[3:0]** : 出力比較 4 モード

OC1M の説明 (TIMx_CCMR1 レジスタのビット 6:4) を参照

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S[1:0]** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti4 に配置されます。

10 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_ti3 に配置されます。

11 : CC4 チャンネルは入力として設定され、tim_ic4 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 16、6:4 **OC3M[3:0]** : 出力比較 3 モード

OC1M の説明 (TIMx_CCMR1 レジスタのビット 6:4) を参照してください。

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S[1:0]** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti3 に配置されます。

10 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_ti4 に配置されます。

11 : CC3 チャンネルは入力として設定され、tim_ic3 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

39.5.11 TIMx キャプチャ／比較有効レジスタ (TIMx_CCER) (x = 2~5)

アドレスオフセット : 0x020

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rW		rW	rW	rW		rW	rW	rW		rW	rW	rW		rW	rW

ビット 15 **CC4NP** : キャプチャ／比較 4 出力極性

CC1NP の説明を参照してください。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P** : キャプチャ／比較 4 出力極性

CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ／比較 4 出力イネーブル。

CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ／比較 3 出力極性

CC1NP の説明を参照してください。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC3P** : キャプチャ／比較 3 出力極性

CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ／比較 3 出力イネーブル。

CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性

CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル。

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 : この場合、CC1NP はクリアされたままでなければなりません。

CC1 チャンネルが入力として設定されている場合 : このビットは、tim_ti1fp1/tim_ti2fp1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

0 : OC1 のアクティブハイ (出力モード) / エッジ感度を選択 (入力モード、下記を参照)

1 : OC1 のアクティブロー (出力モード) / エッジ感度を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されます (ゲートモードまたはエンコーダモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ。この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。

0 : キャプチャモードは無効/OC1 は非アクティブ。

1 : キャプチャモードは有効/OC1 信号は、対応する出力ピンに出力されます。

表 413. 標準 tim_ocx チャンネルの出力制御ビット

CCxE ビット	tim_ocx 出力状態
0	出力無効 (タイマによって駆動されない : ハイインピーダンス)
1	出力は有効 (tim_ocx = tim_ocxref + 極性)

注 : CCxE=0 のとき、標準 tim_ocx チャンネルに接続されている外部 IO ピンの状態は、GPIO レジスタのみに依存します。

39.5.12 TIMx カウンタ (TIMx_CNT) (x = 3、4)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : 値は TIMx_CR1 の UIFREMAP に依存します。

UIFREMAP = 0 の場合 :

予約済みです。

UIFREMAP = 1 の場合 :

UIFCPY : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分を保持します。分数部分は得られません。

39.5.13 TIMx カウンタ (TIMx_CNT) (x = 2、5)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY_ CNT [31]	CNT[30:16]														
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY_CNT[31]** : 値は TIMx_CR1 の UIFREMAP に依存します。

UIFREMAP = 0 の場合 :

CNT[31] : カウンタ値の最上位ビット

UIFREMAP = 1 の場合 :

UIFCPY : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー

ビット 30:0 **CNT[30:0]** : カウンタ値の最下位部分

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[30:0] に非ディザリング部分を保持します。分数部分は得られません。

39.5.14 TIMx プリスケアラ (TIMx_PSC) (x = 2~5)

アドレスオフセット : 0x028

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 $f_{tim_cnt_ck}$ は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

39.5.15 TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 3, 4)

アドレスオフセット : 0x02C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値下位ビット

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 39.4.3 : 1556 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、自動再ロード値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

39.5.16 TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 2、5)

アドレスオフセット : 0x02C

リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **ARR[31:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 39.4.3: 1556 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、自動再ロード値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[31:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

39.5.17 TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 3、4)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR1[19:0] : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

CCR1[15:0] ビットはキャプチャ値を保持します。CCR1[19:16] ビットは予約済みです。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:0] にキャプチャを保持します。CCR1[3:0] ビットはリセットされています。

39.5.18 TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 2、5)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **CCR1[31:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、比較値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[31:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、キャプチャ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[31:0] にキャプチャを保持します。CCR1[3:0] ビットはリセットされています。

39.5.19 TIMx キャプチャ／比較レジスタ 2 (TIMx_CCR2) (x = 3、4)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR2[19:0] : キャプチャ／比較 1 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc2 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] に比較値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] に整数部を保持します。CCR2[3:0] ビットフィールドにはディザリング部分が含まれます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 2 イベント (tim_ic2) によって転送されたカウンタ値です。TIMx_CCR2 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

CCR2[15:0] ビットはキャプチャ値を保持します。CCR2[19:16] ビットは予約済みです。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:0] にキャプチャを保持します。CCR2[3:0] ビットはリセットされています。

39.5.20 TIMx キャプチャ／比較レジスタ 2 (TIMx_CCR2) (x = 2、5)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **CCR2[31:0]** : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc2 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、比較値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[31:4] に整数部を保持します。CCR2[3:0] ビットフィールドにはディザリング部分が含まれます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 2 イベント (tim_ic2) によって転送されたカウンタ値です。TIMx_CCR2 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、キャプチャ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[31:0] にキャプチャを保持します。CCR2[3:0] ビットはリセットされています。

39.5.21 TIMx キャプチャ／比較レジスタ 3 (TIMx_CCR3) (x = 3、4)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR3[19:0] : キャプチャ／比較 3 値

CC3 チャンネルが出力として設定されている場合 :

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値 (プリロード値) です。TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc3 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR3[15:0] に比較値を保持します。CCR3[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[19:4] に整数部を保持します。CCR3[3:0] ビットフィールドにはディザリング部分が含まれます。

CC3 チャンネルが入力として設定されている場合 :

CCR3 は、最後の入力キャプチャ 3 イベント (tim_ic3) によって転送されたカウンタ値です。TIMx_CCR3 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

CCR3[15:0] ビットはキャプチャ値を保持します。CCR3[19:16] ビットは予約済みです。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[19:0] にキャプチャを保持します。CCR3[3:0] ビットはリセットされています。

39.5.22 TIMx キャプチャ／比較レジスタ 3 (TIMx_CCR3) (x = 2、5)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **CCR3[31:0]** : キャプチャ／比較 3 値

CC3 チャンネルが出力として設定されている場合 :

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc3 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、比較値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[31:4] に整数部を保持します。CCR3[3:0] ビットフィールドにはディザリング部分が含まれます。

CC3 チャンネルが入力として設定されている場合 :

CCR3 は、最後の入力キャプチャ 3 イベント (tim_ic3) によって転送されたカウンタ値です。TIMx_CCR3 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、キャプチャ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR3[31:0] にキャプチャを保持します。CCR3[3:0] ビットはリセットされています。

39.5.23 TIMx キャプチャ／比較レジスタ 4 (TIMx_CCR4) (x = 3、4)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR4[19:0] : キャプチャ／比較 4 値

CC4 チャンネルが出力として設定されている場合 :

CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc4 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR4[15:0] に比較値を保持します。CCR4[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[19:4] に整数部を保持します。CCR4[3:0] ビットフィールドにはディザリング部分が含まれます。

CC4 チャンネルが入力として設定されている場合 :

CCR4 は、最後の入力キャプチャ 4 イベント (tim_ic4) によって転送されたカウンタ値です。TIMx_CCR4 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

CCR4[15:0] ビットはキャプチャ値を保持します。CCR4[19:16] ビットは予約済みです。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[19:0] にキャプチャを保持します。CCR4[3:0] ビットはリセットされています。

39.5.24 TIMx キャプチャ／比較レジスタ 4 (TIMx_CCR4) (x = 2、5)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **CCR4[31:0]** : キャプチャ／比較 4 値

CC4 チャンネルが出力として設定されている場合 :

CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc4 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、比較値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[31:4] に整数部を保持します。CCR4[3:0] ビットフィールドにはディザリング部分が含まれます。

CC4 チャンネルが入力として設定されている場合 :

CCR4 は、最後の入力キャプチャ 4 イベント (tim_ic4) によって転送されたカウンタ値です。TIMx_CCR4 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、キャプチャ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR4[31:0] にキャプチャを保持します。CCR4[3:0] ビットはリセットされています。

39.5.25 TIMx タイマエンコーダ制御レジスタ (TIMx_ECR) (x = 2~5)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PWPRSC[2:0]			PW[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPOS[1:0]		FIDX	IBLK[1:0]		IDIR[1:0]		IE
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **PWPRSC[2:0]** : パルス幅プリスケアラ

このビットフィールドは、パルス発生回路用のクロックプリスケアラを次のように設定します。

$$t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$$

ビット 23:16 **PW[7:0]** : パルス幅

このビットフィールドは、パルスの長さを次のように定義します。

$$t_{PW} = PW[7:0] \times t_{PWG}$$

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **IPOS[1:0]** : インデックス位置決め

直交エンコーダモード (SMS[3:0] = 0001, 0010, 0011, 1110, 1111) で、このビットは、インデックスイベントがカウンタをリセットする AB 入力の設定を示します。

00 : AB = 00 のときにインデックスがカウンタをリセットします。

01 : AB = 01 のときにインデックスがカウンタをリセットします。

10 : AB = 10 のときにインデックスがカウンタをリセットします。

11 : AB = 11 のときにインデックスがカウンタをリセットします。

方向クロックモードまたはクロックプラス方向モード (SMS[3:0] = 1010, 1011, 1100, 1101) で、これらのビットは、インデックスイベントがカウンタをリセットするレベルを示します。双方向クロックモードでは、これは両方のクロック入力に適用されます。

x0 : クロックが 0 のときにインデックスがカウンタをリセットします。

x1 : クロックが 1 のときにインデックスがカウンタをリセットします。

注 : IPOS[1] ビットは意味がありません。ビット 5 **FIDX** : 最初のインデックス

このビットは、最初のインデックスのみが考慮されるかどうかを示します。

0 : インデックスは常にアクティブです。

1 : 最初のインデックスのみがカウンタをリセットします。

ビット 4:3 **IBLK[1:0]** : インデックスのブランキング

このビットは、インデックスイベントが tim_ti3 入力によって調整されるかどうかを示します。

00 : インデックスは常にアクティブです。

01 : インデックスは、CC3P ビットフィールドに従って、tim_ti3 入力がアクティブなときに無効になります。

10 : インデックスは、CC4P ビットフィールドに従って、tim_ti4 入力がアクティブなときに無効になります。

11 : 予約済み

ビット 2:1 **IDIR[1:0]** : インデックスの方向

このビットは、どちらの方向のときにインデックスイベントがカウンタをリセットするかを示します。

00 : 方向に関係なく、インデックスがカウンタをリセットします。

01 : アップカウントのときのみ、インデックスがカウンタをリセットします。

10 : ダウンカウントのときのみ、インデックスがカウンタをリセットします。

11 : 予約済み

ビット 0 **IE**インデックスイネーブル

このビットは、インデックスイベントがカウンタをリセットするかどうかを示します。

0 : インデックスは無効です。

1 : インデックスは有効です

39.5.26 TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 2~5)

アドレスオフセット : 0x05C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL [3:0]				Res.	Res.	Res.	Res.	TI3SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **TI4SEL[3:0]** : tim_ti4[15:0] 入力を選択します。

0000 : tim_ti4_in0 : TIMx_CH4

0001 : tim_ti4_in1

.....

1111 : tim_ti4_in15

製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **TI3SEL[3:0]** : tim_ti3[15:0] 入力を選択します。

0000 : tim_ti3_in0 : TIMx_CH3

0001 : tim_ti3_in1

.....

1111 : tim_ti3_in15

製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TI2SEL[3:0]** : tim_ti2[15:0] 入力を選択します。

0000 : tim_ti2_in0 : TIMx_CH2

0001 : tim_ti2_in1

.....

1111 : tim_ti2_in15

製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1[15:0] 入力を選択します。
0000 : tim_ti1_in0 : TIMx_CH1
0001 : tim_ti1_in1
.....
1111 : tim_ti1_in15
製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

39.5.27 TIMx オルタネート機能レジスタ 1 (TIMx_AF1) (x = 2~5)

アドレスオフセット : 0x060
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:14 **ETRSEL[3:0]** : etr_in ソース選択
これらのビットは、etr_in 入力ソースを選択します。
0000 : tim_etr0 : TIMx_ETR 入力
0001 : tim_etr1
.....
1111 : tim_etr15
製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

39.5.28 TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 2~5)

アドレスオフセット : 0x064

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **OCRSEL[2:0]** : ocref_clr ソース選択

これらのビットは、ocref_clr 入力ソースを選択します。

000 : tim_ocref_clr0

001 : tim_ocref_clr1

.....

111 : tim_ocref_clr7

製品固有の実装については[セクション 39.4.2 : TIM2/TIM3/TIM4/TIM5 ピンおよび内部信号](#)を参照してください。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

39.5.29 TIMx DMA 制御レジスタ (TIMx_DCR) (x = 2~5)

アドレスオフセット : 0x3DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **DBSS[3:0]** : DMA パーストソース選択

このビットフィールドは、DMA パースト転送をトリガする割込みソースを定義します（タイマは、TIMx_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにパースト転送を認識します）。

0000 : 予約済み

0001 : 更新

0010 : CC1

0011 : CC2

0100 : CC3

0101 : CC4

0110 : COM

0111 : トリガ

その他 : 予約済み

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA パースト長

この 5 ビットのベクタは、転送数など DMA 転送長（タイマは、TIMx_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにパースト転送を認識します）を指定します。転送は、ハーフワードまたはバイトです（以下の例を参照）。

00000 : 1 回転送

00001 : 2 回転送

00010 : 3 回転送

.....

11010 : 26 回転送

例 : 次の転送を考えます : DBL = 7 バイトかつ DBA = TIM2_CR1。

–DBL = 7 バイトおよび DBA = TIM2_CR1 が転送するバイトのアドレスを表す場合、転送のアドレスは次の式で与えられます。

(TIMx_CR1 アドレス) + DBA + (DMA インデックス)、ここで DMA インデックス = DBL

この例では、(TIMx_CR1 アドレス) + DBA に 7 バイトが追加され、データのコピー元/コピー先アドレスが与えられます。この場合、転送は、以下のアドレスから始めて、7 つのレジスタに対して行われます。(TIMx_CR1 アドレス) + DBA

DMA データサイズの設定に応じて、いくつかのケースが想定されます。

–DMA データサイズをハーフワードで設定した場合、7 つのレジスタにそれぞれ 16 ビットのデータが転送されます。

–DMA データサイズをバイトで設定した場合も、データは 7 つのレジスタに転送されます（最初のレジスタには最初の MSB バイトが含まれ、2 番目のレジスタには最初の LSB バイトが含まれるなど、以降同様）。タイマへの転送で、DMA によって転送されるデータサイズを指定する必要もあります。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し/書込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx_CR1

00001 : TIMx_CR2

00010 : TIMx_SMCR

.....

39.5.30 完全転送の TIMx DMA アドレス (TIMx_DMAR) (x = 2~5)

アドレスオフセット : 0x3E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DMAB[31:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読出しまたは書込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx_DCR 内で設定)。

39.5.31 TIMx レジスタマップ

TIMx レジスタは、次の表のようにマップされます。

表 414. TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	TIMx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DITHEN	UIFREMAP	Res	CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN			
	リセット値																				0	0		0	0	0	0	0	0	0	0	0		
0x004	TIMx_CR2	Res	Res	Res	Res	Res	Res	MMS[3]	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	T1S	MMS[2:0]	CCDS	Res	Res	Res			
	リセット値							0																	0	0	0	0	0					
0x008	TIMx_SMCR	Res	Res	Res	Res	Res	Res	SMSPS	SMSPE	Res	TS[4:3]	Res	Res	Res	Res	Res	SMS[3]	ETP	EOE	ETPS[1:0]	ETF[3:0]	MSM	TS[2:0]	SMS[2:0]										
	リセット値							0	0		0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00C	TIMx_DIER	Res	Res	Res	Res	Res	Res	TERRIE	IERRIE	DIRIE	IDXIE	Res	Res	Res	Res	Res	Res	Res	TDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE		
	リセット値							0	0	0	0								0	0	0	0	0	0		0		0	0	0	0	0		
0x010	TIMx_SR	Res	Res	Res	Res	Res	Res	TERRF	IERRF	DIRF	IDXF	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	Res	Res	TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF	
	リセット値							0	0	0	0									0	0	0	0			0		0	0	0	0	0	0	
0x014	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	CC4G	CC3G	CC2G	CC1G	UG	
	リセット値																									0		0	0	0	0	0	0	
0x018	TIMx_CCMR1 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F [3:0]	IC2 PSC [1:0]	CC2S [1:0]	IC1F [3:0]	IC1 PSC [1:0]	CC1S [1:0]										
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M [2:0]	OC2PE	OC2FE	CC2S [1:0]	OC1CE	OC1M [2:0]	OC1PE	OC1FE	CC1S [1:0]							
	リセット値							0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x01C	TIMx_CCMR2 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F [3:0]	IC4 PSC [1:0]	CC4S [1:0]	IC3F [3:0]	IC3 PSC [1:0]	CC3S [1:0]										
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	TIMx_CCMR2 出力比較モード	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4CE	OC4M [2:0]	OC4PE	OC4FE	CC4S [1:0]	OC3CE	OC3M [2:0]	OC3PE	OC3FE	CC3S [1:0]							
	リセット値							0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x020	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4NP	Res	CC4P	CC4E	CC3NP	Res	CC3P	CC3E	CC2NP	Res	CC2P	CC2E	CC1NP	Res	CC1P	CC1E
	リセット値																		0		0	0	0		0	0		0	0		0	0		
0x024	TIMx_CNT	UIFCPY_CNT[31]	CNT[30:16] (32 ビットタイマでのみ CNT[31:16])															CNT[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x028	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 414. TIM2/TIM3/TIM4/TIM5レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x02C	TIMx_ARR (x = 3、4)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																			
	リセット値													0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x02C	TIMx_ARR (x = 2、5)	ARR[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x030	予約済み																																
0x034	TIMx_CCR1	CCR1 [31:20] (32 ビットタイマのみ)												CCR1 [19:0]																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x038	TIMx_CCR2	CCR2 [31:20] (32 ビットタイマのみ)												CCR2 [19:0]																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x03C	TIMx_CCR3	CCR3 [31:20] (32 ビットタイマのみ)												CCR3 [19:0]																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x040	TIMx_CCR4	CCR4 [31:20] (32 ビットタイマのみ)												CCR4 [19:0]																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x044.. 0x054	予約済み	Res.																															
0x058	TIMx_ECR	Res.	Res.	Res.	Res.	Res.	PWPRSC [2:0]		PW[7:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPOS [1:0]	FIDX	IBLK [1:0]	IDIR [1:0]	IE				
	リセット値						0	0	0	0	0	0	0	0	0	0	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0
0x05C	TIMx_TISEL	Res.	Res.	Res.	Res.	TI4SEL [3:0]				Res.	Res.	Res.	Res.	TI3SEL [3:0]				Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
	リセット値					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x060	TIMx_AF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値															0	0	0	0														
0x064	TIMx_AF2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2: 0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値															0	0	0															
0x068.. 0x3D8	予約済み	Res.																															
0x3DC	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]				Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
	リセット値													0	0	0	0				0	0	0	0	0				0	0	0	0	0
0x3E0	TIMx_DMAR	DMAB[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

40 基本タイマ (TIM6/TIM7)

40.1 TIM6/TIM7 の概要

基本タイマ TIM6/TIM7 は、プログラム可能なプリスケアラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

これらは、タイムベース生成のための汎用タイマとして使用できます。

基本タイマは、デジタル/アナログコンバータをトリガするために使用することもできます。これはタイマのトリガ出力で行われます。

タイマは完全に独立していて、いかなるリソースも共有しません。

40.2 TIM6/TIM7 の主な特徴

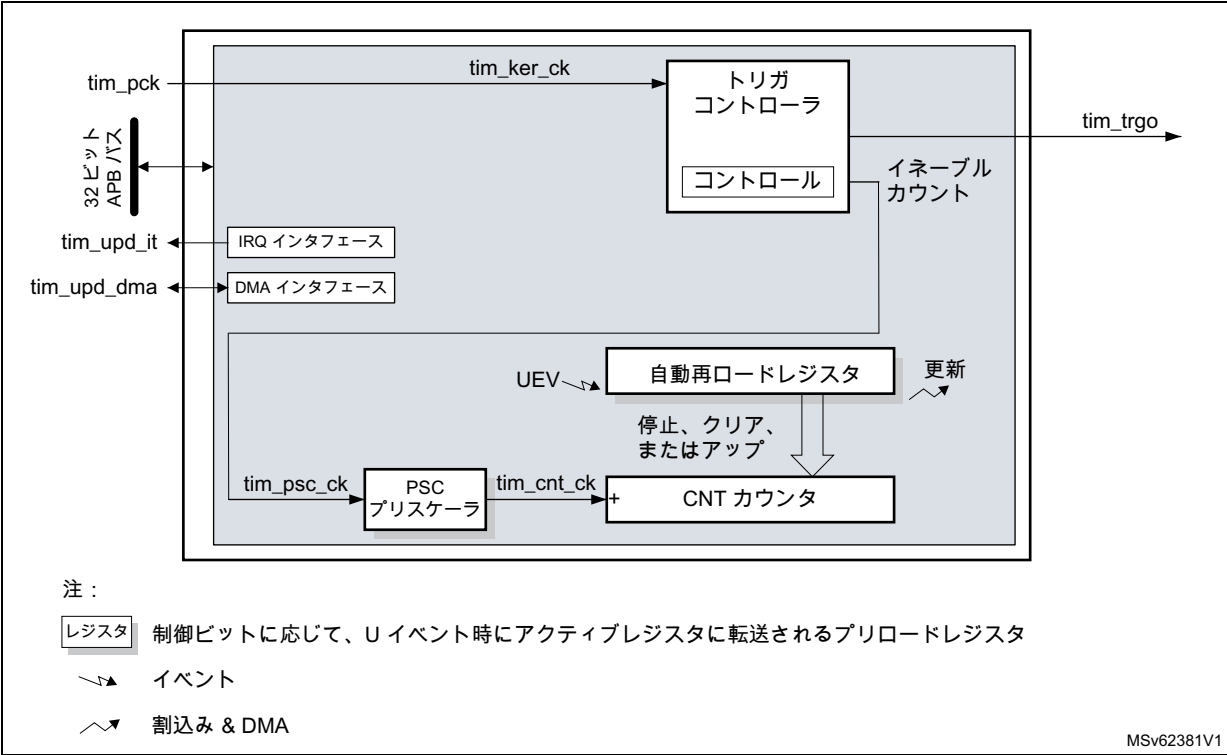
基本タイマ (TIM6/TIM7) の機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能
- DAC をトリガする同期回路
- 更新イベント時の割込み/DMA 生成：カウンタオーバーフロー

40.3 TIM6/TIM7機能詳細

40.3.1 TIM6/TIM7 ブロック図

図 509. 基本タイマブロック図



40.3.2 TIM6/TIM7 内部信号

このセクションの以下の表は、TIM の入力および出力の一覧です。

表 415. TIM 内部入力／出力信号

内部信号名	信号タイプ	説明
tim_pclk	入力	タイマ APB クロック
tim_ker_ck	入力	タイマカーネルクロック。このクロックは、tim_pclk（同じソースから導出）と同期する必要があります。クロック比 $\text{tim_ker_ck}/\text{tim_pclk}$ は整数 1、2、3、・・・16（最大値）でなければなりません。
tim_trgo	出力	内部トリガ出力。このトリガは他の内部ペリフェラル（DAC）をトリガできます。
tim_upd_it	出力	タイマ更新イベント割込み
tim_upd_dma	出力	タイマ更新 DMA リクエスト

40.3.3 TIM6/TIM7 クロック

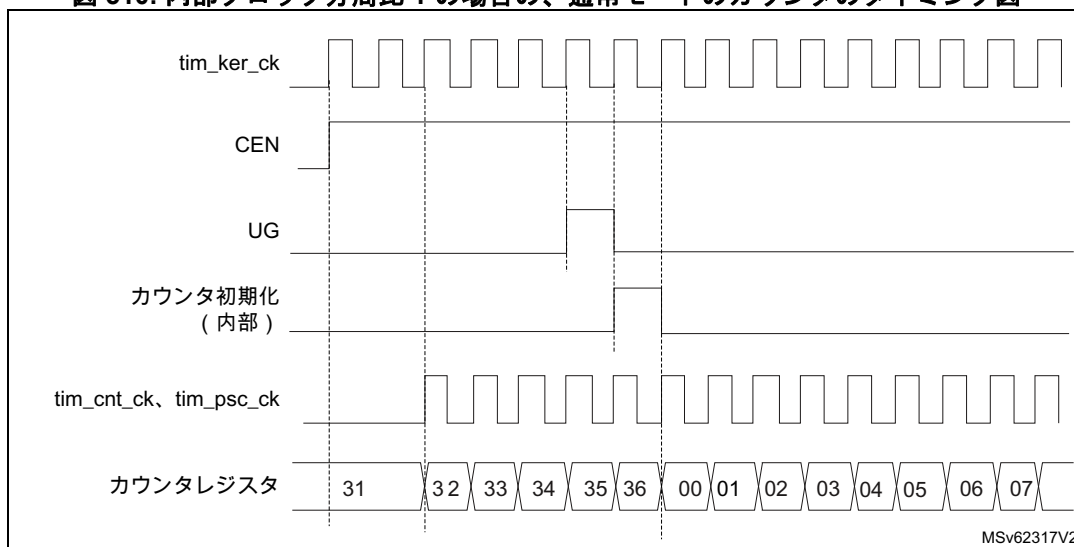
タイマバースインタフェースは `tim_pclk` APB クロックによってクロック供給されます。

カウンタクロック `tim_ker_ck` は `tim_pclk` 入力に接続されます。

TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットは実際の制御ビットであり、ソフトウェアによってのみ変更できます（ただし、自動的にクリア状態が保持される UG ビットを除く）。CEN ビットに 1 が書き込まれるとすぐに、プリスケアラには内部クロック `tim_ker_ck` が供給されます。

図 510 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

図 510. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



40.3.4 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きはプリロードレジスタへのアクセスとなります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント UEV ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、カウンタがオーバーフロー値に達すると、更新イベントが送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウントイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 `tim_cnt_ck` から供給されます。

実際のカウンタイネーブル信号 `tim_cnt_en` は、CEN ビットセットの 1 クロックサイクル後にセットされることにご注意ください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。TIMx_PSC 制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 511 と 図 512 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 511. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

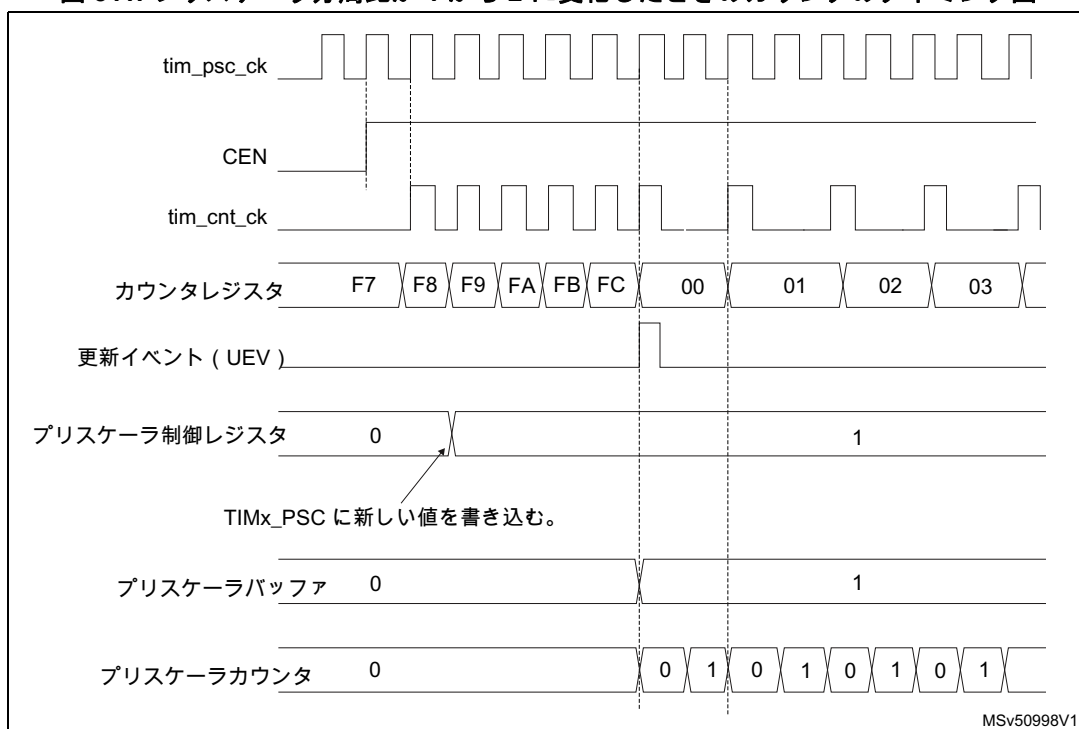
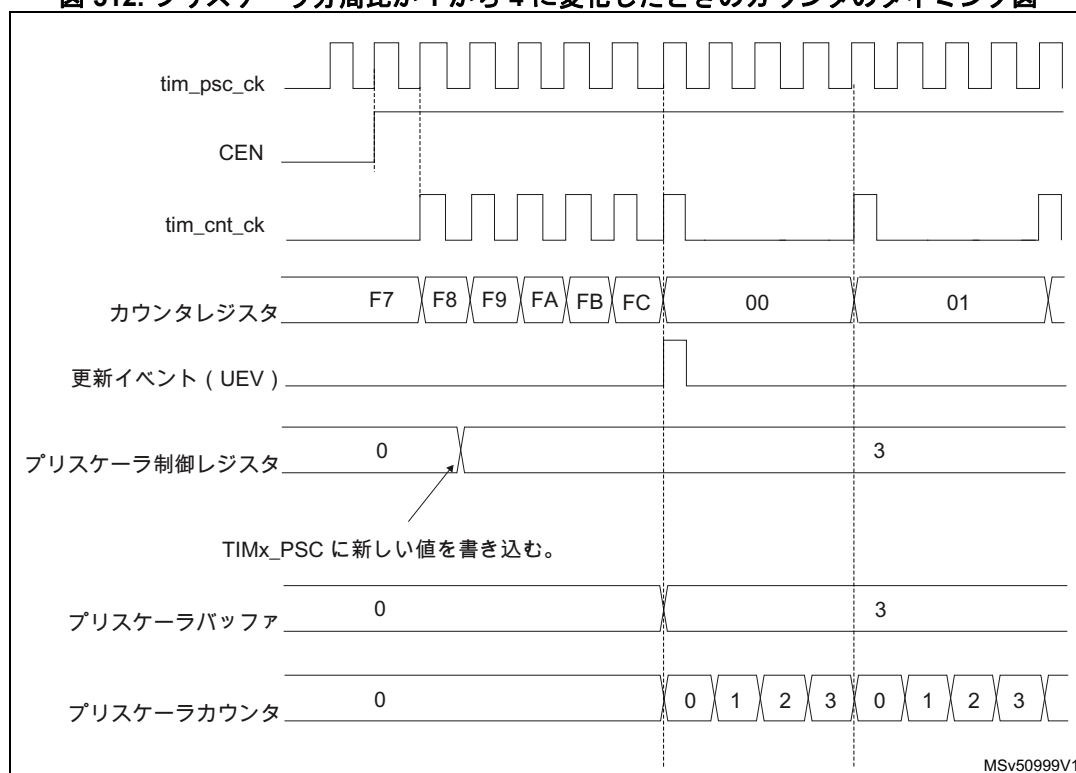


図 512. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



40.3.5 カウントモード

カウンタは、0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントした後、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、TIMx_EGR レジスタの UG ビットをセットすることによって (ソフトウェアで、または、スレーブモードコントローラを使用して) 生成できます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。このようにすると、UDIS ビットに 0 が書き込まれるまで更新イベントは発生しませんが、カウンタとプリスケアラカウンタは両方とも 0 からリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS (更新リクエスト選択) ビットがセットされている場合、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (割り込みや DMA リクエストは送信されません)。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットの設定に応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 513. 内部クロック分周比が 1 の場合のカウンタのタイミング図

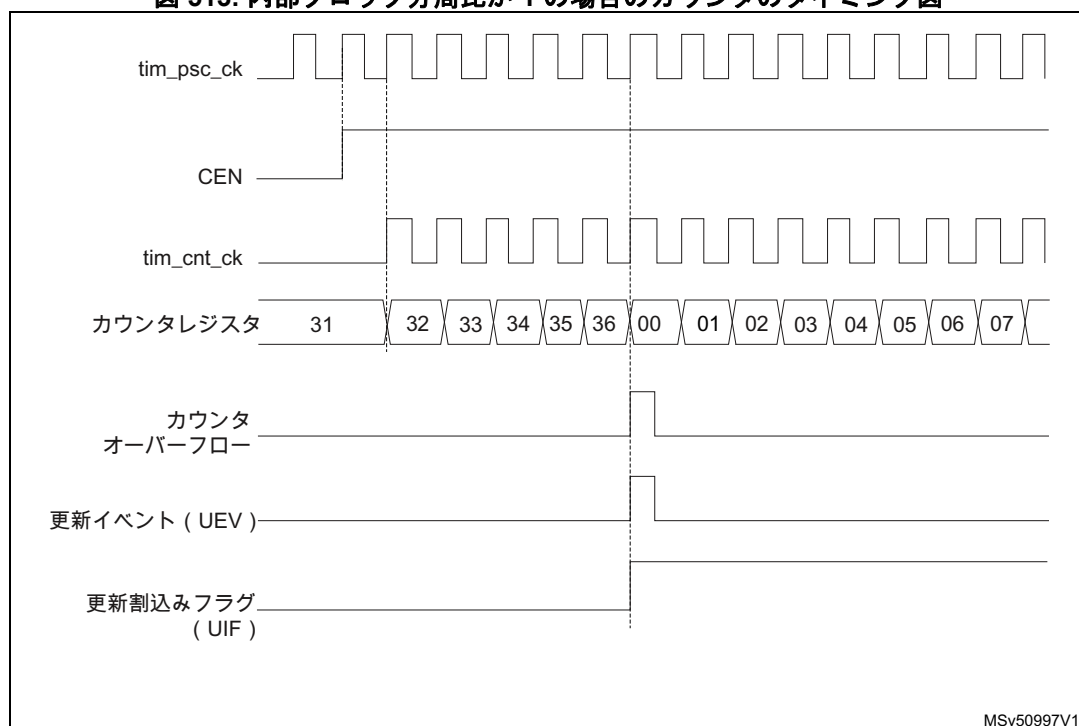


図 514. 内部クロック分周比が 2 の場合のカウンタのタイミング図

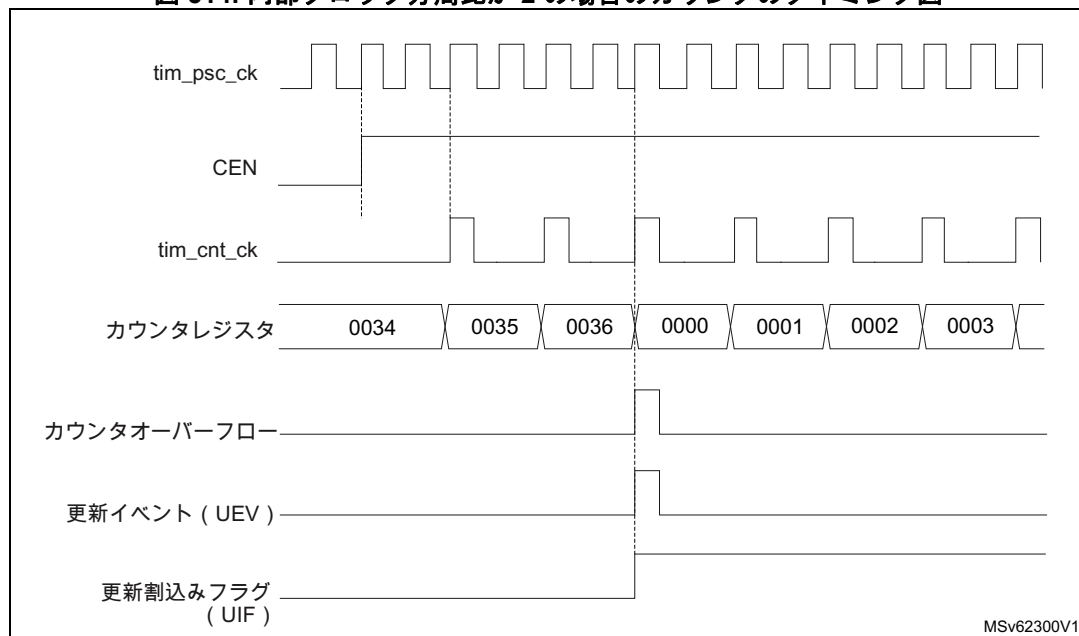


図 515. 内部クロック分周比が 4 の場合のカウンタのタイミング図

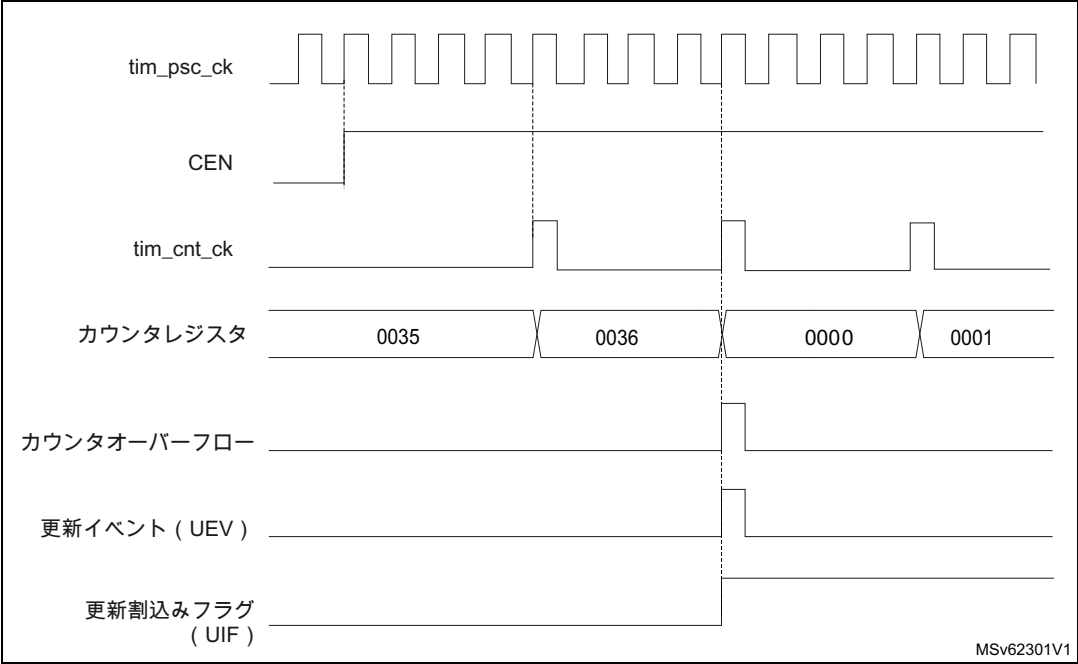


図 516. 内部クロック分周比が N の場合のカウンタのタイミング図

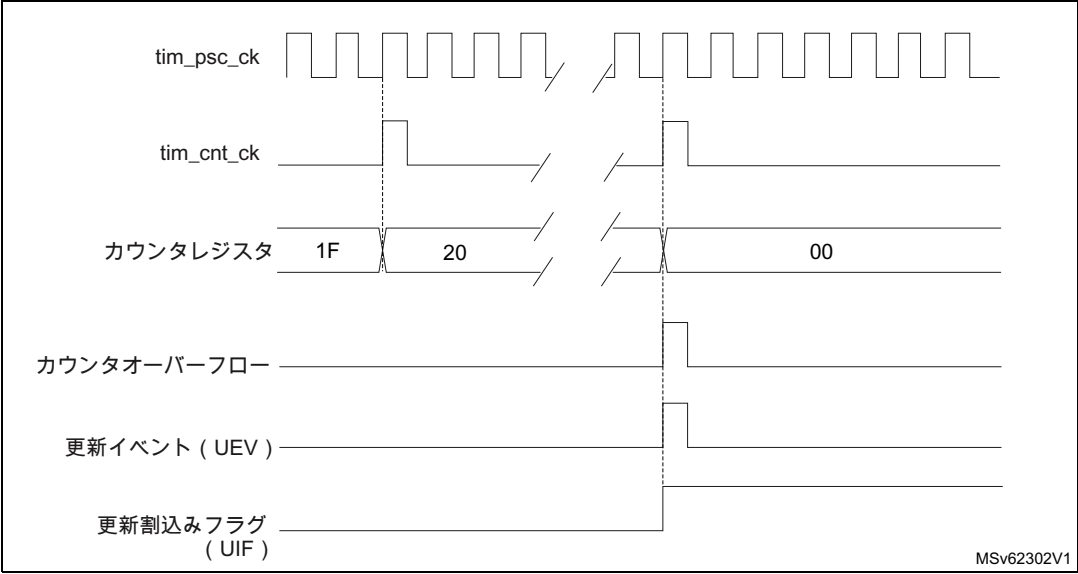


図 517. ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

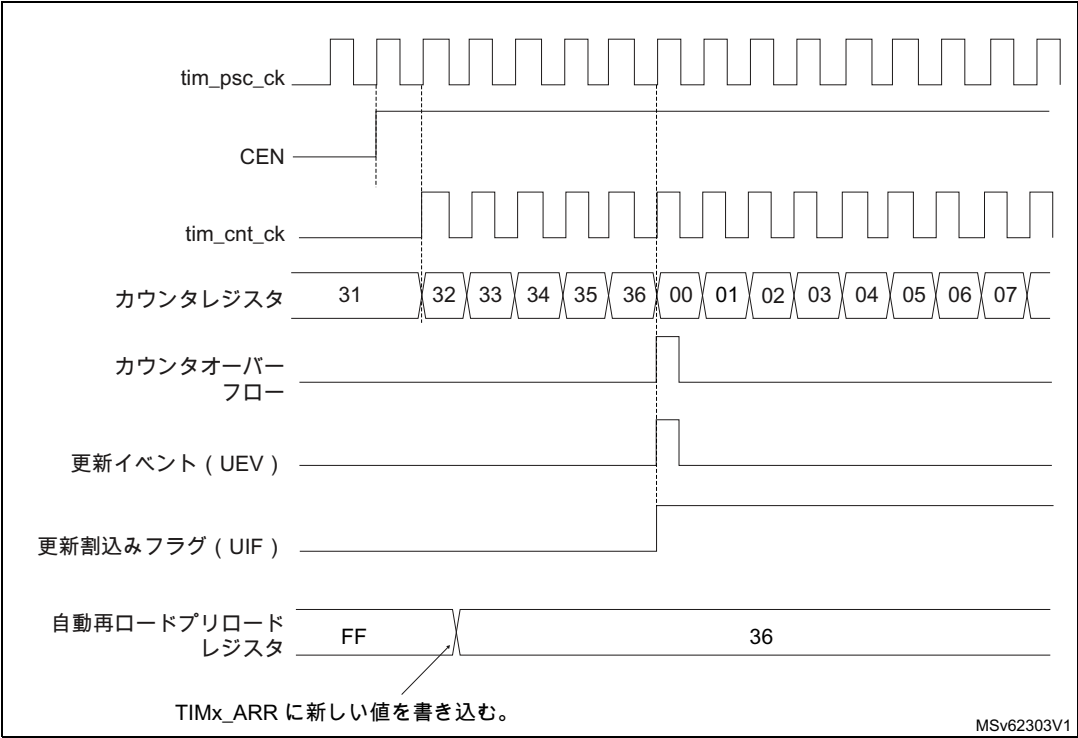
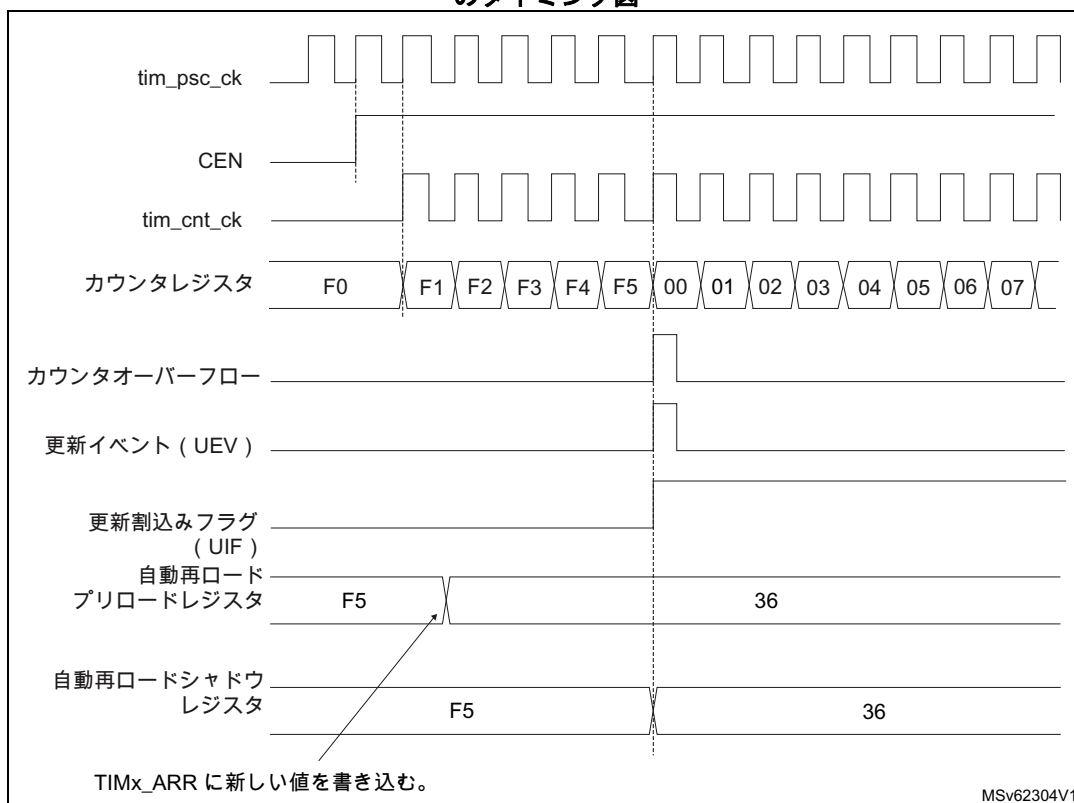


図 518. ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



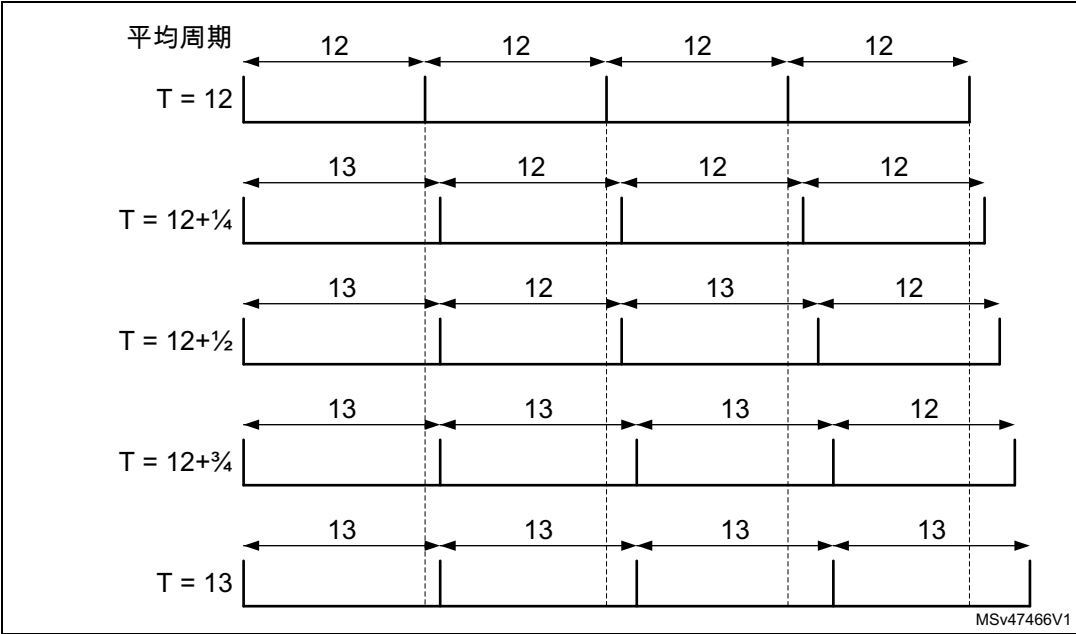
ディザリングモード

タイムベースの有効分解能は、TIMx_CR1 レジスタの DITHEN ビットを使用して、ディザリングモードを有効にすることで増加できます。これは、TIMx_ARR の動作に影響し、タイマがトリガとして使用される場合（通常は DAC の場合）に平均カウンタ周期を調整するのに役立ちます。

動作原理は、事前に定義されたパターンで、16 個の連続した カウント 周期にわたって実際の ARR 値をわずかに変更する（1 タイマクロック周期を追加する、またはしない）ことです。これにより、平均のカウント周期を考えると、16 倍の分解能の向上が可能になります。

下記の図 519 は、4 つの連続するカウント周期に適用されたディザリング原理を示します。

図 519. ディザリング原理

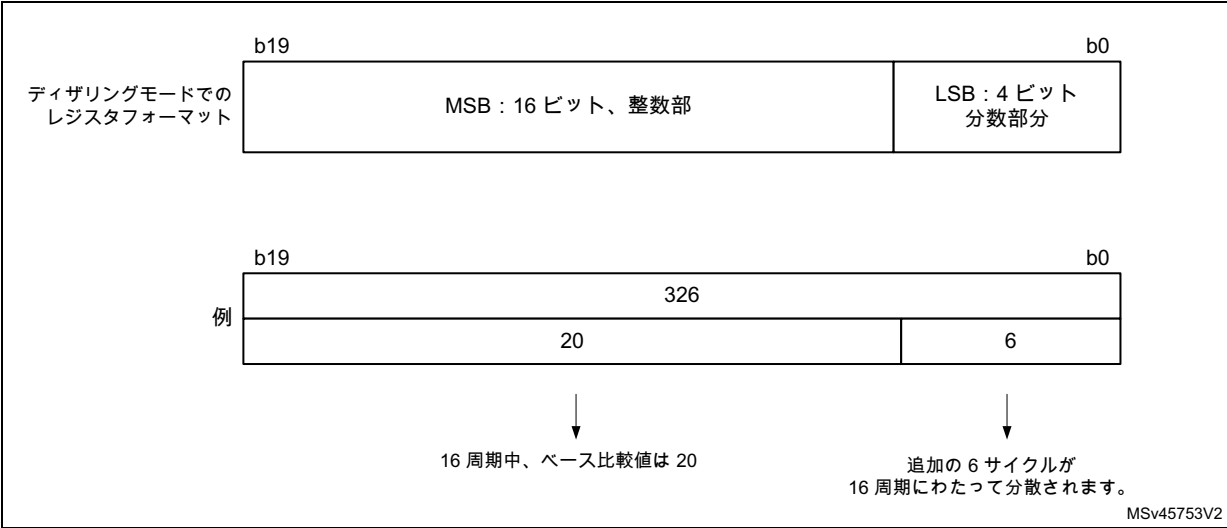


ディザリングモードが有効になっている場合、レジスタのコーディングは以下のように変更されます (たとえば、図 520 を参照)。

- 下位 4 ビットは、拡張分解能部分 (分数部分) 用です。
- 上位ビットは、ビット 19:4 に左シフトされ、ベース値用のコーディングになります。

注 : DITHEN ビットをリセットするときは、次の手順に従う必要があります。
 1を参照してください。CEN および ARPE ビットをリセットする必要があります。
 2を参照してください。DITHEN ビットをリセットする必要があります。
 3を参照してください。CEN ビットをセットすることができます (最終的に ARPE = 1 とともに)。

図 520. ディザリングモードでのデータフォーマットとレジスタのコーディング



最小周波数は次の計算式で与えられます。

$$\text{分解能} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

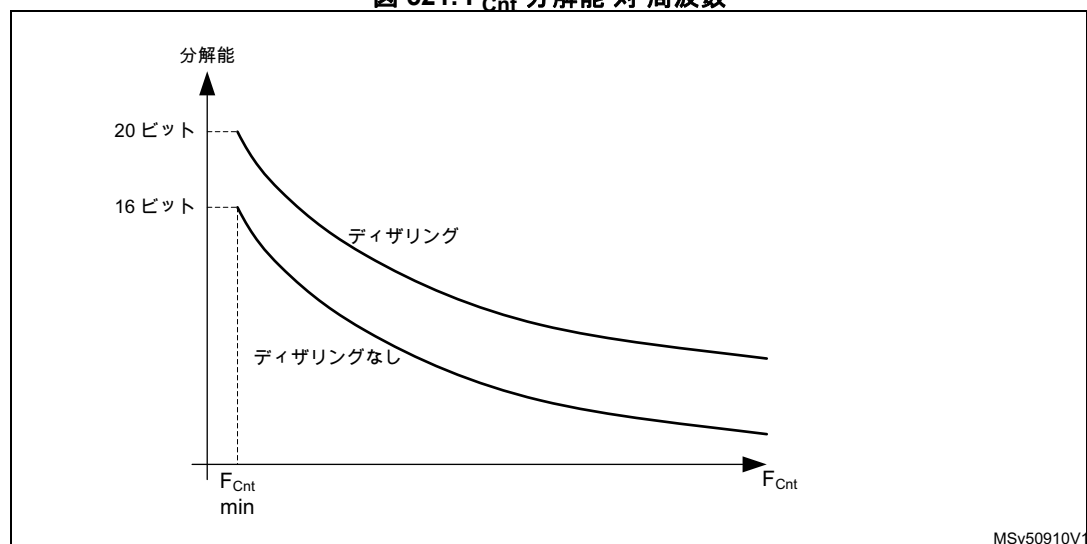
$$\text{ディザリングモード無効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

$$\text{ディザリングモード有効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

注: TIMx_ARR の最大値は、ディザリングモードでは 0xFFFEF に制限されています (整数部では 65534、ディザリング部分では 15 に対応します)。

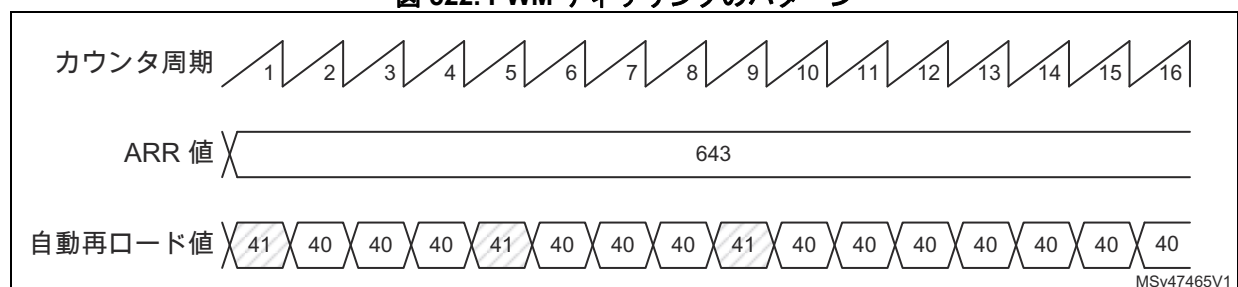
下記の図 521 に示すように、ディザリングモードは PWM 周波数に関係なく、PWM 分解能を増加するために使用されます。

図 521. F_{Cnt} 分解能 対 周波数



下記の図 522 で説明するように、周期の変更は、連続した 16 周期にわたって広がります。

図 522. PWM ディザリングのパターン



自動再ロード値と比較値の増分は、下記の表 416 に記載された特定のパターンに従って分布します。ディザリングシーケンスは、増分が可能な限り均等に分散され、全体のリップルが最小になるように行われます。

表 416. TIMx_ARR レジスタの変更のディザリングパターン

-	PWM 周期															
LSB 値	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

40.3.6 UIF ビットの再配置

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割込みフラグ (UIF) の連続コピーを強制します。これは、UIFCPY フラグによって信号伝達される潜在的なロールオーバー条件とカウンタ値を一度に読み取るために使用されます。特定のケースでは、バックグラウンドタスク（カウンタの読出し）と割込み（更新の割込み）との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

40.3.7 ADC トリガ

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイイベントを生成できます。

注：tim_trgo 信号を受信するスレーブペリフェラル（タイマ、ADC など）のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間はクロック周波数（プリスケアラ）を動作中に変更しないでください。

40.3.8 TIM6/TIM7 DMA リクエスト

表 417 に示すように、TIM6/TIM7 は単一の DMA リクエストを生成できます。

表 417. DMA リクエスト

DMA 項目（略称）	DMA リクエストあり	有効制御ビット
tim_upd_dma	更新	UDE

40.3.9 デバッグモード

マイクロコントローラがデバッグモードになると（Cortex-M33 コアは停止状態）、TIMx カウンタは、通常どおりに動作を続けるか、または停止できます。

デバッグモードでの動作は、デバッグサポート（DBG）モジュールでタイマごとに専用の設定ビットを使用してプログラムできます。

詳細については、デバッグサポート（DBG）のセクションを参照してください。

40.3.10 TIM6/TIM7 低消費電力モード

表 418. 低消費電力モードが TIM6/TIM7 に与える影響

モード	説明
SLEEP	影響はなく、ペリフェラルはアクティブです。割込みによって、デバイスは SLEEP モードから復帰できます。
STOP	タイマの動作は停止され、レジスタの内容は保持されます。割込みは生成できません。
STANDBY	タイマはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

40.3.11 TIM6/TIM7 割込み

表 419 に示すように、TIM6/TIM7 は単一の割込みを生成できます。

表 419. 割込みリクエスト

項目 （割込みの 略称）	割込みイベント	イベント フラグ	有効制御ビット	割込みの クリア方法	SLEEP モードの 終了	STOP および STANDBY モードの 終了
TIM6 TIM7	更新	UIF	UIE	UIF に 0 を 書き込む	可	不可

40.4 TIM6/TIM7 レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

40.4.1 TIMx 制御レジスタ 1 (TIMx_CR1) (x = 6~7)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
			rw	rw				rw				rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。
0 : 次のイベントのいずれかが更新割込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

– カウンタオーバーフロー/アンダーフロー

– UG ビットのセット

– スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

パッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC) はそれぞれの値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケールは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

40.4.2 TIMx 制御レジスタ 2 (TIMx_CR2) (x = 6~7)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **MMS[2:0]** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (TRGO) を選択するために使用します。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (tim_trgo) として使用されます。

001 : **イネーブル** - カウンタイネーブル信号 tim_cnt_en がトリガ出力 (tim_trgo) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、CEN 制御ビットが書き込まれると生成されます。

010 : **更新** - 更新イベントがトリガ出力 (tim_trgo) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

注 : スレーブタイマまたは tim_trgo を受信するペリフェラルのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

40.4.3 TIMx DMA／割込み有効レジスタ (TIMx_DIER) (x = 6~7)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIE** : 更新割込みイネーブル

0 : 更新割込みは無効です。

1 : 更新割込みは有効です。

40.4.4 TIMx ステータスレジスタ（TIMx_SR）（x = 6～7）

アドレスオフセット：0x10

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIF**：更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0：更新は発生していません。

1：更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

–TIMx_CR1 レジスタの UDIS = 0 の場合、カウンタのオーバーフローが発生したとき。

–TIMx_CR1 レジスタの URS = 0 かつ UDIS = 0 の場合に、TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで CNT が再初期化されたとき。

40.4.5 TIMx イベント生成レジスタ（TIMx_EGR）（x = 6～7）

アドレスオフセット：0x14

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UG**：更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0：影響はありません。

1：タイマカウンタを再初期化して、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。

40.4.6 TIMx カウンタ (TIMx_CNT) (x = 6~7)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー。TIMx_CR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分は得られません。

40.4.7 TIMx プリスケアラ (TIMx_PSC) (x = 6~7)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 $f_{tim_cnt_ck}$ は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC には、更新イベントのたびにアクティブプリスケアラレジスタにロードされる値が含まれます。

(更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、カウンタがクリアされる場合も含まれます)。

40.4.8 TIMx 自動再ロードレジスタ（TIMx_ARR）（x = 6～7）

アドレスオフセット：0x2C

リセット値：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]**：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 40.3.4: 1664 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード（DITHEN = 0）

このレジスタは、ARR[15:0] に自動再ロード値を保持します。ARR[19:16] ビットは予約されています。

ディザリングモード（DITHEN = 1）

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

40.4.9 TIMx レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 420. TIMx レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DITHEN	UIFREMAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	リセット値																				0	0				0				0	0	0	0	
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS [2:0]			Res.	Res.	Res.	Res.		
	リセット値																									0	0	0						
0x08	予約済み																																	
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UE	
	リセット値																								0							0		
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF	
	リセット値																															0		
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG	
	リセット値																															0		
0x18-0x20	予約済み																																	
0x24	TIMx_CNT	UFCPY または Res.															CNT[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																				
	リセット値													0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

41 汎用タイマ (TIM12/TIM13/TIM14)

41.1 TIM12/TIM13/TIM14 の概要

TIM12/TIM13/TIM14 汎用タイマは、プログラム可能なプリスケアラによって駆動される 16 ビットの自動再ロードカウンタで構成されています。

これらのカウンタは、入力信号パルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM）など様々な目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

TIM12/TIM13/TIM14 タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 41.4.20: タイマ同期 \(TIM12 のみ\)](#) に示すように、相互に同期させることができます。

41.2 TIM12 の主な特徴

TIM12 汎用タイマ の機能は以下のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ（動作中に変更可能）。
- 次の機能を持つ、最大 2 つの独立チャンネル：
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモード）
 - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み生成。
 - 更新：カウンタオーバーフロー、カウンタ初期化（ソフトウェアまたは内部トリガによる）
 - トリガイベント（内部トリガによるカウンタの開始、停止、初期化、またはカウント）
 - 入力キャプチャ
 - 出力比較

41.3 TIM13/TIM14 の主な特徴

汎用タイマ TIM13/TIM14 の機能は以下のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ（動作中に変更可能）。
- 次の機能を持つ独立チャネル：
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモード）
 - ワンパルスモード出力
- 以下のイベント時の割込み生成。
 - 更新：カウンタオーバーフロー、カウンタの初期化（ソフトウェアによる）
 - 入力キャプチャ
 - 出力比較

41.4 TIM12/TIM13/TIM14機能詳細

41.4.1 ブロック図

図 523. 汎用タイマのブロック図 (TIM12)

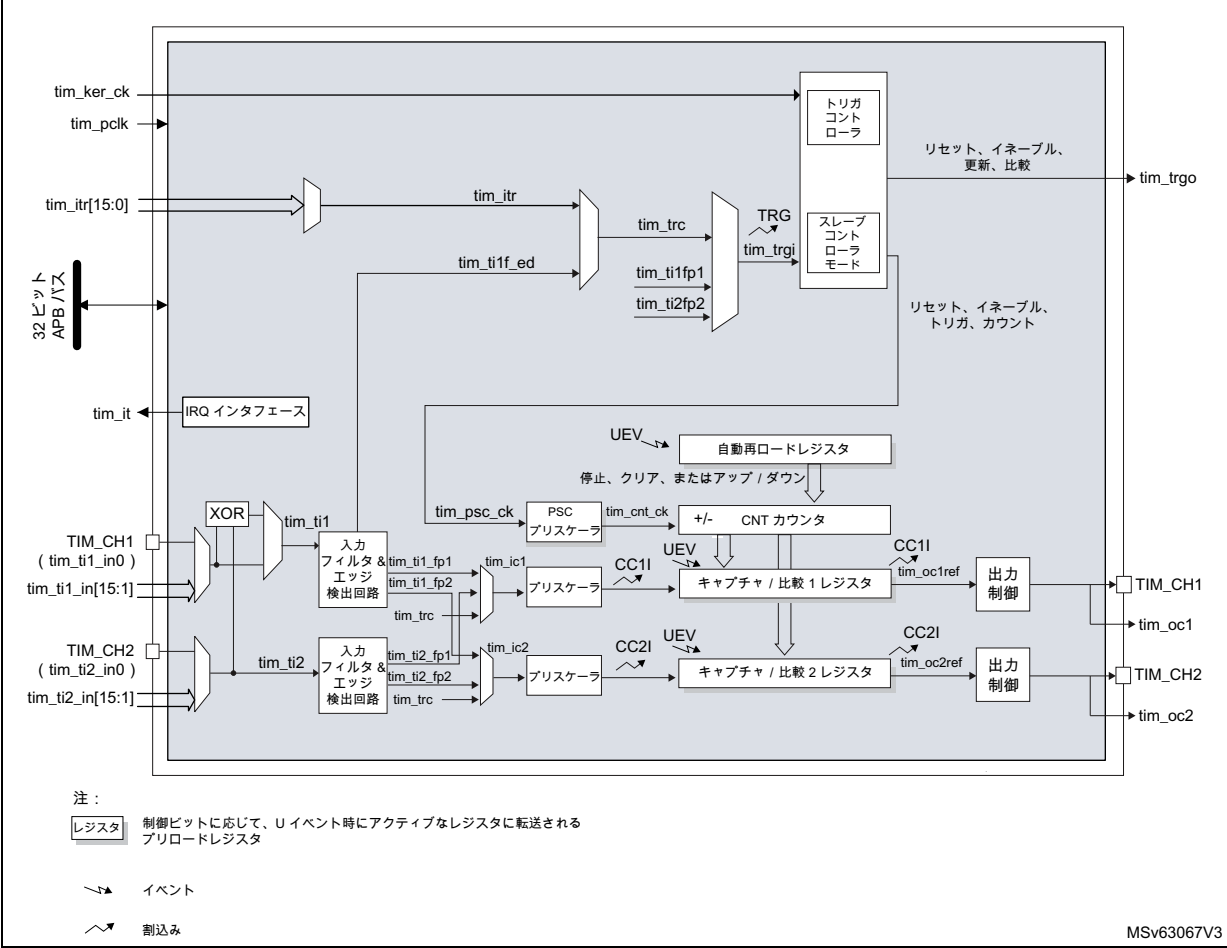
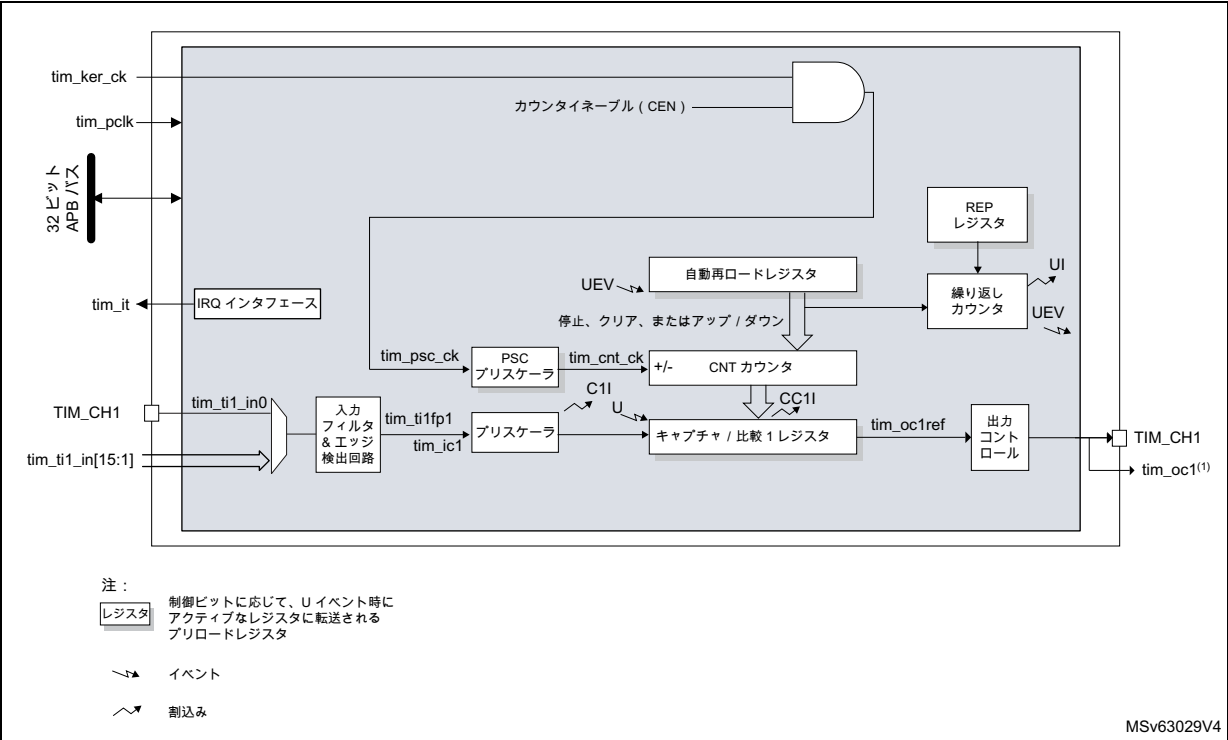


図 524. 汎用タイマのブロック図 (TIM13/TIM14)



1. この信号は一部のスレーブタイマのトリガとして使用できます (次のセクションの内部トリガ接続表を参照)。詳細については、[セクション 41.4.21 : タイマ出力を他のタイマのトリガとして使用 \(TIM13/TIM14 のみ\)](#) を参照してください。

41.4.2 TIM12/TIM13/TIM14 ピンおよび内部信号

このセクションの表 421 および表 422 は、TIM の入力および出力の一覧です。

表 421. TIM の入出力ピン

ピン名	信号タイプ	説明
TIM_CH1 TIM_CH2 ⁽¹⁾	入力／出力	タイマ多目的チャネル。 各チャネルは、キャプチャ、比較、または PWM で使用できます。 TIM_CH1 と TIM_CH2 は、外部クロック (tim_ker_ck クロックの 1/4 未満)、外部トリガの入力としても使用できます。

1. TIM12 でのみ使用できます。

表 422. TIM 内部入力／出力信号

内部信号名	信号タイプ	説明
tim_ti1_in [15:0] tim_ti2_in [15:0] ⁽¹⁾	入力	内部タイマ入力バス。これらの入力は、キャプチャ用に、または外部クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_itr[15:0] ⁽¹⁾	入力	内部トリガ入力バス。これらの入力は、スレーブモードコントローラ用に、または入力クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_oc1 tim_oc2 ⁽¹⁾	出力	内部タイマ出力。他のタイマや ADC のトリガに使用できます。
tim_trgo ⁽¹⁾	出力	内部トリガ出力。このトリガは他のオンチップペリフェラルをトリガできます。



表 422. TIM 内部入力／出力信号 (続き)

内部信号名	信号タイプ	説明
tim_pclk	入力	タイマ APB クロック
tim_ker_ck	入力	タイマカーネルクロック。このクロックは、tim_pclk (同じソースから導出) と同期する必要があります。クロック比 tim_ker_ck/tim_pclk は整数 1、2、3、・・・16 (最大値) でなければなりません。
tim_it	出力	キャプチャ／比較、更新、ブレークトリガ、および転流リクエストを取りまとめるグローバルタイマ割込み。

1. TIM12 でのみ使用できます。

表 423 および表 424 に、tim_ti[2:1] 入力マルチプレクサに接続されるソースを列挙します。

表 423. tim_ti1 入力マルチプレクサへの相互接続

tim_ti1 入力	ソース		
	TIM12	TIM13	TIM14
tim_ti1_in0	TIM12_CH1	TIM13_CH1	TIM14_CH1
tim_ti1_in1	予約済み	予約済み	
tim_ti1_in2			
tim_ti1_in3			
tim_ti1_in4			
tim_ti1_in5	HSI/1024		
	CSI/128		
tim_ti1_in [15:6]	予約済み		

表 424. tim_ti2 入力マルチプレクサへの相互接続

tim_ti2 入力	ソース
	TIM12
tim_ti2_in0	TIM12_CH2
tim_ti2_in [15:1]	予約済み

表 425 に、tim_itr 入力マルチプレクサに接続される内部ソースを列挙します。

表 425. TIMx 内部トリガ接続

TIMx	TIM12
tim_itr0	tim1_trgo
tim_itr1	tim2_trgo
tim_itr2	tim3_trgo
tim_itr3	tim4_trgo
tim_itr4	tim5_trgo
tim_itr5	tim8_trgo
tim_itr6	予約済み

表 425. TIMx 内部トリガ接続 (続き)

TIMx	TIM12
tim_itr7	tim13_oc1
tim_itr8	tim14_oc1
tim_itr9	tim15_trgo
tim_itr10	tim16_oc1
tim_itr11	tim17_oc1
tim_itr[15:12]	予約済み

41.4.3 タイムベースユニット

タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定ごとに詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ有効な、プリスケアラ出力 tim_cnt_ck から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 525 と 図 526 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 525. プリスケール分周比が 1 から 2 に変化したときのカウンタのタイミング図

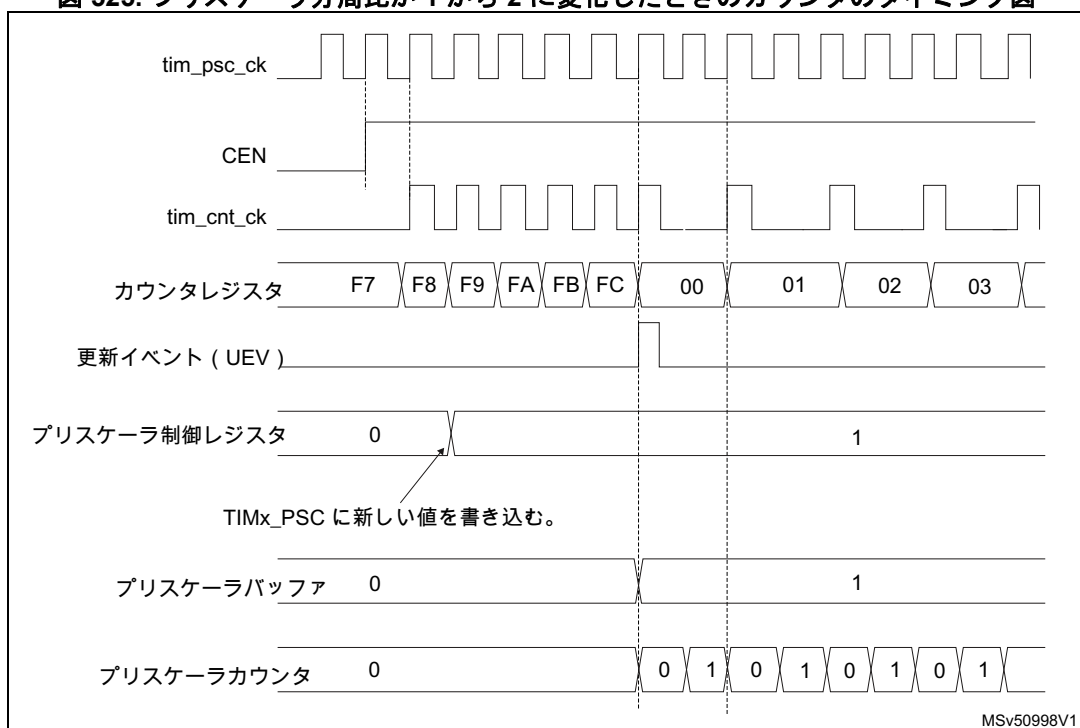
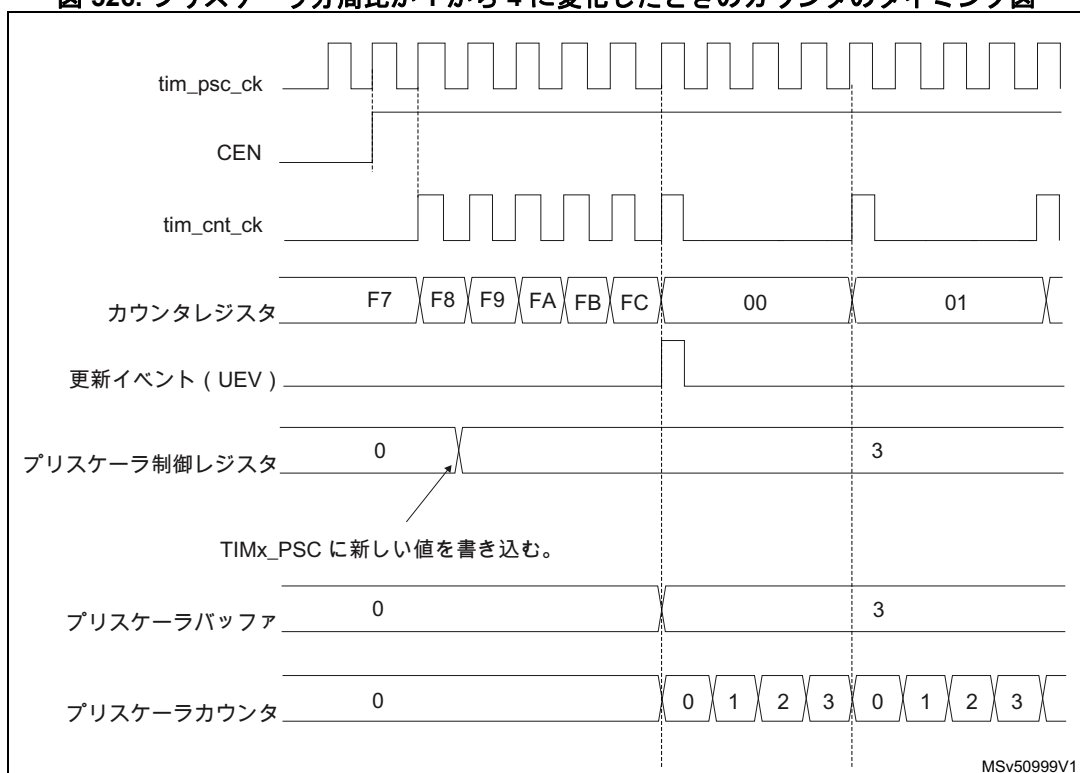


図 526. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



41.4.4 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

(ソフトウェアによって、または TIM12 のスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 527. 内部クロック分周比が 1 の場合のカウンタのタイミング図

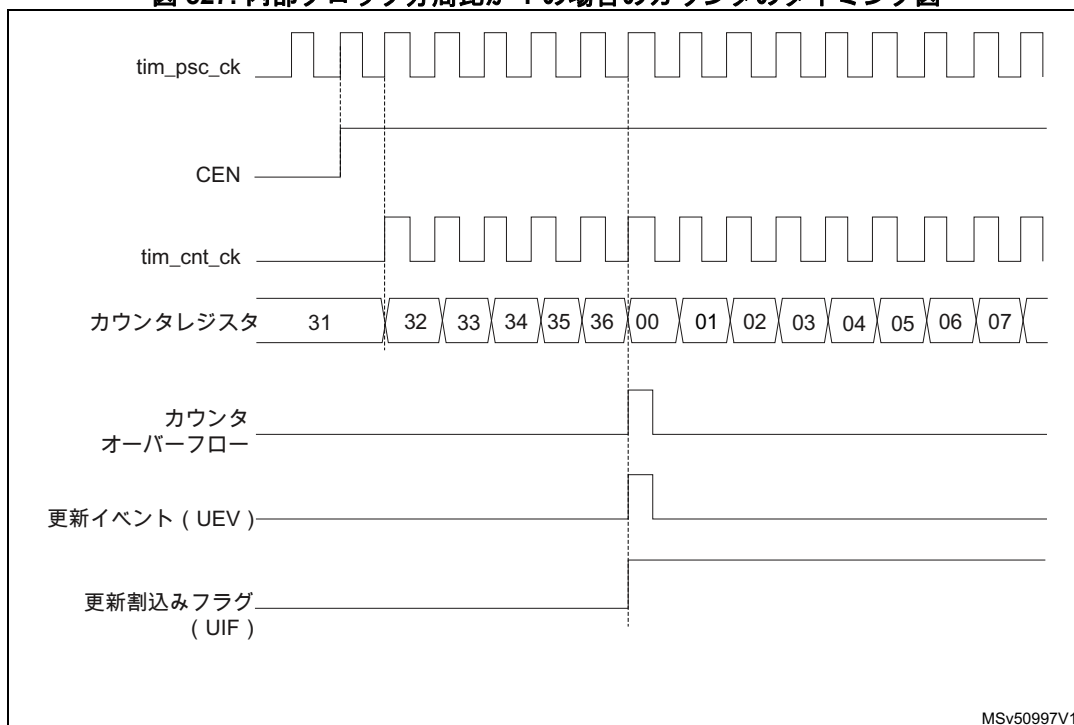


図 528. 内部クロック分周比が 2 の場合のカウンタのタイミング図

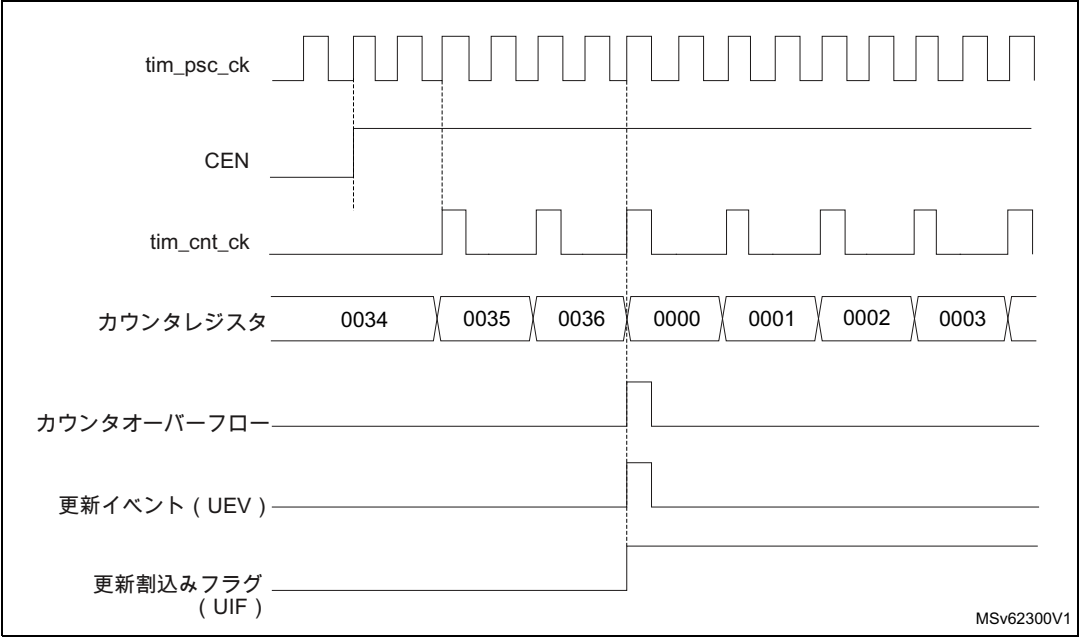


図 529. 内部クロック分周比が 4 の場合のカウンタのタイミング図

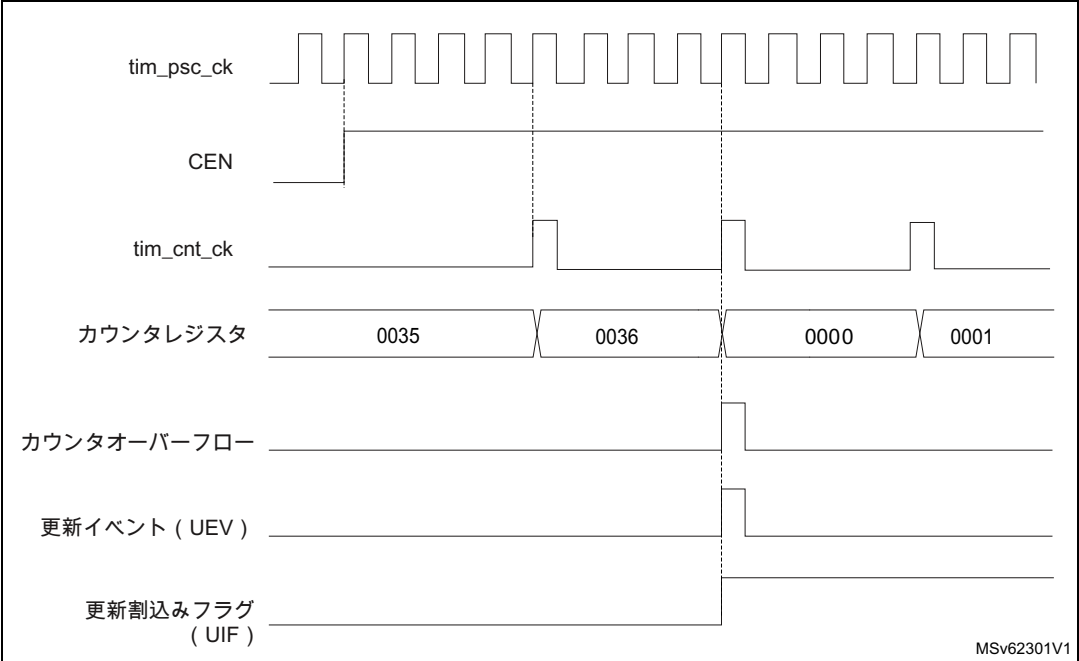


図 530. 内部クロック分周比が N の場合のカウンタのタイミング図

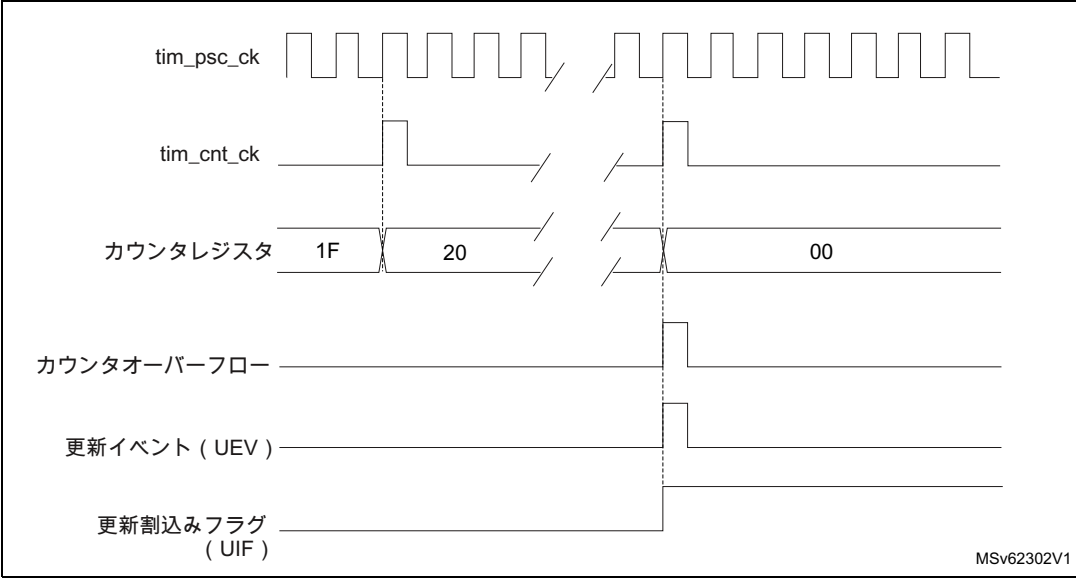


図 531. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

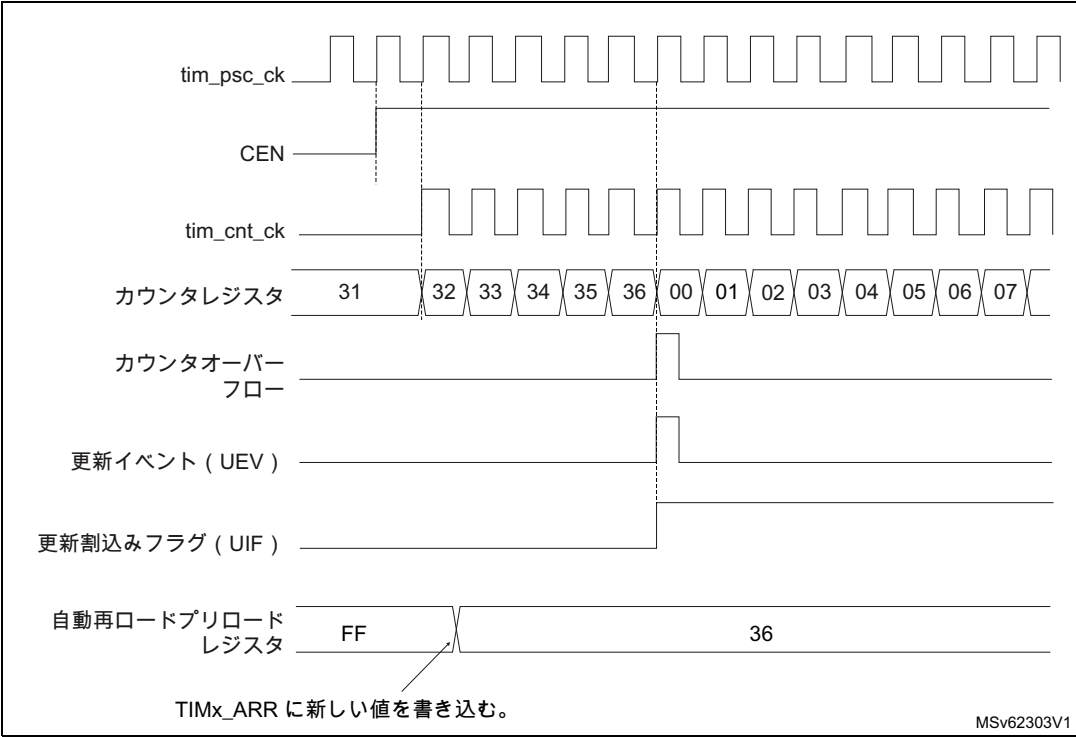
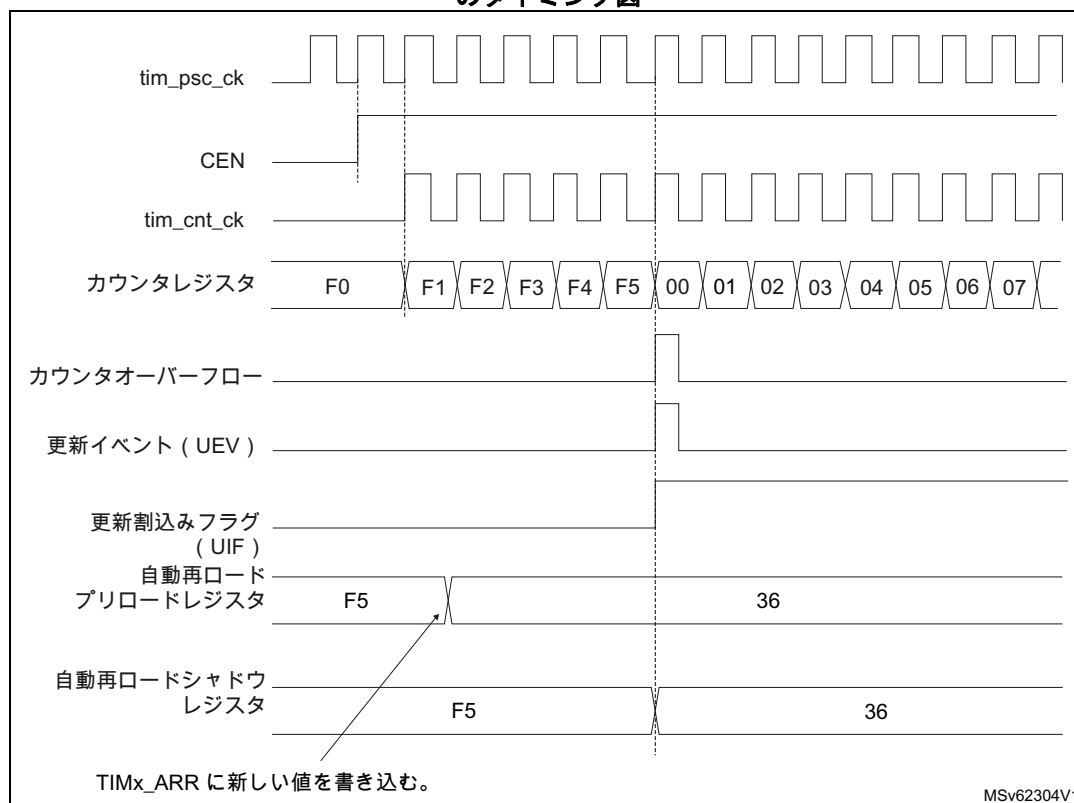


図 532. ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



41.4.5 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

- 内部クロック (tim_ker_ck)
- 外部クロックモード 1 (TIM12 用) : 外部入力ピン (使用可能な場合、tim_ti1 または tim_ti2)
- 内部トリガ入力 (tim_itrx) (TIM12 用) : 他のタイマからトリガ出力を接続。たとえば、他のタイマを TIM12 のプリスケアラとして設定できます。詳細については、[セクション：タイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

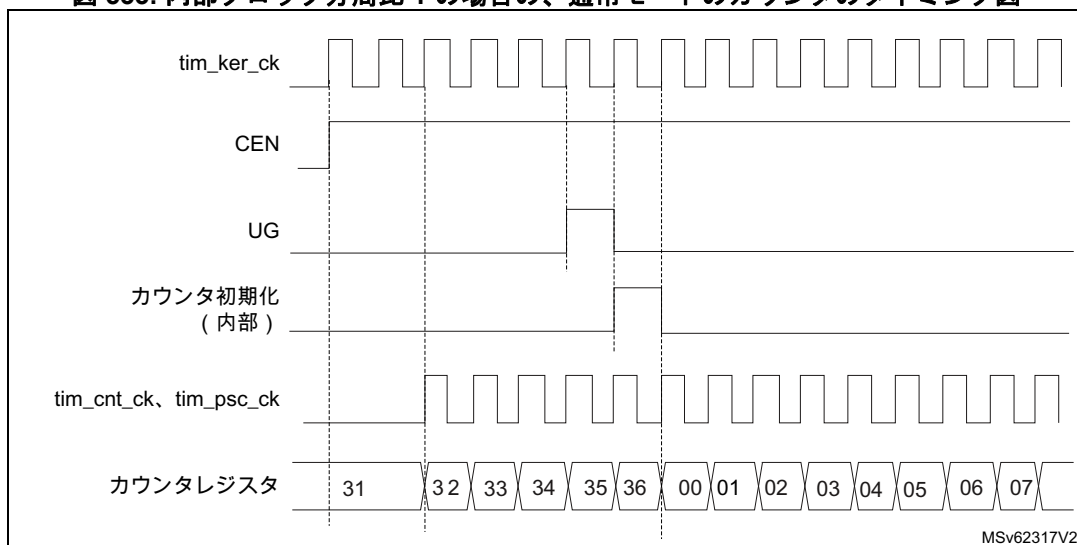
内部クロックソース (tim_ker_ck)

内部クロックソースは、TIM13/TIM14 のデフォルトクロックソースです。

TIM12 では、スレーブモードコントローラが無効の場合 (SMS = 000)、内部クロックソースが選択されます。その際、TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットが制御ビットとして使用され、ソフトウェアからのみ変更できます (クリアされたままの UG ビットは除く)。CEN ビットに 1 がプログラムされるとすぐに、プリスケアラには内部クロック tim_ker_ck が供給されます。

図 533 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

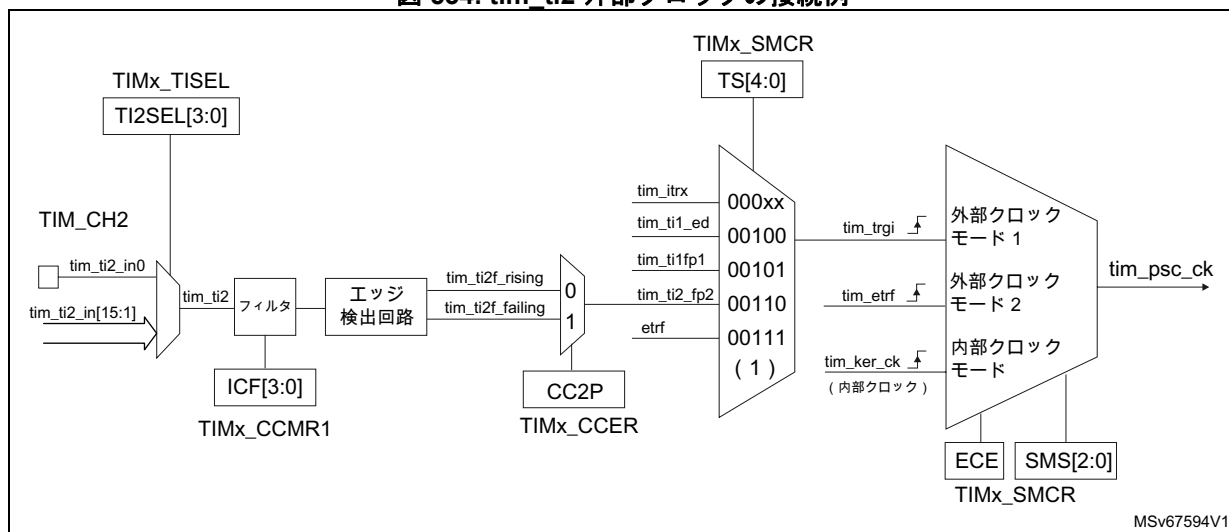
図 533. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS = 111 のときに選択されます。カウンタは、選択された入力の立上がりまたは立下がりエッジでカウントすることができます。

図 534. tim_ti2 外部クロックの接続例



たとえば、tim_ti2 入力の立上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

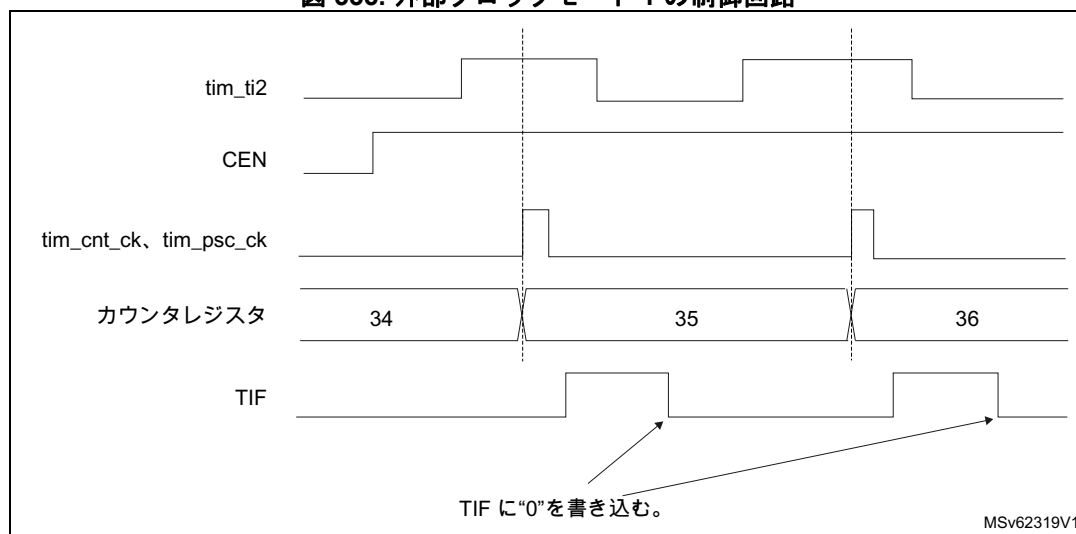
1. TIMx_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:0] ソース（内部または外部）を選択します。
2. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が tim_ti2 入力の立上がりエッジを検出するように設定します。
3. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F = 0000 にしておきます）。
4. TIMx_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込むことによって、立上がりエッジ極性を選択します。
5. TIMx_SMCR レジスタの SMS ビットに“111”を書き込むことによって、タイマを外部クロックモード 1 に設定します。
6. TIMx_SMCR レジスタの TS ビットに“110”を書き込むことによって、トリガ入力ソースとして tim_ti2 を選択します。
7. TIMx_CR1 レジスタの CEN ビットに“1”を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケアラはトリガには使用されないので、設定する必要はありません。

tim_ti2 の立上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際のカウンタクロックまでの間には、tim_ti2 入力の再同期回路による遅延があります。

図 535. 外部クロックモード 1 の制御回路



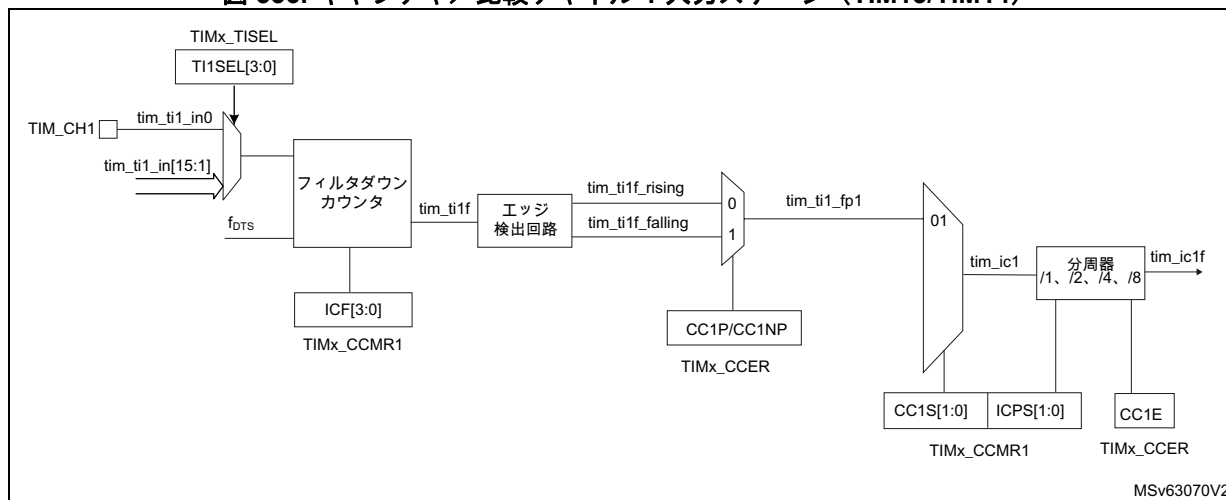
41.4.6 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクス、プリスケアラ）、および出力カステージ（コンパレータと出力制御）から構成されています。

図 536、図 537、図 538、および図 539 に、キャプチャ／比較チャネルの概要を示します。

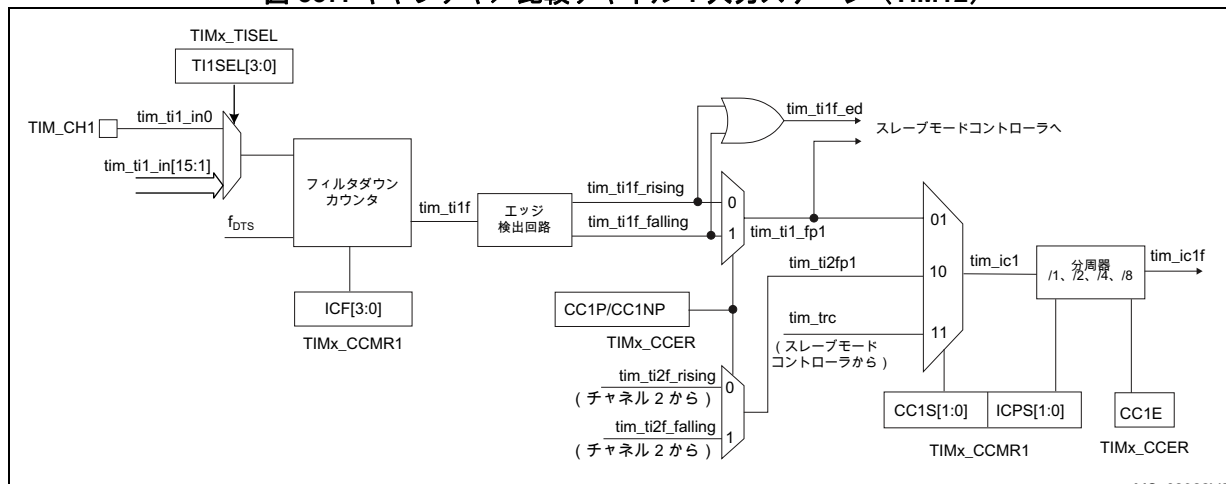
入力カステージは、対応する tim_tix 入力をサンプリングして、フィルタリングを行った信号 tim_tixf を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (tim_tixfpy) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 536. キャプチャ／比較チャネル 1 入力カステージ (TIM13/TIM14)



MSv63070V2

図 537. キャプチャ／比較チャネル 1 入力カステージ (TIM12)



MS-63333V4

出力カステージは、その後、基準として使用される中間波形 tim_ocxref (アクティブハイ) を生成します。信号の極性は最終出力に影響を与えます。

図 538. キャプチャ／比較チャネル 1 メイン回路

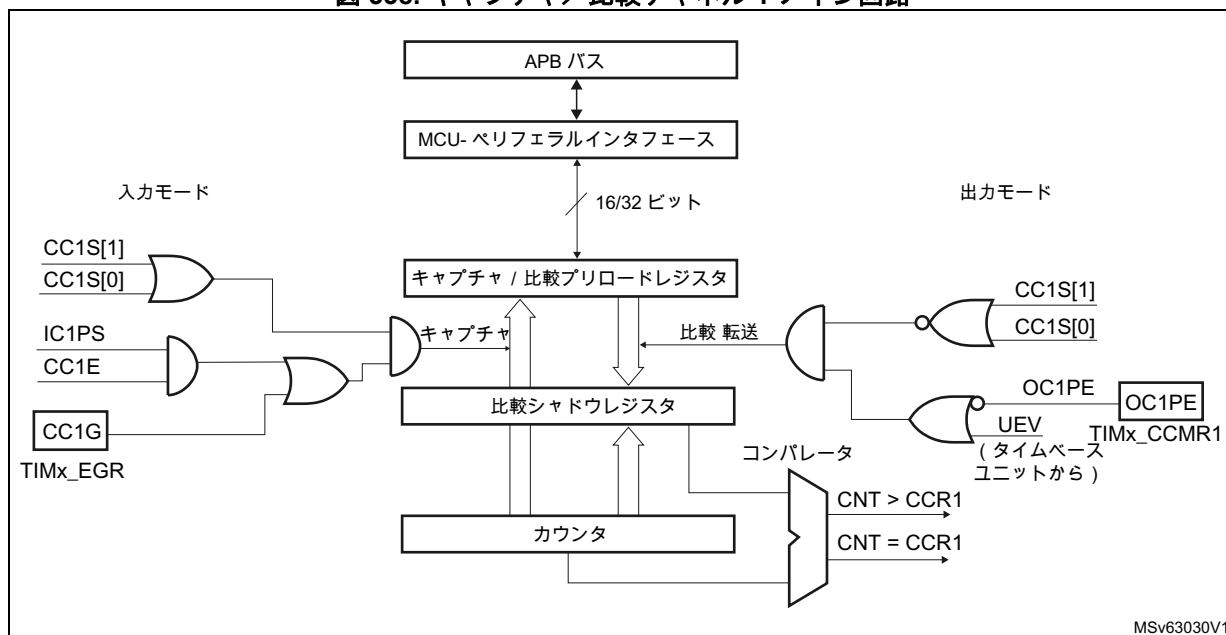
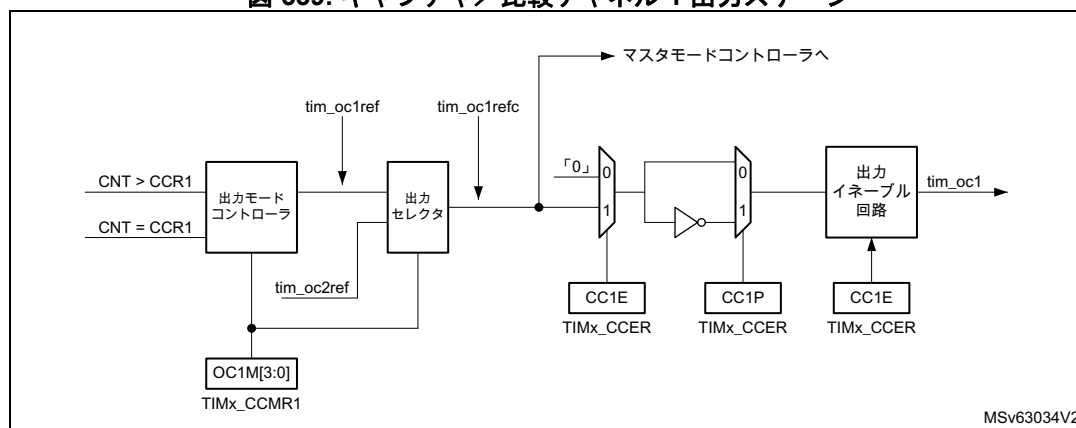


図 539. キャプチャ／比較チャネル 1 出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書込みおよび読出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

41.4.7 入力キャプチャモード

入力キャプチャモードでは、対応する `tim_icx` 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (`TIMx_CCRx`) が使用されます。キャプチャが発生すると、対応する `CCxIF` フラグ (`TIMx_SR` レジスタ) がセットされ、割り込みリクエストを送信できます (有効な場合)。 `CCxIF` フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ `CCxOF` (`TIMx_SR` レジスタ) がセットされます。 `CCxIF` フラグは、ソフトウェアで "0" を書き込むことによって、または、 `TIMx_CCRx` レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。 `CCxOF` は、0 を書き込むとクリアされます。

次の例は、 `tim_ti1` 入力 が立ち上がったときに、カウンタの値を `TIMx_CCR1` にキャプチャする方法を示します。このためには、次の手順を使用します。

1. `TIMx_TISEL` レジスタの `TI1SEL[3:0]` ビットで、適切な `tim_ti1_in[15:0]` ソース (内部または外部) を選択します。
2. アクティブ入力を選択します。 `TIMx_CCR1` は `tim_ti1` 入力にリンクされていないので、 `TIMx_CCMR1` レジスタの `CC1S` ビットに "01" を書き込みます。 `CC1S` の値が "00" から変化するとすぐに、チャンネルは入力モードに設定され、 `TIMx_CCR1` レジスタは読出し専用になります。
3. タイマに接続されている信号に関して適切な入力フィルタ時間を設定します (入力 が `tim_tix` 入力のうちの 1 つである場合、 `TIMx_CCMRx` レジスタの `ICxF` ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、 `tim_ti1` の遷移を検証できます (周波数 f_{DTS} でサンプリング)。次に、 `TIMx_CCMR1` レジスタの `IC1F` ビットに "0011" を書き込みます。
4. `tim_ti1` チャンネルのアクティブ遷移のエッジを選択します。このためには、 `TIMx_CCER` レジスタの `CC1P` ビットと `CC1NP` ビットに "00" を設定します (この場合、立上がりエッジの選択)。
5. 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (`TIMx_CCMR1` レジスタの `IC1PS` ビットに "00" を書き込む)。
6. `TIMx_CCER` レジスタの `CC1E` ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
7. 必要に応じて、 `TIMx_DIER` レジスタの `CC1IE` ビットをセットすることによって、関連する割り込みリクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が `TIMx_CCR1` レジスタに格納されます。
- `CC1IF` フラグがセットされます (割り込みフラグ)。 `CC1OF` ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- `CC1IE` ビットに応じて、割り込みが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これは、フラグ読出し後、データ読出し前に発生するかもしれないオーバキャプチャの見落としを避けるためです。

注： `TIMx_EGR` レジスタの対応する `CCxG` ビットをセットすることで、IC 割り込みリクエストをソフトウェアで発生させることができます。

41.4.8 PWM 入力モード (TIM12 のみ)

このモードでは、単一の `tim_tix` 入力に接続された PWM 信号の周期とデューティサイクルの両方を以下のように測定できます。

- `TIMx_CCR1` レジスタは周期値 (2 つの連続した立上がりエッジ間の間隔) を保持します。
- `TIMx_CCR2` レジスタはパルス幅 (2 つの連続した立上がりと立下がりエッジ間の間隔) を保持します。

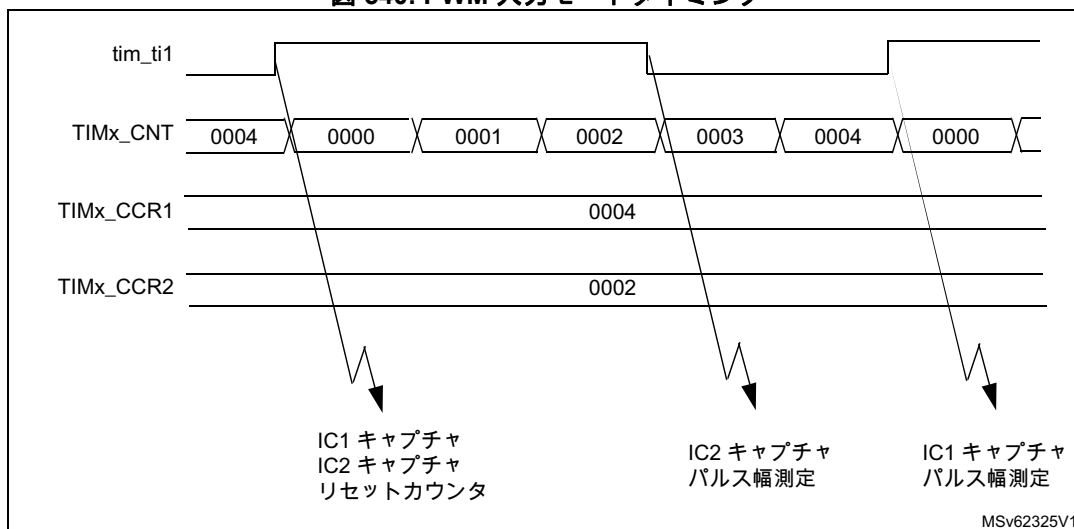
このモードは、入力キャプチャモードの特殊ケースです。設定手順は似ていますが、以下の違いがあります。

- 2 つの `tim_icx` 信号が同じ `tim_tix` 入力にマッピングされます。
- この 2 つの `tim_icx` 信号は、逆の極性のエッジでアクティブです。
- 2 つの `tim_tixfpy` 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、`tim_ti1` に適用された PWM の周期 (`TIMx_CCR1` レジスタ) とデューティサイクル (`TIMx_CCR2` レジスタ) を測定できます (手順は、`tim_ker_ck` 周波数とプリスケール値によって、若干異なることがあります)。

1. `TIMx_TISEL` レジスタの `TI1SEL[3:0]` ビットで、適切な `tim_ti1_in[15:0]` ソース (内部または外部) を選択します。
2. `TIMx_CCMR1` レジスタの `CC1S` ビットに“01”を書き込むことによって (`tim_ti1` を選択)、`TIMx_CCR1` のアクティブ入力を選択します。
3. `CC1P` ビットと `CC1NP` ビットに“00”を書き込むことによって (立上がりエッジでアクティブ)、`tim_ti1fp1` のアクティブな極性を選択します (`TIMx_CCR1` のキャプチャとカウンタクリアの両方に使用)。
4. `TIMx_CCMR1` レジスタの `CC2S` ビットに“10”を書き込むことによって (`tim_ti1` を選択)、`TIMx_CCR2` のアクティブ入力を選択します。
5. `CC2P` ビットと `CC2NP` ビットに“10”を書き込むことによって (立下がりエッジでアクティブ)、`tim_ti1fp2` のアクティブ極性を選択します (`TIMx_CCR2` のキャプチャに使用されます)。
6. `TIMx_SMCR` レジスタの `TS` ビットに“00101”を書き込むことによって (`tim_ti1fp1` を選択)、有効なトリガ入力を選択します。
7. `TIMx_SMCR` レジスタの `SMS` ビットに“100”を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
8. `TIMx_CCER` レジスタの `CC1E` と `CC2E` ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 540. PWM 入力モードタイミング



41.4.9 強制出力モード

このモード (TIMx_CCMRx レジスタの CCxS ビット = "00") では、各出力比較信号 (tim_ocxref、そして tim_ocx) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態にされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (tim_ocxref/tim_ocx) を強制的にアクティブレベルにするには、対応する TIMx_CCMRx レジスタの OCxM ビットに "0101" を書き込む必要があります。これにより、tim_ocxref は強制的にハイレベルになり (tim_ocxref は常にアクティブハイ)、tim_ocx は CCxP 極性ビットと逆の値になります。

例 : CCxP="0" (tim_ocx アクティブハイ) => tim_ocx は強制的にハイレベルになります。

tim_ocxref 信号は、TIMx_CCMRx レジスタの OCxM ビットに "0100" を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

41.4.10 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

1. 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=0000)、アクティブにセットされるか (OCxM=0001)、非アクティブにセットされるか (OCxM=0010)、または反転されます (OCxM=0011)。
2. 割込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCXIF ビット)。
3. 対応する割込みマスク (TIMx_DIER レジスタの CCXIE ビット) がセットされている場合は、割込みを生成します。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

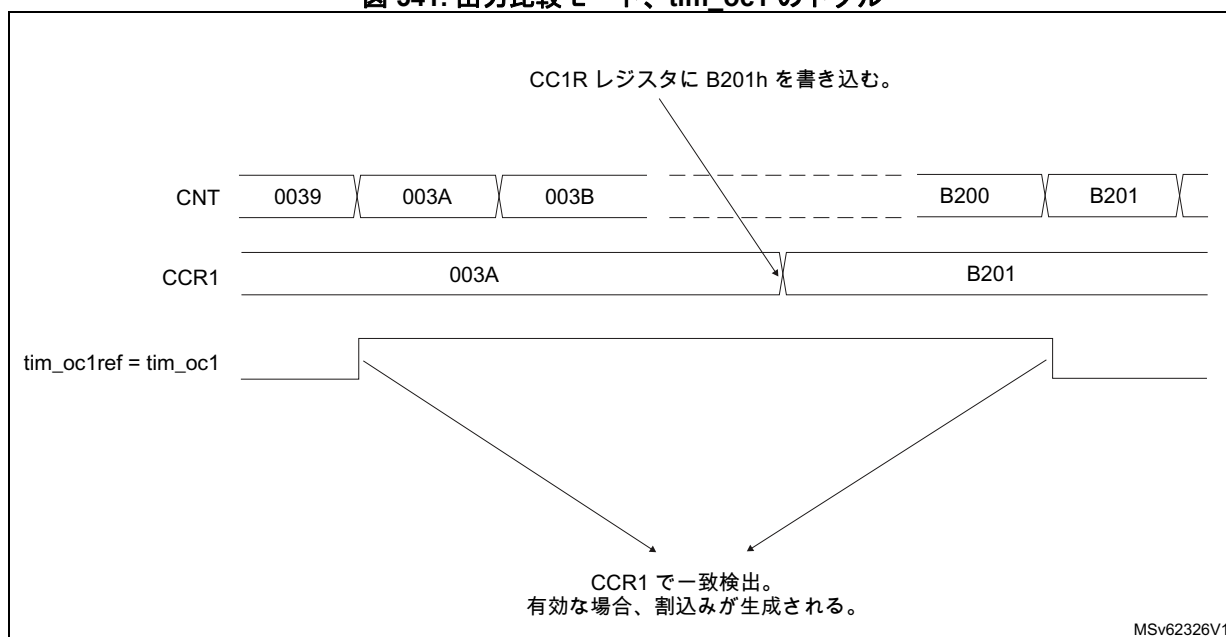
出力比較モードでは、更新イベント UEV は `tim_ocxref` および `tim_ocx` 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます（ワンパルスモード）。

手順：

1. カウンタクロックを選択します（内部、外部、プリスケアラ）。
2. `TIMx_ARR` レジスタと `TIMx_CCRx` レジスタに目的のデータを書き込みます。
3. 割込みリクエストを生成する場合は、`CCxIE` ビットをセットします。
4. 出力モードを選択します。例：
 - CNT と `CCRx` が一致したときに `tim_ocx` 出力ピンを反転するには、`OCxM` ビットに "0011" を書き込みます。
 - プリロードレジスタを無効にするには、`OCxPE` ビットに "0" を書き込みます。
 - アクティブハイ極性を選択するには、`CCxP` ビットに "0" を書き込みます。
 - 出力を有効にするには、`CCxE` ビットに "1" を書き込みます。
5. `TIMx_CR1` レジスタの `CEN` ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって `TIMx_CCRx` レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り（`OCxPE=0`）。そうでない場合、`TIMx_CCRx` シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 541](#) に示します。

図 541. 出力比較モード、`tim_oc1` のトグル



41.4.11 PWM モード

パルス幅変調(PWM)モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“0110” (PWM モード 1) または“0111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (tim_ocx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります、また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

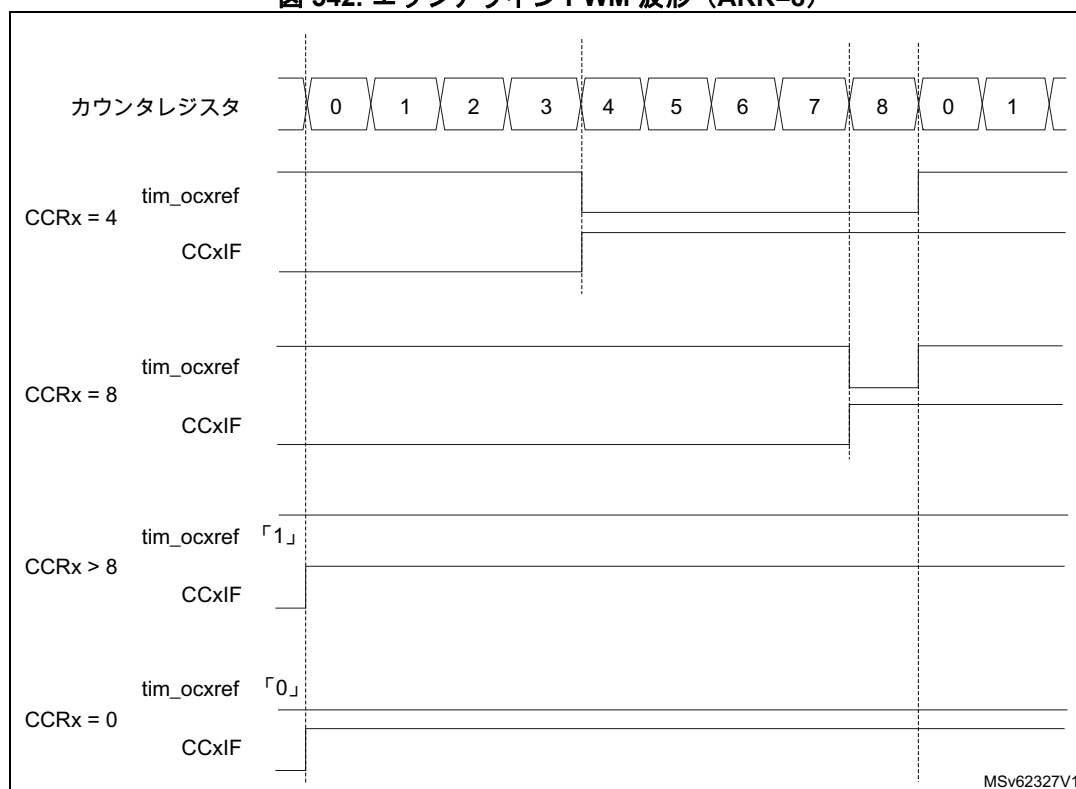
tim_ocx の極性は、TIMx_CCER レジスタの CCxP ビットを使用して、ソフトウェアでプログラムできます。アクティブハイまたはアクティブローとしてプログラムできます。tim_ocx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、 $TIMx_CNT \leq TIMx_CCRx$ かどうか判断されます。

カウンタはカウントアップしているので、タイマはエッジアラインモードでのみ PWM を生成できます。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 tim_ocxref は、 $TIMx_CNT < TIMx_CCRx$ の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、tim_ocxref は“1”に保持されます。比較値が 0 の場合、tim_ocxref は“0”に保持されます。図 542 に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 542. エッジライン PWM 波形 (ARR=8)

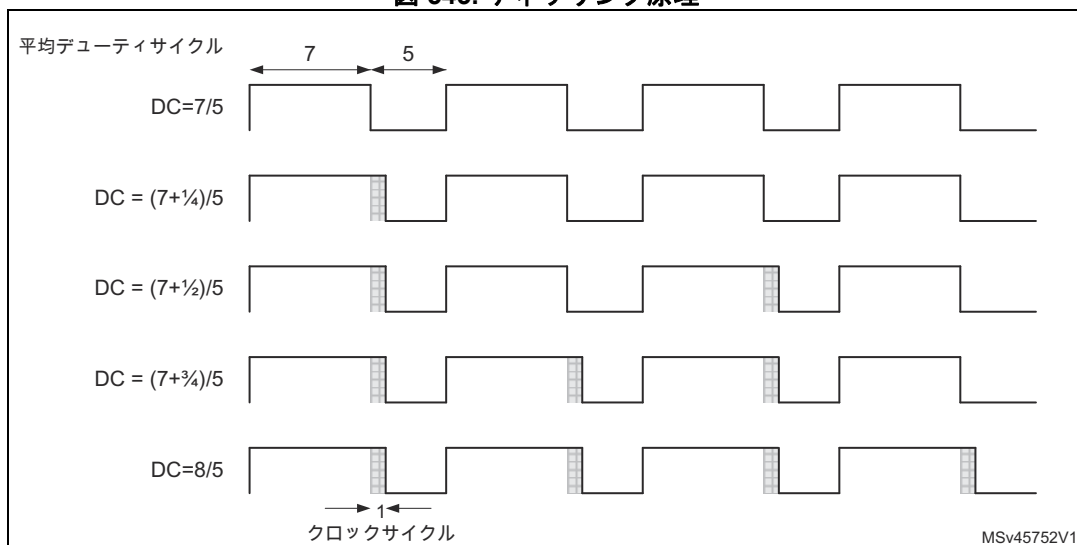


ディザリングモード

PWM モードの有効分解能は、TIMx_CR1 レジスタの DITHEN ビットを使用して、ディザリングモードを有効にすることで増加できます。これは、CCR（デューティサイクル分解能の増加）と ARR（PWM 周波数分解能の増加）の両方に適用されます。

動作原理は、事前に定義されたパターンで、16 個の連続した PWM 周期にわたって実際の CCR（または ARR）値をわずかに変更する（1 タイマクロック周期を追加する、またはしない）ことです。これにより、平均のデューティサイクルまたは PWM 周期を考えると、16 倍の分解能の向上が可能になります。下記の図 543 は、4 つの連続する PWM サイクルに適用されたディザリング原理を示します。

図 543. ディザリング原理



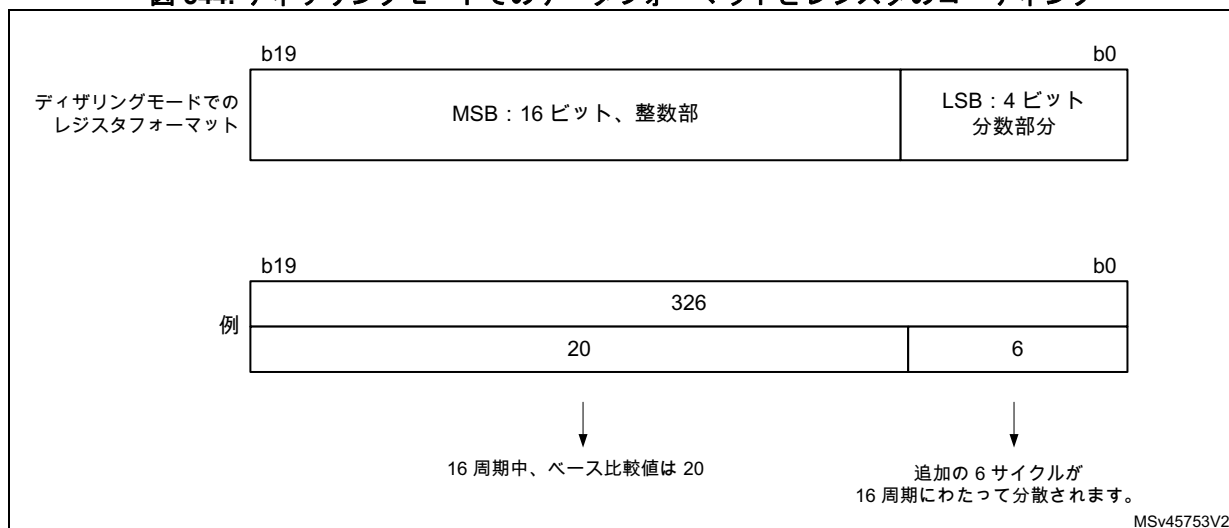
ディザリングモードが有効になっている場合、レジスタのコーディングは以下のように変更されます (たとえば、図 544 を参照)。

- 下位 4 ビットは、拡張分解能部分 (分数部分) 用です。
- 上位ビットは、ビット 19:4 に左シフトされ、ベース値用のコーディングになります。

注：

DITHEN ビットをリセットするときは、次の手順に従う必要があります。
 4を参照してください。CEN および ARPE ビットをリセットする必要があります。
 5を参照してください。DITHEN ビットをリセットする必要があります。
 6を参照してください。CCIF フラグをクリアする必要があります。
 7を参照してください。CEN ビットをセットすることができます (最終的に ARPE = 1 とともに)。

図 544. ディザリングモードでのデータフォーマットとレジスタのコーディング



最小周波数は次の計算式で与えられます。

$$\text{分解能} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

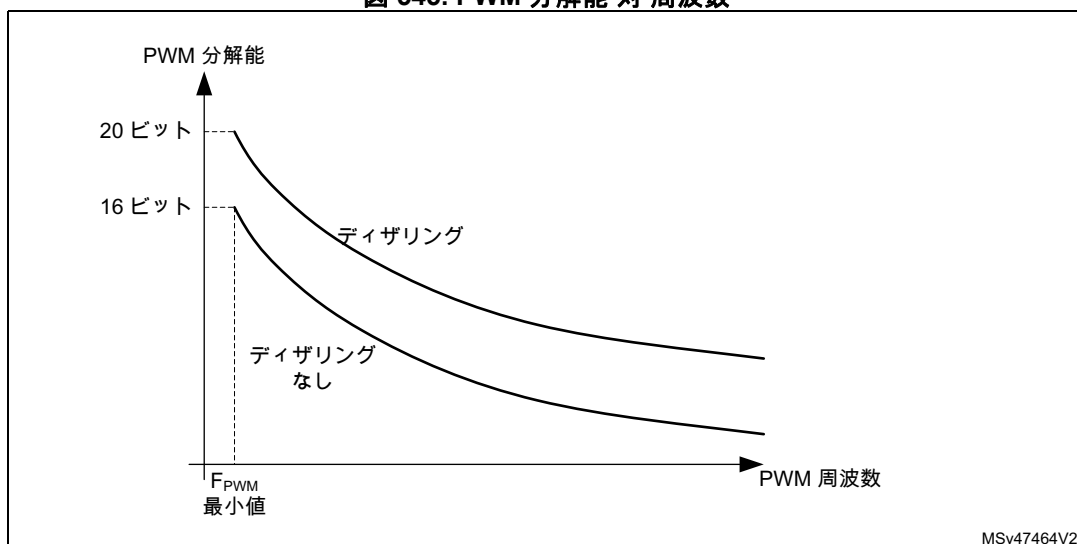
$$\text{ディザリングモード無効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

$$\text{ディザリングモード有効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

注: TIMx_ARR および TIMxCCRy の最大値は、ディザリングモードでは 0xFFFFF に制限されています (整数部では 65534、ディザリング部分では 15 に対応します)。

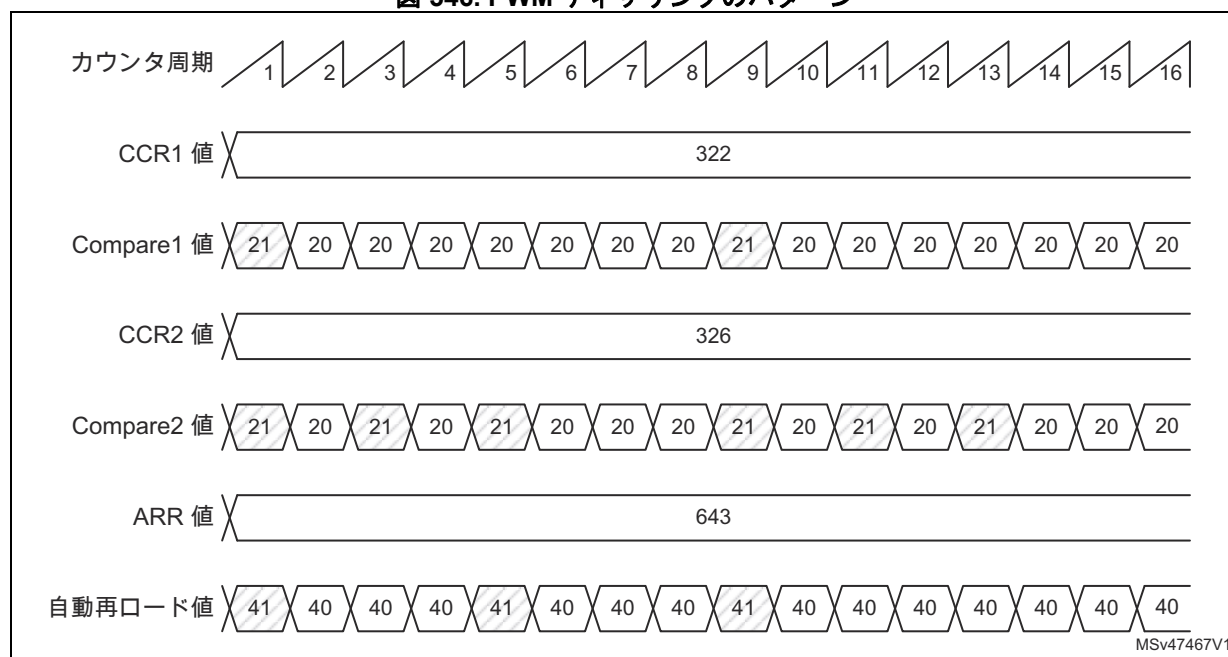
下記の図 545 に示すように、PWM 周波数に関係なく、ディザリングモードによって PWM 分解能を増加することができます。

図 545. PWM 分解能 対 周波数



下記の図 546 で説明するように、デューティサイクルや周期の変更は、連続した 16 周期にわたって広がります。

図 546. PWM ディザリングのパターン



自動再ロード値と比較値の増分は、下記の表 426 に記載された特定のパターンに従って分布します。ディザリングシーケンスは、増分が可能な限り均等に分散され、全体のリップルが最小になるように行われます。

表 426. CCR および ARR レジスタの変更のディザリングパターン

-	PWM 周期															
LSB 値	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-

表 426. CCR および ARR レジスタの変更のディザリングパターン (続き)

-	PWM 周期															
LSB 値	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

41.4.12 組合せ PWM モード (TIM12 のみ)

組み合わせ PWM モードでは、2 つのエッジアラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2 つの TIMx_CCRx レジスタで決定されます。結果として得られる信号 tim_ocxrefc は、2 つの PWM 基準信号の OR または AND による論理結合から成ります。

- tim_oc1refc (または tim_oc2refc) は、TIMx_CCR1 および TIMx_CCR2 レジスタによって制御されます。

組み合わせ PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1100”(組み合わせ PWM モード 1) または“1101”(組み合わせ PWM モード 2) を書き込むことによって、2 チャンネルごとに選択できます (CCR レジスタペアごとに 1 つの tim_ocx 出力)。

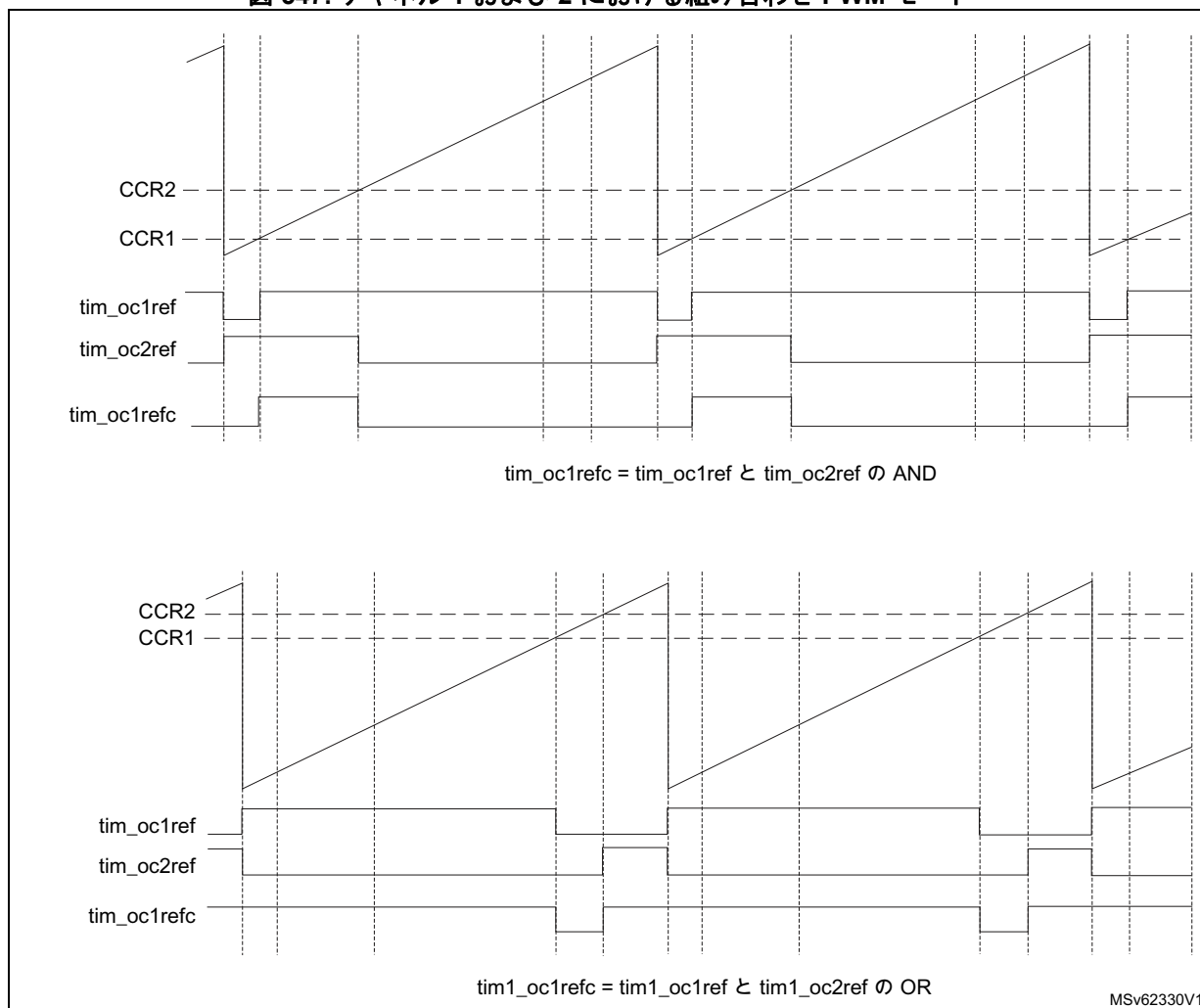
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、相補チャンネルを反対の PWM モードに設定する必要があります (たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします)。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

[図 547](#) は、次の設定で取得可能な組み合わせ PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合

図 547. チャンネル 1 および 2 における組み合わせ PWM モード



41.4.13 ワンパルスモード

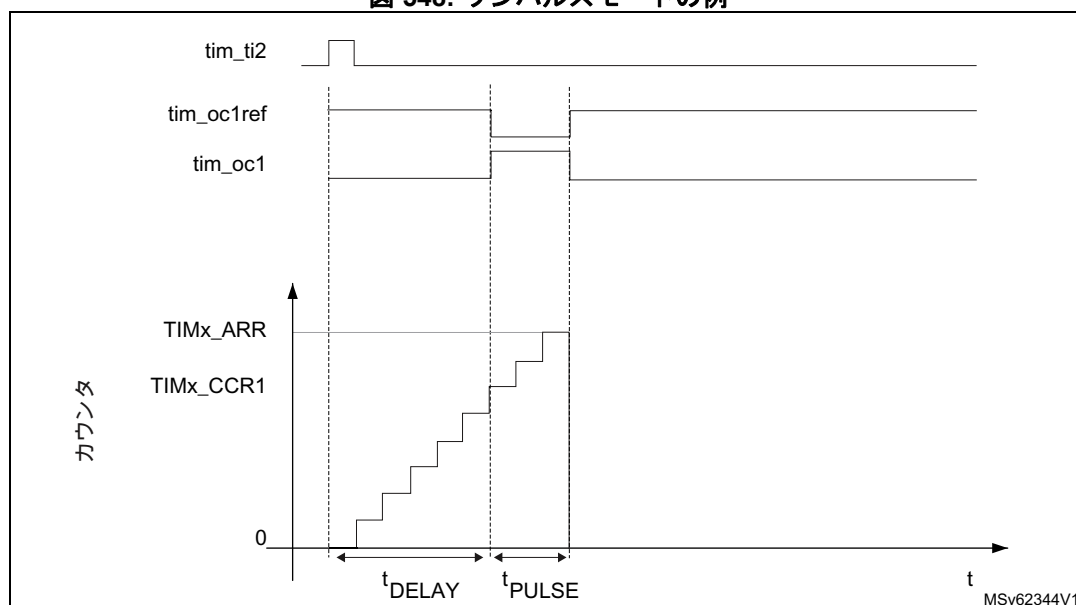
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようでなければなりません。

$$CNT < CCRx \leq ARR \text{ (特に、} 0 < CCRx \text{)}$$

図 548. ワンパルスモードの例



たとえば、tim_ti2 入力ピンで立上がりエッジが検出されたときすぐに、tim_oc1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

tim_ti2fp2 をトリガ 1 として使用します。

1. TIMx_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:0] ソース（内部または外部）を選択します。
2. TIMx_CCMR1 レジスタの CC2S ビットに"01"を書き込むことによって、tim_ti2fp2 を tim_ti2 に配置します。
3. tim_ti2fp2 は、立上がりエッジを検出して、TIMx_CCER レジスタで CC2P="0"と CC2NP="0"を書き込みます。
4. tim_ti2fp2 をスレーブモードコントローラのトリガ (tim_trgi) として構成するために、TIMx_SMCR レジスタの TS="00110"を書き込みます。
5. tim_ti2fp2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに"110"（トリガモード）を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます（クロック周波数とカウンタプリスケアラを考慮に入れて）。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタに OC1M="0111"を書き込むことによって、PWM モード 2 を有効にする必要があります。オプションで、TIMx_CCMR1 レジスタに OC1PE='1'を書き込み、TIMx_CR1 レジスタの ARPE ビットを書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込む必要があります。次に、UG ビットをセットすることによって更新を生成し、tim_ti2 で外部トリガイベントを待つ必要があります。この例では、CC1P に"0"を書き込みます。

必要なパルスは 1 つだけなので（シングルモード）、TIMx_CR1 レジスタの OPM ビットに"1"を書き込みます。こうすると、カウンタは次の更新イベント時に停止します（カウンタが自動再ロード値に

達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊ケース : tim_ocx 高速イネーブル

ワンパルスモードでは、tim_tix 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、tim_ocxref (および tim_ocx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

41.4.14 再トリガ可能なワンパルスモード (TIM12 のみ)

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 41.4.13 : ワンパルスモード](#)で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

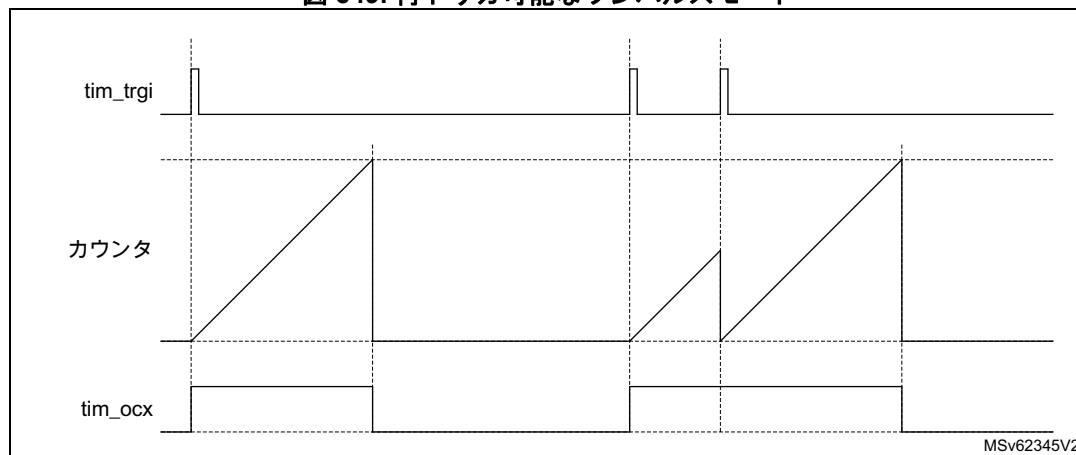
タイマはスレーブモードである必要があり、このときビットは TIMx_SMCR レジスタで SMS[3:0] = “1000” (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が “1000” または “1001” にセットされています。

タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

注 : OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

このモードをセンターアライン PWM モードと組み合わせて使用することはできません。TIMx_CR1 では、CMS[1:0] = 00 にする必要があります。

図 549. 再トリガ可能なワンパルスモード



41.4.15 UIF ビットの再配置

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特定のケースでは、バックグラウンドタスク（カウンタの読出し）と割込み（更新の割込み）との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

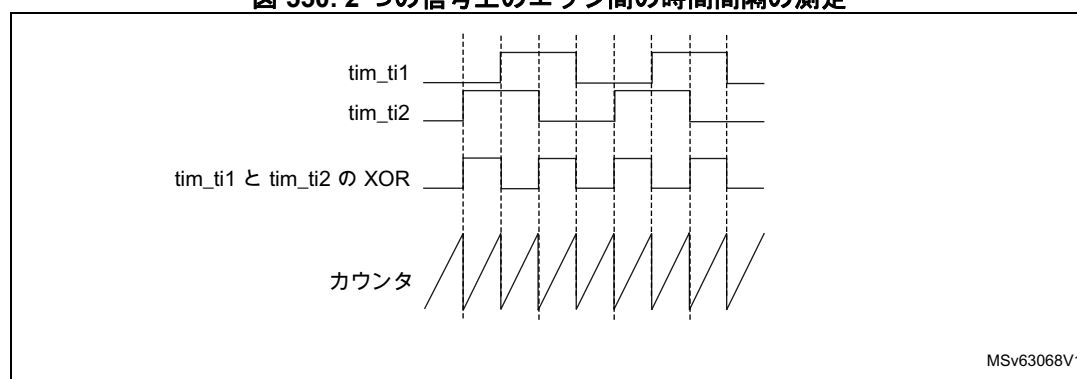
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

41.4.16 タイマ入力 XOR 機能

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 2 つの入力ピン tim_ti1 および tim_ti2 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の図 550 に示すように、2 つの入力信号上のエッジ間の間隔を測定するのに便利です。

図 550. 2 つの信号上のエッジ間の時間間隔の測定



41.4.17 TIM12 外部トリガ同期

TIM12 タイマは、いくつかのモードで外部トリガを使用して同期できます。すなわち、リセットモード、ゲートモード、トリガモード、リセット + トリガモード、およびゲート + リセットモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

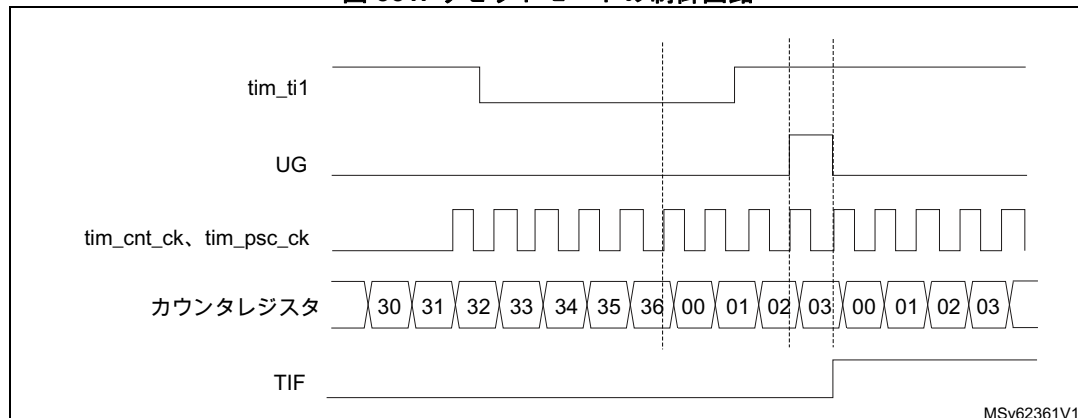
次の例では、tim_ti1 入力の立上がりエッジに応じて、アップカウンタがクリアされます。

1. tim_ti1 の立上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F="0000" のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S = 01）。TIMx_CCER レジスタの CC1P と CC1NP に"00"を書き込んで、極性を有効にします（その後、立上がりエッジのみを検出）。
2. TIMx_SMCR レジスタの SMS ビットに"100"を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタの TS ビットに"00101"を書き込むことによって、入力ソースとして tim_ti1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに"1"を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、tim_ti1 の立上がりエッジまで通常の動作を行います。tim_ti1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、(TIMx_DIER レジスタの TIE ビットが) 有効な場合は割り込みリクエストを送信できます。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。tim_ti1 の立上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_ti1 入力の同期回路によるものです。

図 551. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

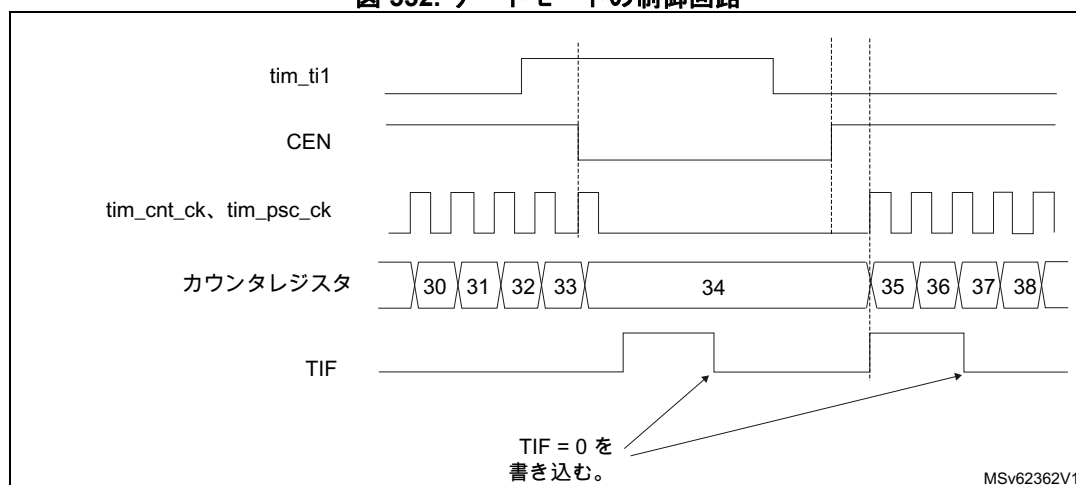
次の例では、アップカウンタは tim_ti1 入力が高レベルのときだけカウントします。

1. tim_ti1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F="0000" のままにしておきます)。キャプチャプリスケラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S = 01)。TIMx_CCER レジスタの CC1P ビットに "1"、CC1NP ビットに "0" を書き込んで、極性を有効にします (その後、ローレベルのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに "101" を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタの TS ビットに "00101" を書き込むことによって、入力ソースとして tim_ti1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに "1" を書き込んで、カウンタを有効にします (ゲートモードでは、CEN = 0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、tim_ti1 がローになると内部クロックでカウントを開始して、tim_ti1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

tim_ti1 の立上がりエッジから実際にカウンタが停止するまでの遅延は、tim_ti1 入力の再同期回路によるものです。

図 552. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

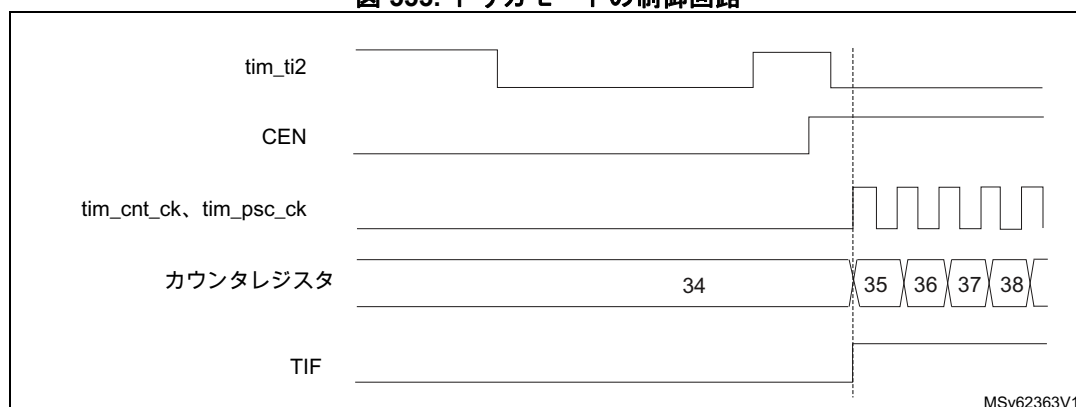
次の例では、アップカウンタは、tim_ti2 入力の立上がりエッジに応じて開始します。

1. tim_ti2 の立上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F="0000" のままにしておきます）。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます（TIMx_CCMR1 レジスタの CC2S = 01）。TIMx_CCER レジスタの CC2P ビットに"1"、CC2NP ビットに"0"を書き込んで、極性を有効にします（その後、ローレベルのみを検出）。
2. TIMx_SMCR レジスタの SMS ビットに"110"を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタの TS ビットに"00110"を書き込むことによって、入力ソースとして tim_ti2 を選択します。

tim_ti2 で立上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際にカウンタが開始するまでの遅延は、tim_ti2 入力の再同期回路によるものです。

図 553. トリガモードの制御回路



41.4.18 スレーブモード - リセットモードとトリガモードの組合せ

この場合、選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

41.4.19 スレーブモード - リセットモードとゲートモードの組合せ

カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

41.4.20 タイマ同期 (TIM12 のみ)

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。詳細については、[セクション 39.4.23 : タイマの同期](#)を参照してください。

注： スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

41.4.21 タイマ出力を他のタイマのトリガとして使用 (TIM13/TIM14 のみ)

1 チャンネルのみのタイマは、マスタモードを備えていません。しかし、OC1 出力信号は、他のいくつかのタイマ (本書の他のセクションで説明されているタイマを含む) のトリガとして使用できます。どのタイマがスレーブとして使用できるかを知るには、デバイスのいずれかのタイマの「TIMx 内部トリガ接続」テーブルを確認してください。

OC1 信号のパルス幅は、スレーブタイマが確実にトリガを検出できるように、転送先タイマの少なくとも 2 クロックサイクルになるようにプログラムする必要があります。

たとえば、転送先タイマの CK_INT クロックが転送元のタイマより 4 倍遅い場合、OC1 のパルス幅は 8 クロックサイクルでなければなりません。

41.4.22 ADC トリガ (TIM12 のみ)

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイイベントを生成できます。

注： tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

41.4.23 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M33 コアは停止状態)、TIMx カウンタは、通常どおりに動作を続けるか、または停止できます。

デバッグモードでの動作は、デバッグサポート (DBG) モジュールでタイマごとに専用の設定ビットを使用してプログラムできます。

詳細については、デバッグセクションを参照してください。

41.5 TIM12/TIM13/TIM14 低消費電力モード

表 427. 低消費電力モードが TIM12/TIM13/TIM14 に与える影響

モード	説明
SLEEP	影響はなく、ペリフェラルはアクティブです。割込みによって、デバイスは SLEEP モードから復帰できます。
STOP	タイマの動作は停止され、レジスタの内容は保持されます。割込みは生成できません。
STANDBY	タイマはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

41.6 TIM12/TIM13/TIM14 割込み

表 428 に示すように、TIM12/TIM13/TIM14 は複数の割込みを生成できます。

表 428. 割込みリクエスト

項目 (割込みの 略称)	割込みイベント	イベント フラグ	有効制御ビット	割込みのクリア方法	SLEEP モードの 終了	STOP および STANDBY モードの 終了
TIM	更新	UIF	UIE	UIF に 0 を書き込む	可	不可
	キャプチャ／比較 1	CC1IF	CC1IE	CC1IF に 0 を書き込む	可	不可
TIM	キャプチャ／比較 2 ⁽¹⁾	CC2IF	CC2IE	CC2IF に 0 を書き込む	可	不可
	トリガ ⁽¹⁾	TIF	TIE	TIF に 0 を書き込む	可	不可

1. TIM12 でのみ使用できます。

41.7 TIM12 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

41.7.1 TIM12 制御レジスタ 1 (TIM12_CR1)

アドレスオフセット : 0x000

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw				rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デジタルフィルタ (tim_tix) によって使用されるサンプリングクロックとの間の分周比を示します。

00 : $t_{DTS} = t_{tim_ker_ck}$

01 : $t_{DTS} = 2 \times t_{tim_ker_ck}$

10 : $t_{DTS} = 4 \times t_{tim_ker_ck}$

11 : 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。
0 : 次のイベントのいずれかが更新割込みを生成します (有効な場合)。これらのイベントは、次のとおりです。

– カウンタオーバーフロー

– UG ビットのセット

– スレーブモードコントローラからの更新生成

1 : カウンタオーバーフローのみが更新割込みを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、更新イベント (UEV) 生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

– カウンタオーバーフロー

– UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

注 : 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。

ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

41.7.2 TIM12 制御レジスタ 2 (TIM12_CR2)

アドレスオフセット : 0x004

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			Res.	Res.	Res.	Res.
								rw	rw	rw	rw				

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S** : tim_ti1 選択

0 : tim_ti1_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます。

1 : tim_ti1_in[15:0] および tim_ti2_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[2:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます

(tim_trgo)。組み合わせは、次のとおりです。

000 : リセット - TIM_x EGR レジスタの UG ビットがトリガ出力 (tim_trgo) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、tim_trgo 信号は実際のリセットより遅延します。

001 : イネーブル - カウンタイネーブル信号 CNT_EN がトリガ出力 (tim_trgo) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (AND) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、tim_trgo には遅延が存在します (TIM_x SMCR レジスタの MSM ビットの説明を参照してください)。

010 : 更新 - 更新イベントがトリガ出力 (tim_trgo) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011 : パルス比較 - キャプチャまたは比較一致が発生するとすぐに、CC1IF フラグがセットされるときに (すでにハイであった場合も)、トリガ出力 (tim_trgo) は正のパルスを送信します。

100 : 比較 - tim_oc1refc 信号がトリガ出力 (tim_trgo) として使用されます。

101 : 比較 - tim_oc2refc 信号がトリガ出力 (tim_trgo) として使用されます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

41.7.3 TIM12 スレーブモード制御レジスタ (TIM12_SMCR)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
								rw	rw	rw	rw		rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (tim_trgi) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (tim_trgo を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期する場合に役立ちます。

ビット 21、20、6、5、**TS[4:0]** : トリガ選択

4 この TS[4:0] ビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

00000 : 内部トリガ 0 (tim_itr0)

00001 : 内部トリガ 1 (tim_itr1)

00010 : 内部トリガ 2 (tim_itr2)

00011 : 内部トリガ 3 (tim_itr3)

00100 : tim_ti1 エッジ検出回路 (tim_ti1f_ed)

00101 : フィルタタイマ入力 1 (tim_ti1fp1)

00110 : フィルタタイマ入力 2 (tim_ti2fp2)

01000 : 内部トリガ 4 (tim_itr4)

01001 : 内部トリガ 5 (tim_itr5)

01010 : 内部トリガ 6 (tim_itr6)

01011 : 内部トリガ 7 (tim_itr7)

01100 : 内部トリガ 8 (tim_itr8)

01101 : 内部トリガ 9 (tim_itr9)

01110 : 内部トリガ 10 (tim_itr10)

01111 : 内部トリガ 10 (tim_itr11)

10000 : 内部トリガ 10 (tim_itr12)

10001 : 内部トリガ 10 (tim_itr13)

10010 : 内部トリガ 10 (tim_itr14)

10011 : 内部トリガ 10 (tim_itr15)

その他 : 予約済み

各タイマにとっての tim_itrx の意味の詳細については、表 425 : TIMx 内部トリガ接続を参照してください。

注： 遷移時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき（たとえば、SMS = "000" のとき）にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 16、2、1、0 **SMS[3:0]** : スレーブモード選択

外部信号が選択されると、トリガ信号 (tim_trgi) のアクティブエッジが、外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : 予約済み

0010 : 予約済み

0011 : 予約済み

0100 : リセットモード - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ tim_trgi の立上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (tim_trgi) の立上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1001 : ゲートモードとリセットモードの組み合わせ - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

その他 : 予約済み

注 : トリガ入力として tim_tif_ed が選択されている場合 (TS="00100")、ゲートモード (ゲート + リセットモードを含む) を使用することはできません。tim_tif_ed は tim_tif の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注 : tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効にする必要があります、マスタタイマからトリガを受信している間は、動作中に変更しないでください。

41.7.4 TIM12 割込み有効レジスタ (TIM12_DIER)

アドレスオフセット : 0x00C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
									rW				rW	rW	rW

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割込みイネーブル

0 : トリガ割込みは無効です。

1 : トリガ割込みは有効です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IE** : キャプチャ/比較 2 割込みイネーブル

0 : CC2 割込みは無効です。

1 : CC2 割込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割込みイネーブル

0 : CC1 割込みは無効です。

1 : CC1 割込みは有効です。

- ビット 0 **UIE** : 更新割込みイネーブル
 0 : 更新割込みは無効です。
 1 : 更新割込みは有効です。

41.7.5 TIM12 ステータスレジスタ (TIM12_SR)

アドレスオフセット : 0x010

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ
 CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。
 0 : オーバーキャプチャは検出されていません。
 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIF** : トリガ割込みフラグ

このフラグは、TRG トリガイイベント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、tim_trgi 入力でアクティブエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。
 0 : トリガイイベントは発生していません。
 1 : トリガ割込みが保留中です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IF** : キャプチャ/比較 2 割込みフラグ
 CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって（入力キャプチャモードまたは出力比較モード）、または TIMx_CCR1 レジスタを読み出すことによって（入力キャプチャモードのみ）クリアされます。

0 : 比較一致/入力キャプチャは発生していません。
 1 : 比較一致または入力キャプチャが発生しました。

チャンネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャンネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた（TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ検知に従って、tim_ic1 でエッジが検出された）ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx_CR1 レジスタで URS="0" かつ UDIS="0" であり、トリガイベントによって CNT が再初期化されたとき ([セクション 41.7.3 : TIM12 スレーブモード制御レジスタ \(TIM12_SMCR\)](#) を参照)。

41.7.6 TIM12 イベント生成レジスタ (TIM12_EGR)

アドレスオフセット : 0x014

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
									w				w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込が発生します。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2G** : キャプチャ/比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

現在のカウンタ値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラのカウンタもクリアされます (分周比は変化しません)。カウンタはクリアされます。

41.7.7 TIM12 キャプチャ／比較モードレジスタ 1 (TIM12_CCMR1)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード（このセクション）または出力比較モード（次のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のすべてのビットは、入力モードと出力モードで機能が異なります。

入力キャプチャモード :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F [3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F[3:0]** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 3:2 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E = 0) のときのみ書き込み可能です。

41.7.8 TIM12 キャプチャ/比較モードレジスタ 1 [オルタネート] (TIM12_CCMR1)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを出力比較モード (このセクション) または入力キャプチャモード (前のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のすべてのビットは、入力モードと出力モードで機能が異なります。

出力比較モード :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 24、14:12 **OC2M[3:0]** : 出力比較 2 モード

ビットの説明は OC1M[3:0] を参照

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

注 : CC2S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E = 0) のときのみ書き込み可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 16、6:4 **OC1M[3:0]** : 出力比較 1 モード (OC1M[3] はビット 16 を参照)

これらのビットは、tim_oc1 が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 のアクティブレベルは CC1P に依存します。

0000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。このモードは、タイマをソフトウェアタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードが有効になると、出力は停止状態になる前の状態（アクティブまたはインアクティブ）を維持します。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。tim_oc1ref 信号は、TIMx_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。tim_oc1ref 信号は、TIMx_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制インアクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111 : PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

1000 : 再トリガ可能な OPM モード 1 - (tim_trgi 信号で) トリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - (tim_trgi 信号で) トリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 OR です。

1101 : 組み合わせ PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 AND です。

1110 : 予約済み。

1111 : 予約済み。

注 : PWM モードでは、比較結果が変化するとき、出力比較モードが「停止」モードから「PWM」モードに切り替わったとき、および出力比較モードが「強制アクティブ/インアクティブ」モードから「PWM」モードに切り替わったときに、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガイベントの効果を加速するために使用されます。

0 : CC1 は、トリガがオンのときでも、カウンタと CCR1 の値に依存して、通常どおりに動作します。
トリガ入力にエッジが発生してから CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E = 0) のときにのみ書き込み可能です。

41.7.9 TIM12 キャプチャ/比較有効レジスタ (TIM12_CCER)

アドレスオフセット : 0x020

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								r/w		r/w	r/w	r/w		r/w	r/w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP** : キャプチャ/比較 2 出力極性

CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ/比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ/比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 : CC1NP はクリア状態に維持する必要があります。

CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、tim_ti1fp1/tim_ti2fp1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

0 : tim_oc1 はアクティブハイ (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

1 : tim_oc1 はアクティブロー (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、トリガまたはキャプチャ動作に対する tim_ti1fp1 と tim_ti2fp1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立上がりエッジ。この回路は tim_tixfp1 の立上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、tim_tixfp1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立下がりエッジ。この回路は tim_tixfp1 の立下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、tim_tixfp1 が反転されます (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ/この回路は tim_tixfp1 の立上がりエッジと立下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、tim_tixfp1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル

0 : キャプチャモードは無効/tim_oc1 は非アクティブ。

1 : キャプチャモードは有効/tim_oc1 信号は、対応する出力ピンに出力されます。

表 429. 標準 tim_ocx チャンネルの出力制御ビット

CCxE ビット	tim_ocx 出力状態
0	出力無効 (タイマによって駆動されない : ハイインピーダンス)
1	出力は有効 (tim_ocx = tim_ocxref + 極性)

注 : 標準 tim_ocx チャンネルに接続されている外部入出力ピンの状態は、tim_ocx チャンネルの状態と、GPIO レジスタに依存します。

41.7.10 TIM12 カウンタ (TIM12_CNT)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN=0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN=1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分は得られません。

41.7.11 TIM12 プリスケアラ (TIM12_PSC)

アドレスオフセット : 0x028

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 $f_{tim_cnt_ck}$ は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC には、更新イベントのたびにアクティブプリスケアラレジスタにロードされる値が含まれます。

(更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

41.7.12 TIM12 自動再ロードレジスタ (TIM12_ARR)

アドレスオフセット : 0x02C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 41.4.3: 1687 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、ARR[15:0] に自動再ロード値を保持します。ARR[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

41.7.13 TIM12 キャプチャ／比較レジスタ 1 (TIM12_CCR1)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR1[19:0]** : キャプチャ／比較 1 値**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] にキャプチャ値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] にキャプチャを保持します。CCR1[3:0] ビットはリセットされています。

41.7.14 TIM12 キャプチャ／比較レジスタ 2 (TIM12_CCR2)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR2[19:0]** : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、tim_oc2 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] に比較値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] に整数部を保持します。CCR2[3:0] ビットフィールドにはディザリング部分が含まれます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 1 イベント (tim_ic2) によって転送されたカウンタ値です。TIMx_CCR2 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] にキャプチャ値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] にキャプチャを保持します。CCR2[3:0] ビットはリセットされています。

41.7.15 TIM12 タイマ入力選択レジスタ (TIM12_TISEL)

アドレスオフセット : 0x05C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TI2SEL[3:0]** : tim_ti2_in[15:0] の入力を選択します。

0000 : TIMx_CH2 入力 (tim_ti2_in0)

0001 : tim_ti2_in1

.....

0100 : tim_ti2_in15

相互接続一覧については、[表 424 : tim_ti2 入力マルチプレクサへの相互接続](#)を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1_in[15:0] の入力を選択します。

0000 : TIMx_CH1 入力 (tim_ti1_in0)

0001 : tim_ti1_in1

.....

1111 : tim_ti1_in15

相互接続一覧については、[表 423 : tim_ti1 入力マルチプレクサへの相互接続](#)を参照してください。

41.7.16 TIM12 レジスタマップ

TIM12 のレジスタは、次のように 16 ビットのアドレス可能レジスタとしてマップされます：

表 430. TIM12レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	TIM12_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DITHEN	UIFREMAP	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	リセット値																				0	0			0	0				0	0	0	0	
0x004	TIM12_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T1S	MMS[2:0]			Res.	Res.	Res.	Res.	
	リセット値				0																					0	0	0	0					
0x008	TIM12_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]	Res.	Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]			
	リセット値										0	0					0									0	0	0	0		0	0	0	
0x00C	TIM12_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	Res.	CC2IE	CC1IE	UIE	
	リセット値																									0					0	0	0	
0x010	TIM12_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIF	Res.	Res.	Res.	Res.	CC2IF	CC1IF	UIF
	リセット値																						0	0			0				0	0	0	
0x014	TIM12_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	Res.	CC2G	CC1G	UG
	リセット値																										0				0	0	0	
0x018	TIM12_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]			IC2 PSC [1:0]		CC2S [1:0]		IC1F [3:0]			IC1 PSC [1:0]		CC1S [1:0]				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	TIM12_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値								0									0		0	0	0	0	0	0		0	0	0	0	0	0	0	
0x01C	予約済み	Res.																																
0x020	TIM12_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	リセット値																									0		0	0	0		0	0	
0x024	TIM12_CNT	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x028	TIM12_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x02C	TIM12_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 430. TIM12レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030	予約済み	予約済み																															
0x034	TIM12_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1 [19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x038	TIM12_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2 [19:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x03C から 0x058	予約済み	Res.																															
0x05C	TIM12_TISEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
	リセット値																					0	0	0	0					0	0	0	0
0x060 - 0x3E8	予約済み	予約済み																															

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

41.8 TIM13/TIM14 レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

41.8.1 TIMx 制御レジスタ 1 (TIMx_CR1) (x = 13, 14)

アドレスオフセット : 0x000

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw				rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デジタルフィルタ (tim_tix) によって使用されるサンプリングクロックとの間の分周比を示します。

00 : $t_{DTS} = t_{tim_ker_ck}$

01 : $t_{DTS} = 2 \times t_{tim_ker_ck}$

10 : $t_{DTS} = 4 \times t_{tim_ker_ck}$

11 : 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、更新割込み (UEV) ソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが UEV を発生します (有効な場合)。

– カウンタオーバーフロー

– UG ビットのセット

1 : カウンタオーバーフローのみが UEV を発生します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、更新割込み (UEV) イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

– カウンタオーバーフロー

– UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注 : 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。

ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

41.8.2 TIMx 割込み有効レジスタ (TIMx_DIER) (x = 13、14)

アドレスオフセット : 0x00C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ/比較 1 割込みイネーブル

0 : CC1 割込みは無効です。

1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル

0 : 更新割込みは無効です。

1 : 更新割込みは有効です。

41.8.3 TIMx ステータスレジスタ (TIMx_SR) (x = 13、14)

アドレスオフセット : 0x010

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって（入力キャプチャモードまたは出力比較モード）、または TIMx_CCR1 レジスタを読み出すことによって（入力キャプチャモードのみ）クリアされます。

0 : 比較一致/入力キャプチャは発生していません。

1 : 比較一致または入力キャプチャが発生しました。

チャンネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャンネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた（TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ検知に従って、tim_ic1 でエッジが検出された）ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

41.8.4 TIMx イベント生成レジスタ (TIMx_EGR) (x = 13、14)

アドレスオフセット : 0x014

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。カウンタはクリアされます。

41.8.5 TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) (x = 13、14)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード (このセクション) または出力比較モード (次のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。

入力キャプチャモード :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 3:2 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : 予約済み

11 : 予約済み

注： **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

41.8.6 TIMx キャプチャ/比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 13、14)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。

出力比較モード :

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
									rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 16、6:4 **OC1M[3:0]** : 出力比較 1 モード (OC1M[3] はビット 16 を参照)

これらのビットは、tim_oc1 が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 のアクティブレベルは CC1P ビットに依存します。

0000 : 停止。出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。このモードは、タイマをソフトウェアタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードが有効になると、出力は停止状態になる前の状態（アクティブまたはインアクティブ）を維持します。

0001 : 一致時にチャンネル 1 をアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 1 をインアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制非アクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111 : PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

その他 : 予約済み

注 : PWM モードでは、比較結果が変化したとき、出力比較モードが「停止」モードから「PWM」モードに切り替わったとき、および出力比較モードが「強制アクティブ/インアクティブ」モードから「PWM」モードに切り替わったときに、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われず、TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、トリガイメントとタイマ出力の遷移の間の待ち時間を短縮します。開始トリガの後、できるだけ早く出力パルスを開始するには、1 パルスモード (TIMx_CR1 レジスタの OPM ビットのセット) で使用する必要があります。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : 予約済み。

11 : 予約済み。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書込み可能です。

41.8.7 TIMx キャプチャ/比較有効レジスタ (TIMx_CCER) (x = 13、14)

アドレスオフセット : 0x020

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 : CC1NP はクリアされたままにする必要があります。

CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、tim_ti1fp1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

0 : tim_oc1 はアクティブハイ (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

1 : tim_oc1 はアクティブロー (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、キャプチャ動作に対する tim_ti1fp1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立上がりエッジ。この回路は tim_ti1fp1 の立上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TlxFP1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立下がりエッジ。この回路は tim_ti1fp1 の立下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、tim_ti1fp1 が反転されます (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ/この回路は tim_ti1fp1 の立上がりエッジと立下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、tim_ti1fp1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。

0 : キャプチャモードは無効/tim_oc1 は非アクティブ。

1 : キャプチャモードは有効/tim_oc1 信号は、対応する出力ピンに出力されます。

表 431. 標準 tim_ocx チャンネルの出力制御ビット

CCxE ビット	tim_ocx 出力状態
0	出力は無効です (tim_ocx="0")
1	tim_ocx=tim_ocxref + 極性

注 : 標準 tim_ocx チャンネルに接続されている外部入出力ピンの状態は、tim_ocx チャンネルの状態と、GPIO レジスタに依存します。

41.8.8 TIMx カウンタ (TIMx_CNT) (x = 13、14)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分は得られません。

41.8.9 TIMx プリスケアラ (TIMx_PSC) (x = 13、14)

アドレスオフセット : 0x028

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 $f_{tim_cnt_ck}$ は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

(更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

41.8.10 TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 13、14)

アドレスオフセット : 0x02C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 41.4.3: 1687 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、ARR[15:0] に自動再ロード値を保持します。ARR[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

41.8.11 TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 13、14)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 CCR1[19:0] : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] にキャプチャ値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] にキャプチャを保持します。CCR1[3:0] ビットはリセットされています。

41.8.12 TIMx タイマ入力選択レジスタ (TIMx_TISEL) (x = 13、14)

アドレスオフセット : 0x05C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL [3:0]			
												rw	rw	rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1_in[15:0] の入力を選択します。

0000 : TIMx_CH1 入力 (tim_ti1_in0)

0001 : tim_ti1_in1

.....

1111 : tim_ti1_in15

相互接続一覧については、表 423 : tim_ti1 入力マルチプレクサへの相互接続を参照してください。

41.8.13 TIM13/TIM14 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットのアドレス可能レジスタとしてマップされます。

表 432. TIM13/TIM14 レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DITHEN	UIFREMAP	Res.	Res.	CKD[1:0]	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	リセット値																				0	0		0	0	0				0	0	0	0
0x004 から 0x008	予約済み	Res.																															
0x00C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
	リセット値																															0	0
0x010	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
	リセット値																							0								0	0
0x014	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
	リセット値																															0	0
0x018	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	リセット値																0										0	0	0	0	0	0	0
	TIMx_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F [3:0]			IC1 PSC [1:0]	CC1S [1:0]			
	リセット値																										0	0	0	0	0	0	0
0x01C	予約済み	Res.																															

表 432. TIM13/TIM14レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x020	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E			
	リセット値																													0		0	0				
0x024	TIMx_CNT	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																			
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x028	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x02C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																							
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x030	予約済み	Res.																																			
0x034	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1 [19:0]																							
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x038 から 0x058	予約済み	Res.																																			
0x05C	TIMx_TISEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL [3:0]							
	リセット値																													0	0	0	0				
0x060 0x3E8	予約済み	Res.																																			

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

42 汎用タイマ (TIM15/TIM16/TIM17)

42.1 TIM15/TIM16/TIM17 の概要

TIM15/TIM16/TIM17 タイマは、プログラム可能なプリスケアラによって駆動される 16 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM、デッドタイムを挿入した相補 PWM）など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

TIM15/TIM16/TIM17 タイマは完全に独立していて、いかなるリソースも共有しません。TIM15 は、[セクション 42.4.26 : タイマ同期 \(TIM15 のみ\)](#) に示すように同期させることができます。

42.2 TIM15 の主な機能

TIM15 には以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能
- 次の機能を持つ、最大 2 つの独立チャンネル：
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジモード）
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力（チャンネル 1 のみ）
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレイク入力。
- 以下のイベント時の割込み/DMA 生成：
 - 更新：カウンタオーバーフロー、カウンタの初期化（ソフトウェアまたは内部/外部トリガによる）
 - トリガイベント（カウンタの開始、停止、初期化、または内部/外部トリガによるカウント）
 - 入力キャプチャ
 - 出力比較
 - ブレイク入力（割込みリクエスト）

42.3 TIM16/TIM17 の主な特徴

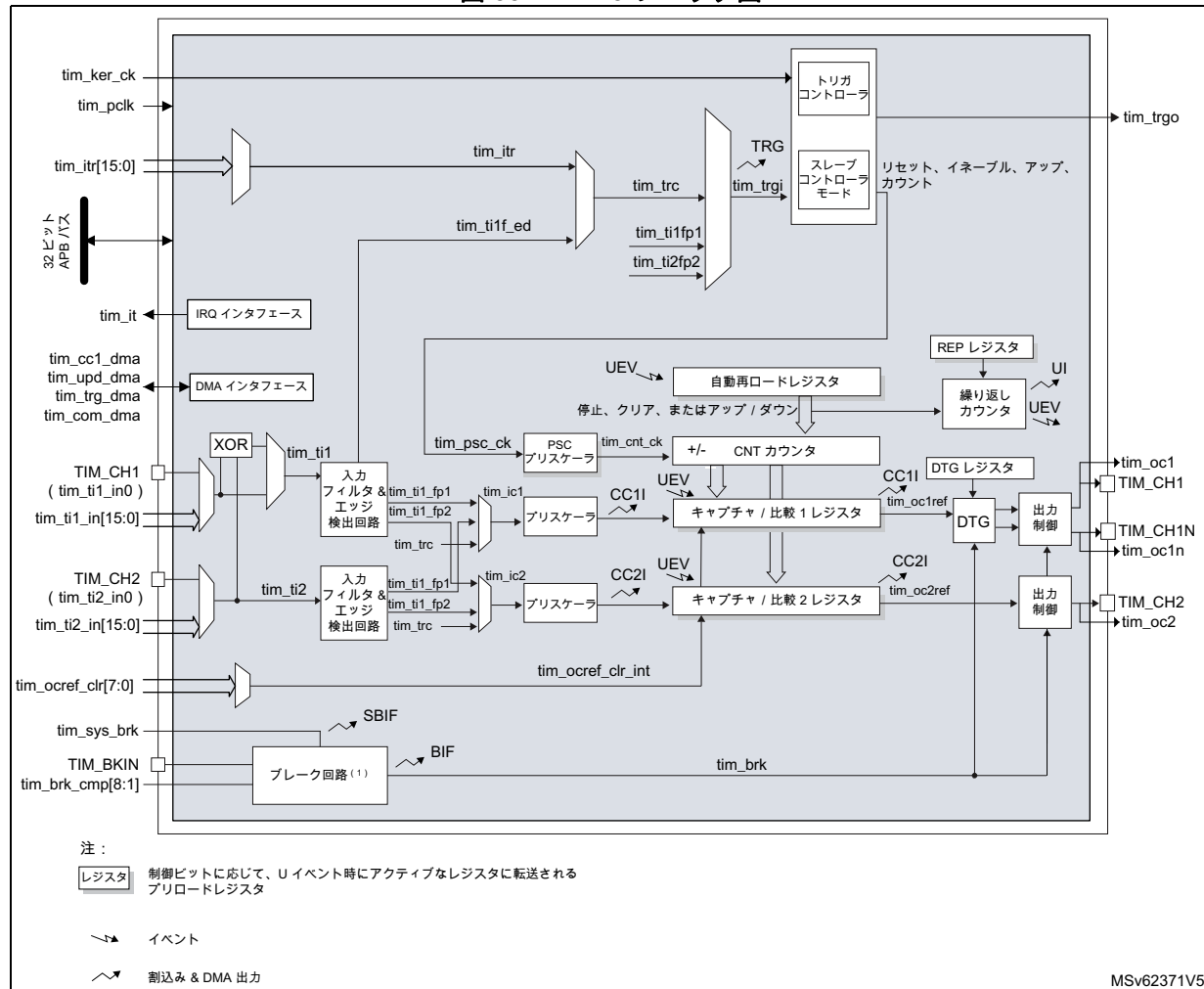
TIM16/TIM17 タイマには以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能
- 次の機能を持つ、1 チャンネルタイマ。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモード）
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレーク入力。
- 以下のイベント時の割込み/DMA 生成：
 - 更新：カウンタオーバーフロー
 - 入力キャプチャ
 - 出力比較
 - ブレーク入力

42.4 TIM15/TIM16/TIM17機能詳細

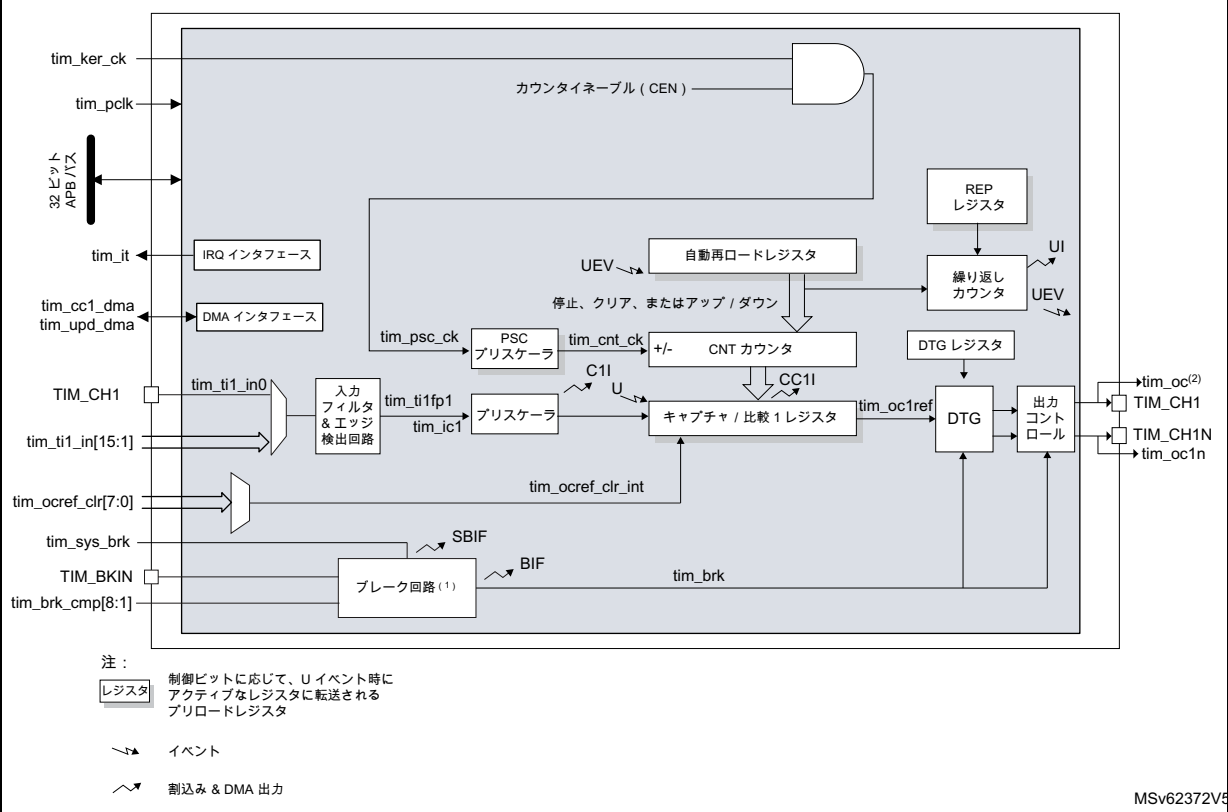
42.4.1 ブロック図

図 554. TIM15 ブロック図



1. 詳細については、[セクション 42.4.15：ブ레이크機能の使用](#)を参照してください。

図 555. TIM16/TIM17 ブロック図



1. 詳細については、[セクション 42.4.15：ブレーク機能の使用](#)を参照してください。
2. この信号は一部のスレーブタイマのトリガとして使用できます（次のセクションの内部トリガ接続表を参照）。詳細については、[セクション 42.4.27：タイマ出力を他のタイマのトリガとして使用（TIM16/TIM17 のみ）](#)を参照してください。

42.4.2 TIM15/TIM16/TIM17 ピンおよび内部信号

このセクションの表 433 および表 434 は、TIM の入力および出力の一覧です。

表 433. TIM の入出力ピン

ピン名	信号タイプ	説明
TIM_CH1 TIM_CH2 ⁽¹⁾	入力／出力	タイマ多目的チャネル。 各チャネルは、キャプチャ、比較、または PWM で使用できます。 TIM_CH1 と TIM_CH2 は、外部クロック（tim_ker_ck クロックの 1/4 未満）、外部トリガの入力としても使用できます。
TIM_CH1N	出力	デッドタイム挿入の可能性がある TIM_CH1 出力から導き出されるタイマ相補出力。
TIM_BKIN	入力／出力	ブレーク入力。この入力は双方向モードにも設定することができます。

1. TIM15 でのみ使用できます。

表 434. TIM 内部入力／出力信号

内部信号名	信号タイプ	説明
tim_ti1_in [15:0] tim_ti2_in [15:0] ⁽¹⁾	入力	内部タイマ入力バス。これらの入力は、キャプチャ用に、または外部クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_itr[15:0] ⁽¹⁾	入力	内部トリガ入力バス。これらの入力は、スレーブモードコントローラ用に、または入力クロック (tim_ker_ck クロックの 1/4 未満) として使用できます。
tim_trgo ⁽¹⁾	出力	内部トリガ出力。このトリガは他のオンチップペリフェラルをトリガできます。
tim_ocref_clr[7:0]	入力	タイマの tim_ocref_clr 入力バス。これらの入力は、通常ハードウェアのサイクルごとのパルス幅制御のために、tim_ocxref 信号をクリアするために使用できます。
tim_brk_cmp[8:1]	入力	内部信号のブレーク入力
tim_sys_brk[n:0]	入力	システムブレーク入力この入力は MCU のシステムレベルのエラーを集めます。
tim_pclk	入力	タイマ APB クロック
tim_ker_ck	入力	タイマカーネルクロック。このクロックは、tim_pclk (同じソースから導出) と同期する必要があります。クロック比 tim_ker_ck/tim_pclk は整数 1、2、3、・・・16 (最大値) でなければなりません。
tim_it	出力	キャプチャ／比較、更新、ブレークトリガ、および転流リクエストを取りまとめるグローバルタイマ割込み。
tim_cc1_dma	出力	タイマキャプチャ／比較 1 DMA リクエスト
tim_upd_dma	出力	タイマ更新 DMA リクエスト
tim_trg_dma	出力	タイマトリガ DMA リクエスト
tim_com_dma	出力	タイマ転流 DMA リクエスト

1. TIM15 でのみ使用できます。

表 435 および表 436 に、tim_ti[2:1] 入力マルチプレクサに接続されるソースを列挙します。

表 435. tim_ti1 入力マルチプレクサへの相互接続

tim_ti1 入力	ソース		
	TIM15	TIM16	TIM17
tim_ti1_in0	TIM15_CH1	TIM16_CH1	TIM17_CH1
tim_ti1_in1	TIM2_CH1	RCC_LSI	予約済み
tim_ti1_in2	TIM3_CH1	RCC_LSE	RCC_HSE_1MHz
tim_ti1_in3	TIM4_CH1	RTC ウェイクアップ	RCC_MCO1
tim_ti1_in4	RCC_LSE	予約済み	予約済み
tim_ti1_in5	RCC_CSI/128		
tim_ti1_in5	RCC_MCO2		
tim_ti1_in [15:6]	予約済み		

表 436. tim_ti2 入力マルチプレクサへの相互接続

tim_ti2 入力	ソース
	TIM15
tim_ti2_in0	TIM15_CH2
tim_ti2_in1	TIM2_CH2
tim_ti2_in2	TIM3_CH2
tim_ti2_in3	TIM4_CH2
tim_ti2_in [15:4]	予約済み

表 437 に、tim_itr 入力マルチプレクサに接続される内部ソースを列挙します。

表 437. TIMx 内部トリガ接続

Tim_itr 入力	TIM15
tim_itr0	tim1_trgo
tim_itr1	tim2_trgo
tim_itr2	tim3_trgo
tim_itr3	tim4_trgo
tim_itr4	tim5_trgo
tim_itr5	tim8_trgo
tim_itr6	tim12_trgo
tim_itr7	tim13_oc1
tim_itr8	tim14_oc1
tim_itr9	予約済み
tim_itr10	tim16_oc1

表 437. TIMx 内部トリガ接続 (続き)

Tim_itr _x 入力	TIM15
tim_itr11	tim17_oc1
tim_itr[15:12]	予約済み

表 438 および表 439 に、tim_brk 入力に接続されるソースを列挙します。

表 438. タイマブレーク相互接続

tim_brk 入力	TIM15	TIM16	TIM17
TIM_BKIN	TIM15_BKIN ピン	TIM16_BKIN ピン	TIM17_BKIN ピン
tim_brk_cmp[8:1]	予約済み		

表 439. システムブレーク相互接続

tim_sys_brk 入力	TIM15/16/17	SBS_CFGR2 レジスタのイネーブルビット
tim_sys_brk0	FLASH ダブル ECC エラー	ECCL
tim_sys_brk1	プログラム可能な電圧検出器 (PVD)	PVDL
tim_sys_brk2	SRAM ダブル ECC エラー	SEL
tim_sys_brk3	Cortex®-M33 LOCKUP	CLL
CSS	クロックセキュリティシステム	なし (常に有効)

42.4.3 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ有効な、プリスケアラ出力 tim_cnt_ck から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 556 と 図 557 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 556. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

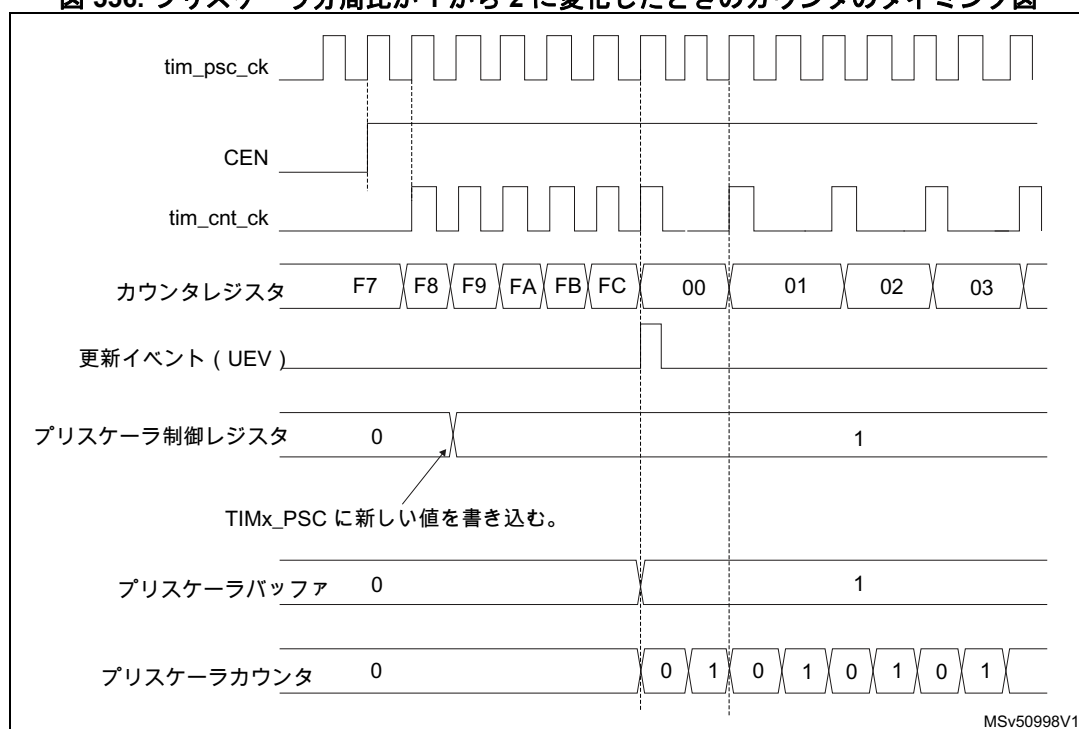
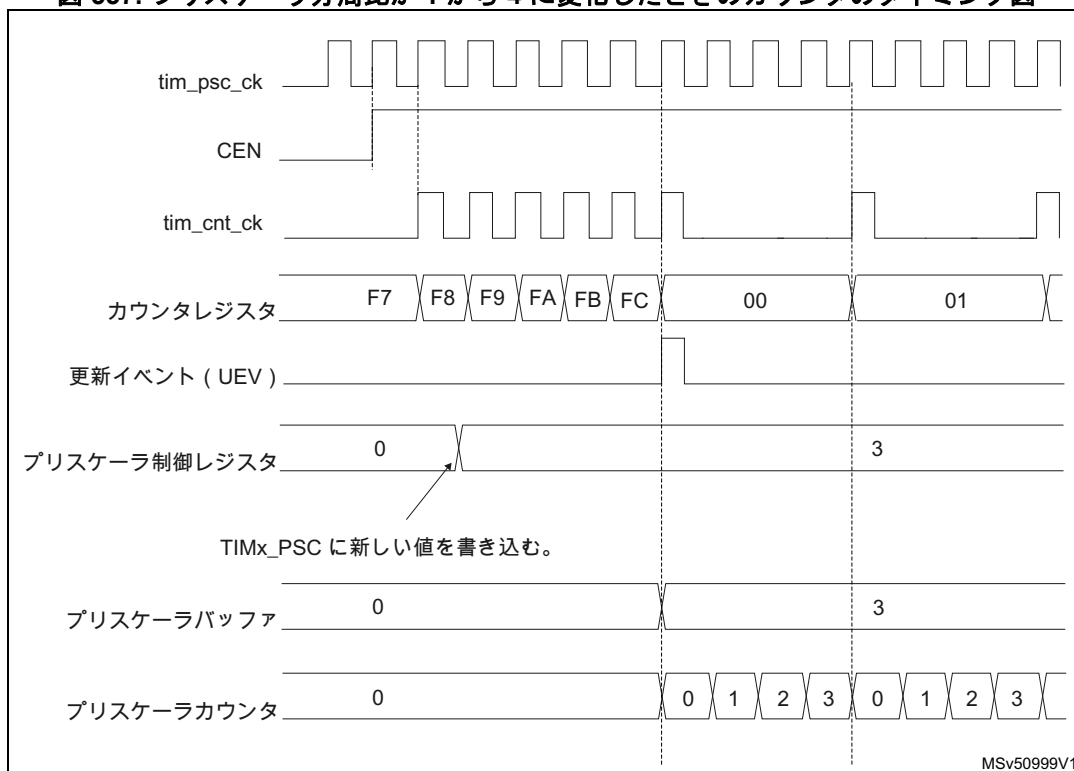


図 557. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



42.4.4 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 558. 内部クロック分周比が 1 の場合のカウンタのタイミング図

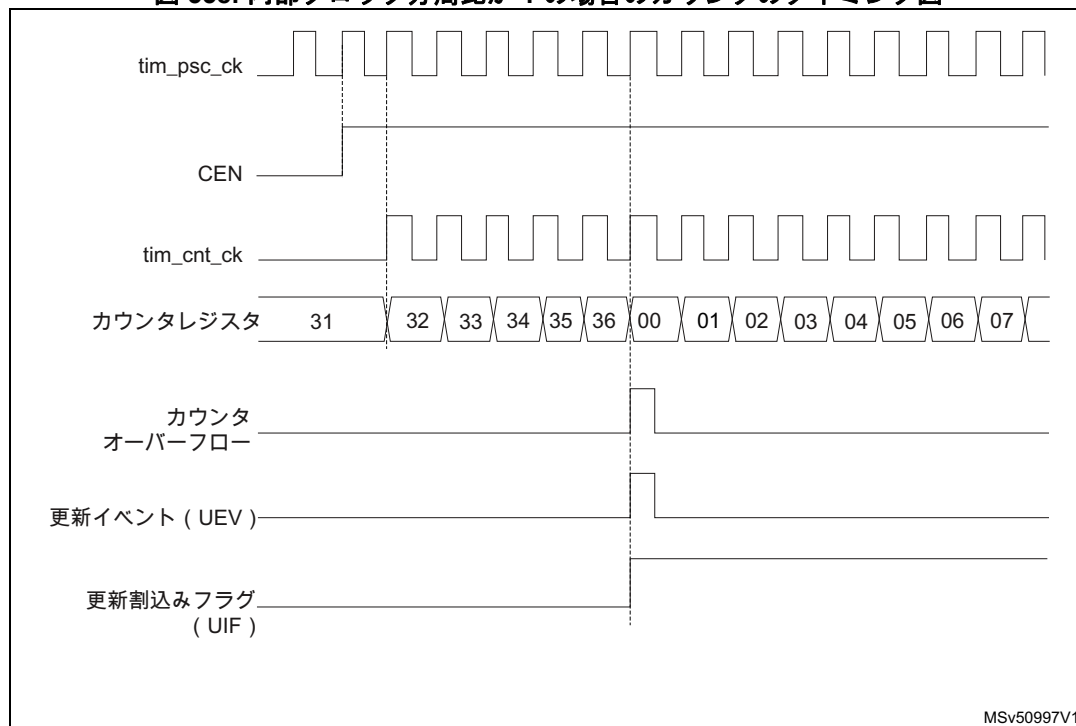


図 559. 内部クロック分周比が 2 の場合のカウンタのタイミング図

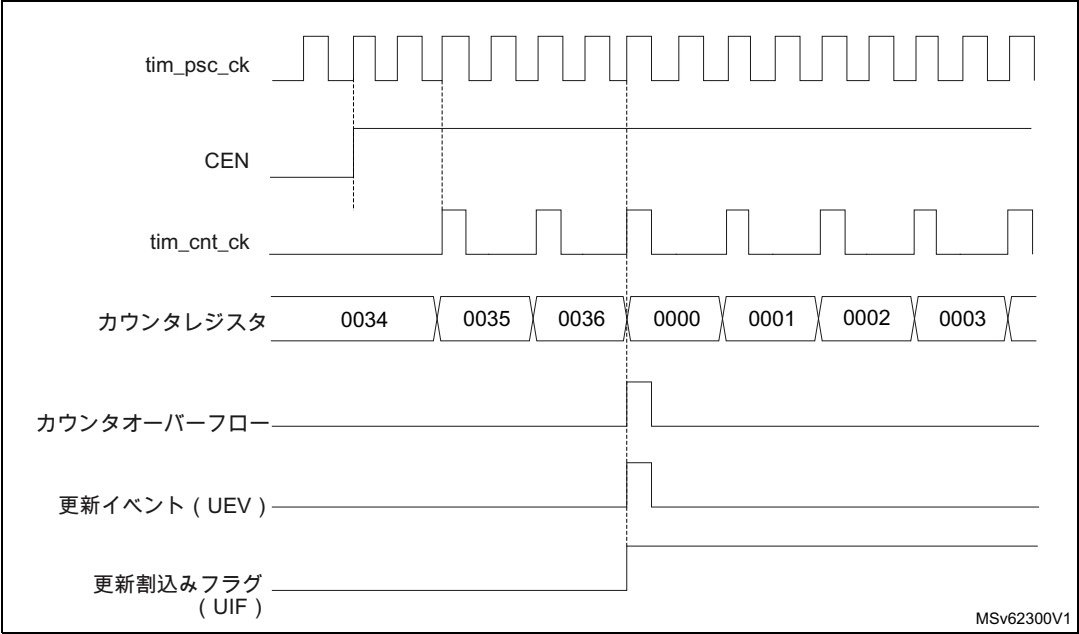


図 560. 内部クロック分周比が 4 の場合のカウンタのタイミング図

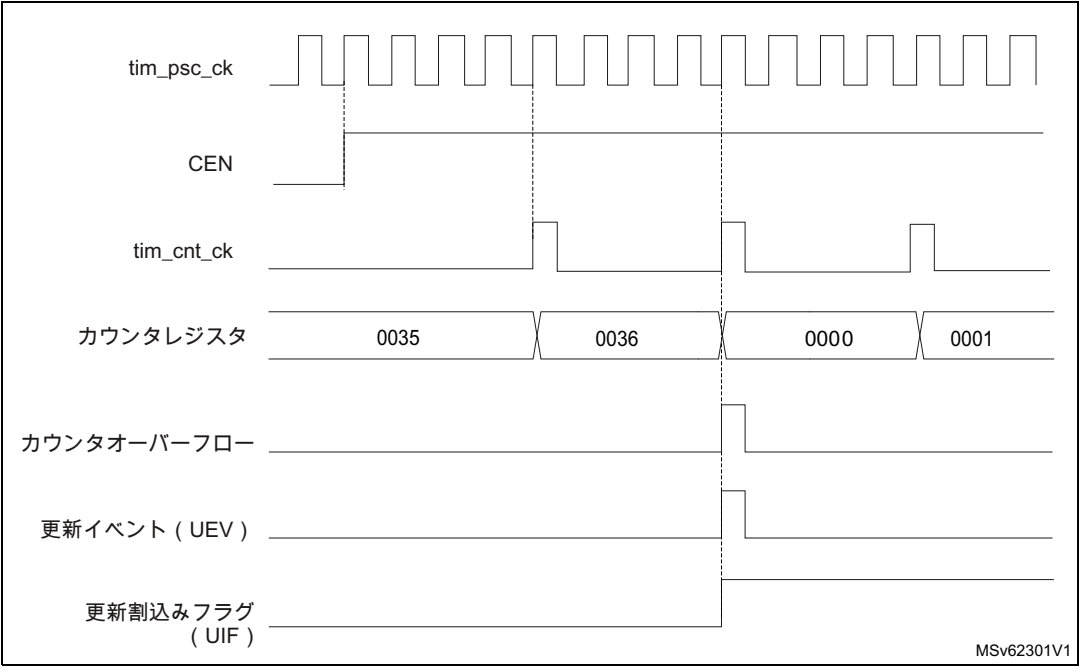


図 561. 内部クロック分周比が N の場合のカウンタのタイミング図

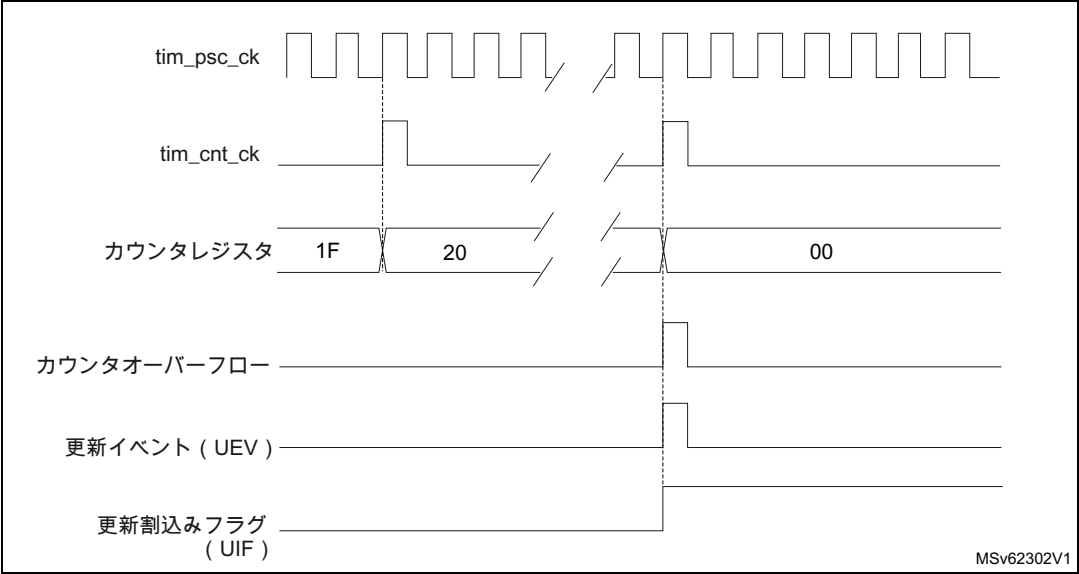


図 562. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

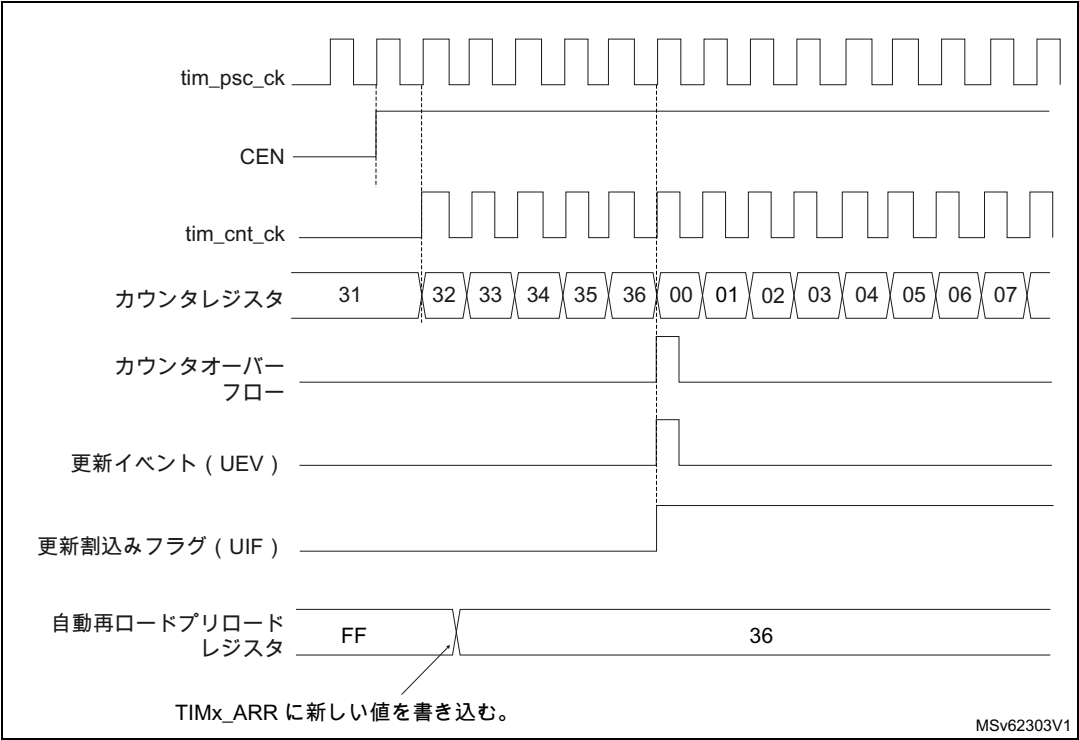
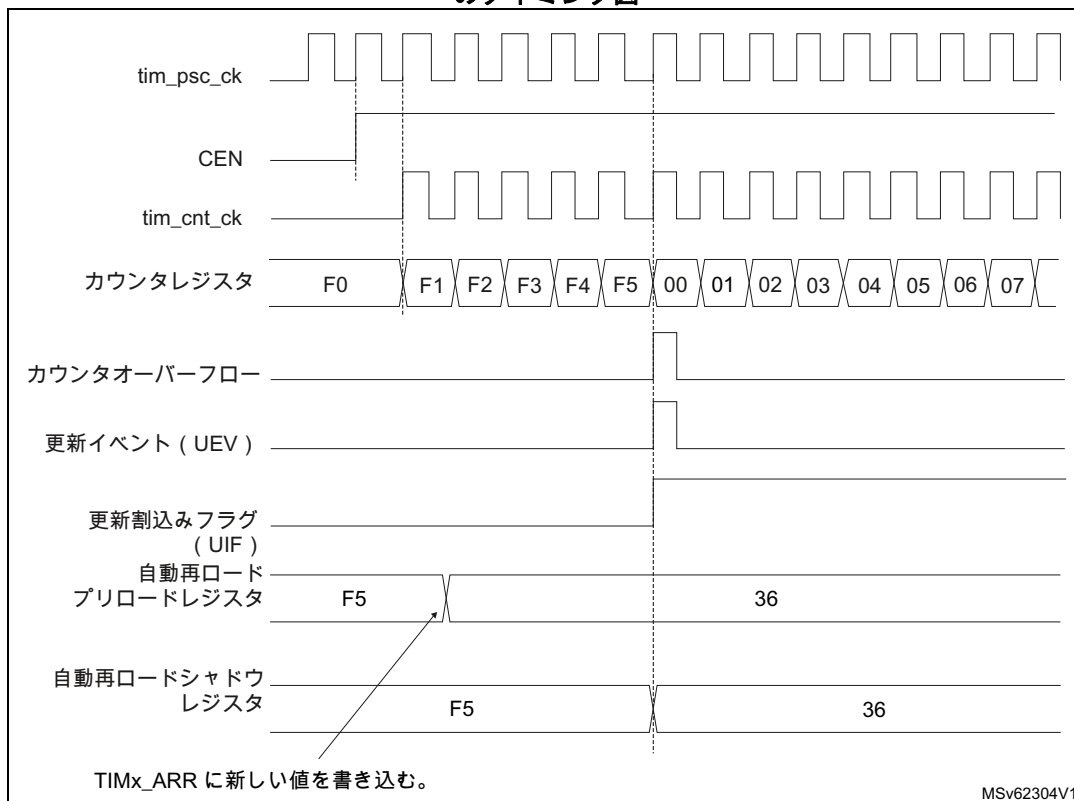


図 563. ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



42.4.5 繰返しカウンタ

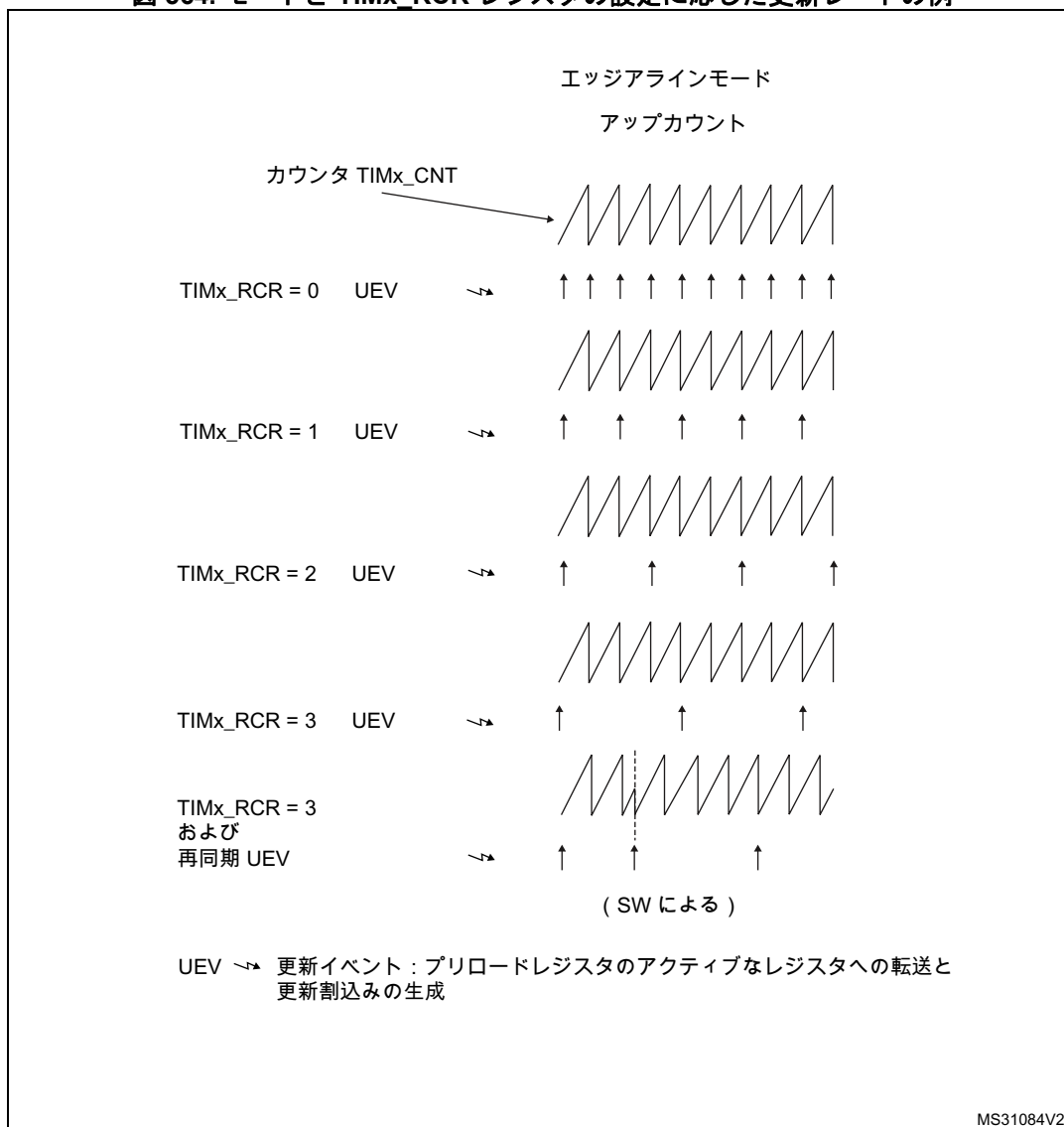
セクション 42.4.3: タイムベースユニットに、カウンタオーバーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx_RCR 繰返しカウンタレジスタの値を N とすると、N 回目のカウンタオーバーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx_ARR 自動再ロードレジスタ、TIMx_PSC プリスケアラレジスタ、比較モードの TIMx_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰返しカウンタは、カウンタオーバーフローごとにデクリメントされます。

繰返しカウンタは自動再ロードタイプです。繰返し回数は、TIMx_RCR レジスタの値によって定義されたとおりに維持されます (図 564 を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰返しカウンタの値にかかわらず直ちにイベントが発生し、繰返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

図 564. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



42.4.6 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

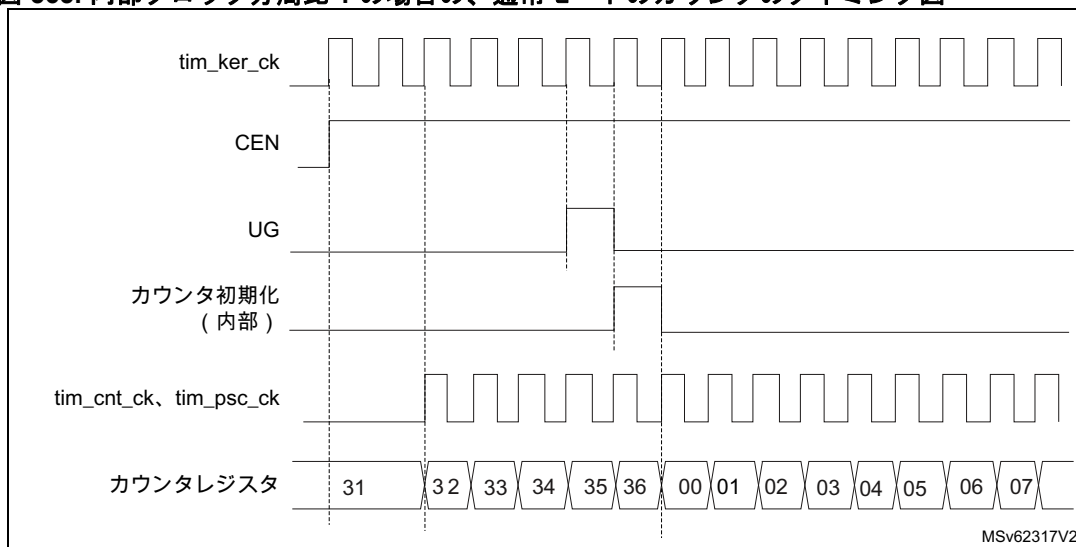
- 内部クロック (tim_ker_ck)
- 外部クロックモード 1：外部入力ピン（使用可能な場合、tim_ti1 または tim_ti2）
- 内部トリガ入力 (tim_itrx) (TIM15 の場合のみ)：あるタイマを別のタイマのプリスケアラとして使用します。たとえば、TIM1 が TIM15 のプリスケアラとして機能するように設定できます。詳細については、[タイマを使用して別のタイマを有効にする](#)を参照してください。

内部クロックソース (tim_ker_ck)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれるとすぐに、プリスケアラには内部クロック tim_ker_ck が供給されます。

図 565 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

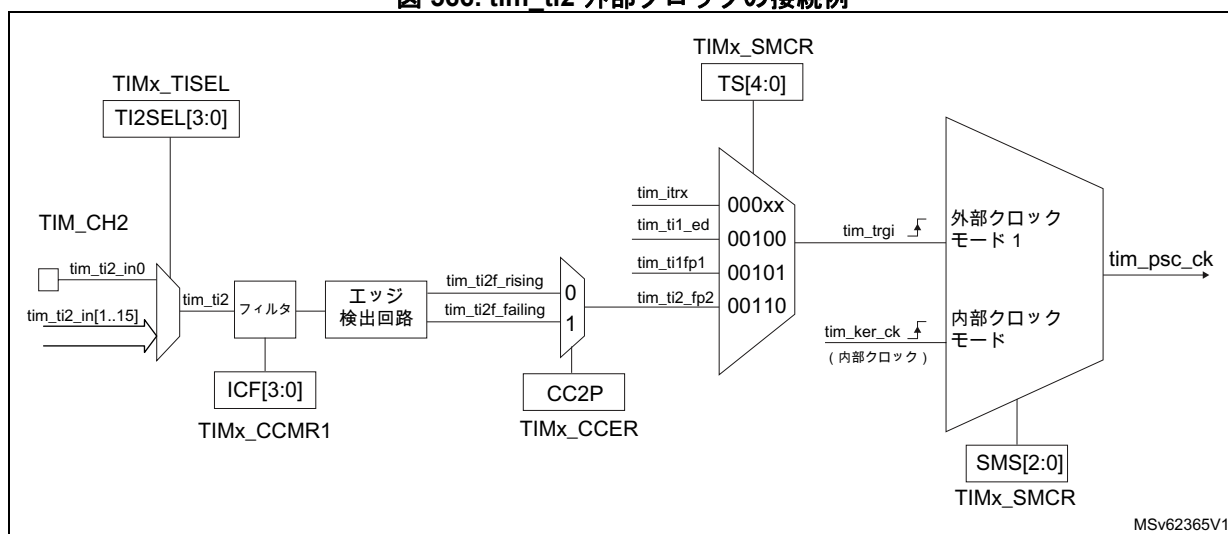
図 565. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立上がりまたは立下がりエッジでカウントすることができます。

図 566. tim ti2 外部クロックの接続例



たとえば、tim_ti2 入力の立上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:0] ソース（内部または外部）を選択します。
2. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャネル 2 が tim_ti2 入力の立上がりエッジを検出するように設定します。
3. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。

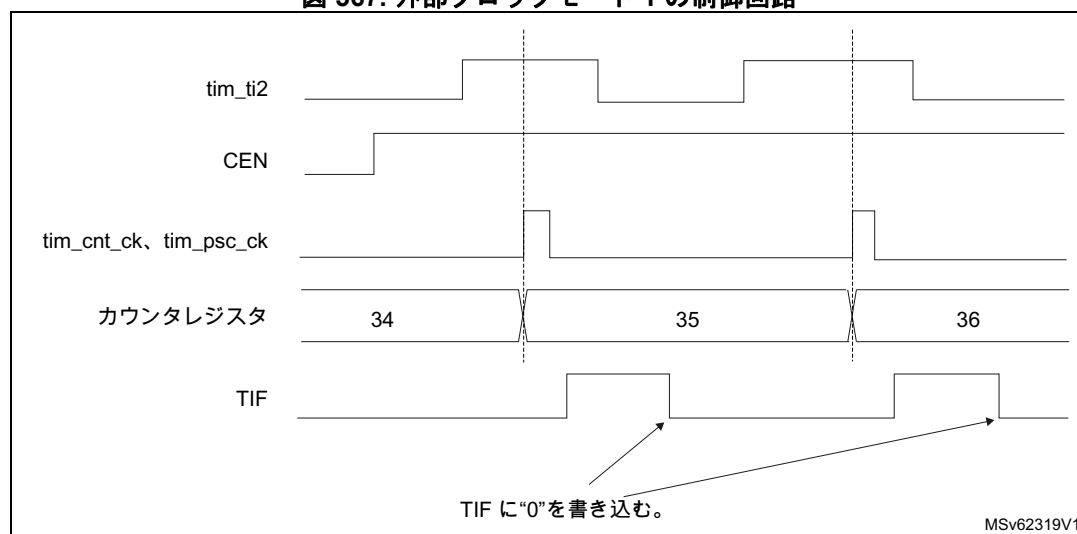
4. CC2P=0 を TIMx_CCER レジスタに書き込んで、立上がりエッジ極性を選択します。
5. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
6. TIMx_SMCR レジスタに TS=00110 を書き込むことによって、トリガ入力ソースとして tim_ti2 を選択します。*/
7. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケアラはトリガには使用されないので、設定する必要はありません。

tim_ti2 の立上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際のカウンタクロックまでの間には、tim_ti2 入力の再同期回路による遅延があります。

図 567. 外部クロックモード 1 の制御回路



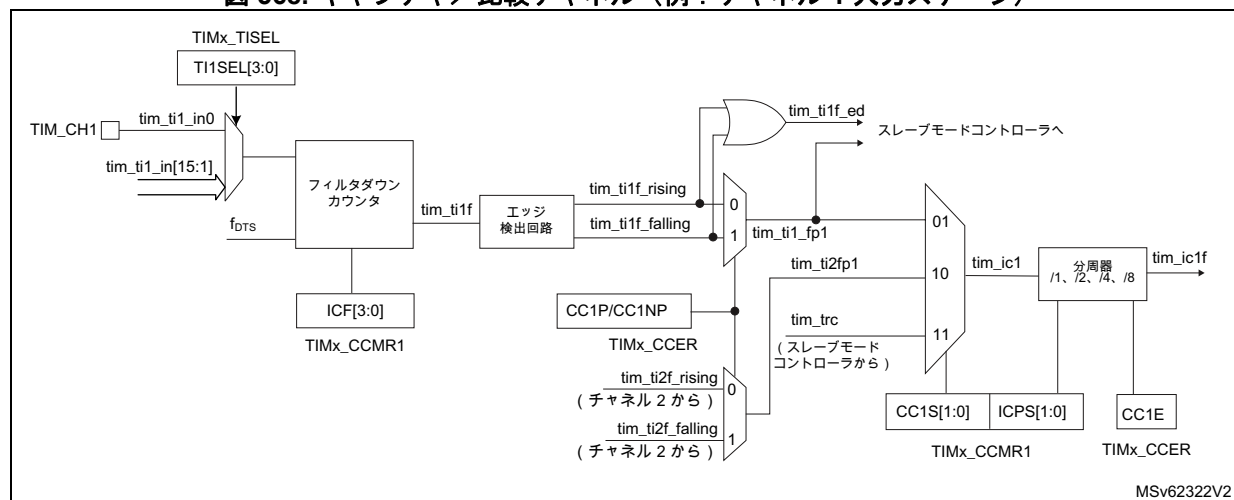
42.4.7 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入力ステージ（デジタルフィルタ、マルチプレクス、プリスケアラ）、および出力ステージ（コンパレータと出力制御）から構成されています。

図 568 から 図 571 に、1 つのキャプチャ／比較チャネルの概要を示します。

入力ステージは、対応する **tim_tix** 入力をサンプリングして、フィルタリングを行った信号 **tim_tixf** を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (**tim_tixfpy**) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 568. キャプチャ／比較チャネル (例：チャネル 1 入力ステージ)



出力ステージは、その後、基準として使用される中間波形 `tim_ocxref`（アクティブハイ）を生成します。信号の極性は最終出力に影響を与えます。

図 569. キャプチャ／比較チャネル 1 メイン回路

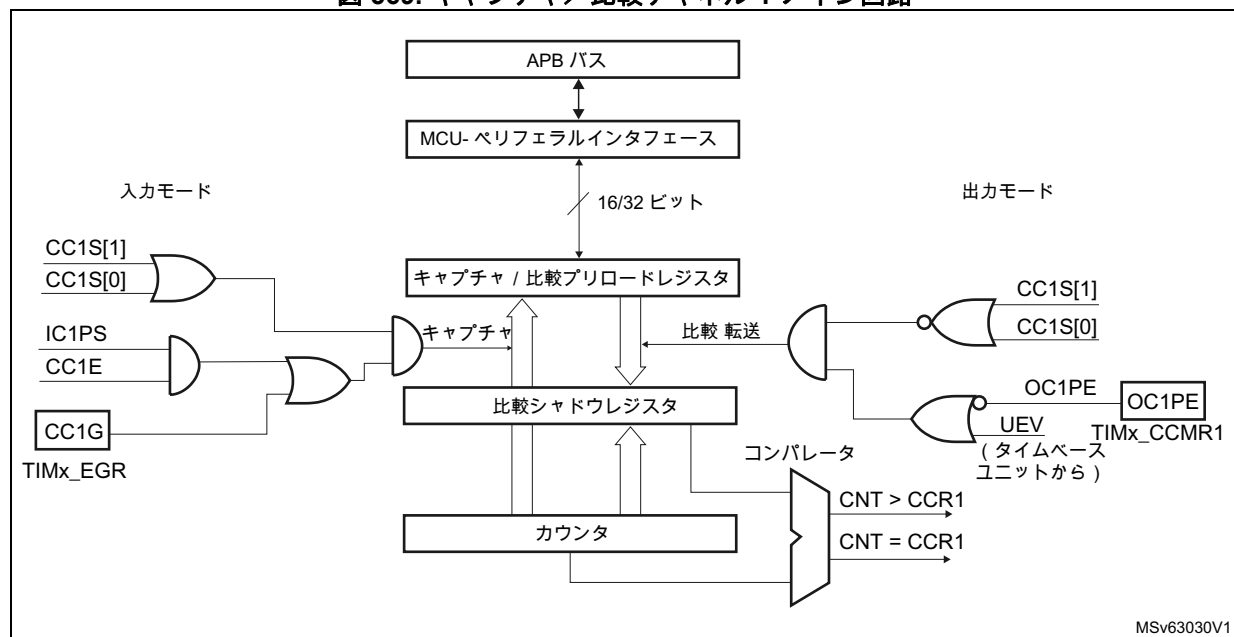


図 570. キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ

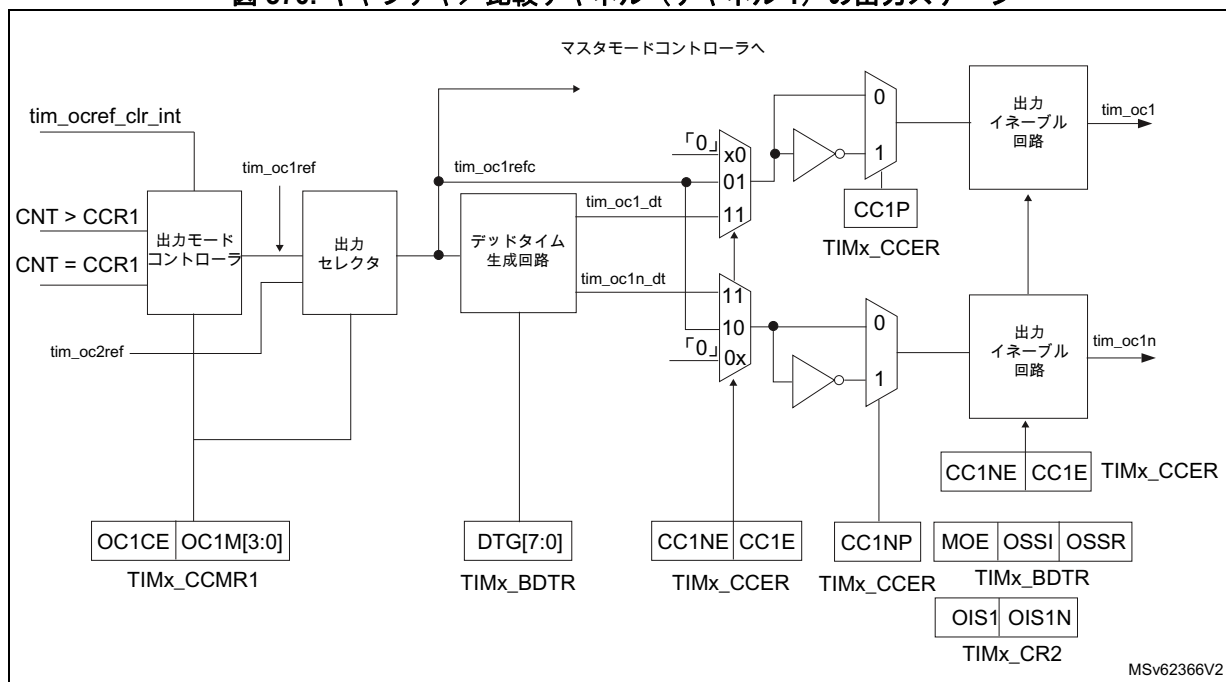
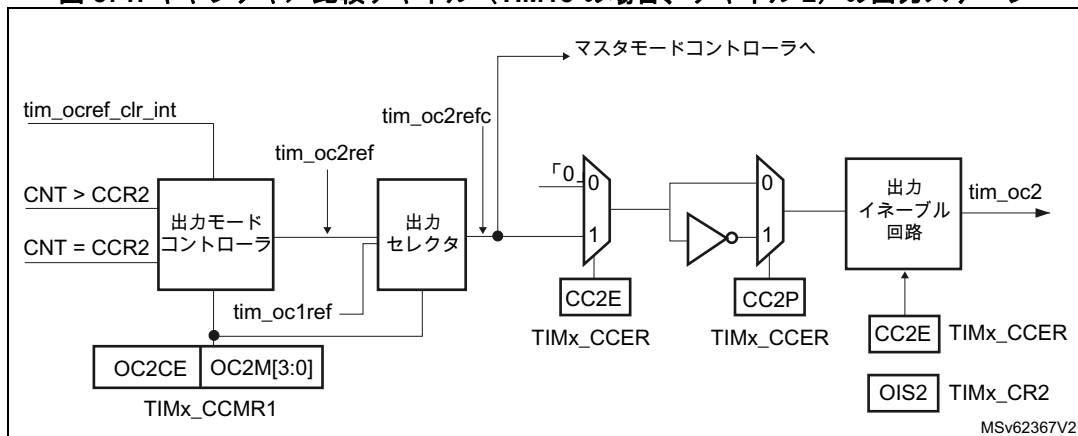


図 571. キャプチャ／比較チャンネル (TIM15 の場合、チャンネル 2) の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

42.4.8 入力キャプチャモード

入力キャプチャモードでは、対応する `tim_icx` 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (`TIMx_CCRx`) が使用されます。キャプチャが発生すると、対応する `CCxIF` フラグ (`TIMx_SR` レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。 `CCxIF` フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ `CCxOF` (`TIMx_SR` レジスタ) がセットされます。 `CCxIF` フラグは、ソフトウェアで "0" を書き込むことによって、または、 `TIMx_CCRx` レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。 `CCxOF` は、0 を書き込むとクリアされます。

次の例は、 `tim_ti1` 入力立ち上がったときに、カウンタの値を `TIMx_CCR1` にキャプチャする方法を示します。このためには、次の手順を使用します。

1. `TIMx_TISEL` レジスタの `TI1SEL[3:0]` ビットで、適切な `tim_ti1_in[15:1]` ソース (内部または外部) を選択します。
2. アクティブ入力を選択します。 `TIMx_CCR1` は `tim_ti1` 入力とリンクされていなければならないので、このためには `TIMx_CCMR1` レジスタの `CC1S` ビットに "01" を書き込みます。 `CC1S` の値が "00" から変化すると、チャンネルは入力に設定され、 `TIMx_CCR1` レジスタは読み出し専用になります。
3. タイマに接続される信号に関して、適切な入力フィルタ時間をプログラムします (入力が `tim_tix` の 1 つである場合、 `TIMx_CCMRx` レジスタの `ICxF` ビットによって)。入力信号の反転時、最低でも内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、 `tim_ti1` の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、 `TIMx_CCMR1` レジスタの `IC1F` ビットに 0011 を書き込みます。
4. `tim_ti1` チャンネルのアクティブ遷移のエッジを選択します。このためには、 `TIMx_CCER` レジスタの `CC1P` ビットに 0 を書き込みます (この場合、立上がりエッジの選択)。
5. 入力プリスケールをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケールを無効にします (`TIMx_CCMR1` レジスタの `IC1PS` ビットに "00" を書き込む)。
6. `TIMx_CCER` レジスタの `CC1E` ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
7. 必要な場合は、 `TIMx_DIER` レジスタの `CC1IE` ビットをセットすることによって、関連する割り込みリクエストを有効にするか、 `TIMx_DIER` レジスタの `CC1DE` ビットをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が `TIMx_CCR1` レジスタに格納されます。
- `CC1IF` フラグがセットされます (割り込みフラグ)。 `CC1OF` ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- `CC1IE` ビットに応じて、割り込みが生成されます。
- `CC1DE` ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これは、フラグ読み出し後、データ読み出し前に発生するかもしれないオーバキャプチャの見落としを避けるためです。

注： IC 割り込みと DMA リクエストは、 `TIMx_EGR` レジスタの対応する `CCxG` ビットをセットすることによって、ソフトウェアによって生成することができます。

42.4.9 PWM 入力モード (TIM15 の場合のみ)

このモードは、単一の `tim_tix` 入力に接続された PWM 信号の周期とデューティサイクルの両方を以下のように測定するために使用されます。

- `TIMx_CCR1` レジスタは周期値 (2 つの連続した立上がりエッジ間の間隔) を保持します。
- `TIMx_CCR2` レジスタはパルス幅 (2 つの連続した立上がりと立下がりエッジ間の間隔) を保持します。

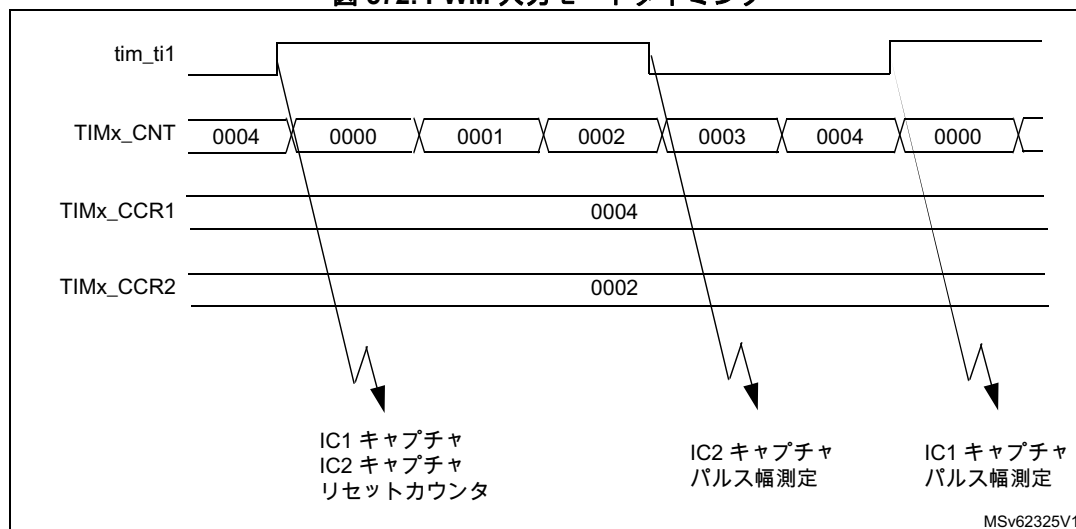
このモードは、入力キャプチャモードの特殊ケースです。設定手順は似ていますが、以下の違いがあります。

- 2 つの `tim_icx` 信号が同じ `tim_tix` 入力にマッピングされます。
- この 2 つの `tim_icx` 信号は、逆の極性のエッジでアクティブです。
- 2 つの `tim_tixfpy` 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、`tim_ti1` に適用された PWM の周期 (`TIMx_CCR1` レジスタ) とデューティサイクル (`TIMx_CCR2` レジスタ) を測定できます (手順は、`tim_ker_ck` 周波数とプリスケアラ値によって、若干異なることがあります)。

1. `TIMx_TISEL` レジスタの `TI1SEL[3:0]` ビットで、適切な `tim_ti1_in[15:0]` ソース (内部または外部) を選択します。
2. `TIMx_CCMR1` レジスタの `CC1S` ビットに "01" を書き込むことによって (`tim_ti1` を選択)、`TIMx_CCR1` のアクティブ入力を選択します。
3. `CC1P` ビットと `CC1NP` ビットに "0" を書き込むことによって (立上がりエッジでアクティブ)、`tim_ti1fp1` のアクティブな極性を選択します (`TIMx_CCR1` のキャプチャとカウンタクリアの両方に使用)。
4. `TIMx_CCMR1` レジスタの `CC2S` ビットに "10" を書き込むことによって (`tim_ti1` を選択)、`TIMx_CCR2` のアクティブ入力を選択します。
5. `CC2P` ビットと `CC2NP` ビットに "10" を書き込むことによって (立下がりエッジでアクティブ)、`tim_ti1fp2` のアクティブ極性を選択します (`TIMx_CCR2` のキャプチャに使用されます)。
6. `TIMx_SMCR` レジスタの `TS` ビットに 00101 を書き込むことによって (`tim_ti1fp1` を選択)、有効なトリガ入力を選択します。
7. `TIMx_SMCR` レジスタの `SMS` ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
8. `TIMx_CCER` レジスタの `CC1E` と `CC2E` ビットに "1" を書き込むことによって、キャプチャを有効にします。

図 572. PWM 入力モードタイミング



1. `tim_ti1fp1` と `tim_ti2fp2` のみがスレーブモードコントローラに接続されているので、PWM 入力モードは `TIMx_CH1/TIMx_CH2`信号でのみ使用できます。

42.4.10 強制出力モード

このモード (`TIMx_CCMRx` レジスタの `CCxS` ビット = 00) では、各出力比較信号 (`tim_ocxref`、そして `tim_ocx/tim_ocxn`) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (`tim_ocxref/tim_ocx`) を強制的にアクティブレベルにするには、対応する `TIMx_CCMRx` レジスタの `OCxM` ビットに 101 を書き込む必要があります。これにより、`tim_ocxref` は強制的にハイレベルになり (`tim_ocxref` は常にアクティブハイ)、`tim_ocx` は `CCxP` 極性ビットと逆の値になります。

例: `CCxP=0` (`tim_ocx` アクティブハイ) => `tim_ocx` は強制的にハイレベルになります。

`tim_ocxref` 信号は、`TIMx_CCMRx` レジスタの `OCxM` ビットに 100 を書き込むことによって、強制的にローにできます。

いずれにしても、`TIMx_CCRx` シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

42.4.11 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (`TIMx_CCMRx` レジスタの `OCxM` ビット) と出力極性 (`TIMx_CCER` レジスタの `CCxP` ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (`OCxM=000`)、アクティブにセットされるか (`OCxM=001`)、非アクティブにセットされるか (`OCxM=010`)、または反転されます (`OCxM=011`)。
- 割込みステータスレジスタのフラグをセットします (`TIMx_SR` レジスタの `CCxIF` ビット)。
- 対応する割込みマスク (`TIMx_DIER` レジスタの `CCXIE` ビット) がセットされている場合は、割込みを生成します。

- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信しません (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

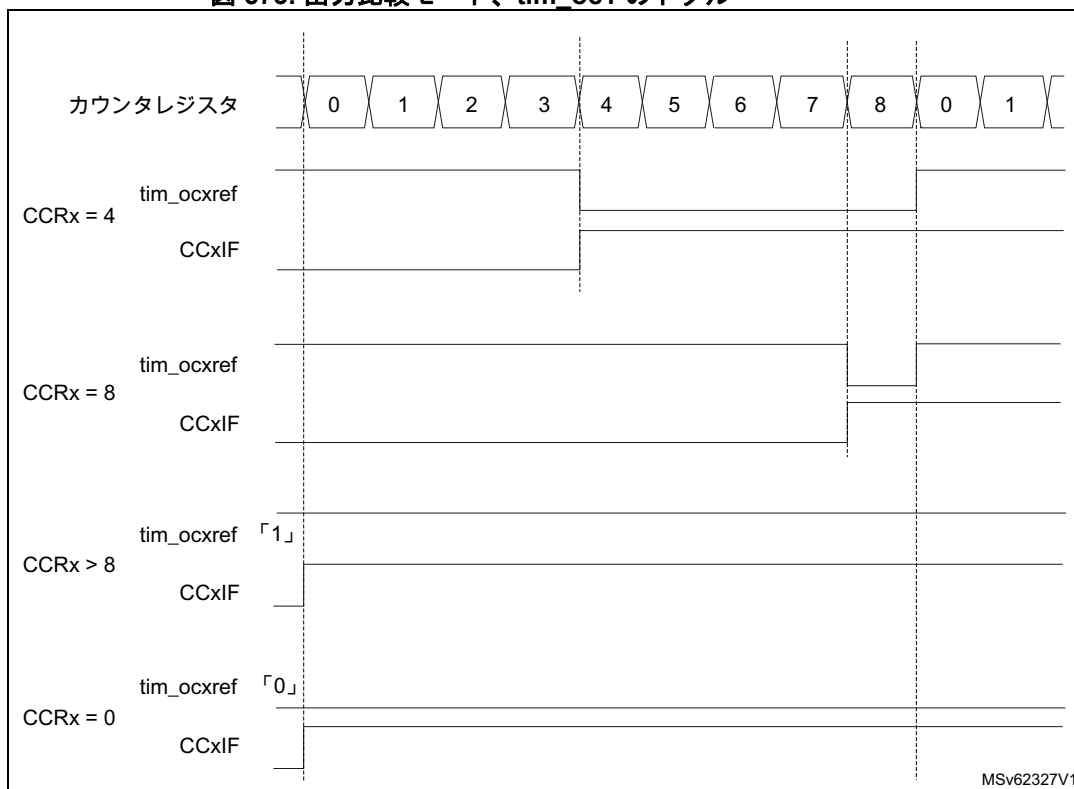
出力比較モードでは、更新イベント UEV は tim_ocxref および tim_ocx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
 - CNT と CCRx が一致したときに tim_ocx 出力ピンを反転するには、OCxM ビットに 011 を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 573](#) に示します。

図 573. 出力比較モード、tim_oc1 のトグル



42.4.12 PWM モード

パルス幅変調 (PWM) モードは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルの信号を生成するために使用されます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (tim_ocx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります、また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

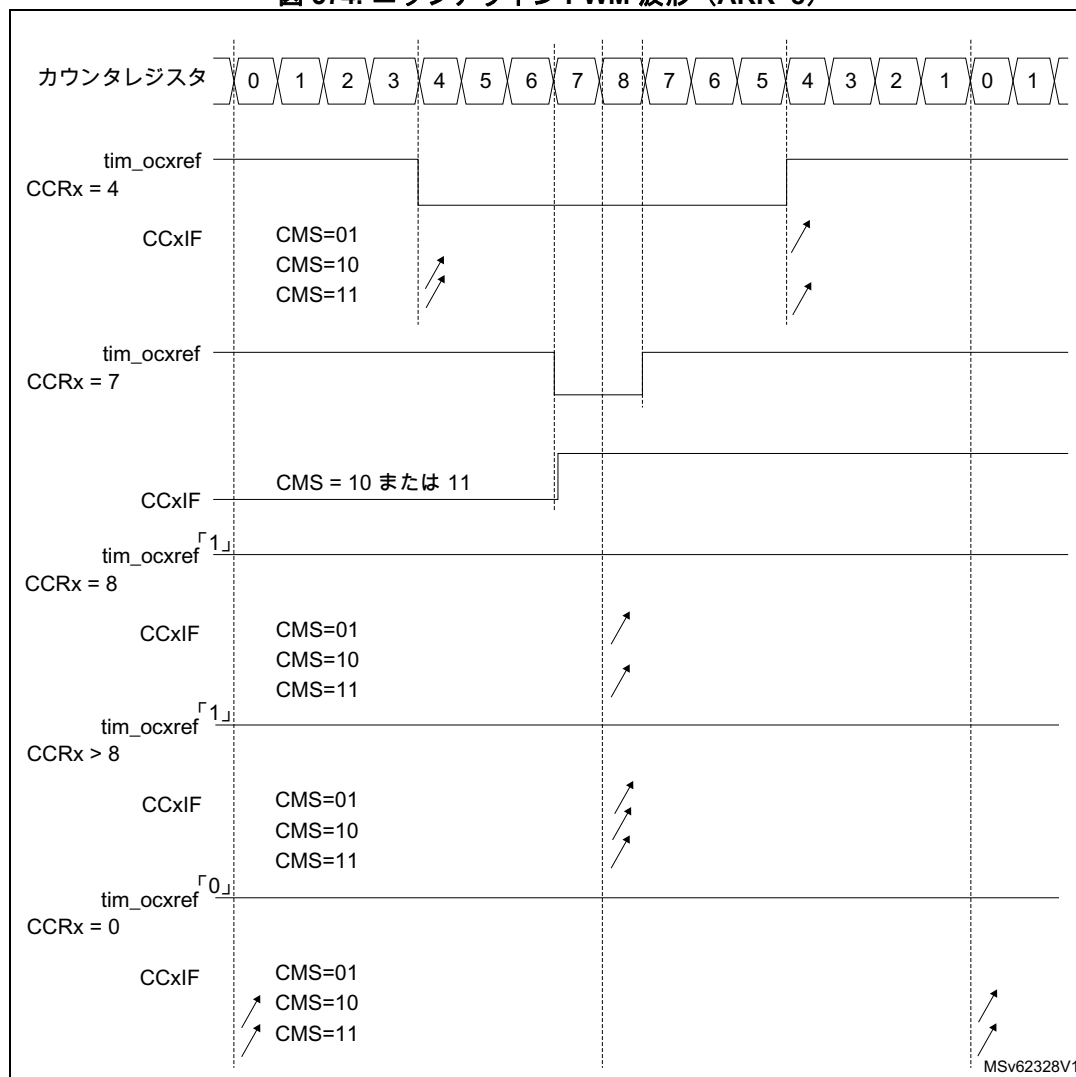
tim_ocx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。これは、アクティブハイまたはアクティブラーとしてプログラムできます。tim_ocx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCER レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx かどうか判断されます (カウントの方向によります)。

TIM15/TIM16/TIM17 はアップカウント動作でのみ使用可能です。1753ページのアップカウントモードを参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 `tim_ocxref` は、`TIMx_CNT < TIMx_CCRx` の間はハイに、そうでない場合はローになります。`TIMx_CCRx` の比較値が `TIMx_ARR` の自動再ロード値より大きい場合、`tim_ocxref` は“1”に保持されます。比較値が 0 の場合、`tim_ocxref` は“0”に保持されます。図 574 に `TIMx_ARR=8` のときのエッジアライン PWM 波形の例を示します。

図 574. エッジアライン PWM 波形 (ARR=8)

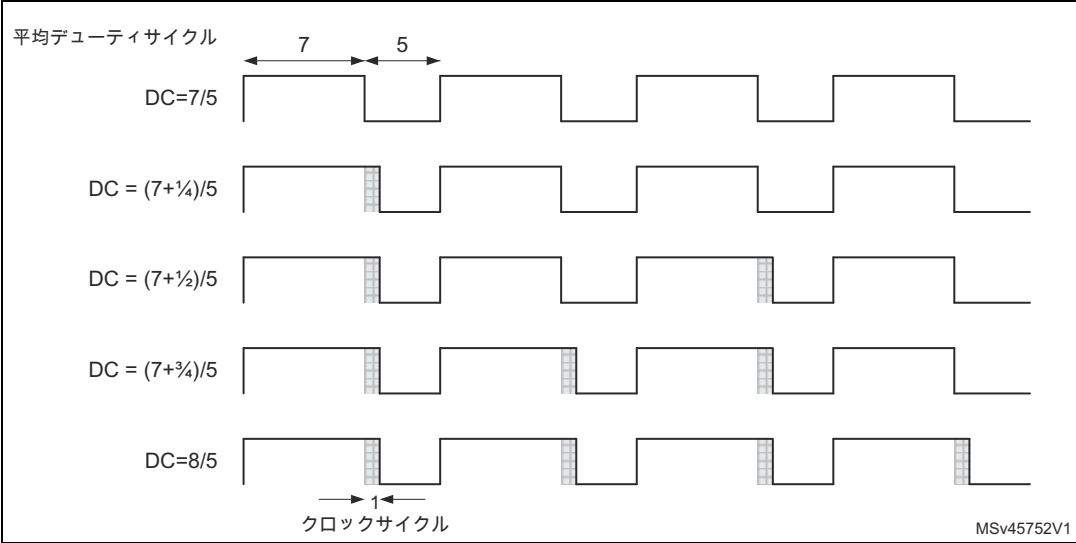


ディザリングモード

PWM モードの有効分解能は、`TIMx_CR1` レジスタの `DITHEN` ビットを使用して、ディザリングモードを有効にすることで増加できます。これは、`CCR`（デューティサイクル分解能の増加）と `ARR`（PWM 周波数分解能の増加）の両方に適用されます。

動作原理は、事前に定義されたパターンで、16 個の連続した PWM 周期にわたって実際の `CCR`（または `ARR`）値をわずかに変更する（1 タイマクロック周期を追加する、またはしない）ことです。これにより、平均のデューティサイクルまたは PWM 周期を考えると、16 倍の分解能の向上が可能になります。下記の図 575 は、4 つの連続する PWM サイクルに適用されたディザリング原理を示します。

図 575. ディザリング原理

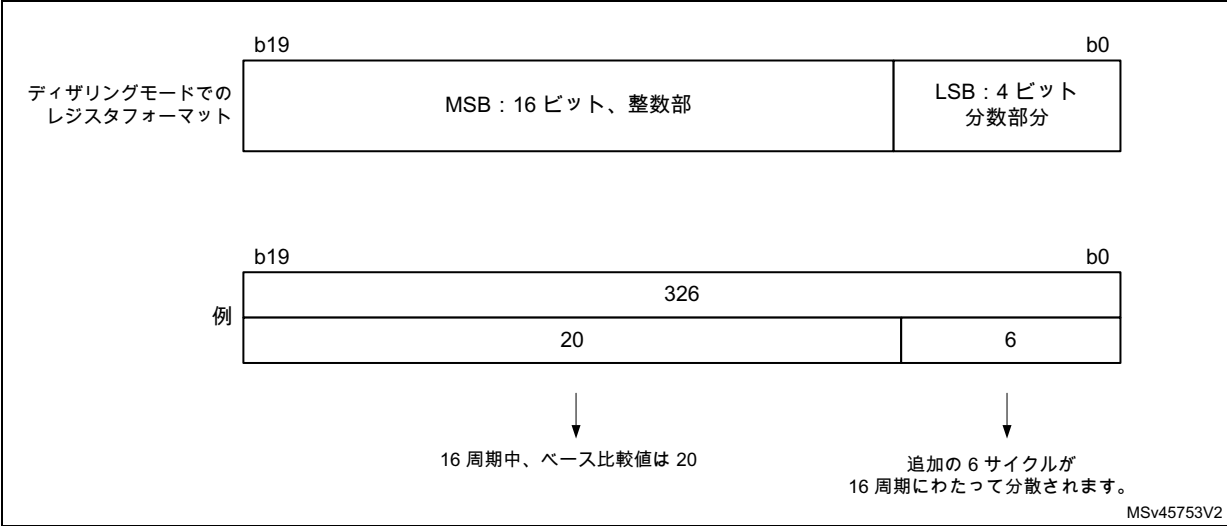


ディザリングモードが有効になっている場合、レジスタのコーディングは以下のように変更されます (たとえば、図 576 を参照)。

- 下位 4 ビットは、拡張分解能部分 (分数部分) 用です。
- 上位ビットは、ビット 19:4 に左シフトされ、ベース値用のコーディングになります。

注 : DITHEN ビットをリセットするときは、次の手順に従う必要があります。
 1を参照してください。CEN および ARPE ビットをリセットする必要があります。
 2を参照してください。ARR[3:0] ビットをリセットする必要があります。
 3を参照してください。CCIF フラグをクリアする必要があります。
 4を参照してください。CEN ビットをセットすることができます (最終的に ARPE = 1 とともに)。

図 576. ディザリングモードでのデータフォーマットとレジスタのコーディング



最小周波数は次の計算式で与えられます。

$$\text{分解能} = \frac{F_{\text{Tim}}}{F_{\text{pwm}}} \Rightarrow F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{\text{Max}_{\text{Resolution}}}$$

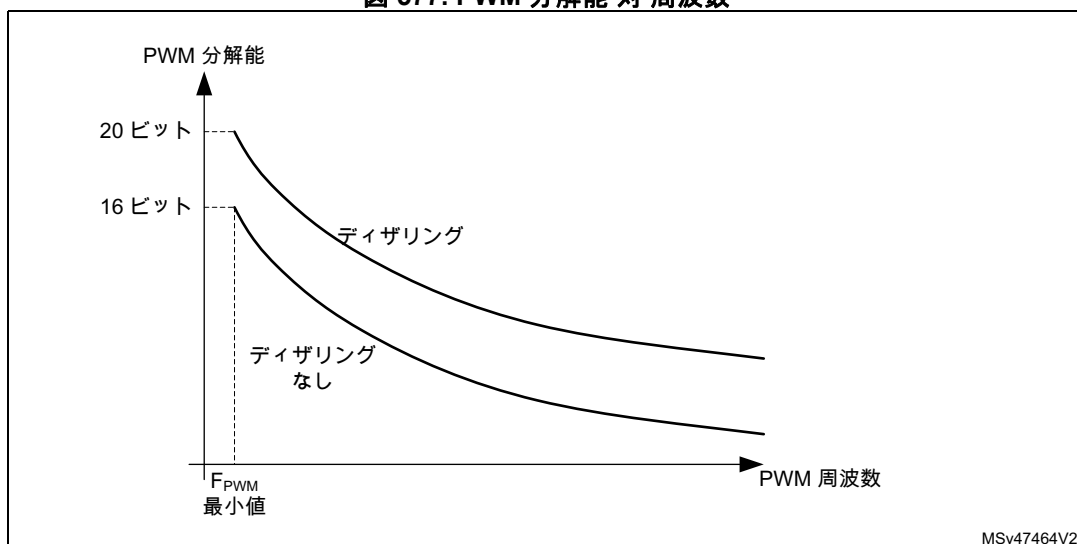
$$\text{ディザリングモード無効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65536}$$

$$\text{ディザリングモード有効時: } F_{\text{pwmMin}} = \frac{F_{\text{Tim}}}{65535 + \frac{15}{16}}$$

注: TIMx_ARR および TIMxCCRy の最大値は、ディザリングモードでは 0xFFFFF に制限されています (整数部では 65534、ディザリング部分では 15 に対応します)。

下記の図 577 に示すように、ディザリングモードは PWM 周波数に関係なく、PWM 分解能を増加するために使用されます。

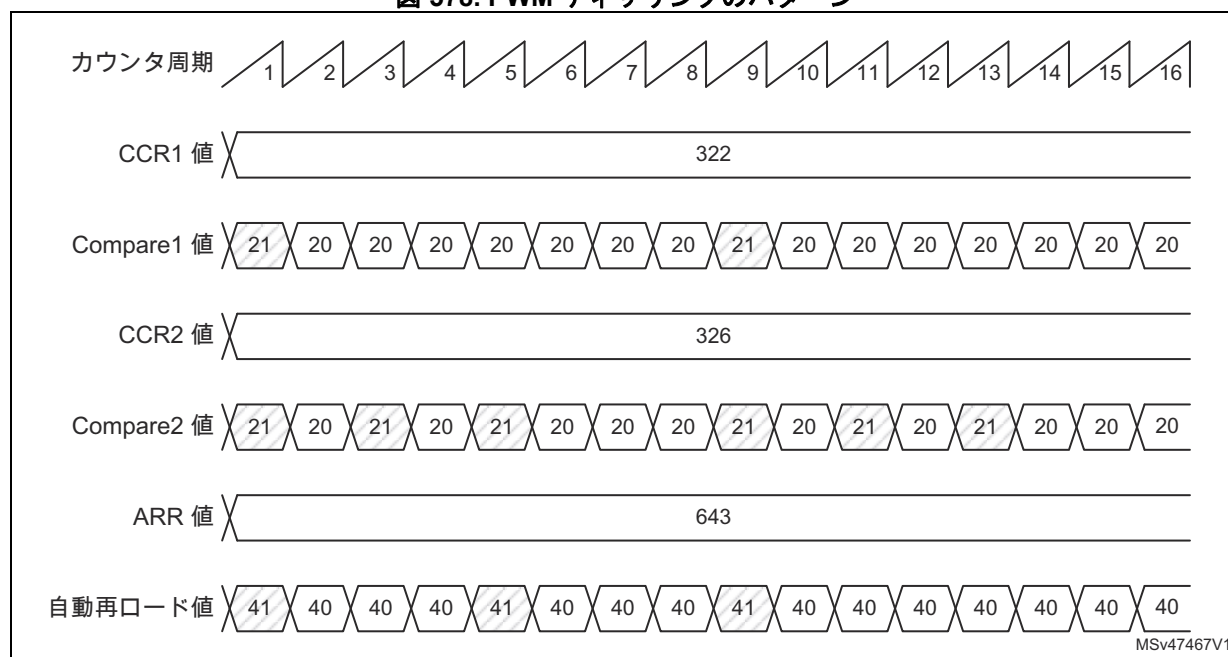
図 577. PWM 分解能 対 周波数



MSv47464V2

下記の図 578 で説明するように、デューティサイクルや周期の変更は、連続した 16 周期にわたって広がります。

図 578. PWM ディザリングのパターン



自動再ロード値と比較値の増分は、下記の表 440 に記載された特定のパターンに従って分布します。ディザリングシーケンスは、増分が可能な限り均等に分散され、全体のリップルが最小になるように行われます。

表 440. CCR および ARR レジスタの変更のディザリングパターン

-	PWM 周期															
LSB 値	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-

表 440. CCR および ARR レジスタの変更のディザリングパターン (続き)

-	PWM 周期															
LSB 値	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

42.4.13 組合せ PWM モード (TIM15 のみ)

組み合わせ PWM モードでは、2 つのエッジアラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2 つの TIMx_CCRx レジスタで決定されます。結果として得られる信号 tim_ocxrefc は、2 つの PWM 基準信号の OR または AND による論理結合から成ります。

- tim_oc1refc (または tim_oc2refc) は、TIMx_CCR1 および TIMx_CCR2 レジスタによって制御されます。

組み合わせ PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“1100”(組み合わせ PWM モード 1) または“1101”(組み合わせ PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの tim_ocx 出力)。

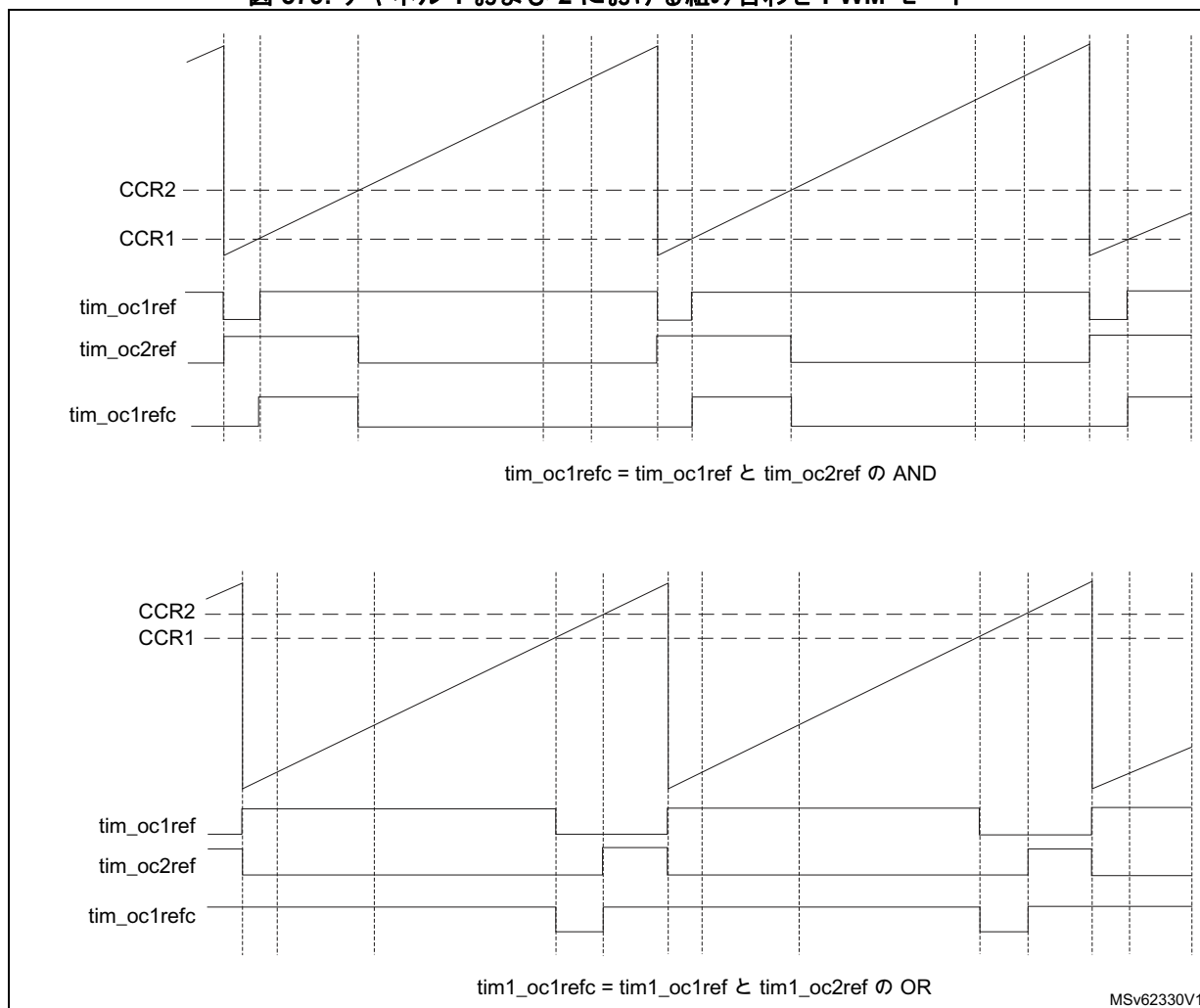
特定のチャネルが組み合わせ PWM チャネルとして使用されている場合、相補チャネルを反対の PWM モードに設定する必要があります (たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします)。

注： OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

[図 579](#) は、次の設定で取得可能な組み合わせ PWM モードを使用して生成される信号の例を表します。

- チャネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャネル 2 が PWM モード 1 で設定されている場合

図 579. チャンネル 1 および 2 における組み合わせ PWM モード



42.4.14 相補出力とデッドタイム挿入

TIM15/TIM16/TIM17 汎用タイマは、1 つの相補信号を出力して、出力時のスイッチオフおよびスイッチオンを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性（レベルシフタの内在的な遅延、電源スイッチによる遅延など）に応じて調整する必要があります。

出力（主出力 tim_ocx または相補 tim_ocxn）の極性は出力ごとに独自に選択できます。これは TIMx_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 tim_ocx および tim_ocxn は、TIMx_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx_BDTR レジスタと TIMx_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、表 447 : 1834 ページのブレーク機能を持つ相補 tim_oc1 および tim_oc1n チャンネルの出力制御ビット (TIM16/TIM17) を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに、デッドタイムが挿入されます。

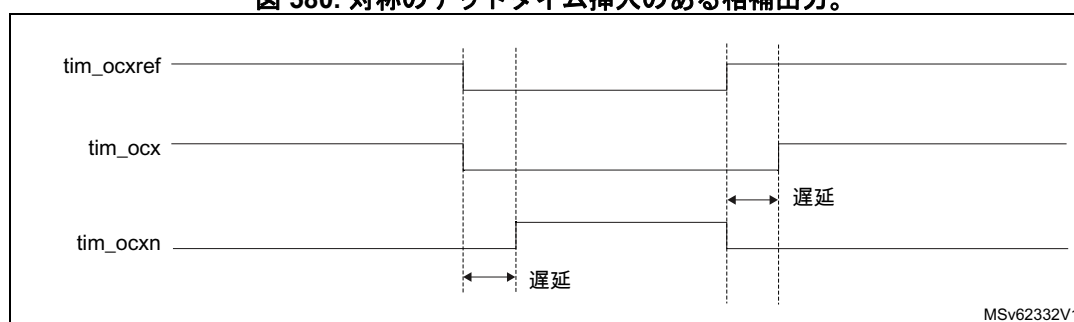
デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッドタイムジェネレータがあります。この回路は、基準波形 `tim_ocxref` から `tim_ocx` と `tim_ocxn` の 2 つの出力を生成します。`tim_ocx` と `tim_ocxn` がアクティブハイの場合、

- `tim_ocx` 出力信号は基準信号と同じですが、立上がりエッジが基準の立上がりエッジより遅い点が異なります。
- `tim_ocxn` 出力信号は、立上がりエッジが基準波形の立下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (`tim_ocx` または `tim_ocxn`) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 `tim_ocxref` との関係を示します。(これらの例では、`CCxP=0`、`CCxNP=0`、`MOE=1`、`CCxE=1`、および `CCxNE=1` を想定しています。)

図 580. 対称のデッドタイム挿入のある相補出力。



TIMx_DTR2 の DTAE ビットは、図 581 に示すように、デッドタイム値を、基準信号の立上がりエッジと立下がりエッジに対して変えるために使用されます。

非対称モード (DTAE = 1) では、立上がりエッジ基準デッドタイムは TIMx_BDTR レジスタの DTG[7:0] ビットフィールドによって定義され、立下がりエッジ基準は TIMx_DTR2 レジスタの DTGF[7:0] ビットフィールドによって定義されます。DTAE ビットは、カウンタを有効にする前に書き込む必要があり、CEN=1 の間には変更してはなりません。

プリロードメカニズムを使用して、PWM 運転中にデッドタイム値を動作中に更新することができます。デッドタイムビットフィールド DTG[7:0] と DTGF[7:0] は、TIMX_DTR2 レジスタで DTPE ビットがセットされるとプリロードされます。プリロード値は、次の更新イベントでアクティブレジスタにロードされます。

注： カウンタが有効なときに DTPE ビットを有効にした場合、最後の更新以降に書き込まれた新しい値は破棄され、以前の値が使用されます。

図 581. 非対称のデッドタイム

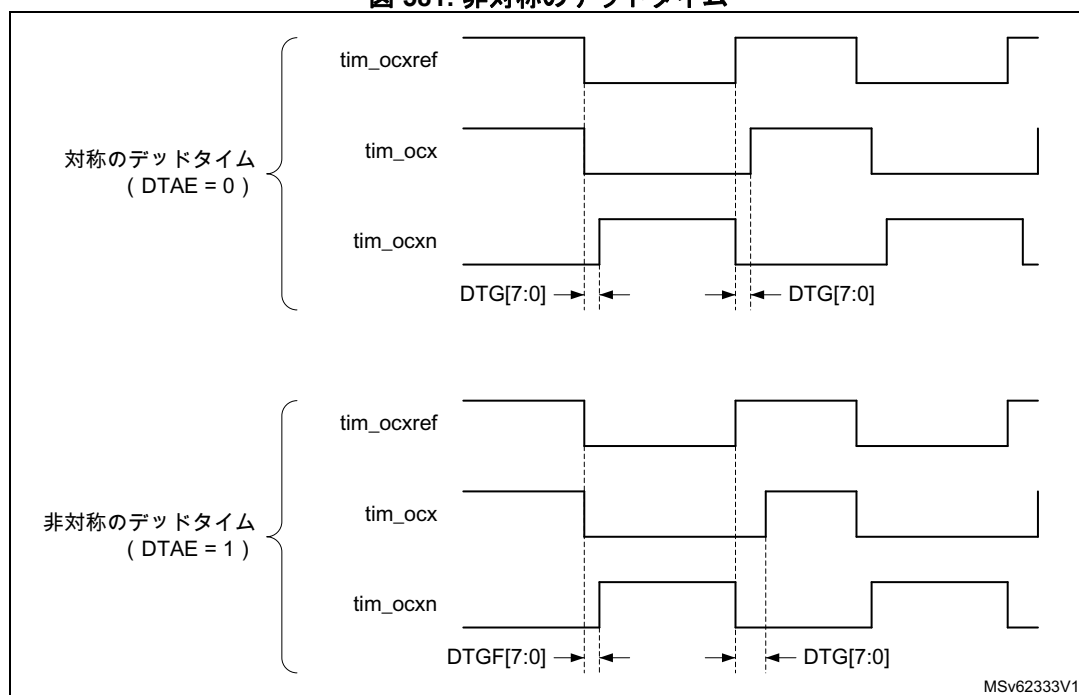


図 582. 負のパルスより長い遅延があるときのデッドタイムの波形

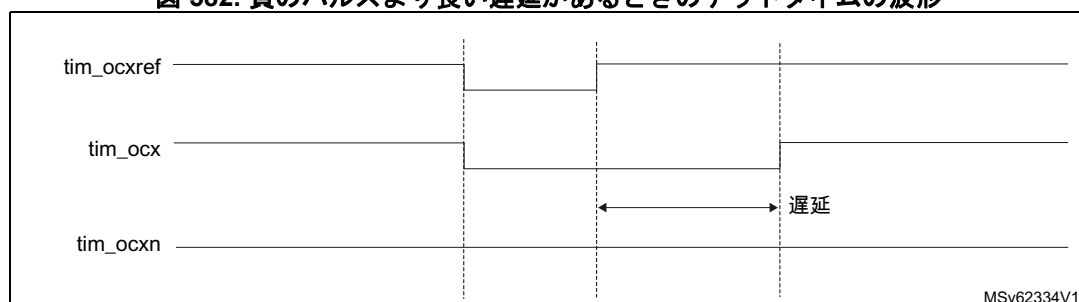
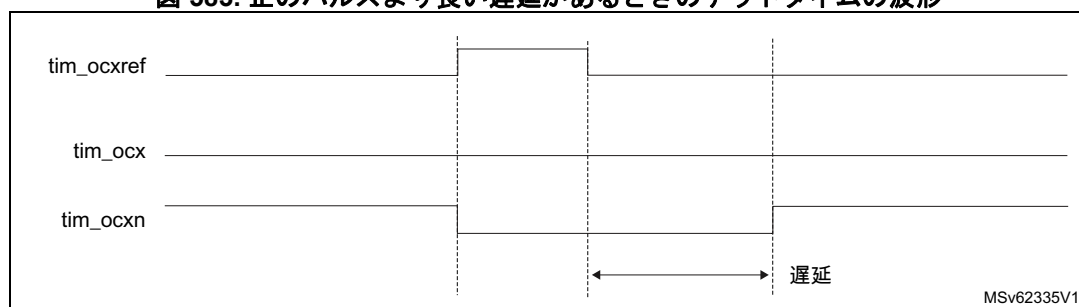


図 583. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx_BDTR レジスタの DTG ビットでプログラム可能です。遅延計算については、[セクション 42.8.14 : 1838 ページのTIMx ブレークおよびデッドタイムレジスタ \(TIMx_BDTR\) \(x = 16~17\)](#) を参照してください。

tim_ocxref 信号の tim_ocx または tim_ocxn へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx_CCER レジスタの CCxE ビットおよび CCxNE ビットを設定することによって、tim_ocxref 信号を tim_ocx 出力または tim_ocxn 出力にリダイレクトできます。

これは、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、相補信号をインアクティブレベルに固定するために使用されます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができ

注： tim_ocxn のみが有効なときには（CCxE=0、CCxNE=1）、相補にならず、tim_ocxref がハイレベルになるとすぐにアクティブになります。たとえば、CCxNP=0 の場合は、tim_ocxn=tim_ocxref になります。他方、tim_ocx と tim_ocxn の両方が有効なときには（CCxE=CCxNE=1）、tim_ocxref がハイになると tim_ocx はアクティブになり、tim_ocxn は相補信号となり、tim_ocxref がローのときにアクティブになります。

42.4.15 ブレーク機能の使用

ブレーク機能の目的は、タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。ブレーク入力通常、パワーステージおよび 3 相インバータの異常出力に接続されています。アクティブ化すると、ブレーク回路は PWM 出力を遮断し、強制的に事前定義された安全な状態に移行させます。

このブレークチャネルは、システムレベル障害（クロック障害、ECC/パリティエラーなど）とアプリケーション障害（入力ピンおよび内蔵比較回路からの障害）の両方を集め、出力をデッドタイムの持続時間経過後に事前定義されたレベル（アクティブまたはインアクティブ）に強制できます。

ブレーク時の出力有効信号および出力レベルは、いくつかの制御ビットに依存しています。

- TIMx_BDTR レジスタの MOE ビット。ソフトウェアによって出力を有効/無効にするために使用され、ブレーク または ブレーク 2 イベント時にリセットされます。
- TIMx_BDTR レジスタの OSSR ビット。出力をインアクティブ状態で制御するか、GPIO コントローラへの制御を解除するかについて、タイマを定義します（通常、ハイインピーダンスモードにするため）。
- TIMx_CR2 レジスタの OISx および OISxN ビット。アクティブまたはインアクティブな出力遮断レベルをセットします。OISx および OISxN の値にかかわらず、一度に tim_ocx 出力と tim_ocxn 出力を両方ともアクティブレベルにセットすることはできません。詳細については、[表 447：1834 ページのブレーク機能を持つ相補 tim_oc1 および tim_oc1n チャンネルの出力制御ビット \(TIM16/TIM17\)](#) を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。ブレーク機能は、TIMx_BDTR レジスタの BKE ビットをセットすることによって有効になります。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立下がりエッジは非同期のことがあるので、実際の信号（出力に作用する信号）と同期制御ビット（TIMx_BDTR レジスタからアクセスできる）の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 に設定された場合、MOE を正しく読み出すためには、遅延（ダミー命令）を挿入する必要があります。これは、書き込みは非同期信号に対応しますが、読出しは同期信号を反映するからです。

ブレークは、次を持つ tim_brk 入力によって生成されます。

- プログラム可能な極性 (TIMx_BDTR レジスタの BKP ビット)
- プログラム可能なイネーブルビット (TIMx_BDTR レジスタの BKE ビット)
- 誤ったイベントを避けるためのプログラム可能なフィルタ (TIMx_BDTR レジスタの BKF[3:0] ビット)

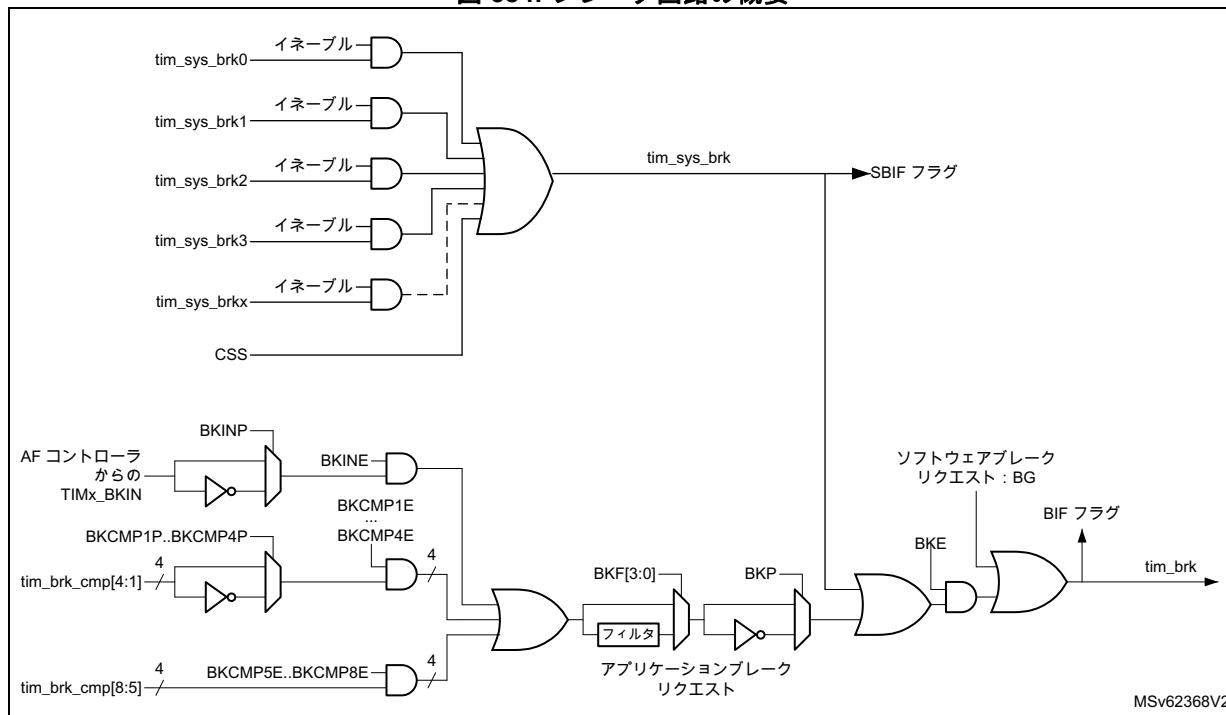
ブレークは、TIMx_AF1 レジスタを使用して、個別に有効化でき、プログラム可能なエッジ検知を持つ複数のソースから生成できます。

ブレーク (tim_brk) チャンネルのソースは以下のいずれかです。

- (GPIO オルタネート機能選択レジスタでの選択に従って) TIM_BKIN ピンの 1 つに接続された外部ソースで、極性選択およびオプションのデジタルフィルタリングあり
- 内部ソース :
 - tim_brk_cmpx 入力から着信するもの (製品固有の実装については [セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#) を参照)
 - システムブレークリクエストから tim_sys_brk 入力へ着信するもの (製品固有の実装については [セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#) を参照)

ブレークイベントは、TIMx_EGR レジスタで BG ビットを使用して、ソフトウェアによって生成することもできます。以下の [図 584](#) のとおり、すべてのソースはタイマの tim_brk 入力に入る前に論理和がとられます。

図 584. ブレーク回路の概要



注意 : 非同期 (クロックなし) 動作は、プログラム可能なフィルタが無効な場合にのみ保証されます。有効になっている場合は、必ずブレークイベントが処理されるように、フェイルセーフクロックモード (たとえば、内部 PLL や CSS を使用) を使用する必要があります。

ブレークが発生すると（ブレーク入力で選択されたレベル）、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態またはアイドル状態になるか、GPIO への制御が解除されます（OSSI ビットで選択）。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャネルは、MOE=0 になったとき、TIMx_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマは出力の制御（GPIO によって引き継がれた）を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
 - 出力は、まずリセット状態のインアクティブ状態に置かれます（極性に依存します）。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、tim_ocx と tim_ocxn を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください（約 2 tim_ker_ck クロックサイクル）。
 - OSSI=0 の場合、タイマはイネーブル出力（ハイインピーダンス状態を強制する GPIO によって引き継がれた）を解除し、そうでない場合、イネーブル出力はハイのままか、CCxE または CCxNE ビットのどちらかがハイになったときにハイになります。
- ブレーク状態フラグ（TIMx_SR レジスタの BIF ビット）がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合は、割込みを生成できます。TIMx_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント UEV で再び自動的にセットされます。これを使用して、たとえば、レギュレーションを行うことができます。そうでない場合、MOE は、再び 1 が書き込まれるまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

注： AOE ビットが設定されている状態で CPU により MOE がリセットされると、出力はアイドル状態になり、OSSI 値に応じてインアクティブレベルまたはハイインピーダンスに強制されます。MOE ビットと AOE ビットの両方が CPU によってリセットされた場合、出力は無効状態になり、TIMx_CR2 レジスタの OISx ビットでプログラムされたレベルで駆動されます。

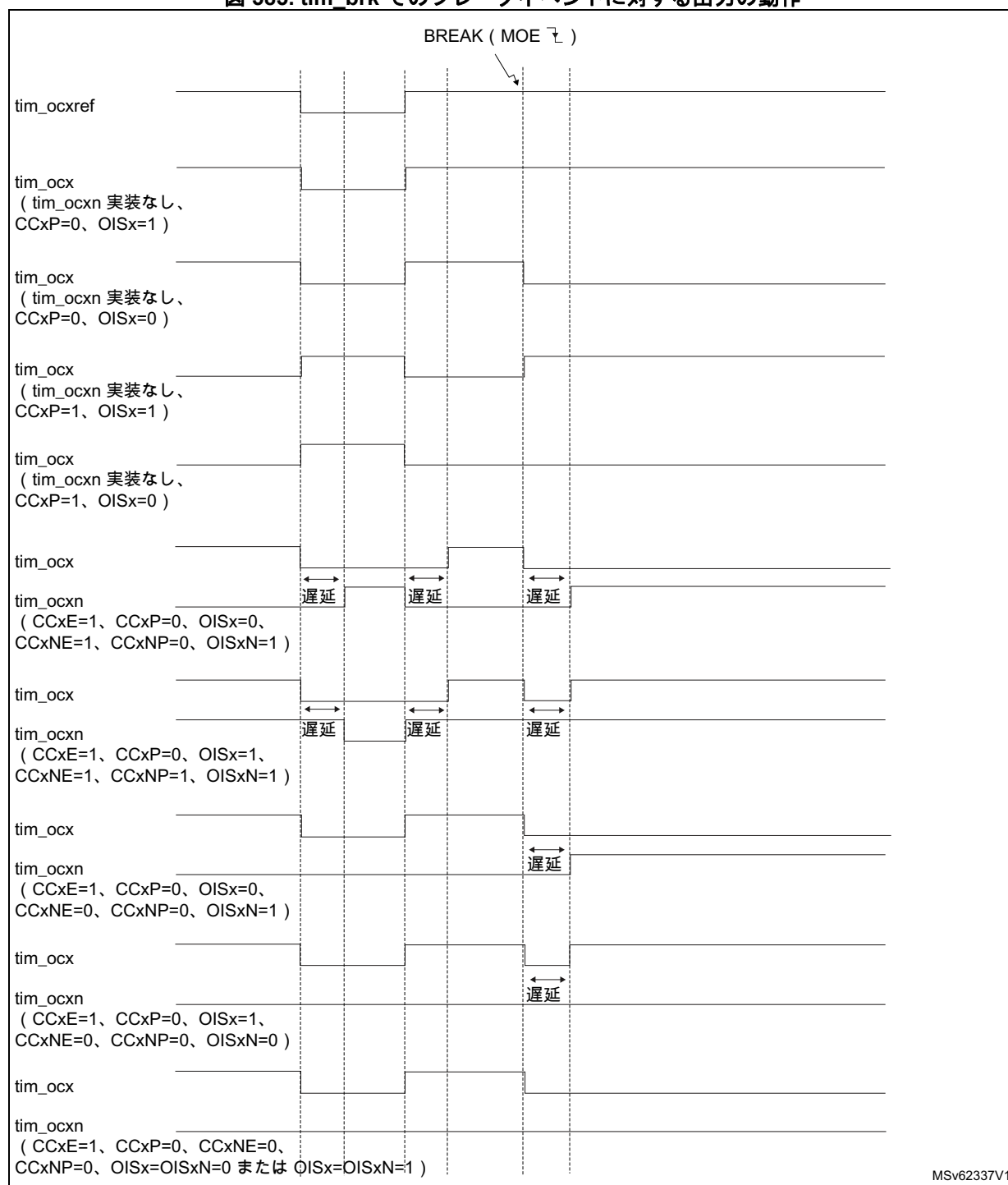
注： ブレーク入力は、信号レベルに対して動作します。このため、ブレーク入力がアクティブな間は、MOE をセットできません（自動的に、ソフトウェアによっても）。この間、ステータスフラグ BIF をクリアできません。

ブレークは、tim_brk 入力によって生成でき、tim_brk はプログラミング可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書込み保護機能を設けてあります。これは、いくつかのパラメータ（デッドタイムの長さ、tim_ocx/tim_ocxn 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性）の設定を固定するために使用されます。TIMx_BDTR レジスタの LOCK ビットによって、保護を 3 つのレベルから選択することができます。セクション 42.8.14：TIMx ブレークおよびデッドタイムレジスタ (TIMx_BDTR) (x = 16~17) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。

図 585 に、ブレークに対する出力の動作例を示します。

図 585. tim_brk でのブレーキイベントに対する出力の動作



42.4.16 双方向ブレーク入力

TIM15/TIM16/TIM17 には、図 586 に示すとおり、双方向ブレーク I/O が搭載されています。

それらは下記のために使用されます。

- 入力および出力ステータスピンドちらにもなる単一のピンによる、外部の MCU やゲートドライバに障害を通知するために使用できる幅広いレベルのグローバルブレーク信号
- 複数の内部および外部ブレークソースを統合する必要がある場合、互いに論理和をとり、単一のブレークイベントをトリガする内部ブレークソースおよび複数の外部オープンドレインソース

tim_brk 入力は、TIMxBDTR レジスタの BKBID ビットを使用して双方向モードに設定されます。BKBID プログラミングビットは、TIMxBDTR レジスタの LOCK ビットを使用して読出し専用モードにロックできます (LOCK レベル 1 以上)。

双方向モードでは、I/O をアクティブロー極性でオープンドレインモードに設定する必要があります (BKINP および BKP ビットを使用)。システム (たとえば CSS)、内部ペリフェラル、またはブレーク入力からのブレークリクエストによって、障害イベントを通知するブレーク入力のレベルが強制的にローにされます。安全性のため、極性ビットが正しくセット (アクティブハイ極性) されていない場合の双方向モードは禁止されています。

(BG ビットのセットによってトリガされる) ブレークソフトウェアイベントも、ブレーク I/O を強制的に 0 にして、タイマがブレーク状態に移行したことを外部コンポーネントに示します。ただし、これはブレークが有効になっている場合 (BKE = 1) にのみ有効です。ソフトウェアブレークイベントが BKE = 0 で生成された場合、出力は安全な状態になり、ブレークフラグはセットされますが、TIM_BKIN I/O には効果がありません。

安全解除メカニズムは、システムが完全にロックされることを防ぎます (ブレーク入力でレベルがローになると、同じ入力でレベルをローに強制するブレークがトリガされます)。

BKDSRM ビットが 1 にセットされると、ブレーク出力が解放され、障害信号をクリアしてシステムを再設定できます。

ブレーク保護回路はどの時点においても無効にできません。

- ブレーク入力パスは常にアクティブです。BKDSRM ビットがセットされており、オープンドレイン制御が解放されている場合でもブレークイベントはアクティブです。これによって、ブレーク条件が存在する限り、PWM 出力の再開を防ぐことができます。
- 出力が有効になっている (MOE ビットがセットされている) 限り、BKDSRM ビットでブレーク保護を解除できません (表 441 を参照)。

表 441. ブレーク保護解除条件

MOE	BKBID	BKDSRM	ブレーク保護状態
0	0	X	設定
0	1	0	設定
0	1	1	解除
1	X	X	設定

ブレーク回路の設定および再設定

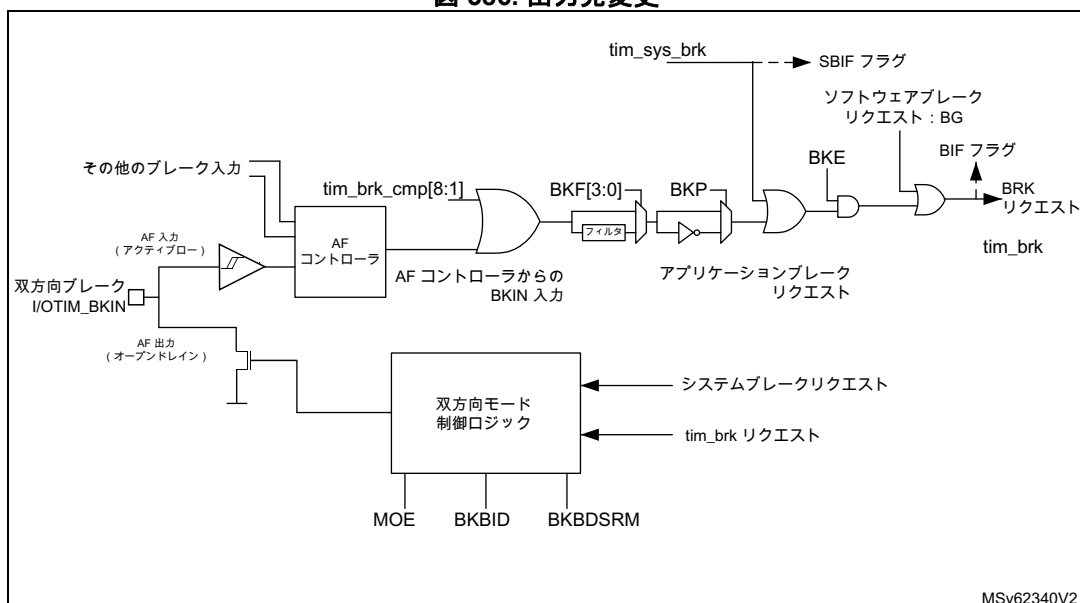
ブレーク回路（入力モードまたは双方向モード）はデフォルト（ペリフェラルリセット設定）で設定されています。

ブレークイベント後に保護を再び設定するには次の手順に従う必要があります。

- BKDSRM ビットをセットして、出力制御を解放する必要があります。
- ソフトウェアは、システムブレーク条件（ある場合）がなくなるまで待ってから、SBIF ステータスフラグをクリアする必要があります（または再設定の前にシステムでクリアします）。
- ソフトウェアは、BKDSRM ビットをハードウェアによってクリアされるまでポーリングする必要があります（アプリケーションブレーク条件がなくなるとき）。

この時点から、ブレーク回路は設定されアクティブになり、MOE ビットをセットして、PWM 出力を再び有効にできます。

図 586. 出力先変更

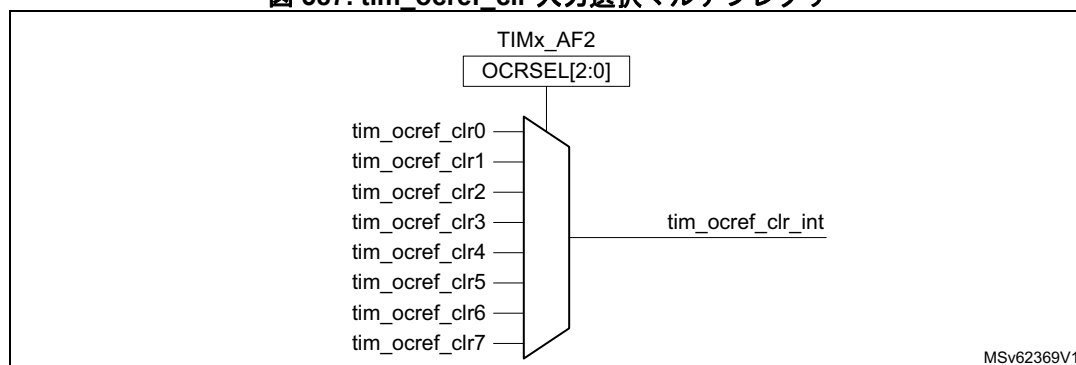


42.4.17 外部イベントによる tim_ocxref 信号のクリア

特定のチャンネルの tim_ocxref 信号は tim_ocref_clr_int 入力にハイレベルを適用するとクリアされます (対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを 1 にセットする)。tim_ocxref は、次の PWM サイクルでアクティブ状態に遷移するまでローのままです。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

tim_ocref_clr_int 入力は、以下の図 587 に示すように、複数の入力から選択できます。

図 587. tim_ocref_clr 入力選択マルチプレクサ



MSv62369V1

42.4.18 6 ステップ PWM 生成

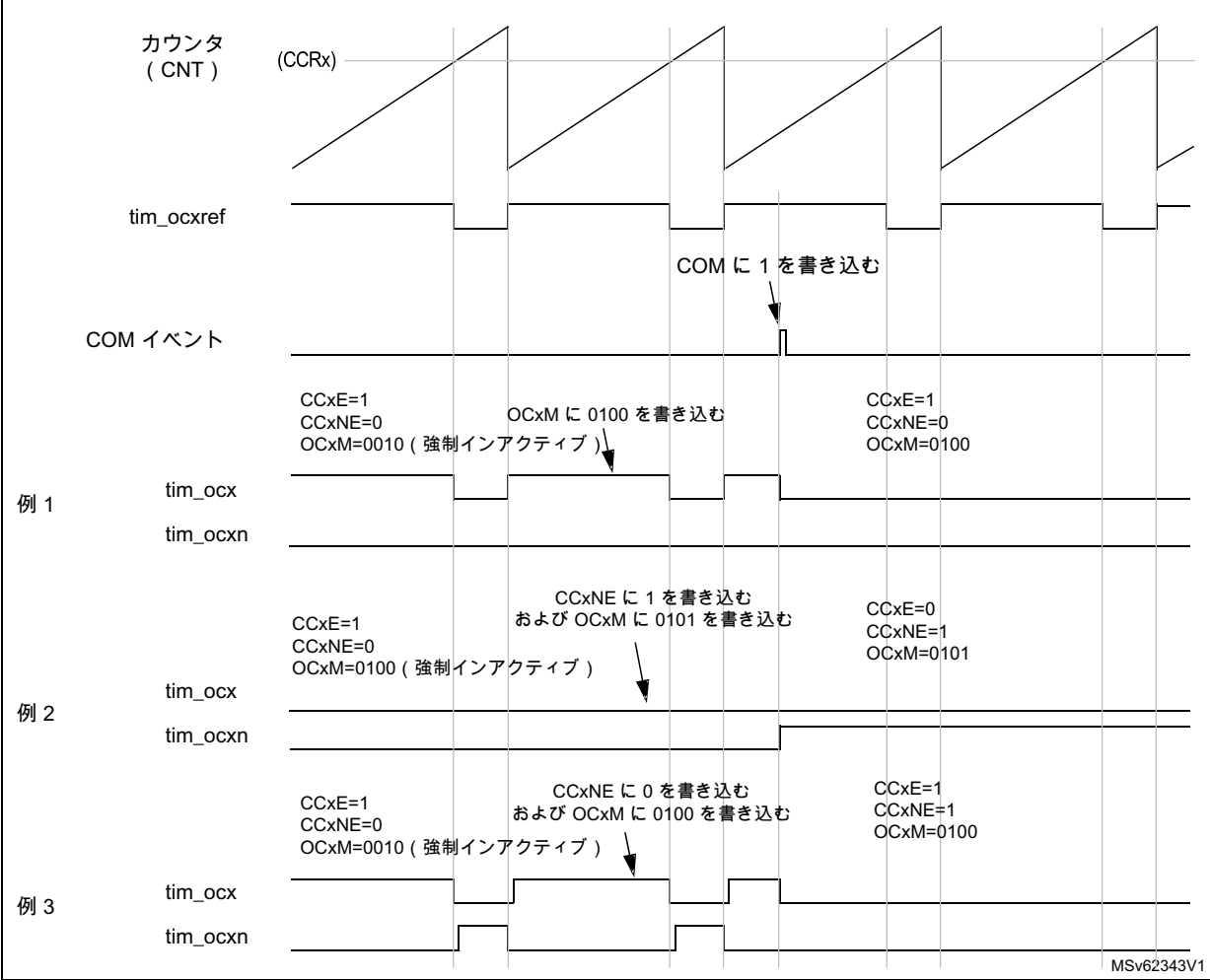
チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転流イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (tim_trgi 立上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx_SR レジスタの COMIF ビット)、これによって割込み (TIMx_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 588 に、COM イベントが発生したときの tim_ocx と tim_ocxn 出力の動作を、

3 種類のプログラミング構成の例で示します。

図 588. 6 ステップ生成 COM の例 (OSSR=1)



42.4.19 ワンパルスモード

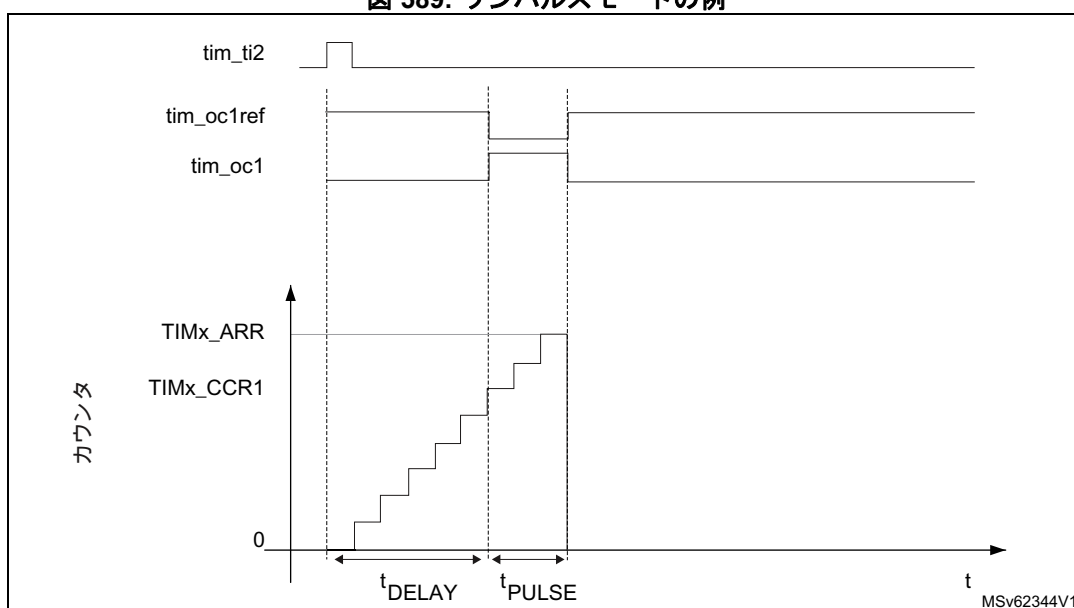
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)

図 589. ワンパルスモードの例



たとえば、tim_ti2 入力ピンで立上がりエッジが検出されたときに、tim_oc1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

tim_ti2fp2 をトリガ 1 として使用します。

1. TIMx_TISEL レジスタの TI2SEL[3:0] ビットで、適切な tim_ti2_in[15:1] ソース (内部または外部) を選択します。
2. TIMx_CCMR1 レジスタの CC2S ビットに "01" を書き込むことによって、tim_ti2fp2 を tim_ti2 に配置します。
3. tim_ti2fp2 は、立上がりエッジを検出して、TIMx_CCER レジスタで CC2P="0" と CC2NP="0" を書き込みます。
4. tim_ti2fp2 をスレーブモードコントローラのトリガ (tim_trgi) として構成するために、TIMx_SMCR レジスタの TS="00110" を書き込みます。
5. tim_ti2fp2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタに OC1M=111 を書き込むことによって、PWM モード 2 を有効にする必要があります。オプションで、TIMx_CCMR1 レジスタに OC1PE='1' を書き込み、TIMx_CR1 レジスタの ARPE ビットを書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込む必要があります。次に、UG ビットをセットすることによって更新を生成し、tim_ti2 で外部トリガイベントを待つ必要があります。この例では、CC1P に“0”を書き込みます。

必要なパルスは 1 つだけなので、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。

特殊ケース : tim_ocx 高速イネーブル

ワンパルスモードでは、tim_tix 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、tim_ocxref (および tim_ocx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

42.4.20 再トリガ可能なワンパルスモード (TIM15 のみ)

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 42.4.19](#) で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

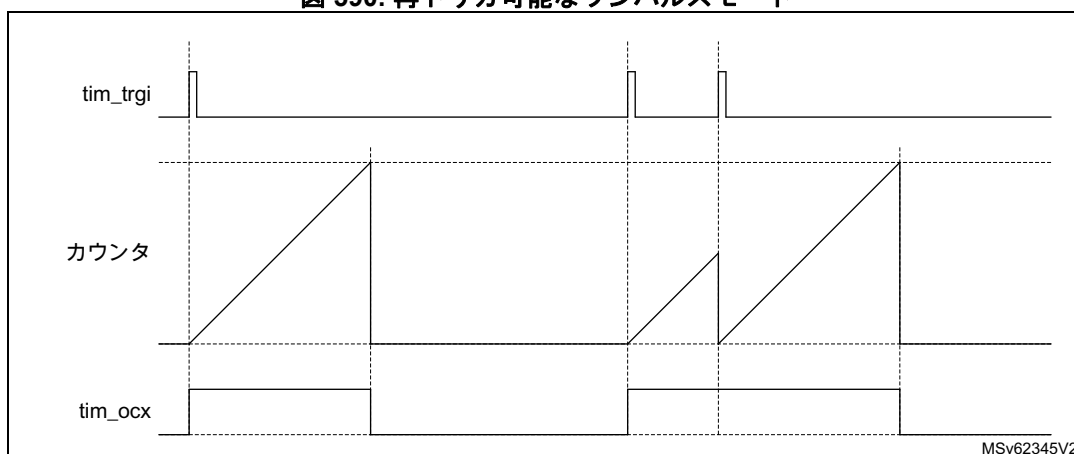
タイマはスレーブモードである必要があり、このときビットは TIMx_SMCR レジスタで SMS[3:0] = “1000” (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が “1000” または “1001” にセットされています。

タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

注 : OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

このモードをセンターアライン PWM モードと組み合わせて使用することはできません。TIMx_CR1 では、CMS[1:0] = 00 にする必要があります。

図 590. 再トリガ可能なワンパルスモード



MSv62345V2

42.4.21 UIF ビットの再配置

TIMx_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割込みフラグ (UIF) の連続コピーを強制します。これは、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取るために使用されます。特定のケースでは、バックグラウンドタスク（カウンタの読出し）と割込み（更新の割込み）との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

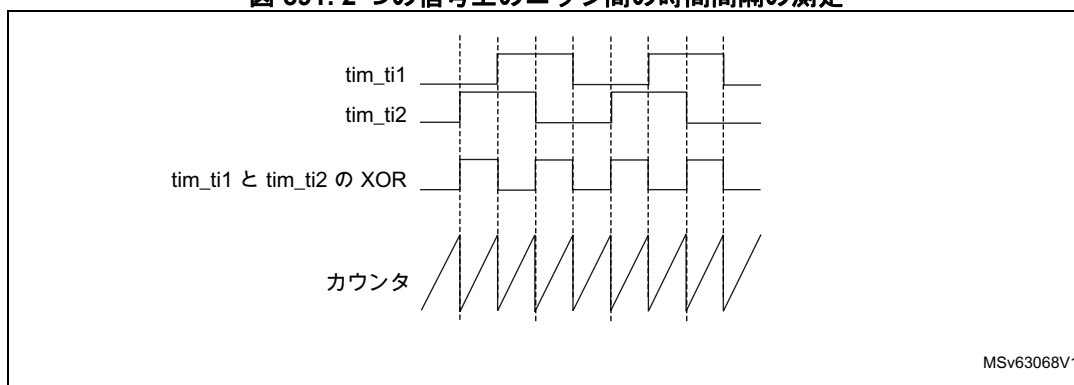
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

42.4.22 タイマ入力 XOR 機能 (TIM15 のみ)

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 2 つの入力ピン tim_ti1 および tim_ti2 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の図 591 に示すように、2 つの入力信号上のエッジ間の時間間隔を測定するのに便利です。

図 591. 2 つの信号上のエッジ間の時間間隔の測定



MSv63068V1

42.4.23 外部トリガ同期 (TIM15 のみ)

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。

TIM15 タイマは、いくつかのモードで外部トリガを使用して同期できます。すなわち、リセットモード、ゲートモード、トリガモード、リセット + トリガモード、およびゲート + リセットモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

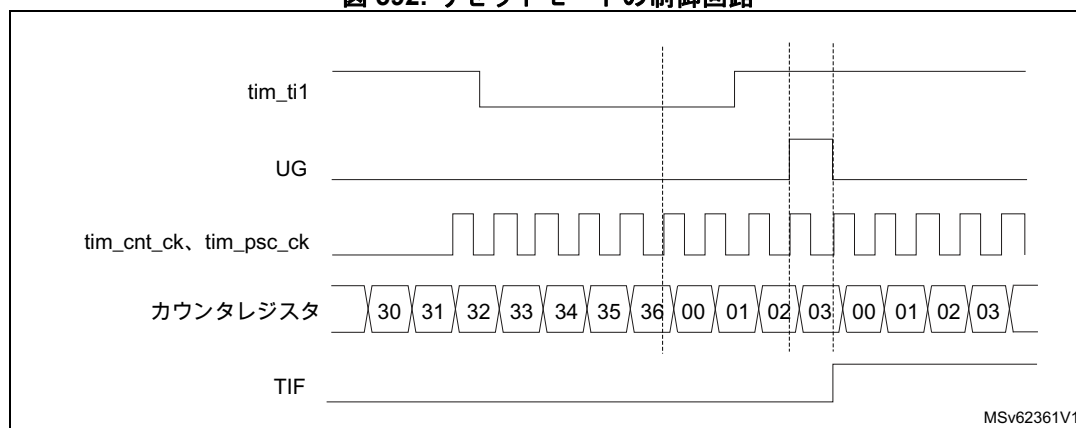
次の例では、tim_ti1 入力の立上がりエッジに応じて、アップカウンタがクリアされます。

1. tim_ti1 の立上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないなので、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立上がりエッジのみを検出します)。
2. TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして tim_ti1 を選択します。
3. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、tim_ti1 の立上がりエッジまで通常の動作を行います。tim_ti1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビットに依存)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。tim_ti1 の立上がりエッジから実際にカウンタがリセットされるまでの遅延は、tim_ti1 入力の同期回路によるものです。

図 592. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

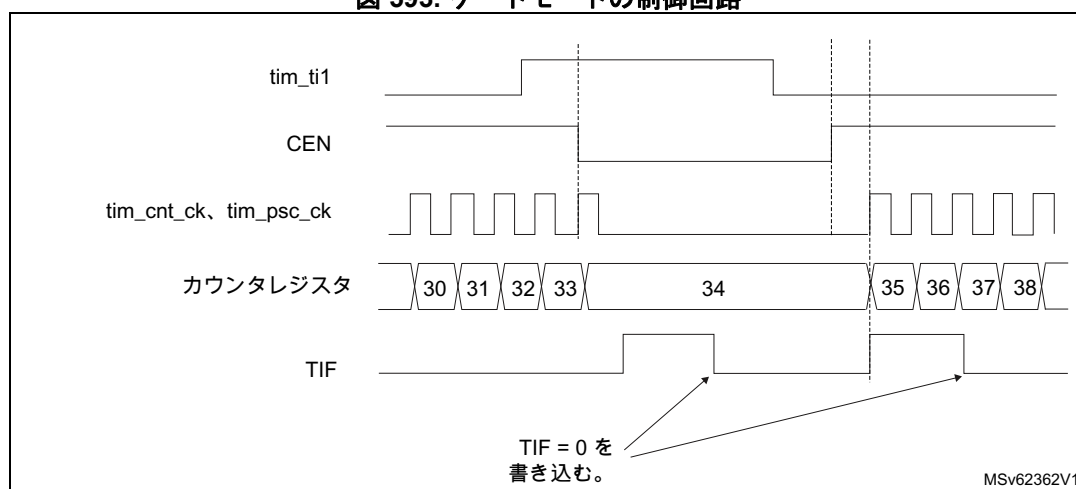
次の例では、アップカウンタは `tim_ti1` 入力が高レベルのときだけカウントします。

1. `tim_ti1` のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、`IC1F=0000` のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01 ビット）。TIMx_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
2. TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=00101 を書き込むことによって、入力ソースとして `tim_ti1` を選択します。
3. TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします（ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

カウンタは、`tim_ti1` がローになると内部クロックでカウントを開始して、`tim_ti1` がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

`tim_ti1` の立上がりエッジから実際にカウンタが停止するまでの遅延は、`tim_ti1` 入力の再同期回路によるものです。

図 593. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

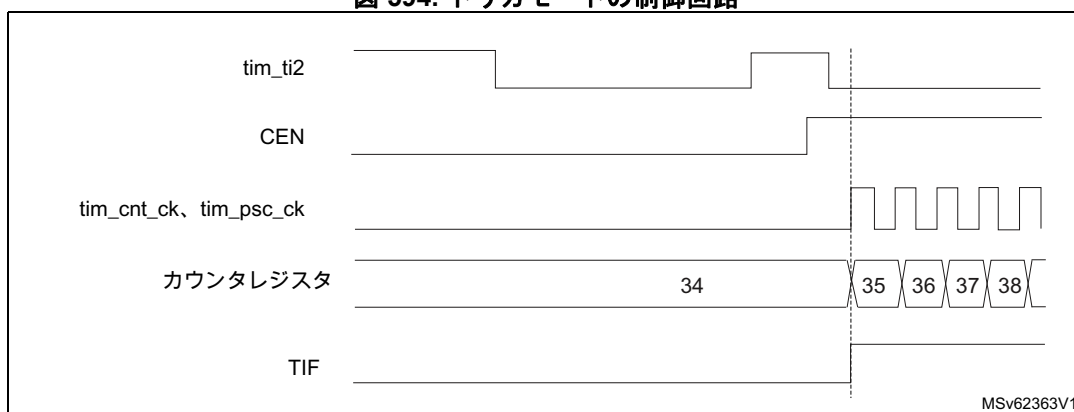
次の例では、アップカウンタは、tim_ti2 入力の立上がりエッジに応じて開始します。

1. tim_ti2 の立上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないなので、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで CC2P に「1」、CC2NP に「0」を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
2. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=00110 を書き込むことによって、入力ソースとして tim_ti2 を選択します。

tim_ti2 で立上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

tim_ti2 の立上がりエッジから実際にカウンタが開始するまでの遅延は、tim_ti2 入力の再同期回路によるものです。

図 594. トリガモードの制御回路



実行時更新のためのスレーブモード選択プリロード

SMS[3:0] ビットをプリロードすることができます。これは、TIMx_SMCR レジスタの SMSPE イネーブルビットをセットすることで有効になります。SMS[3:0] プリロードからアクティブ値へ転送するトリガは、カウンタがオーバーフローするときに発生する更新イベント (UEV) です。

42.4.24 スレーブモード - リセットモードとトリガモードの組合せ (TIM15 のみ)

この場合、選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

42.4.25 スレーブモード - リセットモードとゲートモードの組合せ (TIM15 のみ)

カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

このモードは、範囲外の PWM 信号 (最大期待値を超えるデューティサイクル) を検出するために使用されます。

42.4.26 タイマ同期 (TIM15 のみ)

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。詳細については、[セクション 39.4.23 : タイマの同期](#)を参照してください。

注 : tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

42.4.27 タイマ出力を他のタイマのトリガとして使用 (TIM16/TIM17 のみ)

1 チャンネルのみのタイマは、マスタモードを備えていません。しかし、OC1 出力信号は、他のいくつかのタイマ (本書の他のセクションで説明されているタイマを含む) のトリガとして使用できます。どのタイマがスレーブとして使用できるかを知るには、デバイスのいずれかのタイマの「TIMx 内部トリガ接続」テーブルを確認してください。

OC1 信号のパルス幅は、スレーブタイマが確実にトリガを検出できるように、転送先タイマの少なくとも 2 クロックサイクルになるようにプログラムする必要があります。

たとえば、転送先タイマの CK_INT クロックが転送元のタイマより 4 倍遅い場合、OC1 のパルス幅は 8 クロックサイクルでなければなりません。

42.4.28 ADC トリガ (TIM15 のみ)

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイイベントを生成できます。

注 : tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

42.4.29 DMA バーストモード

TIMx タイマには、1 つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、いくつかのタイマレジスタをソフトウェアのオーバヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx_DMAR を示している必要があります。特定のタイマイイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx_DMAR レジスタへの各書込みは、実際にタイマレジスタの 1 つにリダイレクトされます。

TIMx_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワード単位またはバイト単位) を示します。

TIMx_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し/書込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx_CR1

00001 : TIMx_CR2

00010 : TIMx_SMCR

TIMx_DCR レジスタの DBSS[3:0] ビットでは、DMA バースト転送をトリガする割り込みソースを定義します（詳細は[セクション 42.8.19: TIMx DMA 制御レジスタ \(TIMx_DCR\) \(x = 16~17\)](#)を参照）。

たとえば、更新イベント時に CCRx レジスタ値 (x = 2、3、4) の内容を更新するためにタイマ DMA バースト機能を使用できます。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 回転送、DBA = 0xE、および DBSS = 1。
3. TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
4. TIMx を有効化
5. DMA チャンネルを有効化

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 でなければなりません。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

注： null 値を予約済みレジスタに書き込むことができます。

42.4.30 TIM15/TIM16/TIM17 DMA リクエスト

[表 442](#) に示すように、TIM15/TIM16/TIM17 は DMA リクエストを生成できます。

表 442. DMA リクエスト

DMA リクエスト信号	DMA リクエストあり	有効制御ビット
tim_upd_dma	更新	UDE
tim_cc1_dma	キャプチャ/比較 1	CC1DE
tim_com_dma ⁽¹⁾	転流 (COM)	COMDE
tim_trg_dma ⁽¹⁾	トリガ	TDE

1. TIM15 でのみ使用できます。

42.4.31 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M33 コアは停止状態)、TIMx カウンタは、通常どおりに動作を続けるか、または停止できます。

デバッグモードでの動作は、デバッグサポート (DBG) モジュールでタイマごとに専用の設定ビットを使用してプログラムできます。

安全のため、カウンタが停止すると、出力は無効になります (MOE ビットのリセット時と同じ)。強制的にハイインピーダンスにするために、出力を強制的にインアクティブ状態にするか (OSSI ビット = 1)、GPIO コントローラで制御することができます (OSSI ビット = 0)。

詳細については、デバッグセクションを参照してください。

42.5 TIM15/TIM16/TIM17 低消費電力モード

表 443. 低消費電力モードが TIM15/TIM16/TIM17 に与える影響

モード	説明
SLEEP	影響はなく、ペリフェラルはアクティブです。割込みによって、デバイスは SLEEP モードから復帰できます。
STOP	タイマの動作は停止され、レジスタの内容は保持されます。割込みは生成できません。
STANDBY	タイマはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

42.6 TIM15/TIM16/TIM17 割込み

表 444 に示すように、TIM15/TIM16/TIM17 は複数の割込みを生成できます。

表 444. 割込みリクエスト

項目 (割込みの 略称)	割込みイベント	イベント フラグ	有効制御ビット	割込みのクリア方法	SLEEP モードの 終了	STOP および STANDBY モードの 終了
TIM	更新	UIF	UIE	UIF に 0 を書き込む	可	不可
	キャプチャ/比較 1	CC1IF	CC1IE	CC1IF に 0 を書き込む	可	不可
	キャプチャ/比較 2 ⁽¹⁾	CC2IF	CC2IE	CC2IF に 0 を書き込む	可	不可
	転流 (COM)	COMIF	COMIE	COMIF に 0 を書き込む	可	不可
	トリガ ⁽¹⁾	TIF	TIE	TIF に 0 を書き込む	可	不可
	ブレーク	BIF	BIE	BIF に 0 を書き込む	可	不可

1. TIM15 のみ使用できます。

42.7 TIM15 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2](#) を参照してください。

42.7.1 TIM15 制御レジスタ 1 (TIM15_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw				rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIM15_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIM15_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デッドタイムジェネレータとデジタルフィルタ (tim_tix) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

00 : $t_{DTS} = t_{tim_ker_ck}$

01 : $t_{DTS} = 2 * t_{tim_ker_ck}$

10 : $t_{DTS} = 4 * t_{tim_ker_ck}$

11 : 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIM15_ARR レジスタはバッファされません。

1 : TIM15_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。
0 : 次のイベントのいずれかが更新割込みを生成します (有効な場合)。これらのイベントは、次のとおりです。

– カウンタオーバーフロー/アンダーフロー

– UG ビットのセット

– スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割込みを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注： 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

42.7.2 TIM15 制御レジスタ 2 (TIM15_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OIS2** : 出力アイドル状態 2 (tim_oc2 出力)

0 : MOE = 0 のとき、tim_oc2=0

1 : MOE = 0 のとき、tim_oc2=1

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM15_BKR レジスタの LOCK ビット)、変更できません。

ビット 9 **OIS1N** : 出力アイドル状態 1 (tim_oc1n 出力)

0 : MOE=0 のとき、デッドタイム後に tim_oc1n=0

1 : MOE=0 のとき、デッドタイム後に tim_oc1n=1

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM15_BKR レジスタの LOCK ビット)、変更できません。

ビット 8 **OIS1** : 出力アイドル状態 1 (tim_oc1 出力)

0 : MOE=0 のとき、デッドタイム後に tim_oc1=0

1 : MOE=0 のとき、デッドタイム後に tim_oc1=1

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM15_BKR レジスタの LOCK ビット)、変更できません。

ビット 7 **TI1S** : tim_ti1 選択

0 : tim_ti1_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます。

1 : tim_ti1_in[15:0] および tim_ti2_in[15:0] マルチプレクサ出力が tim_ti1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[2:0]** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (tim_trgo) を選択するために使用します。組み合わせは、次のとおりです。

- 000 : **リセット** - TIM15_EGR レジスタの UG ビットがトリガ出力 (tim_trgo) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、tim_trgo 信号は実際のリセットより遅延します。
- 001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (tim_trgo) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理積 (AND) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、tim_trgo には遅延が存在します (TIM15_SMCR レジスタの MSM ビットの説明を参照してください)。
- 010 : **更新** - 更新イベントがトリガ出力 (tim_trgo) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。
- 011 : **パルス比較** - キャプチャまたは比較一致が発生すると (tim_trgo)、CC1IF フラグがセットされるとき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します
- 100 : **比較** - tim_oc1refc 信号がトリガ出力 (tim_trgo) として使用されます。
- 101 : **比較** - tim_oc2refc 信号がトリガ出力 (tim_trgo) として使用されます。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

- 0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。
- 1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

- 0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。
- 1 : キャプチャ/比較制御ビットがプリロードされているときに (CCPC=1)、COMG ビットをセットすることによって、または tim_trgi の立上がりエッジで更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

- 0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。
- 1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書込みの後、これらのビットは、転流イベント (COM) が発生したときにのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは tim_trgi で立上がりエッジが検出されたとき)。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

42.7.3 TIM15 スレーブモード制御レジスタ (TIM15_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMSPE	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
							rw			rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
								rw	rw	rw	rw		rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **SMSPE** : SMS プリロードイネーブル

このビットは、SMS[3:0] ビットフィールドをプリロードするかどうかを選択します。

0 : SMS[3:0] ビットフィールドはプリロードされません。

1 : SMS[3:0] プリロードは有効です。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (tim_trgi) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (tim_trgo を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 21、20、6、**TS[4:0]** : トリガ選択

5、4 このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

00000 : 内部トリガ 0 (tim_itr0)
00001 : 内部トリガ 1 (tim_itr1)
00010 : 内部トリガ 2 (tim_itr2)
00011 : 内部トリガ 3 (tim_itr3)
00100 : tim_ti1 エッジ検出回路 (tim_ti1f_ed)
00101 : フィルタタイマ入力 1 (tim_ti1fp1)
00110 : フィルタタイマ入力 2 (tim_ti2fp2)
00111 : 予約済み
01000 : 内部トリガ 4 (tim_itr4)
01001 : 内部トリガ 5 (tim_itr5)
01010 : 内部トリガ 6 (tim_itr6)
01011 : 内部トリガ 7 (tim_itr7)
01100 : 内部トリガ 8 (tim_itr8)
01101 : 内部トリガ 9 (tim_itr9)
01110 : 内部トリガ 10 (tim_itr10)
10000 : 内部トリガ 12 (tim_itr12)
10001 : 内部トリガ 13 (tim_itr13)
10010 : 内部トリガ 14 (tim_itr14)
10011 : 内部トリガ 15 (tim_itr15)

その他 : 予約済み

各タイマでの tim_itrx の詳細については、[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

注： 遷移時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき（たとえば、SMS = 000 のとき）にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 16、2、1、0 **SMS[3:0]** : スレーブモード選択

外部信号が選択されると、トリガ信号 (tim_trgi) のアクティブエッジが、外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : 予約済み

0010 : 予約済み

0011 : 予約済み

0100 : リセットモード - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ tim_trgi の立上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (tim_trgi) の立上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (tim_trgi) の立上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1001 : ゲートモードとリセットモードの組み合わせ - カウンタクロックは、トリガ入力 (tim_trgi) がハイのときに有効になります。トリガがローになると、カウンタは停止し、リセットされます。カウンタの開始と停止の両方が制御されます。

その他 : 予約済み。

注 : トリガ入力として tim_ti1f_ed が選択されている場合 (TS=00100)、ゲートモードを使用することはできません。tim_ti1f_ed は tim_ti1f の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注 : tim_trgo 信号を受信するスレーブペリフェラル (タイマ、ADC など) のクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間はクロック周波数 (プリスケアラ) を動作中に変更しないでください。

42.7.4 TIM15 DMA／割込み有効レジスタ (TIM15_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	rw	rw				rw	rw	rw	rw	rw			rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

0 : COM DMA リクエストは無効です。

1 : COM DMA リクエストは有効です。

ビット 12:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル
 0 : 更新 DMA リクエストは無効です。
 1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割込みイネーブル
 0 : ブレーク割込みは無効です。
 1 : ブレーク割込みは有効です。

ビット 6 **TIE** : トリガ割込みイネーブル
 0 : トリガ割込みは無効です。
 1 : トリガ割込みは有効です。

ビット 5 **COMIE** : COM 割込み有効
 0 : COM 割込み無効
 1 : COM 割込み有効

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IE** : キャプチャ/比較 2 割込みイネーブル
 0 : CC2 割込みは無効です。
 1 : CC2 割込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割込みイネーブル
 0 : CC1 割込みは無効です。
 1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル
 0 : 更新割込みは無効です。
 1 : 更新割込みは有効です。

42.7.5 TIM15 ステータスレジスタ (TIM15_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ
 CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ
 このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。
 0 : オーバキャプチャは検出されていません。
 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIM15_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割込みフラグ
 このフラグは、ブレーク入力がアクティブになると、ハードウェアによってセットされます。ブレーク入力がアクティブでない場合、ソフトウェアによってクリアできます。
 0 : ブレークイベントは発生していません。
 1 : ブレーク入力でアクティブレベルが検出されました。

ビット 6 **TIF** : トリガ割込みフラグ

このフラグは、TRG トリガイイベント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、tim_trgi 入力でアクティブエッジが検出されたとき、またはゲートモードが選択されている場合は、両方のエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

0 : トリガイイベントは発生していません。

1 : トリガ割込みが保留中です。

ビット 5 **COMIF** : COM 割込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます（キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき）。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割込みがペンディング中です。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IF** : キャプチャ/比較 2 割込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって（入力キャプチャモードまたは出力比較モード）、または TIMx_CCR1 レジスタを読み出すことによって（入力キャプチャモードのみ）クリアされます。

0 : 比較一致/入力キャプチャは発生していません。

1 : 比較一致または入力キャプチャが発生しました。

チャンネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャンネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた（TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ検知に従って、IC1 でエッジが検出された）ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフロー（繰り返しカウンタ=0 の場合の更新）、および TIM15_CR1 レジスタの UDIS=0 の場合。

- TIM15_CR1 レジスタの URS=0 かつ UDIS=0 の場合に、TIM15_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

- TIM15_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイイベントによって CNT が再初期化されたとき（[セクション 42.7.3 : TIM15 スレーブモード制御レジスタ \(TIM15_SMCR\)](#) を参照）。

42.7.6 TIM15 イベント生成レジスタ (TIM15_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIM15_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2G** : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIM15_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

42.7.7 TIM15 キャプチャ／比較モードレジスタ 1 (TIM15_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード（このセクション）または出力比較モード（次のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F [3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F[3:0]** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIM15_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIM15_CCER レジスタの CC2E="0") のときのみ、書込みができます。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力の変化を検証するには N 個の連続イベントが必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 3:2 IC1PSC[1:0] : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E="0" (TIM15_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 CC1S[1:0] : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIM15_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIM15_CCER レジスタの CC1E="0") のときのみ、書込みができます。

42.7.8 TIM15 キャプチャ/比較モードレジスタ 1 [オルタネート] (TIM15_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

同じレジスタを出力比較モード (このセクション) または入力キャプチャモード (前のセクション) に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます (たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M [2:0]			OC2 PE	OC2 FE	CC2S [1:0]		OC1 CE	OC1M [2:0]			OC1 PE	OC1 FE	CC1S [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード：

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 24、14:12 **OC2M[3:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_ti2 に配置されます。

10 : C2 チャンネルは入力として設定され、tim_ic2 は tim_ti1 に配置されます。

11 : CC2 チャンネルは入力として設定され、tim_ic2 は tim_trc に配置されます。このモードは、TS ビット (TIM15_SMCR レジスタ) を通じて内部トリガ入力を選択されている場合のみ機能します。

注： **CC2S** ビットは、チャンネルがオフ (TIM15_CCER レジスタの CC2E="0") のときのみ、書き込みができます。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : tim_oc1ref は tim_ocref_clr_int 入力の影響を受けません。

1 : tim_oc1ref は tim_ocref_clr_int 入力のハイレベルが検出されるとクリアされます。

ビット 16、6:4 OC1M[3:0] : 出力比較 1 モード

これらのビットは、tim_oc1 および tim_oc1n が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 および tim_oc1n のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIM15_CCR1 とカウンタ TIM15_CNT との間の比較結果は出力に影響しません。このモードは、タイマをソフトウェアタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードが有効になると、出力は停止状態になる前の状態（アクティブまたはインアクティブ）を維持します。

0001 : 一致時にチャンネル 1 をアクティブレベルにセットします。カウンタ TIM15_CNT がキャプチャ/比較レジスタ 1 (TIM15_CCR1) と一致すると、tim_oc1ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 1 をインアクティブレベルにセットします。カウンタ TIM15_CNT がキャプチャ/比較レジスタ 1 (TIM15_CCR1) と一致すると、tim_oc1ref 信号は強制的にローになります。

0011 : 反転 - TIM15_CNT = TIM15_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制非アクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - チャンネル 1 は、TIM15_CNT < TIM15_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111 : PWM モード 2 - チャンネル 1 は、TIM15_CNT < TIM15_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、tim_trgi 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み

1011 : 予約済み

1100 : 組み合わせ PWM モード 1 - tim_oc1ref は、PWM モード 1 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 OR です。

1101 : 組み合わせ PWM モード 2 - tim_oc1ref は、PWM モード 2 と同様に挙動します。tim_oc1refc は tim_oc1ref と tim_oc2ref の論理 AND です。

1110 : 予約済み。

1111 : 予約済み。

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIM15_BDTR レジスタの LOCK ビット)、CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

PWM モードでは、比較結果が変化したとき、出力比較モードが「停止」モードから「PWM」モードに切り替わったとき、および出力比較モードが「強制アクティブ/インアクティブ」モードから「PWM」モードに切り替わったときに、OCREF のレベルが変化します。

相補出力を持つチャンネルでは、このビットはプリロードされます。TIM15_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとります。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIM15_CCR1 のプリロードレジスタは無効です。TIM15_CCR1 は、いつでも書込み可能であり、新しい値はただちに有効になります。

1 : TIM15_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIM15_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIM15_BDTR レジスタの LOCK ビット)、CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、トリガイイベントとタイマ出力の遷移の間の待ち時間を短縮します。開始トリガの後、できるだけ早く出力パルスを開始するには、1 パルスモード (TIMx_CR1 レジスタの OPM ビットのセット) で使用する必要があります。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、tim_ocx は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S[1:0]** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

10 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti2 に配置されます。

11 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_trc に配置されます。このモードは、TS ビット (TIM15_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIM15_CCER レジスタの CC1E="0") のときのみ、書き込みができます。

42.7.9 TIM15 キャプチャ/比較有効レジスタ (TIM15_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP** : キャプチャ/比較 2 相補出力極性

CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ/比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ/比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : tim_oc1n はアクティブハイです。

1 : tim_oc1n はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、tim_ti1fp1 と tim_ti2fp1 の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ (TIM15_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後に書き込みできなくなります。相補出力を持つチャンネルでは、このビットはプリロードされます。TIM15_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転流イベントが発生したときのみ、プリロードされたビットから新しい値を取り込みます。

ビット 2 CC1NE : キャプチャ / 比較 1 相補出力カインープル

0 : オフ - tim_oc1n はアクティブではありません。tim_oc1n のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - tim_oc1n 信号は、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

ビット 1 CC1P : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 のアクティブハイ (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

1 : OC1 のアクティブロー (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立上がりエッジ。この回路は TIxFP1 の立上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立下がりエッジ。この回路は TIxFP1 の立下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 が反転されます (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ/この回路は TIxFP1 の立上がりエッジと立下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた (TIM15_BDTR レジスタの LOCK ビット) 直後は書き込みできません。

相補出力を持つチャンネルでは、このビットはプリロードされます。TIM15_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 CC1E : キャプチャ / 比較 1 出力カインープル

0 : キャプチャモードは無効/OC1 は非アクティブ (下記を参照)。

1 : キャプチャモードは有効/OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが出力として設定されている場合、OC1 レベルは、CC1E ビットの状態にかかわらず、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1NE ビットに依存します。詳細については、表 445 を参照してください。

表 445. ブレーク機能を持つ相補 tim_ocx および tim_ocxn チャンネルの出力制御ビット (TIM15)

制御ビット					出力状態 ⁽¹⁾		
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	tim_ocx 出力状態	tim_ocxn 出力状態	
1	X	X	0	0	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocx=0 tim_ocxn=0		
		0	0	1	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocx=0	tim_ocxref + 極性 tim_ocxn = tim_ocxref XOR CCxNP	
		0	1	0	tim_ocxref + 極性 tim_ocx=tim_ocxref XOR CCxP	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_ocxn=0	
		X	1	1	tim_ocxref + 極性 + デッドタイム	tim_ocxref に対する相補（OCREF ではなく）+ 極性 + デッドタイム	
		1	0	1	オフ状態（インアクティブ状態で出力有効） tim_ocx=CCxP	tim_ocxref + 極性 tim_ocxn=tim_ocxref XOR CCxNP	
		1	1	0	tim_ocxref + 極性 tim_ocx=tim_ocxref xor CCxP	オフ状態 （インアクティブ状態で出力有効） tim_ocxn=CCxNP	
0	0	X	X	X	出力無効（タイマによって駆動されない：ハイインピーダンス）		
	1		0	0			
			0	1	オフ状態（インアクティブ状態で出力有効） 非同期に：tim_ocx=CCxP、tim_ocxn=CCxNP クロックが存在する場合：デッドタイム後、tim_ocx=OISx および tim_ocxn=OISxN。ただし、OISx と OISxN は、アクティブ状態における tim_ocx と tim_ocxn の両方に対応しないことを前提とします。		
			1	0			
			1	1			

1. チャンネルの両方の出力が使用されないとき (GPIO コントローラが制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP ビットはクリアされたままでなければなりません。

注 : 相補 tim_ocx および tim_ocxn チャンネルに接続されている外部入出力ピンの状態は、tim_ocx および tim_ocxn チャンネルの状態と、GPIO 制御 および オルタネート機能選択レジスタに依存します。

42.7.10 TIM15 カウンタ (TIM15_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : UIF コピー

このビットは TIM15_ISR レジスタの UIF ビットのコピーで読出し専用です。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分は得られません。

42.7.11 TIM15 プリスケアラ (TIM15_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 ($f_{tim_cnt_ck}$) は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIM15_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

42.7.12 TIM15 の自動再ロードレジスタ (TIM15_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 42.4.3 : 1751 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、ARR[15:0] に自動再ロード値を保持します。ARR[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

42.7.13 TIM15 繰返しカウンタレジスタ (TIM15_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰返しカウンタの再ロード値

このビットフィールドによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの周期的な転送）を定義します。また、更新割込み生成の頻度も定義します（この割込みが有効な場合）。

繰返しダウンカウンタがゼロに達すると、更新イベントが生成され、REP 値からカウントをリスタートします。繰返しカウンタは、繰返し更新イベント UEV のみ、REP 値で再ロードされるので、TIM15_RCR レジスタへの書込みは、次の繰返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

- エッジアラインモードでは、PWM 周期の数
- センターアラインモードでは、PWM の 1/2 周期の数

42.7.14 TIM15 のキャプチャ／比較レジスタ 1 (TIM15_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR1[19:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIM15_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIM15_CNT と比較され、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。TIMx_CCR1 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] にキャプチャ値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] にキャプチャ値を保持します。CCR1[3:0] ビットはリセットされています。

42.7.15 TIM15 のキャプチャ／比較レジスタ 2 (TIM15_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR2[19:0]** : キャプチャ／比較 2 値**CC2 チャンネルが出力として設定されている場合 :**

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIM15_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIM15_CNT と比較され、tim_oc2 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] に比較値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] に整数部を保持します。CCR2[3:0] ビットフィールドにはディザリング部分が含まれます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 1 イベント (tim_ic2) によって転送されたカウンタ値です。TIMx_CCR2 レジスタは読み出し専用レジスタで、プログラムできません。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR2[15:0] にキャプチャ値を保持します。CCR2[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR2[19:4] にキャプチャ値を保持します。CCR2[3:0] ビットはリセットされています。

42.7.16 TIM15 ブレークおよびデッドタイムレジスタ (TIM15_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
			rw		rw							rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： BKBID、BKDSRM、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] ビットは、LOCK 設定に応じて書き込みがロックされるので、TIM15_BDTR レジスタへの最初の書き込みアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **BKBID** : ブレーク双方向

0 : ブレーク入力 tim_brk は入力モードです。

1 : ブレーク入力 tim_brk は双方向モードです。

双方向モード (BKBID ビットが 1 にセット) では、ブレーク入力が入力モードとオープンドレイン出力モード両方で設定されます。アクティブなブレークイベントで、ブレーク入力の低ロジックレベルをアサートし、外部デバイスに対する内部ブレークイベントを表します。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

注： このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **BKDSRM** : ブレーク解除

0 : ブレーク入力 tim_brk が設定されます。

1 : ブレーク入力 tim_brk は解除されます。

このビットは、アクティブなブレークソースがない場合、ハードウェアによってクリアされます。

BKDSRM ビットは、双方向出力制御 (ハイインピーダンス状態でのオープンドレイン出力) を解放するためにソフトウェアでセットしてから、ハードウェアによってリセットされ、障害状態がなくなったことを示すまでポーリングする必要があります。

注： このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 25:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **BKF[3:0]** : ブレークフィルタ

このビットフィールドは、tim_brk 入力信号をサンプリングする周波数と、tim_brk に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、tim_brk は非同期で動作します。

0001 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 2$

0010 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 4$

0011 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 8$

0100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、 $N = 6$

0101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、 $N = 8$

0110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、 $N = 6$

0111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、 $N = 8$

1000 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、 $N = 6$

1001 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、 $N = 8$

1010 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、 $N = 5$

1011 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、 $N = 6$

1100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、 $N = 8$

1101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 5$

1110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 6$

1111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 8$

注 : このビットは、**LOCK レベル 1** がプログラムされているとき (TIM15_BDTR レジスタの **LOCK** ビット) には変更できません。

ビット 15 **MOE** : メイン出カインーブル

このビットは、tim_brk 入力が入アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : OSSR ビットに応じて、tim_ocx および tim_ocxn 出力が無効か、強制的にアイドル状態になります。

1 : tim_ocx および tim_ocxn 出力は、それぞれのインーブルビット (TIM15_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、tim_ocx/tim_ocxn インーブルの説明を参照してください ([セクション 42.7.9 : 1806 ページのTIM15 キャプチャ/比較有効レジスタ \(TIM15_CCER\)](#))。

ビット 14 **AOE** : 自動出カインーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力が入アクティブでない場合)。

注 : このビットは、**LOCK レベル 1** がプログラムされている場合 (TIM15_BDTR レジスタの **LOCK** ビット)、変更できません。

ビット 13 **BKP** : ブレーク極性

0 : ブレーク入力 tim_brk はアクティブローです。

1 : ブレーク入力 tim_brk はアクティブハイです。

注 : このビットは、**LOCK レベル 1** がプログラムされている場合 (TIM15_BDTR レジスタの **LOCK** ビット)、変更できません。

このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 **BKE** : ブレーキイネーブル

0 : ブレーキ入力 (tim_brk および tim_sys_brk クロック障害イベント) は無効です。

1 : ブレーキ入力 (tim_brk および tim_sys_brk クロック障害イベント) は有効です。

このビットは、LOCK レベル 1 がプログラムされているとき (TIM15_BDTR レジスタの LOCK ビット) には変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 **OSSR** : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、tim_ocx/tim_ocxn イネーブルの説明を参照してください ([セクション 42.7.9 : 1806 ページのTIM15 キャプチャ/比較有効レジスタ \(TIM15_CCER\)](#))。

0 : インアクティブのとき、tim_ocx/tim_ocxn 出力は無効です (タイマは出力の制御を解放し、制御はハイインピーダンス状態を強制する GPIO によって引き継がれます)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、tim_ocx/tim_ocxn 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御されます)。

注 : このビットは、LOCK レベル 2 がプログラムされているとき (TIM15_BDTR レジスタの LOCK ビット) には変更できません。

ビット 10 **OSSI** : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、tim_ocx/tim_ocxn イネーブルの説明を参照してください ([セクション 42.7.9 : 1806 ページのTIM15 キャプチャ/比較有効レジスタ \(TIM15_CCER\)](#))。

0 : インアクティブのとき、tim_ocx/tim_ocxn 出力は無効です (tim_ocx/tim_ocxn イネーブル出力信号 = 0)。

1 : インアクティブのとき、tim_ocx/tim_ocxn 出力は、CCxE=1 または CCxNE=1 になると、まず、強制的にアイドルレベルになります (tim_ocx/tim_ocxn イネーブル出力信号 = 1)

注 : このビットは、LOCK レベル 2 がプログラムされているとき (TIM15_BDTR レジスタの LOCK ビット) には変更できません。

ビット 9:8 **LOCK[1:0]** : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 - TIM15_BDTR レジスタの DTG ビット、TIM15_CR2 レジスタの OISx および OISxN ビット、および TIM15_BDTR レジスタの BKBID/BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIM15_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込みできなくなります。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIM15_CCMRx レジスタの OCxM および OCxPE ビット) が書き込みできなくなります。

注 : LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIM15_BDTR レジスタに書き込みが行われると、それらの内容は次のリセットまで凍結されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$

例 : $T_{DTS}=125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされている (TIM15_BDTR レジスタの LOCK ビット) と、変更できません。

42.7.17 TIM15 タイマデッドタイムレジスタ 2 (TIM15_DTR2)

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE
														r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGF[7:0]							
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DTPE** : デッドタイムプリロードイネーブル

0 : デッドタイム値はプリロードされません。

1 : デッドタイム値プリロードは有効です。

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 16 **DTAE** : デッドタイム非対称イネーブル

0 : 立上がりおよび立下がりエッジでのデッドタイムは同一で、DTG[7:0] レジスタで定義されます。

1 : 立上がりエッジのデッドタイムは DTG[7:0] レジスタで定義され、立下がりエッジのデッドタイムは DTGF[7:0] ビットで定義されます。

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DTGF[7:0]** : デッドタイム立下がりエッジジェネレータのセットアップ

このビットフィールドでは、立下がりエッジにおいて、相補出力の間に挿入されるデッドタイムの長さを指定します。

$DTGF[7:5]=0xx \Rightarrow DTF=DTGF[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$ 。

$DTGF[7:5]=10x \Rightarrow DTF=(64+DTGF[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$ 。

$DTGF[7:5]=110 \Rightarrow DTF=(32+DTGF[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$ 。

$DTGF[7:5]=111 \Rightarrow DTF=(32+DTGF[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$ 。

例 : $T_{DTS}=125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされている (TIM15_BDTR レジスタの LOCK ビット) とき、変更できません。

42.7.18 TIM15 入力選択レジスタ (TIM15_TISEL)

アドレスオフセット : 0x5C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TI2SEL[3:0]** : tim_ti2_in[15:0] の入力を選択します。

0000 : TIM15_CH2 入力 (tim_ti2_in0)

0001 : tim_ti2_in1

.....

1111 : tim_ti2_in15

相互接続一覧については、[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1_in[15:0] の入力を選択します。

0000 : TIM15_CH1 入力 (tim_ti1_in0)

0001 : tim_ti1_in1

.....

1111 : tim_ti1_in15

相互接続一覧については、[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

42.7.19 TIM15 オルタネート機能レジスタ 1 (TIM15_AF1)

アドレスオフセット : 0x060

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	BK CMP4P	BK CMP3P	BK CMP2P	BK CMP1P	BKINP	BK CMP8E	BK CMP7E	BK CMP6E	BK CMP5E	BK CMP4E	BK CMP3E	BK CMP2E	BK CMP1E	BKINE
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

製品固有の実装については[セクション 42.4.2:TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **BKCOMP4P** : tim_brk_cmp4 入力極性

このビットは、tim_brk_cmp4 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp4 入力はアクティブハイです。

1 : tim_brk_cmp4 入力はアクティブローです。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 12 **BKCOMP3P** : tim_brk_cmp3 入力極性

このビットは、tim_brk_cmp3 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp3 入力はアクティブハイです。

1 : tim_brk_cmp3 入力はアクティブローです。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 11 **BKCOMP2P** : tim_brk_cmp2 入力極性

このビットは、tim_brk_cmp2 入力検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp2 入力はアクティブハイです。

1 : tim_brk_cmp2 入力はアクティブローです。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 10 **BKCOMP1P** : tim_brk_cmp1 入力極性

このビットは、tim_brk_cmp1 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp1 入力はアクティブハイです。

1 : tim_brk_cmp1 入力はアクティブローです。

注： このビットは、**LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 9 BKINP : TIMx_BKIN 入力の極性

このビットは、TIMx_BKIN オルタネート機能入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : TIMx_BKIN 入力はアクティブハイです。

1 : TIMx_BKIN 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 8 BKCMP8E : tim_brk_cmp8 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp8 を有効化します。mdf_brkx 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp8 入力は無効です。

1 : tim_brk_cmp8 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 7 BKCMP7E : tim_brk_cmp7 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp7 を有効化します。COMP7 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp7 入力は無効です。

1 : tim_brk_cmp7 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 6 BKCMP6E : tim_brk_cmp6 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp6 を有効化します。tim_brk_cmp6 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp6 入力は無効です。

1 : tim_brk_cmp6 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 5 BKCMP5E : tim_brk_cmp5 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp5 を有効化します。tim_brk_cmp5 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp5 入力は無効です。

1 : tim_brk_cmp5 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 4 BKCMP4E : tim_brk_cmp4 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp4 を有効化します。tim_brk_cmp4 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp4 入力は無効です。

1 : tim_brk_cmp4 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 3 BKCMP3E : tim_brk_cmp3 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp3 を有効化します。tim_brk_cmp3 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp3 入力は無効です。

1 : tim_brk_cmp3 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 2 **BKCMP2E** : tim_brk_cmp2 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp2 を有効化します。tim_brk_cmp2 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp2 入力は無効です。

1 : tim_brk_cmp2 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 1 **BKCMP1E** : tim_brk_cmp1 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp1 を有効化します。tim_brk_cmp1 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp1 入力は無効です。

1 : tim_brk_cmp1 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 0 **BKINE** : TIMx_BKIN 入力イネーブル

このビットは、タイマの tim_brk 入力に対して TIMx_BKIN オルタネート機能入力を有効化します。TIMx_BKIN 入力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : TIMx_BKIN 入力は無効です。

1 : TIMx_BKIN 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

42.7.20 TIM15 オルタネート機能レジスタ 2 (TIM15_AF2)

アドレスオフセット : 0x064

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OCRSEL[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **OCRSEL[2:0]** : ocref_clr ソース選択

これらのビットは、ocref_clr 入力ソースを選択します。

000 : tim_ocref_clr0

001 : tim_ocref_clr1

010 : tim_ocref_clr2

011 : tim_ocref_clr3

100 : tim_ocref_clr4

101 : tim_ocref_clr5

110 : tim_ocref_clr6

111 : tim_ocref_clr7

製品固有の実装については[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

注： これらのビットは、LOCK レベル 1 がプログラムされている場合 (TIM15_BDTR レジスタの LOCK ビット)、変更できません。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

42.7.21 TIM15 DMA 制御レジスタ (TIM15_DCR)

アドレスオフセット : 0x3DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **DBSS[3:0]** : DMA パーストソース選択

このビットフィールドは、DMA パースト転送をトリガする割込みソースを定義します（タイマは、TIMx_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにパースト転送を認識します）。

0000 : 予約済み

0001 : 更新

0010 : CC1

0110 : COM

0111 : トリガ

その他 : 予約済み

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA パースト長

この 5 ビットのフィールドは、DMA 転送長（タイマは、TIM15_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにパースト転送を認識します）を指定します。

00000 : 1 回転送

00001 : 2 回転送、

00010 : 3 回転送、

.....

10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのフィールドは、DMA 転送のベースアドレスを指定します（TIM15_DMAR アドレスを通じて読出し／書込みアクセスが行われるとき）。DBA は、TIM15_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIM15_CR1

00001 : TIM15_CR2

00010 : TIM15_SMCR

.....

42.7.22 完全転送の TIM15 DMA アドレス (TIM15_DMAR)

アドレスオフセット : 0x3E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DMAB[31:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読みまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$(\text{TIM15_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$

ここで、TIM15_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIM15_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIM15_DCR 内で設定)。

42.7.23 TIM15 レジスタマップ

TIM15 レジスタは、次の表のように、16 ビットのアドレス可能レジスタとしてマップされます。

表 446. TIM15 レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIM15_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	CKD[1:0]		ARPE	Res	Res	Res	OPM	URS	UDIS	CEN	
	リセット値																					0		0	0	0				0	0	0	0	
0x04	TIM15_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OIS2	OIS1N	OIS1	THS	MMS[2:0]			CCDS	CCUS	Res	CCPC	
	リセット値																						0	0	0	0	0	0	0	0	0			0
0x08	TIM15_SMCRR	Res	Res	Res	Res	Res	Res	Res	SMSPE	Res	Res	TS[4:3]		Res	Res	Res	SMS[3]	Res	Res	Res	Res	Res	Res	Res	Res	MSM	TS[2:0]			Res	SMS[2:0]			
	リセット値								0			0	0				0									0	0	0	0			0	0	0
0x0C	TIM15_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	Res	Res	Res	Res	CC1DE	UDE	BIE	TIE	COMIE	Res	Res	CC2IE	CC1IE	UIE
	リセット値																		0	0					0	0	0	0				0	0	0
0x10	TIM15_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC2OF	CC1OF	Res	BIF	TIF	COMIF	Res	Res	CC2IF	CC1IF	UIF
	リセット値																		0	0				0	0	0	0	0				0	0	0
0x14	TIM15_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BG	TG	COMG	Res	Res	CC2G	CC1G	UG	
	リセット値																									0	0	0				0	0	0

表 446. TIM15 レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x18	TIM15_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F [3:0]				IC1 PSC [1:0]		CC1S [1:0]				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	TIM15_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE		OC2M [2:0]		OC2PE OC2FE		CC2S [1:0]		OC1CE		OC1M [2:0]		OC1PE OC1FE		CC1S [1:0]				
	リセット値								0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x20	TIM15_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E			
	リセット値																									0		0	0	0	0	0	0			
0x24	TIM15_CNT	UFCPY または Res.																																		
	リセット値	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x28	TIM15_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x2C	TIM15_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																						
	リセット値													0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
0x30	TIM15_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]										
	リセット値																									0	0	0	0	0	0	0	0			
0x34	TIM15_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1 [19:0]																						
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x38	TIM15_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2 [19:0]																						
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x38 ~ 0x40	予約済み	Res.																																		
0x44	TIM15_BDTR	Res.	Res.	Res.	BKBD	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]	DT[7:0]												
	リセット値				0		0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x48 ~ 0x50	予約済み	Res.																																		
0x54	TIM15_DTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGF[7:0]										
	リセット値																0	0								0	0	0	0	0	0	0	0			
0x58	予約済み	Res.																																		

表 446. TIM15 レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	TIM15_TISEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL [3:0]				Res.	Res.	Res.	Res.	TI1SEL [3:0]				
	リセット値																					0	0	0	0					0	0	0	0
0x60	TIM15_AF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCOMP4P	BKCOMP3P	BKCOMP2P	BKCOMP1P	BKINP	BKCOMP8E	BKCOMP7E	BKCOMP6E	BKCOMP5E	BKCOMP4E	BKCOMP3E	BKCOMP2E	BKCOMP1E	BKINE
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x64	TIM15_AF2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCR SEL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値														0	0	0																
0x68 0x3D8	予約済み	Res.																															
0x3DC	TIM15_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]						
	リセット値														0	0	0	0				0	0	0	0	0				0	0	0	0
0x3E0	TIM15_DMAR	DMAB[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

42.8 TIM16/TIM17 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2](#) を参照してください。

42.8.1 TIMx 制御レジスタ 1 (TIMx_CR1) (x = 16~17)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DITH EN	UIFRE MAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw				rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DITHEN** : ディザリングイネーブル

0 : ディザリングは無効です。

1 : ディザリングは有効です。

注 : **DITHEN** ビットは、**CEN** ビットがリセットされているときのみ変更できます。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (tim_ker_ck) 周波数と、デッドタイムジェネレータとデジタルフィルタ (tim_tix) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

00 : $t_{DTS}=t_{tim_ker_ck}$

01 : $t_{DTS}=2*t_{tim_ker_ck}$

10 : $t_{DTS}=4*t_{tim_ker_ck}$

11 : 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

– カウンタオーバーフロー/アンダーフロー

– UG ビットのセット

– スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注 : 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

42.8.2 TIMx 制御レジスタ 2 (TIMx_CR2) (x = 16~17)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **OIS1N** : 出力アイドル状態 1 (tim_oc1n 出力)

0 : MOE=0 のとき、デッドタイム後に tim_oc1n=0

1 : MOE=0 のとき、デッドタイム後に tim_oc1n=1

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BKR レジスタの LOCK ビット)。

ビット 8 **OIS1** : 出力アイドル状態 1 (tim_oc1 出力)

0 : MOE=0 のとき、デッドタイム後に tim_oc1=0

1 : MOE=0 のとき、デッドタイム後に tim_oc1=1

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BKR レジスタの LOCK ビット)。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CCDS** : キャプチャ／比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ／比較制御更新選択

0 : キャプチャ／比較制御ビットがプリロードされるときには (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ／比較制御ビットがプリロードされているときに (CCPC=1)、COMG ビットをセットすることによって、または tim_trgi の立上がりエッジで (使用可能な場合) 更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ / 比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書込みの後、これらのビットは COM ビットがセットされたときにのみ更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

42.8.3 TIMx DMA／割込み有効レジスタ (TIMx_DIER) (x = 16~17)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
						rw	rw	rw		rw				rw	rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1DE** : キャプチャ / 比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割込みイネーブル

0 : ブレーク割込みは無効です。

1 : ブレーク割込みは有効です。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **COMIE** : COM 割込み有効

0 : COM 割込み無効

1 : COM 割込み有効

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ / 比較 1 割込みイネーブル

0 : CC1 割込みは無効です。

1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル

0 : 更新割込みは無効です。

1 : 更新割込みは有効です。

42.8.4 TIMx ステータスレジスタ (TIMx_SR) (x = 16~17)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバークャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。これは、“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバークャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割込みフラグ

このフラグは、tim_brk 入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力でアクティブレベルが検出されました。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **COMIF** : COM 割込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE, CCxNE, OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割込みがペンディング中です。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ/比較 1 割込みフラグ

このフラグはハードウェアによってセットされます。ソフトウェアによって (入力キャプチャモードまたは出力比較モード)、または TIMx_CCR1 レジスタを読み出すことによって (入力キャプチャモードのみ) クリアされます。

0 : 比較一致/入力キャプチャは発生していません。

1 : 比較一致または入力キャプチャが発生しました。

チャンネル CC1 が出力として設定されている場合 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致したとき、このフラグがセットされます。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。センターアラインモードでのフラグの設定には 3つのオプションがあります。詳細については、TIMx_CR1 レジスタの CMS ビットを参照してください。

CC1 チャンネルが入力として設定されている場合 : このビットは、TIMx_CCR1 レジスタでカウンタ値がキャプチャされた (TIMx_CCER の CC1P および CC1NP ビット設定で定義されたエッジ検知に従って、IC1 でエッジが検出された) ときに設定されます。

ビット 0 **UIF** : 更新割込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフロー（繰り返しカウンタ=0 の場合の更新）、および TIMx_CR1 レジスタの UDIS=0 の場合。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

42.8.5 TIMx イベント生成レジスタ (TIMx_EGR) (x = 16~17)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割込みまたは DMA 転送が発生します。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

42.8.6 TIMx キャプチャ／比較モードレジスタ 1 (TIMx_CCMR1) (x = 16~17)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

同じレジスタを入力キャプチャモード（このセクション）または出力比較モード（次のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F [3:0]				IC1PSC [1:0]		CC1S [1:0]	
								rW	rW	rW	rW	rW	rW	rW	rW

入力キャプチャモード

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、tim_ti1 入力をサンプリングする周波数と、tim_ti1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 2$

0010 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 4$

0011 : $f_{SAMPLING} = f_{tim_ker_ck}$, $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$, $N =$

0101 : $f_{SAMPLING} = f_{DTS}/2$, $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$, $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$, $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$, $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$, $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$, $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$, $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$, $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$, $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$, $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$, $N = 8$

ビット 3:2 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (tim_ic1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S[1:0]** : キャプチャ／比較 1 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

その他 : 予約済み

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

42.8.7 TIMx キャプチャ／比較モードレジスタ 1 [オルタネート] (TIMx_CCMR1) (x = 16~17)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

同じレジスタを出力比較モード（このセクション）または入力キャプチャモード（前のセクション）に使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力キャプチャモードと出力比較モードで異なる機能を持ちます。両方のモードを個別に組み合わせることができます（たとえば、チャンネル 1 は入力キャプチャモードで、チャンネル 2 は出力比較モード）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード :

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : tim_oc1ref は tim_ocref_clr 入力の影響を受けません。

1 : tim_oc1ref は tim_ocref_clr 入力のハイレベルが検出されるとクリアされます。

ビット 16、6:4 OC1M[3:0] : 出力比較 1 モード

これらのビットは、tim_oc1 および tim_oc1n が導き出される出力基準信号 tim_oc1ref の動作を定義します。tim_oc1ref はアクティブハイですが、tim_oc1 および tim_oc1n のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。このモードは、タイマをソフトウェアタイムベースとして使用する場合に使用できます。タイマ動作中に停止モードが有効になると、出力は停止状態になる前の状態（アクティブまたはインアクティブ）を維持します。

0001 : 一致時にチャンネル 1 をアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にハイになります。

0010 : 一致時にチャンネル 1 をインアクティブレベルにセットします。カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致すると、tim_oc1ref 信号は強制的にローになります。

0011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、tim_oc1ref は反転します。

0100 : 強制非アクティブレベル - tim_oc1ref は強制的にローになります。

0101 : 強制アクティブレベル - tim_oc1ref は強制的にハイになります。

0110 : PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111 : PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

その他 : 予約済み

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

PWM モードでは、比較結果が変化したとき、出力比較モードが「停止」モードから「PWM」モードに切り替わったとき、および出力比較モードが「強制アクティブ/インアクティブ」モードから「PWM」モードに切り替わったときに、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われません。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、トリガイベントとタイマ出力の遷移の間の待ち時間を短縮します。開始トリガの後、できるだけ早く出力パルスを開始するには、1 パルスモード (TIMx_CR1 レジスタの OPM ビットのセット) で使用する必要があります。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、tim_ocx は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S[1:0] : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、tim_ic1 は tim_ti1 に配置されます。

その他 : 予約済み

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書込み可能です。

42.8.8 TIMx キャプチャ／比較有効レジスタ (TIMx_CCER) (x = 16~17)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP** : キャプチャ／比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : tim_oc1n はアクティブハイです。

1 : tim_oc1n はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、tim_ti1fp1 の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転流イベント (COM) が発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力カインエーブル

0 : オフ - tim_oc1n はアクティブではありません。tim_oc1n のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - tim_oc1n 信号は、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

0 : OC1 のアクティブハイ (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

1 : OC1 のアクティブロー (出力モード) / エッジ検知を選択 (入力モード、下記を参照)

CC1 チャンネルが入力として設定されているときは、CC1NP/CC1P の両ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

CC1NP=0、CC1P=0 : 非反転/立上がりエッジ。この回路は TIxFP1 の立上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=0、CC1P=1 : 反転/立下がりエッジ。この回路は TIxFP1 の立下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 が反転されます (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=1 : 非反転/両エッジ/この回路は TIxFP1 の立上がりエッジと立下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

CC1NP=1、CC1P=0 : この設定は予約済みです。使用できません。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転流イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル

0 : キャプチャモードは無効/OC1 は非アクティブ (下記を参照)。

1 : キャプチャモードは有効/OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが出力として設定されている場合、OC1 レベルは、CC1E ビットの状態にかかわらず、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1NE ビットに依存します。詳細については、表 447 を参照してください。

表 447. ブレーク機能を持つ相補 tim_oc1 および tim_oc1n チャンネルの出力制御ビット (TIM16/TIM17)

制御ビット					出力状態 ⁽¹⁾		
MOE ビット	OSSI ビット	OSSR ビット	CC1E ビット	CC1NE ビット	tim_oc1 出力状態	tim_oc1n 出力状態	
1	X	X	0	0	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_oc1 = 0 tim_oc1n = 0		
		0	0	1	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_oc1 = 0	tim_oc1ref + 極性 tim_oc1n = tim_oc1ref XOR CC1NP	
		0	1	0	tim_oc1ref + 極性 tim_oc1=tim_oc1ref XOR CC1P	出力無効（タイマによって駆動されない：ハイインピーダンス） tim_oc1n = 0	
		X	1	1	tim_oc1ref + 極性 + デッドタイム	tim_oc1ref に対する相補 （tim_oc1ref ではなく） + 極性 + デッドタイム	
		1	0	1	オフ状態 （インアクティブ状態で出力有効） tim_oc1=CC1P	tim_oc1ref + 極性 tim_oc1n=tim_oc1ref XOR CC1NP	
		1	1	0	tim_oc1ref + 極性 tim_oc1=tim_oc1ref XOR CC1P	オフ状態 （インアクティブ状態で出力有効） tim_oc1n=CC1NP	
0	0	X	X	X	出力無効（タイマによって駆動されない：ハイインピーダンス）		
	1		0	0			
			0	1	オフ状態（インアクティブ状態で出力有効） 非同期に：tim_oc1=CC1P、tim_oc1n=CC1NP クロックが存在する場合：デッドタイム後、tim_oc1=OIS1 および tim_oc1n=OIS1N。ただし、OIS1 と OIS1N は、アクティブ状態における tim_oc1 と tim_oc1n の両方に対応しないことを前提とします。		
			1	0			
			1	1			

1. チャンネルの両方の出力が使用されないとき (GPIO コントローラが制御を引き継いだ場合)、OIS1、OIS1N、CC1P、および CC1NP ビットはクリアされたままでなければなりません。

注 : 相補 tim_oc1 および tim_oc1n チャンネルに接続されている外部入出力ピンの状態は、tim_oc1 および tim_oc1n チャンネルの状態と、GPIO 制御 および オルタネート機能選択レジスタに依存します。

42.8.9 TIMx カウンタ (TIMx_CNT) (x = 16~17)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx_ISR レジスタの UIF ビットの読出し専用コピー。TIMx_CR1 の UIFREMAP ビットがリセットされている場合、ビット 31 は予約済みです。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

非ディザリングモード (DITHEN = 0)

このレジスタは、カウンタ値を保持します。

ディザリングモード (DITHEN = 1)

このレジスタは、CNT[15:0] に非ディザリング部分のみを保持します。分数部分はありません。

42.8.10 TIMx プリスケアラ (TIMx_PSC) (x = 16~17)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 ($f_{tim_cnt_ck}$) は $f_{tim_psc_ck} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

42.8.11 TIMx 自動再ロードレジスタ (TIMx_ARR) (x = 16~17)

アドレスオフセット : 0x2C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **ARR[19:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 42.4.3 : 1751 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

非ディザリングモード (DITHEN = 0)

このレジスタは、ARR[15:0] に自動再ロード値を保持します。ARR[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、ARR[19:4] に整数部を保持します。ARR[3:0] ビットフィールドにはディザリング部分が含まれます。

42.8.12 TIMx 繰返しカウンタレジスタ (TIMx_RCR) (x = 16~17)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰返しカウンタの再ロード値

このビットフィールドによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの周期的な転送）を定義します。また、更新割込み生成の頻度も定義します（この割込みが有効な場合）。

繰返しダウンカウンタがゼロに達すると、更新イベントが生成され、REP 値からカウントをリスタートします。繰返しカウンタは、繰返し更新イベント UEV のみ、REP 値で再ロードされるので、TIMx_RCR レジスタへの書込みは、次の繰返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

- エッジアラインモードでは、PWM 周期の数
- センターアラインモードでは、PWM の 1/2 周期の数

42.8.13 TIMx キャプチャ／比較レジスタ 1 (TIMx_CCR1) (x = 16~17)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[19:16]			
												rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **CCR1[19:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、tim_oc1 出力に送信される値を含みます。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] に比較値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] に整数部を保持します。CCR1[3:0] ビットフィールドにはディザリング部分が含まれます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (tim_ic1) によって転送されたカウンタ値です。

非ディザリングモード (DITHEN = 0)

このレジスタは、CCR1[15:0] にキャプチャ値を保持します。CCR1[19:16] ビットはリセットされています。

ディザリングモード (DITHEN = 1)

このレジスタは、CCR1[19:4] にキャプチャ値を保持します。CCR1[3:0] ビットはリセットされています。

42.8.14 TIMx ブレークおよびデッドタイムレジスタ (TIMx_BDTR) (x = 16~17)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
			rw		rw							rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注 : BKBID、BKDSRM、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] ビットは、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初の書き込みアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 BKBID : ブレーク双方向

0 : ブレーク入力 tim_brk は入力モードです。

1 : ブレーク入力 tim_brk は双方向モードです。

双方向モード (BKBID ビットが 1 にセット) では、ブレーク入力が入力モードとオープンドレイン出力モード両方で設定されます。アクティブなブレークイベントで、ブレーク入力の低ロジックレベルをアサートし、外部デバイスに対する内部ブレークイベントを表します。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 BKDSRM : ブレーク解除

0 : ブレーク入力 tim_brk が設定されます。

1 : ブレーク入力 tim_brk は解除されます。

このビットは、アクティブなブレークソースがない場合、ハードウェアによってクリアされます。

BKDSRM ビットは、双方向出力制御 (ハイインピーダンス状態でのオープンドレイン出力) を解放するためにソフトウェアでセットしてから、ハードウェアによってリセットされ、障害状態がなくなったことを示すまでポーリングする必要があります。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 25:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **BKF[3:0]** : ブレークフィルタ

このビットフィールドは、tim_brk 入力をサンプリングする周波数と、tim_brk に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、tim_brk は非同期で動作します。

0001 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 2$

0010 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 4$

0011 : $f_{\text{SAMPLING}} = f_{\text{tim_ker_ck}}$ 、 $N = 8$

0100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、 $N = 6$

0101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、 $N = 8$

0110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、 $N = 6$

0111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、 $N = 8$

1000 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、 $N = 6$

1001 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、 $N = 8$

1010 : $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、 $N = 16$

1011 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、 $N = 6$

1100 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、 $N = 8$

1101 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 5$

1110 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 6$

1111 : $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、 $N = 8$

このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 15 **MOE** : メイン出力イネーブル

このビットは、tim_brk 入力アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : OSSR ビットに応じて、tim_oc1 および tim_oc1n 出力が無効か、強制的にアイドル状態になります。

1 : tim_oc1 および tim_oc1n 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CC1E、CC1NE ビット) がセットされている場合は有効です。

詳細については、tim_oc1/tim_oc1n イネーブルの説明を参照してください ([セクション 42.8.8 : 1833 ページのTIMx キャプチャ/比較有効レジスタ \(TIMx_CCER\) \(x = 16~17\)](#))。

ビット 14 **AOE** : 自動出力イネーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (tim_brk 入力アクティブでない場合)。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 **BKP** : ブレーク極性

0 : ブレーク入力 tim_brk はアクティブラーです。

1 : ブレーク入力 tim_brk はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 **BKE** : ブレークイネーブル

0 : ブレーク入力 (tim_brk および tim_sys_brk イベント) は無効です。

1 : ブレーク入力 (tim_brk および tim_sys_brk イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 **OSSR** : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、tim_oc1/tim_oc1n イネーブルの説明を参照してください（[セクション 42.8.8 : 1833 ページのTIMx キャプチャ/比較有効レジスタ \(TIMx_CCER\) \(x = 16~17\)](#)）。

0 : インアクティブのとき、tim_oc1/tim_oc1n 出力は無効です（タイマは出力の制御を解放し、制御はハイインピーダンス状態を強制する GPIO によって引き継がれます）。

1 : インアクティブのとき、CC1E=1 または CC1NE=1 になると、tim_oc1/tim_oc1n 出力は、インアクティブレベルで有効になります（出力は引き続きタイマで制御されます）。

注 : このビットは、**LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。**

ビット 10 **OSSI** : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、tim_oc1/tim_oc1n イネーブルの説明を参照してください（[セクション 42.8.8 : 1833 ページのTIMx キャプチャ/比較有効レジスタ \(TIMx_CCER\) \(x = 16~17\)](#)）。

0 : インアクティブのとき、tim_oc1/tim_oc1n 出力は無効です (tim_oc1/tim_oc1n イネーブル出力信号 = 0)。

1 : インアクティブのとき、tim_oc1/tim_oc1n 出力は、CC1E=1 または CC1NE=1 になると、まず、強制的にアイドルレベルになります (tim_oc1/tim_oc1n イネーブル出力信号 = 1)

注 : このビットは、**LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。**

ビット 9:8 **LOCK[1:0]** : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 — TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKBID/BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 — LOCK レベル 1 に加えて、CC 極性ビット（関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット）と OSSR および OSSI ビットも書き込みできなくなります。

11 : LOCK レベル 3 — LOCK レベル 2 に加えて、CC 制御ビット（関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMRx レジスタの OCxM および OCxPE ビット）が書き込みできなくなります。

注 : **LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで停止されます。**

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

DTG[7:5]=0xx => $DT = DTG[7:0] \times t_{dtg}$ 、ここで $t_{dtg} = t_{DTS}$

DTG[7:5]=10x => $DT = (64 + DTG[5:0]) \times t_{dtg}$ 、ここで $T_{dtg} = 2 \times t_{DTS}$

DTG[7:5]=110 => $DT = (32 + DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg} = 8 \times t_{DTS}$

DTG[7:5]=111 => $DT = (32 + DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg} = 16 \times t_{DTS}$

例 : $T_{DTS} = 125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、**LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。**

42.8.15 TIMx タイマデッドタイムレジスタ 2 (TIMx_DTR2) (x = 16~17)

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGF[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DTPE** : デッドタイムプリロードイネーブル

0 : デッドタイム値はプリロードされません。

1 : デッドタイム値プリロードは有効です。

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 16 **DTAE** : デッドタイム非対称イネーブル

0 : 立上がりおよび立下がりエッジでのデッドタイムは同一で、DTG[7:0] レジスタで定義されます。

1 : 立上がりエッジのデッドタイムは DTG[7:0] レジスタで定義され、立下がりエッジのデッドタイムは DTGF[7:0] ビットで定義されます。

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DTGF[7:0]** : デッドタイム立下がりエッジジェネレータのセットアップ

このビットフィールドでは、立下がりエッジにおいて、相補出力の間に挿入されるデッドタイムの長さを指定します。

$DTGF[7:5]=0xx \Rightarrow DTF=DTGF[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$ 。

$DTGF[7:5]=10x \Rightarrow DTF=(64+DTGF[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$ 。

$DTGF[7:5]=110 \Rightarrow DTF=(32+DTGF[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$ 。

$DTGF[7:5]=111 \Rightarrow DTF=(32+DTGF[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$ 。

例 : $T_{DTS}=125\text{ns}$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

42.8.16 TIMx 入力選択レジスタ (TIMx_TISEL) (x = 16~17)

アドレスオフセット : 0x5C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL [3:0]			
													rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TI1SEL[3:0]** : tim_ti1_in[15:0] の入力を選択します。

0000 : TIMx_CH1 入力 (tim_ti1_in0)

0001 : tim_ti1_in1

.....

1111 : tim_ti1_in15

相互接続一覧については、[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

42.8.17 TIMx オルタネート機能レジスタ 1 (TIMx_AF1) (x = 16~17)

アドレスオフセット : 0x060

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	BK CMP4P	BK CMP3P	BK CMP2P	BK CMP1P	BKINP	BK CMP8E	BK CMP7E	BK CMP6E	BK CMP5E	BK CMP4E	BK CMP3E	BK CMP2E	BK CMP1E	BKINE
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

製品固有の実装については[セクション 42.4.2: TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **BKCOMP4P** : tim_brk_cmp4 入力極性

このビットは、tim_brk_cmp4 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp4 入力はアクティブハイです。

1 : tim_brk_cmp4 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 12 BKCMP3P : tim_brk_cmp3 入力極性

このビットは、tim_brk_cmp3 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp3 入力はアクティブハイです。

1 : tim_brk_cmp3 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 11 BKCMP2P : tim_brk_cmp2 入力極性

このビットは、tim_brk_cmp2 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp2 入力はアクティブハイです。

1 : tim_brk_cmp2 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 10 BKCMP1P : tim_brk_cmp1 入力極性

このビットは、tim_brk_cmp1 入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : tim_brk_cmp1 入力はアクティブハイです。

1 : tim_brk_cmp1 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 9 BKINP : TIMx_BKIN 入力の極性

このビットは、TIMx_BKIN オルタネート機能入力の検知極性を選択します。BKP 極性ビットとともにプログラムする必要があります。

0 : TIMx_BKIN 入力はアクティブハイです。

1 : TIMx_BKIN 入力はアクティブローです。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 8 BKCMP8E : tim_brk_cmp8 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp8 を有効化します。mdf_brkx 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp8 入力は無効です。

1 : tim_brk_cmp8 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 7 BKCMP7E : tim_brk_cmp7 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp7 を有効化します。tim_brk_cmp7 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp7 入力は無効です。

1 : tim_brk_cmp7 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 6 BKCMP6E : tim_brk_cmp6 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp6 を有効化します。tim_brk_cmp6 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp6 入力は無効です。

1 : tim_brk_cmp6 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 5 **BKCMP5E** : tim_brk_cmp5 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp5 を有効化します。tim_brk_cmp5 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp5 入力は無効です。

1 : tim_brk_cmp5 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 4 **BKCMP4E** : tim_brk_cmp4 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp4 を有効化します。tim_brk_cmp4 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp4 入力は無効です。

1 : tim_brk_cmp4 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 3 **BKCMP3E** : tim_brk_cmp3 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp3 を有効化します。tim_brk_cmp3 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp3 入力は無効です。

1 : tim_brk_cmp3 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 2 **BKCMP2E** : tim_brk_cmp2 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp2 を有効化します。tim_brk_cmp2 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp2 入力は無効です。

1 : tim_brk_cmp2 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 1 **BKCMP1E** : tim_brk_cmp1 イネーブル

このビットは、タイマの tim_brk 入力に対して tim_brk_cmp1 を有効化します。tim_brk_cmp1 出力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : tim_brk_cmp1 入力は無効です。

1 : tim_brk_cmp1 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 0 **BKINE** : TIMx_BKIN 入力イネーブル

このビットは、タイマの tim_brk 入力に対して TIMx_BKIN オルタネート機能入力を有効化します。TIMx_BKIN 入力は、ほかの tim_brk ソースとの「論理和」がとられます。

0 : TIMx_BKIN 入力は無効です。

1 : TIMx_BKIN 入力は有効です。

注： このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

42.8.18 TIMx オルタネート機能レジスタ 2 (TIMx_AF2) (x = 16~17)

アドレスオフセット : 0x064

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCRSEL[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **OCRSEL[2:0]** : tim_ocref_clr ソース選択

これらのビットは、tim_ocref_clr 入力ソースを選択します。

000 : tim_ocref_clr0

001 : tim_ocref_clr1

010 : tim_ocref_clr2

011 : tim_ocref_clr3

100 : tim_ocref_clr4

101 : tim_ocref_clr5

110 : tim_ocref_clr6

111 : tim_ocref_clr7

製品固有の実装については[セクション 42.4.2 : TIM15/TIM16/TIM17 ピンおよび内部信号](#)を参照してください。

注： これらのビットは、LOCK レベル 1 がプログラムされている場合(TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

42.8.19 TIMx DMA 制御レジスタ (TIMx_DCR) (x = 16~17)

アドレスオフセット : 0x3DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **DBSS[3:0]** : DMA パーストソース選択

このビットフィールドは、DMA パースト転送をトリガする割込みソースを定義します (タイマは、TIMx_DMAR アドレスに対して読出しまたは書込みアクセスが行われるときにパースト転送を認識します)。

0000 : 予約済みです。

0001 : 更新

0010 : CC1

その他 : 予約済み

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのフィールドは、転送数など DMA 転送長（タイマは、TIMx_DMAR アドレスに対して読みまたは書き込みアクセスが行われるときにバースト転送を認識します）を指定します。転送は、ハーフワードまたはバイトです（以下の例を参照）。

00000 : 1 回転送
00001 : 2 回転送、
00010 : 3 回転送、
.....
10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのフィールドは、DMA 転送のベースアドレスを指定します（TIMx_DMAR アドレスを通じて読みし／書き込みアクセスが行われるとき）。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx_CR1
00001 : TIMx_CR2
00010 : TIMx_SMCR
.....

例 : 次の転送を考えます : DBL = 7 転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

42.8.20 完全転送の TIM16/TIM17 DMA アドレス (TIMx_DMAR) (x = 16~17)

アドレスオフセット : 0x3E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAB[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **DMAB[31:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読みまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL は TIMx_DCR 内で設定）。

42.8.21 TIM16/TIM17 レジスタマップ

TIM16/TIM17 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 448. TIM16/TIM17レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UJFREMAP	Res.		CKD[1:0]		ARPE		Res.	Res.		OPM	URS	UDIS	CEN
	リセット値																					0		0	0	0				0	0	0	0		
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC		
	リセット値																							0	0					0	0		0		
0x08	予約済みです。	Res.																																	
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE		COMIE		Res.	Res.	Res.	CC1IE	UIE	
	リセット値																							0	0	0		0				0	0		
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF		
	リセット値																							0		0		0				0	0		
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG		
	リセット値																									0		0				0	0		
0x18	TIMx_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F [3:0]				IC1 PSC [1:0]		CC1S [1:0]				
	リセット値																									0	0	0	0	0	0	0	0		
	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1CE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値																0										0	0	0	0	0	0	0	0	
0x1C	予約済みです。	Res.																																	
0x20	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E	
	リセット値																													0	0	0	0		
0x24	TIMx_CNT	UJFCPY または Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[19:0]																					
	リセット値													0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

表 448. TIM16/TIM17レジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x30	TIMx_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]													
	リセット値																									0	0	0	0	0	0	0	0	0					
0x34	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1 [19:0]																									
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x38 - 0x40	予約済みです。	Res.																																					
0x44	TIMx_BDTR	Res.	Res.	Res.	BKID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]														
	リセット値				0		0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x48 - 0x50	予約済みです。	Res.																																					
0x54	TIMx_DTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTPE	DTAE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTGF[7:0]													
	リセット値														0	0									0	0	0	0	0	0	0	0	0						
0x58	予約済みです。	Res.																																					
0x5C	TIMx_TISEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T1SEL [3:0]								
	リセット値																													0	0	0	0						
0x60	TIMx_AF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCOMP4P	BKCOMP3P	BKCOMP2P	BKCOMP1P	BKINP	BKCOMP8E	BKCOMP7E	BKCOMP6E	BKCOMP5E	BKCOMP4E	BKCOMP3E	BKCOMP2E	BKCOMP1E	BKINE							
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1						
0x64	TIMx_AF2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OCR SEL[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.						
	リセット値													0	0	0																							
0x68 - 0x3D8	予約済みです。	Res.																																					
0x3DC	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBSS[3:0]				Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]											
	リセット値													0	0	0	0				0	0	0	0	0				0	0	0	0	0						
0x3E0	TIMx_DMAR	DMAB[31:0]																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

43 低消費電力タイマ (LPTIM)

43.1 概要

LPTIM は、消費電力削減の究極的な進展を利用した 16 ビットタイマです。クロックソースの多様性により、LPTIM は STANDBY モードを除くすべての電力モードで実行し続けることができます。内部クロックソースがなくても実行できるため、LPTIM は、一部のアプリケーションで役立つ「パルスカウンタ」として使用することができます。また、LPTIM はシステムを低消費電力モードからウェイクアップできるため、消費電力が極端に低い「タイムアウト機能」の実現に適しています。

LPTIM の柔軟性の高いクロック方式は、必要な機能性とパフォーマンスを提供しながら、消費電力を最小化します。

43.2 LPTIM の主な機能

- 16 ビットアップカウンタ
- 8 つの分周比 (1、2、4、8、16、32、64、128) を持つ 3 ビットプリスケアラ
- 選択可能なクロック
 - 内部クロックソース：設定可能な内部クロックソース (RCC セクションを参照)
 - LPTIM 入力経由の外部クロックソース (LP オシレータが実行していないときに動作、パルスカウンタアプリケーションによって使用)
- 16 ビット ARR 自動再ロードレジスタ
- 16 ビットキャプチャ／比較レジスタ
- 連続／ワンショットモード
- 選択可能なソフトウェア／ハードウェア入力トリガ
- プログラム可能なデジタルグリッチフィルタ
- 設定可能な出力：パルス、PWM
- 設定可能な I/O 極性
- エンコーダモード
- 繰り返しカウンタ
- 次の機能を持つ、最大 2 つの独立チャネル：
 - 入力キャプチャ
 - PWM 生成 (エッジアラインモード)
 - ワンパルスモード出力
- 10 種のイベント時の割込み生成
- 以下のイベント時の DMA リクエスト生成：
 - 更新イベント
 - 入力キャプチャ

43.3 LPTIM の実装

表 449 に、STM32H563/H573および STM32H562 デバイスでの LPTIM の実装を示します。LPTIM1、2、3、5、および 6 にはすべての機能セットが実装されています。LPTIM4 では、それよりも少ない機能がサポートされています。

表 449. STM32H563/H573および STM32H562 LPTIM 機能

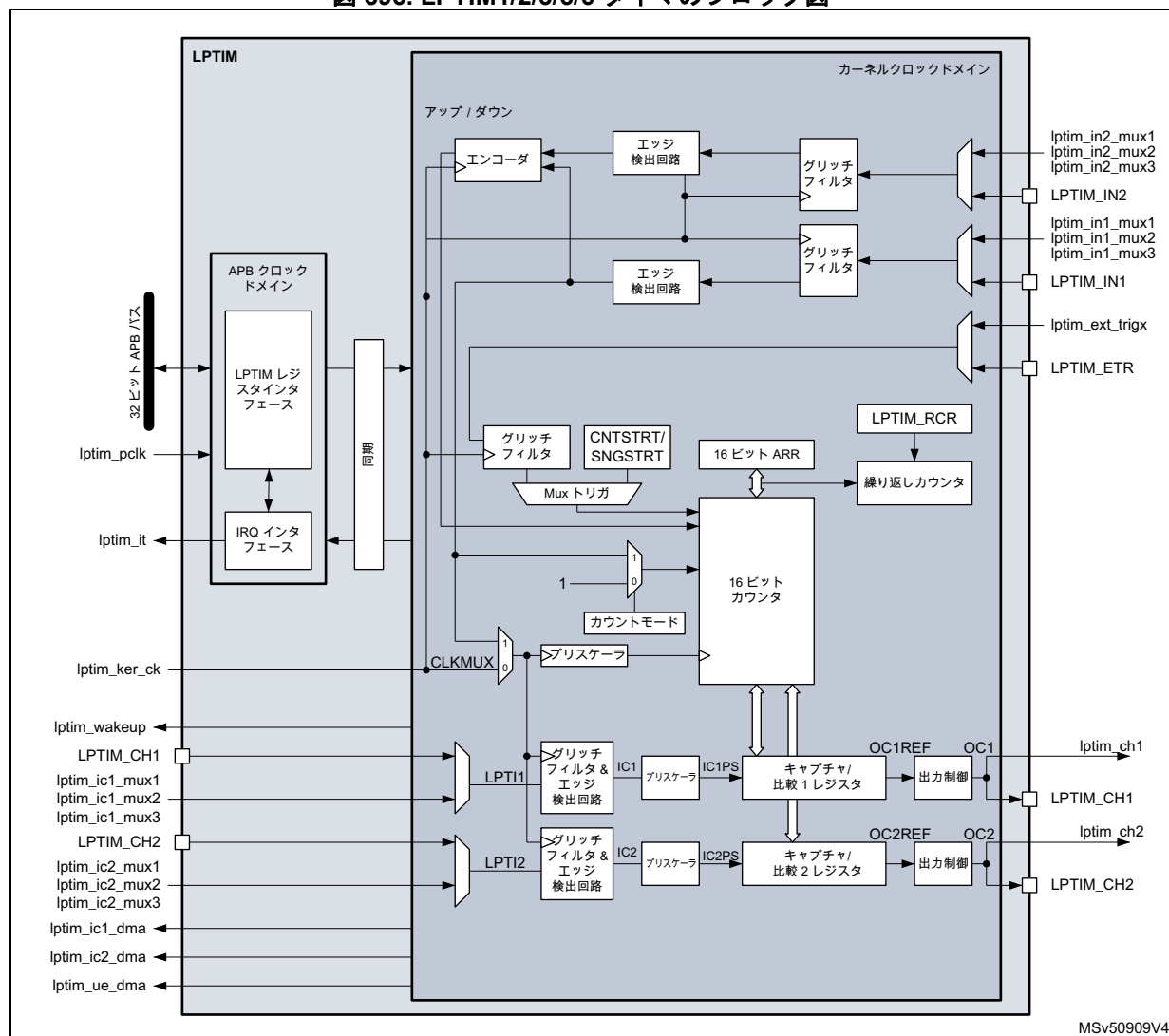
LPTIM のモード／機能 ⁽¹⁾	LPTIM1	LPTIM2	LPTIM3	LPTIM4	LPTIM5	LPTIM6
エンコーダモード	X	X	X	-	X	X
PWM モード	X	X	X	X	X	X
入力キャプチャ	X	X	X	-	X	X
チャネル数	2	2	2	0	2	2
DMA リクエスト数	3	3	3	0	3	3
STOP モードからのウェイクアップ	X	X	X	X	X	X
自律モード	-	-	-	-	-	-

1. X: サポートされています。

43.4 LPTIM の機能説明

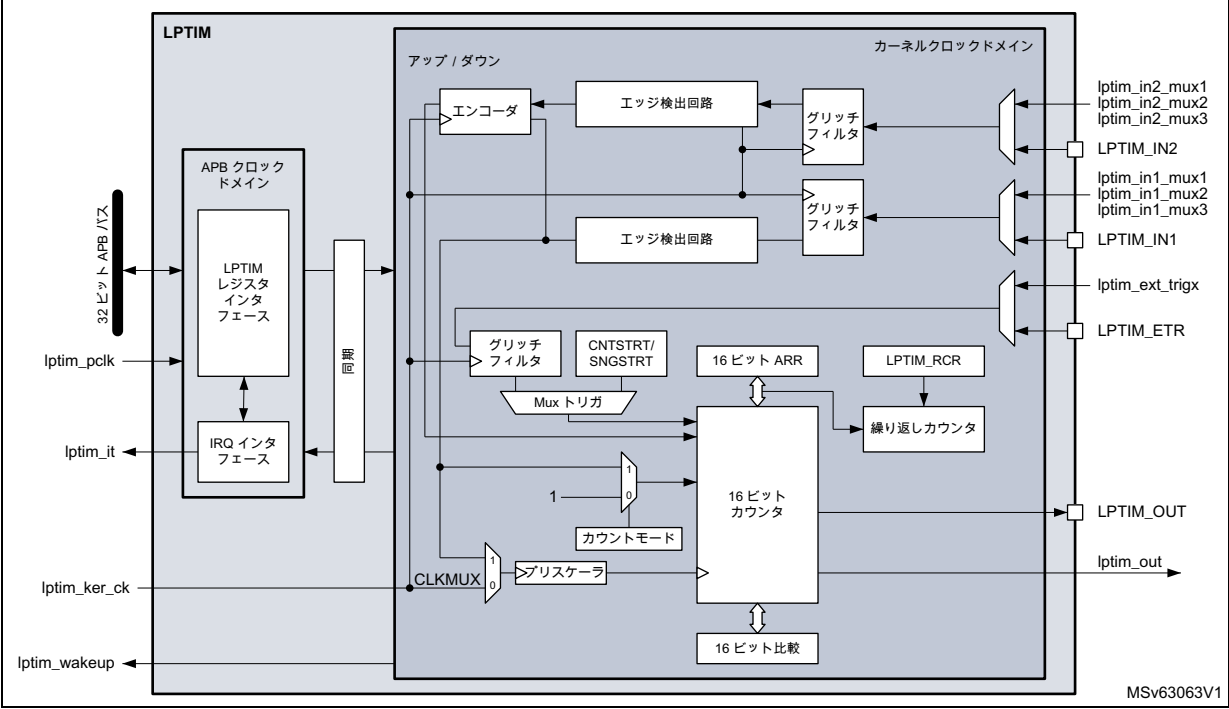
43.4.1 LPTIM ブロック図

図 595. LPTIM1/2/3/5/6 タイマのブロック図⁽¹⁾



1. 一部の IO は利用できない場合があります。セクション 43.4.2 : LPTIM ピンおよび内部信号を参照してください。

図 596. LPTIM4 タイマのブロック図⁽¹⁾



1. 一部の IO は利用できない場合があります。セクション 43.4.2 : LPTIM ピンおよび内部信号を参照してください。

43.4.2 LPTIM ピンおよび内部信号

次の表に、LPTIM のピンと内部信号の一覧を示します。

表 450. LPTIM1/2/3/5/6 の入出力ピン

名前	信号タイプ	説明
LPTIM_IN1	デジタル入力	GPIO ピンからマルチプレクサ入力 0 への LPTIM 入力 1
LPTIM_IN2	デジタル入力	GPIO ピンからマルチプレクサ入力 0 への LPTIM 入力 2
LPTIM_ETR	デジタル入力	LPTIM 外部トリガ GPIO ピン
LPTIM_CH1	デジタル入出力	LPTIM チャンネル 1 入出力 GPIO ピン
LPTIM_CH2	デジタル入出力	LPTIM チャンネル 2 入出力 GPIO ピン

表 451. LPTIM4 の入出力ピン

名前	信号タイプ	説明
LPTIM_IN1	デジタル入力	GPIO ピンからマルチプレクサ入力 0 への LPTIM 入力 1
LPTIM_ETR	デジタル入力	LPTIM 外部トリガ GPIO ピン
LPTIM_OUT	デジタル出力	LPTIM 出力 GPIO ピン

表 452. LPTIM1/2/3/5/6 の内部信号

名前	信号タイプ	説明
lptim_pclk	デジタル入力	LPTIM APB クロックドメイン
lptim_ker_ck	デジタル入力	LPTIM カーネルクロック
lptim_in1_mux1	デジタル入力	マルチプレクサ入力 1 に接続される内部 LPTIM 入力 1
lptim_in1_mux2	デジタル入力	マルチプレクサ入力 2 に接続される内部 LPTIM 入力 1
lptim_in1_mux3	デジタル入力	マルチプレクサ入力 3 に接続される内部 LPTIM 入力 1
lptim_in2_mux1	デジタル入力	マルチプレクサ入力 1 に接続される内部 LPTIM 入力 2
lptim_in2_mux2	デジタル入力	マルチプレクサ入力 2 に接続される内部 LPTIM 入力 2
lptim_in2_mux3	デジタル入力	マルチプレクサ入力 3 に接続される内部 LPTIM 入力 2
lptim_ic1_mux1	デジタル入力	マルチプレクサ入力 1 に接続される内部 LPTIM 入力キャプチャ 1
lptim_ic1_mux2	デジタル入力	マルチプレクサ入力 2 に接続される内部 LPTIM 入力キャプチャ 1
lptim_ic1_mux3	デジタル入力	マルチプレクサ入力 3 に接続される内部 LPTIM 入力キャプチャ 1
lptim_ic2_mux1	デジタル入力	マルチプレクサ入力 1 に接続される内部 LPTIM 入力キャプチャ 2
lptim_ic2_mux2	デジタル入力	マルチプレクサ入力 2 に接続される内部 LPTIM 入力キャプチャ 2
lptim_ic2_mux3	デジタル入力	マルチプレクサ入力 3 に接続される内部 LPTIM 入力キャプチャ 2
lptim_ext_trigx	デジタル入力	LPTIM 外部トリガ入力 x
lptim_it	デジタル出力	LPTIM グローバル割込み
lptim_wakeup	デジタル出力	LPTIM ウェイクアップイベント
lptim_ic1_dma	デジタル出力	LPTIM 入力キャプチャ 1 DMA リクエスト
lptim_ic2_dma	デジタル出力	LPTIM 入力キャプチャ 2 DMA リクエスト
lptim_ue_dma	デジタル出力	LPTIM 更新イベントの DMA リクエスト

表 453. LPTIM4 内部信号

名前	信号タイプ	説明
lptim_pclk	デジタル入力	LPTIM APB クロックドメイン
lptim_ker_ck	デジタル入力	LPTIM カーネルクロック
lptim_in1_mux1	デジタル入力	マルチプレクサ入力 1 に接続される内部 LPTIM 入力 1
lptim_in1_mux2	デジタル入力	マルチプレクサ入力 2 に接続される内部 LPTIM 入力 1
lptim_in1_mux3	デジタル入力	マルチプレクサ入力 3 に接続される内部 LPTIM 入力 1
lptim_ext_trigx	デジタル入力	LPTIM 外部トリガ入力 x
lptim_out	デジタル出力	LPTIM カウンタ出力
lptim_it	デジタル出力	LPTIM グローバル割込み
lptim_wakeup	デジタル出力	LPTIM ウェイクアップイベント

43.4.3 LPTIM 入力およびトリガマッピング

LPTIM 外部トリガおよび入力の接続について、以下に詳しく説明します。

表 454. LPTIM1/2/3/4/5/6 外部トリガ接続

TRIGSEL	外部トリガ					
	LPTIM1	LPTIM2	LPTIM3	LPTIM4	LPTIM5	LPTIM6
lptim_ext_trig0	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO
lptim_ext_trig1	rtc_alra_trg	rtc_alra_trg	rtc_alra_trg	rtc_alra_trg	rtc_alra_trg	rtc_alra_trg
lptim_ext_trig2	rtc_alrb_trg	rtc_alrb_trg	rtc_alrb_trg	rtc_alrb_trg	rtc_alrb_trg	rtc_alrb_trg
lptim_ext_trig3	tamp_trg1	tamp_trg1	tamp_trg1	tamp_trg1	tamp_trg1	tamp_trg1
lptim_ext_trig4	tamp_trg2	gpdma_ch0_tcf	-	-	-	-
lptim_ext_trig5	gpdma_ch1_tcf	gpdma_ch4_tcf	-	-	-	-
lptim_ext_trig6	-	-	-	-	-	-
lptim_ext_trig7	-	-	-	-	-	-

表 455. LPTIM1/2/3/5/6 の入力 1 の接続

lptim_in1_mux	LPTIM1/2/3/4 の入力 1 の接続先
lptim_in1_mux0	GPIO
lptim_in1_mux1	-
lptim_in1_mux2	-
lptim_in1_mux3	-

表 456. LPTIM1/2/3/5/6 の入力 2 の接続

lptim_in2_mux	LPTIM1/2 の入力 2 の接続先
lptim_in2_mux0	GPIO
lptim_in2_mux1	-
lptim_in2_mux2	-
lptim_in2_mux3	-

表 457. LPTIM1/2/3/5/6 の入力キャプチャ 1 の接続

lptim_ic1_mux	LPTIM1/2/3 入力キャプチャ 1 の接続先
lptim_ic1_mux0	GPIO
lptim_ic1_mux1	-
lptim_ic1_mux2	-
lptim_ic1_mux3	-

表 458. LPTIM1 入力キャプチャ 2 の接続

lptim_ic2_mux	LPTIM1 入力キャプチャ 2 の接続先
lptim_ic2_mux0	I/O
lptim_ic2_mux1	LSI
lptim_ic2_mux2	LSE
lptim_ic2_mux3	-

表 459. LPTIM2 入力キャプチャ 2 の接続

lptim_ic2_mux	LPTIM2 入力キャプチャ 2 の接続先
lptim_ic2_mux0	I/O
lptim_ic2_mux1	HSI/1024
lptim_ic2_mux2	CSI/128
lptim_ic2_mux3	HSI/8

表 460. LPTIM3/5/6 入力キャプチャ 2 の接続

lptim_ic2_mux	LPTIM3 入力キャプチャ 2 の接続先
lptim_ic2_mux0	I/O
lptim_ic2_mux1	-
lptim_ic2_mux2	-
lptim_ic2_mux3	-

43.4.4 LPTIM のリセットとクロック

LPTIM のクロックには、いくつかのクロックソースを使用できます。RCC を介して選択できる任意の設定可能な内部クロックソースになることができる内部クロック信号を使用してクロックを供給できます（詳細については、RCC のセクションを参照してください）。また、LPTIM のクロックには、外部 Input1 に入力された外部クロック信号を使用することもできます。外部クロックソースでクロック供給されるときには、LPTIM は、次の 2 つの構成のいずれかで動作します。

- 最初の構成では、LPTIM のクロックは外部信号によって供給されますが、同時に内部クロック信号も、設定可能な内部クロックソースから LPTIM に供給されます（RCC のセクションを参照）。
- 2 番目の構成では、LPTIM のクロックは外部 Input1 を通じて外部クロックソースによってのみ供給されます。この構成は、低消費電力モードになった後、すべての埋め込みオシレータがオフになるときに、タイムアウト機能またはパルスカウンタ機能を実現するために使用されます。

CKSEL ビットおよび COUNTMODE ビットに書き込むことによって、LPTIM が外部クロックソースと内部クロックソースのいずれを使用するかを決めることができます。

外部クロックソースを使用するように設定されたときには、CKPOL ビットを使用して外部クロック信号のアクティブエッジを選択します。両方のエッジがアクティブとして設定された場合は、内部クロック信号も供給する必要があります（最初の構成）。この場合、内部クロック信号の周波数は、外部クロック信号の周波数の 4 倍以上である必要があります。

43.4.5 グリッチフィルタ

LPTIM 入力の外部入力 (GPIO にマップされたもの) または内部入力 (チップレベルで他の内蔵ペリフェラルにマップされたもの) はデジタルフィルタによって保護され、グリッチとノイズの影響が LPTIM 内部に伝播されるのを防ぎます。これは、誤ったカウントまたはトリガを避けるためです。

デジタルフィルタをアクティブにする前に、内部クロックソースを LPTIM に供給する必要があります。これは、フィルタの正しい動作を保証するために必要です。

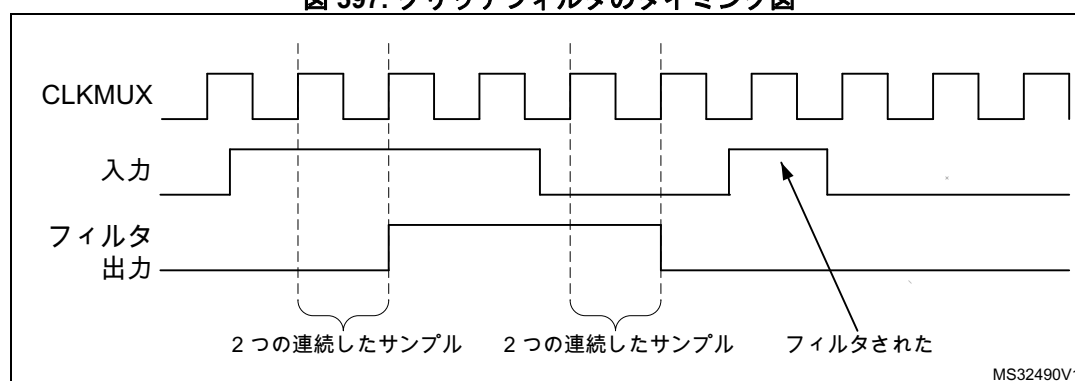
デジタルフィルタは、3 つのグループに分けられます。

- 最初のグループのデジタルフィルタは、LPTIM の内部または外部入力を保護します。デジタルフィルタの期間は CKFLT ビットによって制御されます。
- 2 番目のグループのデジタルフィルタは、LPTIM の内部または外部トリガ入力を保護します。デジタルフィルタの期間は TRGFLT ビットによって制御されます。
- 3 番目のグループのデジタルフィルタは、LPTIM の内部または外部入力キャプチャを保護します。デジタルフィルタの期間は ICxF ビットによって制御されます。

注： デジタルフィルタの期間は、グループごとに制御されます。同じグループ内の各デジタルフィルタの期間を個別に設定することはできません。

フィルタの期間は、信号のレベル変化を有効な遷移とみなすために、LPTIM 入力の 1 つで検出される連続した等しいサンプルの数に基づいて決められます。図 597 に、2 つの連続サンプルがプログラムされた場合のグリッチフィルタの動作例を示します。

図 597. グリッチフィルタのタイミング図



注： 内部クロック信号が供給されない場合は、CKFLT、ICxF、および TRGFLT ビットを 0 にセットすることによって、デジタルフィルタを無効にする必要があります。その場合、外部アナログフィルタを使用して、LPTIM の外部入力をグリッチから保護できます。

43.4.6 プリスケアラ

LPTIM 16 ビットカウンタの前には、設定可能な 2 のべき乗プリスケアラがあります。プリスケアラの分周比は PRESC[2:0] 3 ビットフィールドによって制御されます。下の表に、可能な分周比を示します。

表 461. プリスケアラ分周比

プログラミング	分周比
000	/1
001	/2
010	/4

表 461. プリスケアラ分周比（続き）

プログラミング	分周比
011	/8
100	/16
101	/32
110	/64
111	/128

43.4.7 トリガマルチプレクサ

LPTIM カウンタは、ソフトウェアによって、または 8 つのトリガ入力の 1 つのアクティブエッジの検出後に開始できます。

LPTIM のトリガソースを決めるには、TRIGEN[1:0] が使用されます。

- TRIGEN[1:0] が 00 の場合、LPTIM カウンタは、CNTSTRT または SNGSTRT ビットがソフトウェアによってセットされるとすぐに開始します。TRIGEN[1:0] の残りの 3 つの可能な値は、トリガ入力によって使用されるアクティブエッジを設定するために使用されます。LPTIM カウンタは、アクティブエッジが検出されるとすぐに開始します。
- TRIGEN[1:0] が 00 以外のときには、カウンタの開始に使用される 8 つのトリガ入力の 1 つを選択するために、TRIGSEL[2:0] が使用されます。

外部トリガは、LPTIM の場合、非同期信号とみなされます。したがって、同期のために、トリガ検出後、タイマが実行を開始するまでに 2 カウンタクロック周期の遅延が必要です。

タイマがすでに開始しているときに新しいトリガイベントが発生した場合、無視されます（タイムアウト機能が有効な場合を除きます）。

注： SNGSTRT/CNTSTRT ビットをセットする前に、タイマが有効になっている必要があります。タイマが無効なときにこれらのビットに書き込むと、ハードウェアによって破棄されます。

注： ソフトウェアによってカウンタを開始する場合（TRIGEN[1:0] = 00）、LPTIM_CR レジスタが更新（SNGSTRT または CNTSTRT ビットのいずれかをセット）されてからカウンタが有効に開始されるまで 3 カーネルクロックサイクルの遅延が発生します。

43.4.8 動作モード

LPTIM には 2 つの動作モードがあります。

- 連続モード：タイマはフリーランニングし、トリガイベントによって開始し、タイマが無効にされるまで停止しません。
- ワンショットモード：タイマはトリガイベントによって開始し、LPTIM 更新イベントが生成されると停止します。

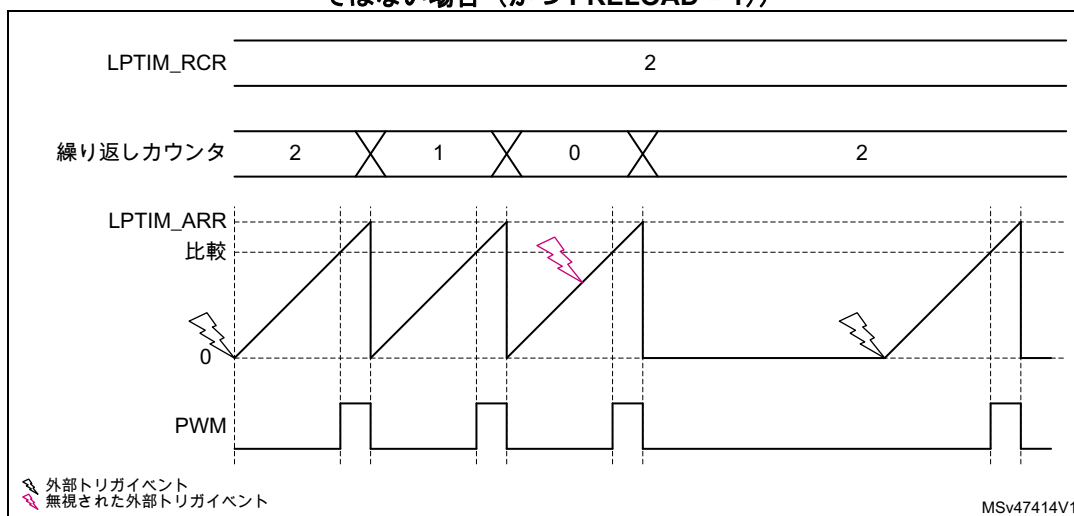
ワンショットモード

ワンショットカウントを有効にするには、SNGSTRT ビットをセットする必要があります。

新しいトリガイベント発生はタイマを再開します。カウンタが開始してから、次の LPTIM 更新イベントまでに発生したトリガイベントは破棄されます。

外部トリガが選択された場合、SNGSTRT ビットがセットされた後で外部トリガイベントを受信し、繰り返しカウンタが停止した後（更新イベント後）、かつ繰り返しレジスタの内容が 0 ではない場合、繰り返しカウンタは繰り返しレジスタによってすでに入力された値を再ロードし、新たなワンショットカウントサイクルを開始します（図 598 を参照）。

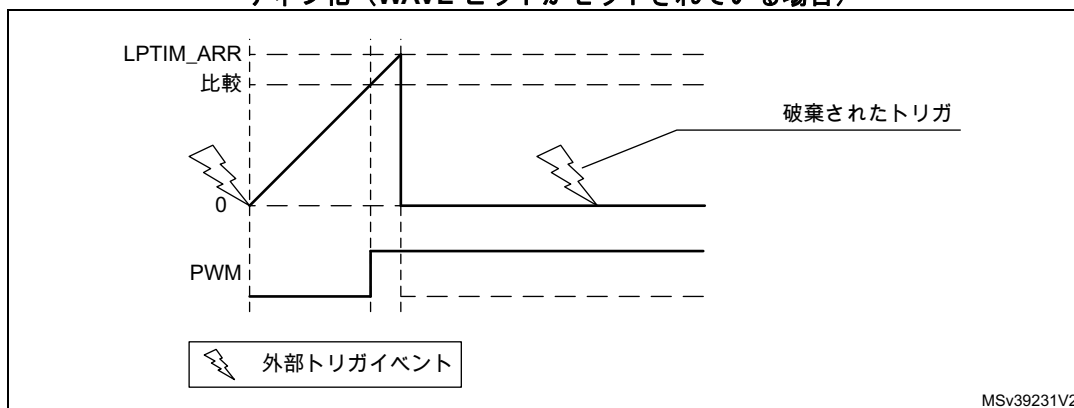
図 598. LPTIM 出力波形、シングルカウントモードの設定（繰り返しレジスタの内容が 0 ではない場合（かつ PRELOAD = 1））



- セットワンスモードのアクティブ化 :

LPTIM_CFGR レジスタの WAVE ビットフィールドがセットされると、セットワンスモードがアクティブになることに注意してください。この場合、最初のトリガに続いてカウンタが一度だけ開始され、その後発生したトリガイベントはすべて破棄されます（図 599 を参照）。

図 599. LPTIM 出力波形、シングルカウントモードの設定、セットワンスモードのアクティブ化（WAVE ビットがセットされている場合）



ソフトウェアによる開始の場合（TRIGEN[1:0] = 00）、SNGSTRT をセットすると、カウンタはワンショットカウントを開始します。

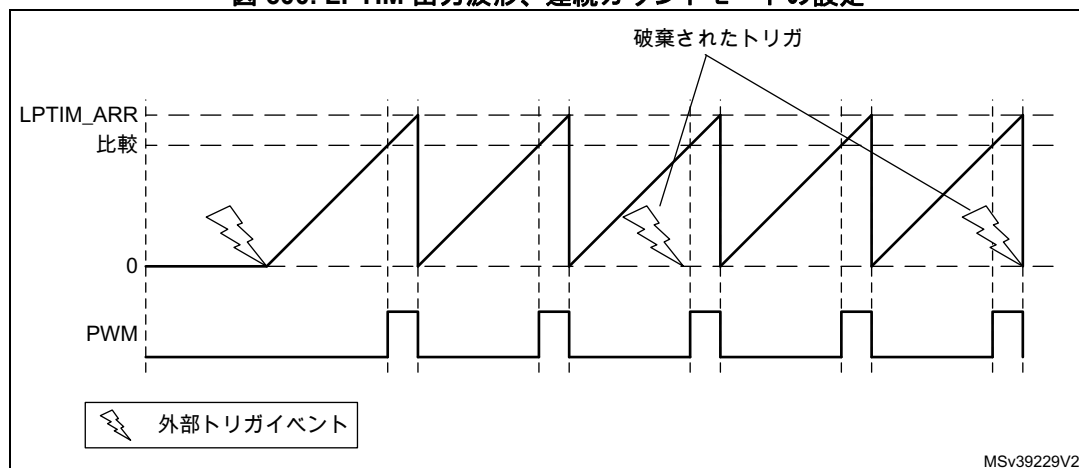
連続モード

連続カウントを有効にするには、CNTSTRT ビットをセットする必要があります。

外部トリガが選択された場合、CNTSTRT がセットされた後に外部トリガイベントが着信すると、カウンタは連続カウントを開始します。その後発生した外部トリガイベントはすべて、図 600 に示すように破棄されます。

ソフトウェアによる開始の場合（TRIGEN[1:0] = 00）、CNTSTRT をセットすると、カウンタは連続カウントを開始します。

図 600. LPTIM 出力波形、連続カウントモードの設定



SNGSTRT および CNTSTRT ビットは、タイマが有効なときのみ（イネーブルビットが1にセットされている）、セットできます。動作中にワンショットモードから連続モードに変更することが可能です。

以前に連続モードが選択されていた場合、SNGSTRT をセットすると、LPTIM はワンショットモードに切り替わります。カウンタ（アクティブな場合）は、LPTIM 更新イベントが生成されるとすぐに停止します。

以前にワンショットモードが選択されていた場合、CNTSTRT をセットすると、LPTIM は連続モードに切り替わります。カウンタ（アクティブな場合）は、ARR に達するとすぐに再開始します。

43.4.9 タイムアウト機能

選択されたトリガ入力のアクティブエッジの検出を使用して、LPTIM カウンタをリセットできます。この機能は、TIMOUT ビットで制御されます。

最初のトリガイイベントでタイマが開始し、その後のトリガイイベントで LPTIM カウンタと繰り返しカウンタがリセットされ、タイマが再開始します。

低消費電力タイムアウト機能を実現できます。タイムアウト値は、比較値に対応します。予期された時間内にトリガが発生しなかった場合、比較一致イベントによって MCU がウェイクアップします。

43.4.10 波形生成

2 つの 16 ビットレジスタ LPTIM_ARR（自動再ロードレジスタ）と LPTIM_CCRx（キャプチャ／比較レジスタ）は、LPTIM 出力にいくつかの異なった波形を生成するために使用されます。

タイマは次の波形を生成できます。

- PWM モード：LPTIM 出力は、LPTIM_CNT のカウンタ値が LPTIM_CCRx の比較値を超えるとすぐにセットされます。LPTIM 出力は、LPTIM_ARR レジスタと LPTIM_CNT レジスタの間で一致が発生するとすぐにリセットされます。詳細については、[セクション 43.4.19：PWM モード](#)を参照してください。
- ワンパルスモード：出力波形は、最初のパルスについては PWM モードの波形と同様であり、その後、出力はリセットされたままです。
- セットワンスモード：出力波形はワンパルスモードと同様ですが、出力は最後の信号レベルに保たれます（設定された出力極性に応じて）。

上記のモードでは、LPTIM_ARR レジスタの値は LPTIM_CCRx レジスタの値より厳密に大きい必要があります。

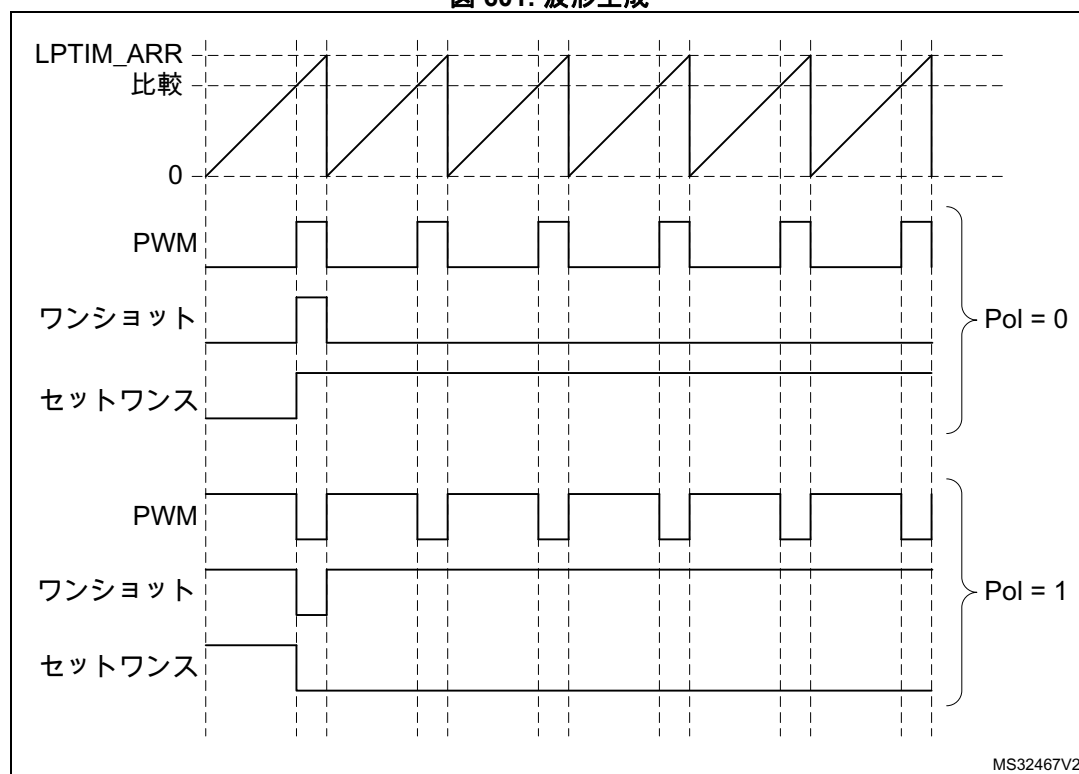
LPTIM 出力波形は、WAVE ビットによって次のように設定できます。

- WAVE ビットを 0 にリセットすると、LPTIM は CNTSTRT または SNGSTRT のセットされているビットに応じて、PWM 波形またはワンパルス波形のいずれかを生成します。
- WAVE ビットを 1 にセットすると、LPTIM はセットワンスモード波形を生成します。

WAVPOL/CCxP ビットは、LPTIM 出力の極性を制御します。変更はただちに有効になるので、極性が再設定されると、タイマが有効になる前でも、出力のデフォルト値はただちに変更されます。

LPTIM クロック周波数を 2 分周した周波数までの信号を生成できます。図 601 に、LPTIM 出力で生成できる 3 つの波形を示します。また、WAVPOL/CCxP ビットを使用して極性を変更したときの効果も示します。

図 601. 波形生成



43.4.11 レジスタの更新

LPTIM_ARR レジスタ、LPTIM_RCR レジスタ、LPTIM_CCRx レジスタは、APB バス書き込み操作の直後、またはタイマがすでに開始していた場合は次の LPTIM 更新イベントでの同期時に更新されます。

PRELOAD ビットによって、LPTIM_ARR、LPTIM_RCR、および LPTIM_CCRx レジスタの更新方法を制御します。

- PRELOAD ビットを“0”にリセットすると、LPTIM_ARR レジスタ、LPTIM_RCR レジスタ、および LPTIM_CCRx レジスタは書き込みアクセスの直後に更新されます。
- PRELOAD ビットを“1”にセットすると、LPTIM_ARR レジスタ、LPTIM_RCR レジスタ、および LPTIM_CCRx レジスタは、次の LPTIM 更新イベント時に更新されます（タイマがすでに開始していた場合）。

LPTIM APB インタフェースと LPTIM カーネルロジックは異なるクロックを使用するので、APB 書込みから、これらの値がカウンタコンパレータで使用可能になるまで、遅延があります。この遅延時間内は、これらのレジスタに追加で書き込めません。

LPTIM_ISR レジスタの ARROK フラグ、REPOK フラグ、CMPxOK フラグは、それぞれ、LPTIM_ARR レジスタ、LPTIM_RCR レジスタ、LPTIM_CCRx レジスタへの書込み操作が完了したことを示します。

LPTIM_ARR、LPTIM_RCR、または LPTIM_CCRx レジスタへの書込みの後、同じレジスタへの新しい書込み操作は、前の書込み操作が完了してからでなければ実行できません。ARROK フラグ、REPOK フラグ、または CMPxOK フラグがそれぞれセットされる前に、続けて書込みを行うと、予測不能な結果になります。

43.4.12 カウンタモード

LPTIM カウンタを使用して、LPTIM Input1 の外部イベントをカウントするか、内部クロックサイクルをカウントすることができます。CKSEL ビットおよび COUNTMODE ビットは、カウンタの更新にどのソースを使用するかを決定します。

LPTIM が Input1 の外部イベントをカウントするように設定された場合、カウンタは、CKPOL[1:0] ビットに書き込まれた値に応じて、立上がりエッジ、立下がりエッジ、または両方のエッジで更新できます。

CKSEL および COUNTMODE の値に応じて、以下に示すカウントモードを選択できます。

- CKSEL = 0 : LPTIM のクロックは、内部クロックソースによって供給されます。
 - COUNTMODE = 0
LPTIM が内部クロックソースによってクロック供給されるように設定され、LPTIM カウンタが各内部クロックパルス後に更新されるように設定されます。
 - COUNTMODE = 1
LPTIM 外部 Input1 は、LPTIM に供給される内部クロックでサンプリングされます。
したがって、イベントをミスしないためには、外部 Input1 信号の変化の周波数が、LPTIM に供給される内部クロックの周波数を超えてはなりません。また、LPTIM に供給される内部クロックを分周してはなりません (PRESC[2:0] = 000)。
- CKSEL = 1 : LPTIM のクロックは、外部クロックソースによって供給されます。
COUNTMODE の値は無視されます。
この構成では、LPTIM は内部クロックソースを必要としません (グリッチフィルタ有効である場合を除く)。LPTIM 外部 Input1 に入力された信号が LPTIM のシステムクロックとして使用されます。この構成は、埋め込みオシレータを有効にしない動作モードに適しています。
この構成の場合、LPTIM カウンタは、Input1 クロック信号の立上がりエッジまたは立下がりエッジで更新できますが、立上がりと立下がりの両方のエッジで更新することはできません。
LPTIM 外部 Input1 に入力された信号は LPTIM カーネルロジックのクロック動作にも使用されるので、カウンタがインクリメントされる前 (LPTIM が有効にされた後)、初期遅延があります。より正確には、LPTIM 外部 Input1 の (LPTIM が有効になった後の) 最初の 5 つのアクティブエッジは失われます。

43.4.13 タイマ有効

LPTIM_CR レジスタのイネーブルビットは、LPTIM カーネルロジックを有効化/無効化するために使用されます。イネーブルビットをセットした後、LPTIM が実際に有効になるまで、2 カウンタクロックの遅延が必要です。

LPTIM_CFGR レジスタの変更は、LPTIM が無効なときにのみ行う必要があります。

43.4.14 タイマカウンタのリセット

LPTIM_CNT レジスタの内容をゼロにリセットするために、以下の 2 つのリセットメカニズムが実装されています。

- 同期リセットメカニズム：同期リセットは、LPTIM_CR レジスタの COUNTRST ビットによって制御されます。COUNTRST ビットフィールドを“1”にセットした後、リセット信号が LPTIM カーネルクロックドメイン内に伝播されます。したがって、リセットが考慮される前に、LPTIM カーネルロジックの若干のクロックパルスが経過することに注意することが重要です。このため、リセットがトリガされてから有効になるまでの間に、LPTIM カウンタは若干の余分なパルスをカウントすることになります。COUNTRST ビットは APB クロックドメインに位置し、LPTIM カウンタは LPTIM カーネルクロックドメインに位置しているため、COUNTRST ビットに“1”を書き込んだ時に APB クロックによって発行されるリセット信号に同期するのに、カーネルクロックの 3 クロックサイクルの遅延が必要となります。
- 非同期リセットメカニズム：非同期リセットは、LPTIM_CR レジスタにある RSTARE ビットによって制御されます。このビットが“1”にセットされると、LPTIM_CNT レジスタに対するあらゆる読出しアクセスによって、その内容がゼロにリセットされます。非同期リセットは、LPTIM コアクロックが提供されない時間枠内にトリガする必要があります。たとえば、LPTIM 入力 1 が外部クロックソースとして使用されているとき、非同期リセットは、LPTIM 入力 1 に反転が起きないことが十分確保されるときにのみ適用する必要があります。
LPTIM_CNT レジスタの内容を信頼できるように読み出すためには、2 回の連続した読出しアクセスを実行して比較する必要があることに注意してください。2 回の読出しアクセスで得られた値が同じであるとき、各読出しアクセスは信頼できると考えられます。しかし残念ながら、非同期リセットが有効になっているとき、LPTIM_CNT レジスタを 2 回読み出すことはできません。

警告： LPTIM 内部には、2 つのリセットメカニズムを同時に使用することを防ぐメカニズムはありません。したがって、開発者はこれらの 2 つのメカニズムを排他的に使用することを確実にする必要があります。

43.4.15 エンコーダモード

このモードでは、ロータリー素子の角度位置の検出に使用される直交エンコーダからの信号を処理できます。エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。これは、カウンタが 0 と LPTIM_ARR レジスタでプログラムされた自動再ロード値の間で（方向に応じて、0 から ARR まで、または ARR から 0 まで）連続的にカウントすることを意味します。したがって、カウンタを開始する前に LPTIM_ARR を設定する必要があります。Input1 と Input2 の 2 つの外部入力信号から、LPTIM カウンタのクロックのためのクロック信号が生成されます。この 2 つの信号の間の位相によって、カウント方向が決まります。

エンコーダモードは、LPTIM が内部クロックソースからクロック供給されるときにのみ使用できます。Input1 と Input2 の両方の信号周波数は、LPTIM 内部クロック周波数を 4 分周したものを超えてはなりません。これは、LPTIM の正しい動作を保証するために必要です。

方向の変更は、LPTIM_ISR レジスタの Down と Up の 2 つのフラグによって通知されます。また、DOWNIE ビットを通じて有効化された場合、両方の方向変更イベントで割り込みを生成できます。

エンコーダモードを有効にするには、ENC ビットを 1 にセットする必要があります。LPTIM を、まず、連続モードに設定する必要があります。

エンコーダモードがアクティブなとき、LPTIM カウンタはインクリメンタルエンコーダの速度と方向に従って自動的に変更されます。したがって、その内容は常にエンコーダの位置を表します。カウント方向は、Up および Down フラグによって通知され、エンコーダのロータの回転方向に対応します。

CKPOL[1:0] ビットを使用して設定されたエッジ検出に応じて、さまざまなカウントシナリオが可能です。次の表に、可能な組み合わせを示します (Input1 と Input2 は同時に切り替わらないと想定しています)。

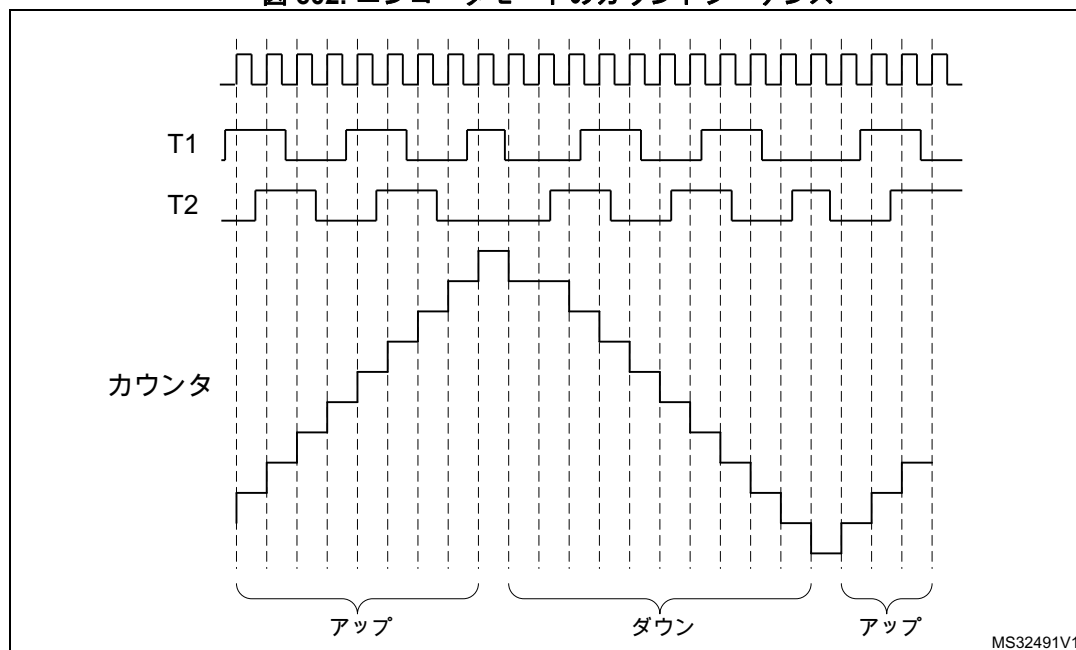
表 462. エンコーダのカウントシナリオ

アクティブエッジ	他方の信号のレベル (Input2 に対する Input1、Input1 に対する Input2)	Input1 信号		Input2 信号	
		立上がり	立下がり	立上がり	立下がり
立上がりエッジ	高	ダウン	カウントなし	アップ	カウントなし
	ロー	アップ	カウントなし	ダウン	カウントなし
立下がりエッジ	高	カウントなし	アップ	カウントなし	ダウン
	ロー	カウントなし	ダウン	カウントなし	アップ
両エッジ	高	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

次の図に、両方のエッジ検出が設定された場合のエンコーダモードのカウントシーケンスを示します。

注意： このモードでは、LPTIM のクロックは内部クロックソースによって供給される必要があるため、CKSEL ビットをリセット値 (0) に維持する必要があります。また、プリスケアラの分周比はリセット値である 1 に等しくなければなりません (PRESC[2:0] ビットが 000 である必要があります)。

図 602. エンコーダモードのカウントシーケンス



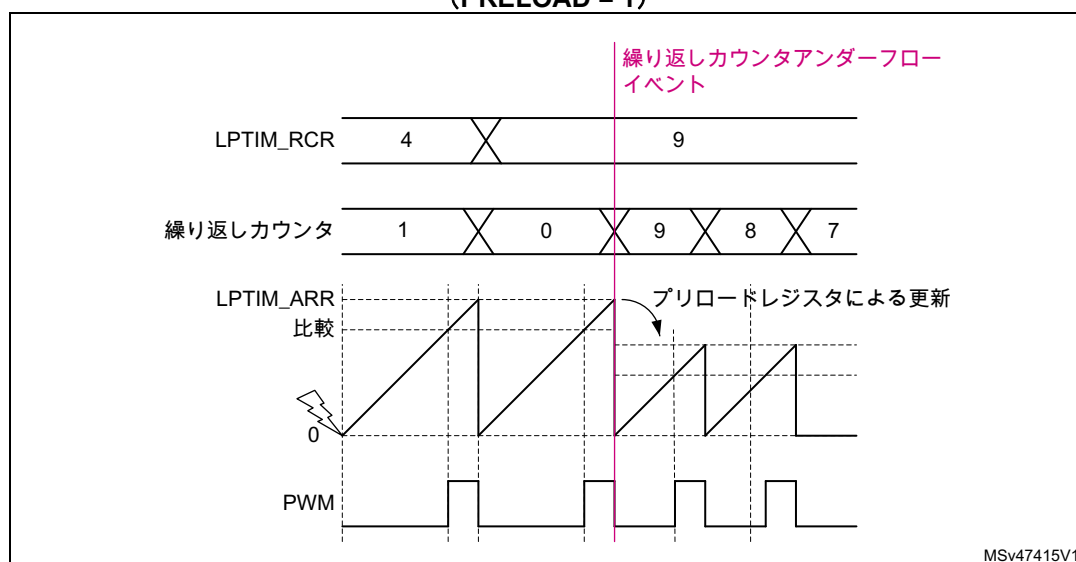
43.4.16 繰返しカウンタ

LPTIM には、LPTIM カウンタオーバーフローイベントが発生するたびに 1 ずつデクリメントする繰返しカウンタが搭載されています。繰返しカウンタアンダーフローイベントは、繰返しカウンタに 0 が含まれ、LPTIM カウンタがオーバーフローすると生成されます。各繰返しカウンタアンダーフローイベントの後、繰返しカウンタには、繰返しレジスタ LPTIM_RCR に属する REP[7:0] ビットフィールドの内容がロードされます。

繰返しアンダーフローイベントは、REP[7:0] レジスタが 0 にセットされる場合、各 LPTIM カウンタオーバーフロー時に生成されます。

PRELOAD = 1 の場合、REP[7:0] ビットフィールドに書き込んでも、次の繰返しアンダーフローイベントが発生するまで繰返しカウンタの内容に影響しません。繰返しカウンタが各 LPTIM カウンタオーバーフローイベントでデクリメントし続け、繰返しアンダーフローイベントが生成された場合のみ、REP[7:0] に書き込まれた新しい値が繰返しカウンタにロードされます。この動作を図 603 に示します。

図 603. 繰返しレジスタ LPTIM_RCR が 0 ではない場合の連続カウントモード (PRELOAD = 1)



繰返しカウンタアンダーフローイベントは、LPTIM プリロードレジスタの更新と体系的に関連付けられています（詳細はセクション「レジスタの更新」を参照）。

繰返しカウンタアンダーフローイベントは、LPTIM_ISR レジスタに配置された更新イベント (UE) フラグでソフトウェアに通知されます。LPTIM_DIER レジスタに配置された、それぞれの更新イベント割込みイネーブル (UEIE) 制御ビットがセットされていれば、UE フラグがセットされたとき、LPTIM 割込みがトリガされます。

繰返しレジスタ LPTIM_RCR は APB バスインタフェースクロックドメインにあり、繰返しカウンタ自体は LPTIM カーネルクロックドメインにあります。新しい値が LPTIM_RCR レジスタに書き込まれるたびに、新しい内容が APB バスインタフェースクロックドメインから LPTIM カーネルクロックドメインに伝播され、新たに書き込まれた値は繰返しカウンタアンダーフローイベント直後に繰返しカウンタにロードされます。新たに書き込まれた内容の同期遅延は 4 APB クロックサイクルと 3 LPTIM カーネルクロックサイクルとなり、経過すると LPTIM_ISR レジスタにある REPOK フラグによって通知されます。LPTIM カーネルが LSI クロックソースからクロック供給されている場合など、LPTIM カーネルクロックサイクルが比較的遅くなる場合、LPTIM_RCR レジスタの内容の同期が完了したことをソフトウェアで検出する REPOK フラグでしばらくポーリングすることができ

ます。そのため、LPTIM_DIER レジスタの関連する REPOKIE 制御ビットをセットしておけば、REPOK フラグがセットされた時に割り込みを生成することができます。

注： LPTIM_RCR レジスタへの書き込みの後、同じレジスタへの新しい書き込み操作は、前の書き込み操作が完了してからでなければ実行できません。REPOK フラグがセットされる前に続けて書き込みを行うと、予測不能な結果になります。

注意： PRELOAD = 0 で繰り返しカウンタを使用する場合、自動再ロード一致イベントの少なくとも 5 カウンタサイクル前に LPTIM_RCR レジスタを変更する必要があります。そうしないと、予測不能な動作が発生することがあります。

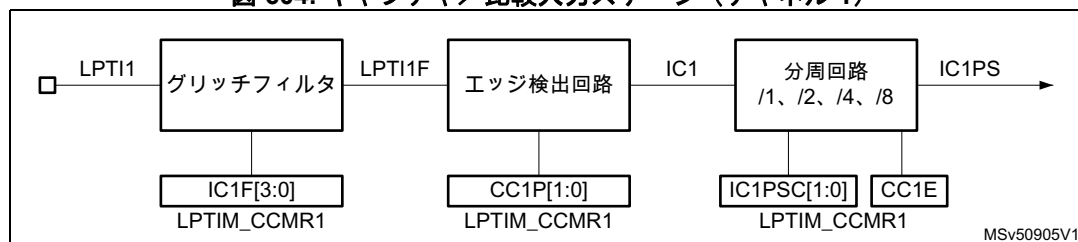
43.4.17 キャプチャ／比較チャネル

各キャプチャ / 比較チャネルは、PWM 用の、キャプチャ / 比較レジスタ、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケアラ）、および出カステージ（比較回路と出力制御）から構成されています。

入カステージ

入カステージは、対応する LPTIx 入力をサンプリングして、フィルタリングを行った信号 LPTiF を生成します。次に、極性選択付きのエッジ検出回路が、キャプチャコマンドとして使用される ICx 信号を生成します。この信号はプリスケアラを通じて、キャプチャコマンド信号 (ICxPS) を生成します。

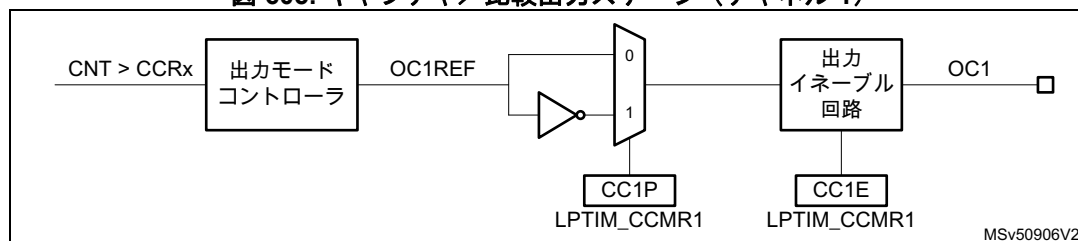
図 604. キャプチャ／比較入カステージ (チャネル 1)



出カステージ

出カステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します OCxREF（アクティブハイ）信号の極性は最終出力に影響を与えます。

図 605. キャプチャ／比較出カステージ (チャネル 1)



43.4.18 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって遷移が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (LPTIM_CCRx) が使用されます。チャンネル x で入力キャプチャが有効 (CCxE セット) になっていて、キャプチャが発生すると、対応する CCxIF フラグ (LPTIM_ISR レジスタ) がセットされ、割込みまたは DMA リクエストが送信されます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (LPTIM_ISR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで CCxICF に 1 を書き込むことによって、または、LPTIM_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、CCxOCF に 1 を書き込むことによってクリアされます。

注： DMA モードでは、IC DMA リクエストを有効にした後に、カウンタを開始した後、最後に入力キャプチャチャンネルを有効にする (CCxE ビットをセットする) 必要があります。これは、カウンタがまだ開始されていないときに、入力キャプチャの DMA リクエストが発生するのを防ぐためです。

入力キャプチャのグリッチフィルタの遅延

トリガーイベントがチャンネル x 入力 (LPTIx) に到着する時、設定されたグリッチフィルタ (CCMRx レジスタの ICxF[1:0] フィールド) とカーネルクロックプリスケール値 (CFGR レジスタの PRESC[2:0] フィールド) に応じて、可変の遅延が発生します。これによって、CCRx レジスタに格納されたキャプチャ値とキャプチャトリガに対応した実際の値との間に系統的なオフセット (表 463 を参照) が生じます。

このオフセットは系統的であって、2 つのキャプチャ間で補正されるため、パルス幅測定には影響しません。

入力キャプチャトリガに対応する実際のキャプチャ値は、以下の式を使用して計算できます。

実際のキャプチャ値 = キャプチャされた (LPTIM_CCRx) 値 - オフセット

グリッチフィルタとカーネルクロックのプリスケール値 (CFGR レジスタの PRESC フィールド) に応じて、関連するオフセットを使用する必要があります。

例：PRESC[2:0] = 0x2 で ICxF = 0x3 の場合の実際のキャプチャ値の決定

この設定 (PRESC[2:0] = 0x2 で ICxF = 0x3) の場合、表 463 によれば、オフセットは 5 です。

CCRx でキャプチャされた値が 9 (LPTIM_CNT = 9) であるとすれば、これは、LPTIM_CNT が 9 - 5 = 4 に等しかったときにキャプチャトリガが発生したことになります。

表 463. 入力キャプチャのグリッチフィルタの遅延 (カウンタステップ単位)

プリスケールの PRESC[2:0]	ICxF[1:0]	オフセット
0	0	2
	1	7
	2	9
	3	13
1	0	3
	1	5
	2	6
	3	8

表 463. 入力キャプチャのグリッチフィルタの遅延 (カウンタステップ単位) (続き)

プリスケアラの PRESC[2:0]	ICxF[1:0]	オフセット
2	0	2
	1	3
	2	4
	3	5
3	0	2
	1	2
	2	3
	3	3
4	0	2
	1	2
	2	2
	3	2
5	0	2
	1	2
	2	2
	3	2
6	0	2
	1	2
	2	2
	3	2
7	0	2
	1	2
	2	2
	3	2

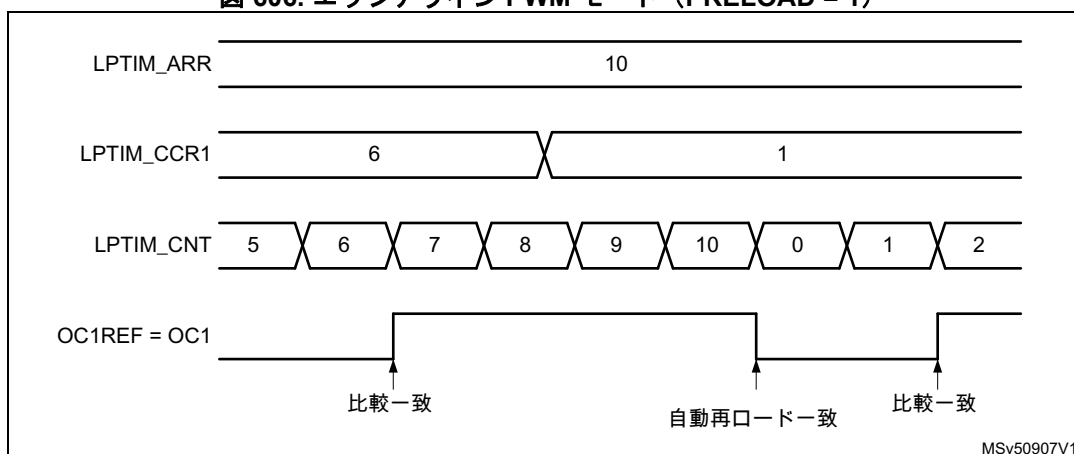
43.4.19 PWM モード

PWM モードでは、LPTIM_ARR レジスタの値によって決められた周波数と LPTIM_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。LPTIM はエッジアラインモードで PWM を生成できます。

OCx 極性は、LPTIM_CCMRx レジスタの CCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、LPTIM_CCMRx レジスタの CCxE ビットによって有効になります。詳細については、LPTIM_CCMRx レジスタの説明を参照してください。

図 606 に、LPTIM チャンネル 1 が、LPTIM_CCR1 = 6、その後 1、および LPTIM_ARR=10 で PWM モードに設定されている例を示します。

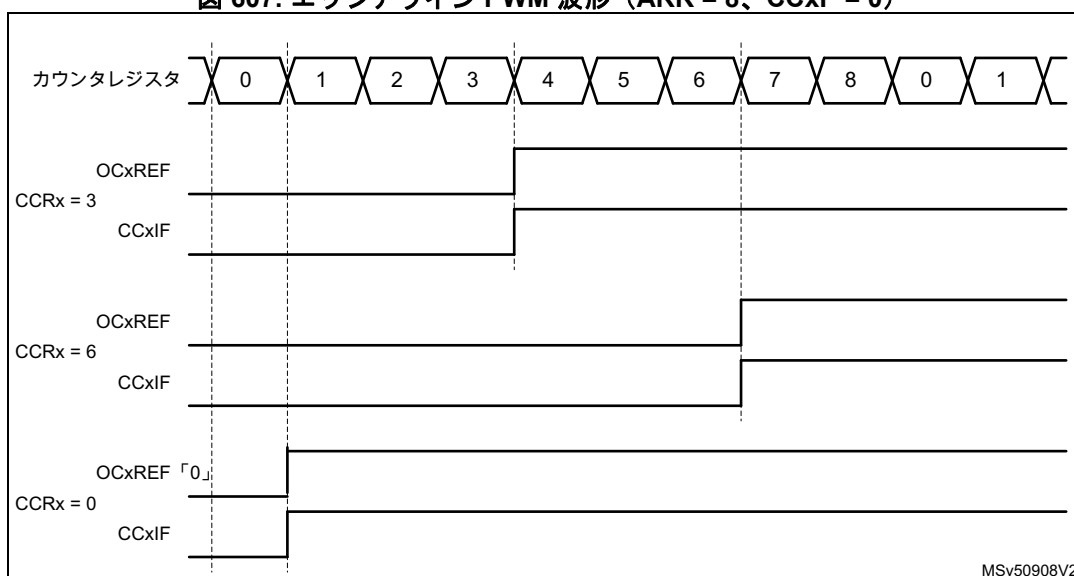
図 606. エッジアライン PWM モード (PRELOAD = 1)



次の例では、基準 PWM 信号 OCxREF は、 $LPTIM_CNT \leq LPTIM_CCR_x$ の間はローであり、そうでない場合はハイになります。

図 607 に $LPTIM_ARR = 8$ の場合の例のいくつかのエッジアライン PWM 波形を示します。

図 607. エッジアライン PWM 波形 (ARR = 8、CCxP = 0)



即時更新 (PRELOAD = 0) の PWM モード

PRELOAD = 0 の PWM モードでは、現在の PWM サイクル内で出力レベルを早期に変更できます。LPTIM_CCRx レジスタの即時更新 (PRELOAD = 0)、および LPTIM_CNT レジスタと LPTIM_CCRx レジスタの継続的な比較に基づいて、現在の PWM 周期の完了を待たずに、PWM サイクル内でできるだけ早く新しいデューティサイクル値を適用させることができます。

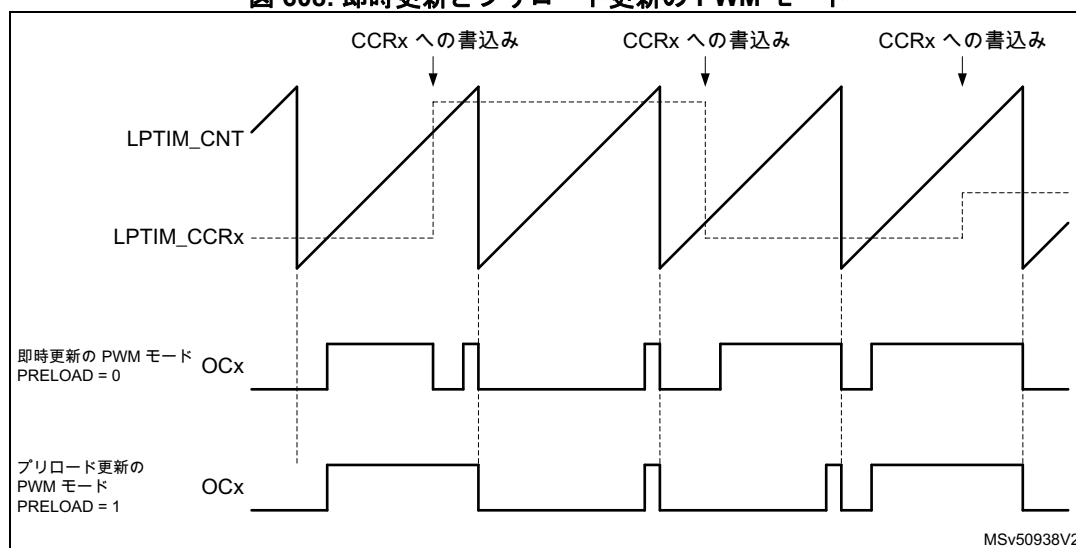
(PRELOAD = 0) の場合、LPTIM_CCRx レジスタの比較値を更新することにより、ソフトウェア（または DMA）によって OCxREF 信号レベルを動作中に変更できます。

書き込まれた比較値、および現在のカウンタ値と比較値に応じて、OCxREF レベルは次のように再割り当てされます。

- 新しい比較値が現在のカウンタ値を超えておらず、現在の比較値がカウンタ値を超えている場合、新しい比較値が書き込まれるとすぐに OCxREF レベルはハイに再割り当てされます。
- 新しい比較値がカウンタ値を超えており、現在の比較値がカウンタ値を超えていない場合、新しい比較値が書き込まれるとすぐに OCxREF レベルはローに再割り当てされます。

新しい比較値と現在の比較値が両方ともカウンタ値を超えていない場合、出力基準信号 OCxREF レベルは変更されません。図 608 に、PRELOAD = 0 および PRELOAD = 1 の場合の OCxREF 信号レベルの動作を示します。

図 608. 即時更新とプリロード更新の PWM モード



注： どちらの PWM モードでも、比較一致、自動再ロード一致、更新イベントの各フラグは、対応するイベントの 1 LPTIM カウンタサイクル後にセットされ、OCxREF レベルも、対応するイベントの 1 LPTIM カウンタサイクル後に変更されます。たとえば、LPTIM_CCRx が 3 にセットされている場合、CCxIF は LPTIM_CNT = 4 の時にセットされます。この動作を図 606 に示します。

43.4.20 DMA リクエスト

LPTIM には、次の 2 つのカテゴリの DMA リクエストを生成する機能があります。

- 入力キャプチャカウンタ値を取得するために使用される DMA リクエスト
- DMA 更新リクエストは、ソフトウェアのオーバーヘッドを発生させずに、LPTIMER の一部を定期的に複数回再プログラムするために使用されます。

入力キャプチャ DMA リクエスト

各 LPTIM チャンネルには専用の入力キャプチャ DMA リクエストがあります。CCRx レジスタでキャプチャの準備ができるたびに、DMA リクエストが生成され (LPTIM_DIER で CCxDE ビットがセットされている場合)、CCxIF がセットされます。その後、CCRx にキャプチャされた値は、DMA によって目的のメモリの転送先に定期的に転送できます。CCRx レジスタにキャプチャされた値が読み出されると、CCxIF はハードウェアによって自動的にクリアされます。

- 注： ICx DMA リクエスト信号 `lptim_icx_dma` は、次の状況でリセットされます。
- 該当する DMA リクエストが無効になった場合 (LPTIM_DIER レジスタの CCxDE ビットのクリア)
 - または、チャンネル x が無効になった場合 (CCxE ビットのクリア)
 - または、LPTIM が無効になった場合 (LPTIM_CR レジスタの ENABLE ビットのクリア)

更新イベント DMA リクエスト

更新イベントごとに DMA リクエストが生成され (LPTIM_DIER に UEDE が設定されている場合)、UE フラグがセットされます。DMA リクエストを使用すると、LPTIM_ARR、LPTIM_RCR、または LPTIM_CCRx レジスタを定期的に更新し、カスタム PWM 波形を生成できるようになります。

UE は、LPTIM_ARR レジスタへのバスマスタ (CPU や DMA など) の書き込みアクセス時にハードウェアによって自動的にクリアされます。

- 注： UE DMA リクエスト信号 `lptim_ue_dma` は、次の状況でリセットされます。
- 該当する DMA リクエストが無効になった場合 (LPTIM_DIER レジスタの UEDE ビットのクリア)
 - または、LPTIM が無効になった場合 (LPTIM_CR レジスタの ENABLE ビットのクリア)

43.4.21 デバッグモード

マイクロコントローラがデバッグモードになると (コアは停止状態)、LPTIM カウンタは、DBG モジュールの DBG_LPTIM_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

43.5 LPTIM 低消費電力モード

表 464. 低消費電力モードが LPTIM に与える影響

モード	説明
SLEEP	影響はありません。LPTIM 割込みによって、デバイスは SLEEP モードを終了します。
STOP	STOP モードで使用可能なオシレータから LPTIM のクロックが供給されている場合、LPTIM は動作を続け、割込みによりデバイスは STOP モードを終了します。
STANDBY	LPTIM ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

- 注： SLEEP、STOP、STANDBY モードに入る前に、すべての DMA リクエストを無効にする (UEDE ビットと CCxDE ビットをリセットする) 必要があります。

43.6 LPTIM 割込み

LPTIM_DIER レジスタで有効化されていた場合、次のイベントが発生すると、割込み／ウェイクアップイベントが生成されます。

- 比較一致
- 自動再ロード一致（エンコーダモードの場合は方向にかかわらず）
- 外部トリガイベント
- 自動再ロードレジスタへの書き込み完了
- 比較レジスタへの書き込み完了
- 方向変更（エンコーダモード）、プログラム可能（アップ／ダウン／両方）
- 更新イベント
- 繰り返しレジスタ更新 OK
- 入力キャプチャの発生
- オーバーキャプチャの発生
- 割込みイネーブルレジスタ更新 OK

注： LPTIM_ISR レジスタ（ステータスレジスタ）の対応するフラグがセットされた後に、LPTIM_DIER レジスタのビットをセットしても、割込みはアサートされません。

表 465. 割込みイベント

割込みベクタ	割込みイベント	イベントフラグ	イネーブル制御ビット	割込みのクリア方法	SLEEP モードの終了	STOP モードの終了 ⁽¹⁾
LPTIMx	比較一致	CCxIF	CCxIE	CCxCF に 1 を書き込む	可	はい
	入力キャプチャ ⁽²⁾	CCxIF	CCxIE	CCxCF に 1 を書き込む	可	はい
	オーバーキャプチャ ⁽²⁾	CCxOF	CCxOIE	CCxOCF に 1 を書き込む	可	はい
	自動再ロード一致	ARRM	ARRMIE	ARRMCF に 1 を書き込みます。	可	はい
	外部トリガイベント	EXTTRIG	EXTTRIGIE	EXTTRIGCF に 1 を書き込む	可	はい
	自動再ロードレジスタ更新 OK	ARROK	ARROKIE	ARROKCF に 1 を書き込む	可	はい
	キャプチャ／比較レジスタ更新 OK	CMPxOK	CMPxOKIE	CMPxOKCF に 1 を書き込む	可	はい
	アップ方向への変更 ⁽³⁾	UP	UPIE	UPCF に 1 を書き込みます。	可	はい
	ダウン方向への変更 ⁽³⁾	DOWN	DOWNIE	DOWNCF に 1 を書き込みます。	可	はい
	更新イベント	UE	UEIE	UECF に 1 を書き込みます。	可	はい
	繰り返しレジスタ更新 OK	REPOK	REPOKIE	REPOKCF に 1 を書き込む	可	はい

1. 各 LPTIM イベントは、LPTIM インスタンスが STOP モードからのウェイクアップ機能をサポートしている場合にのみデバイスを STOP モードからウェイクアップすることができます。セクション 43.3 : LPTIM の実装を参照してください。
2. LPTIM がチャネルを実装していない場合、このイベントは存在しません。セクション 43.3 : LPTIM の実装を参照してください。
3. LPTIM がエンコーダモード機能をサポートしていない場合、このイベントは存在しません。セクション 43.3 : LPTIM の実装を参照してください。

43.7 LPTIM レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.2 : 100 ページのレジスタに関する略記](#)を参照してください。

ペリフェラルレジスタには、ワード (32-bit) 単位でアクセスすることができます。

43.7.1 LPTIM4 割込みおよびステータスレジスタ (LPTIM4_ISR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIER OK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP OK	UE	DOWN	UP	ARR OK	CMP1 OK	EXT TRIG	ARRM	CC1IF
							r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROK** : 割込みイネーブルレジスタ更新 OK

DIEROK は、LPTIM_DIER レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。DIEROK フラグは、LPTIM_ICR レジスタの DIEROKCF ビットに 1 を書き込むことでクリアできます。

ビット 23:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **REPOK** : 繰り返しレジスタ更新 OK

REPOK は、LPTIM_RCR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。REPOK フラグは、LPTIM_ICR レジスタの REPOKCF ビットに 1 を書き込むことでクリアできます。

ビット 7 **UE** : LPTIM 更新イベントの発生

UE は、更新イベントが生成されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。UE フラグは、LPTIM_ICR レジスタの UECF ビットに 1 を書き込むことでクリアできます。

ビット 6 **DOWN** : カウンタの方向をアップからダウンへ変更

エンコーダモードでは、DOWN ビットは、カウンタの方向がアップからダウンに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。DOWN フラグは、LPTIM_ICR レジスタの DOWNCF ビットに 1 を書き込むことでクリアできます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。[セクション 43.3 : LPTIM の実装](#)を参照してください。

ビット 5 **UP** : カウンタの方向をダウンからアップへ変更

エンコーダモードでは、UP ビットは、カウンタの方向がダウンからアップに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。UP フラグは、LPTIM_ICR レジスタの UPCF ビットに 1 を書き込むことでクリアできます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。[セクション 43.3 : LPTIM の実装](#)を参照してください。

ビット 4 **ARROK** : 自動再ロードレジスタ更新 OK

ARROK は、LPTIM_ARR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARROK フラグは、LPTIM_ICR レジスタの ARROKCF ビットに 1 を書き込むことでクリアできます。

ビット 3 **CMP1OK** : 比較レジスタ 1 更新 OK

CMP1OK は、LPTIM_CCR1 レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。CMP1OK フラグは、LPTIM_ICR レジスタの CMP1OKCF ビットに 1 を書き込むことでクリアできます。

ビット 2 **EXTTRIG** : 外部トリガエッジイベント

EXTTRIG は、選択された外部トリガ入力で有効なエッジが発生したことをアプリケーションに知らせるために、ハードウェアによってセットされます。タイマがすでに開始していたためにトリガが無視された場合、このフラグはセットされません。EXTTRIG フラグは、LPTIM_ICR レジスタの EXTTRIGCF ビットに 1 を書き込むことでクリアできます。

ビット 1 **ARRM** : 自動再ロード一致

ARRM は、LPTIM_CNT レジスタの値が LPTIM_ARR レジスタの値に達したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARRM フラグは、LPTIM_ICR レジスタの ARRMCF ビットに 1 を書き込むことでクリアできます。

ビット 0 **CC1IF** : 比較 1 割込みフラグ

CC1IF フラグは、LPTIM_CNT レジスタの値が比較レジスタの値に一致したことをアプリケーションに知らせるために、ハードウェアによってセットされます。CC1IF フラグは、LPTIM_ICR レジスタの CC1CF ビットに 1 を書き込むことでクリアできます。

0 : 一致していません。

1 : カウンタ LPTIM_CNT レジスタの値の内容が LPTIM_CCR1 レジスタの値と一致しました。

43.7.2 LPTIMx 割込みおよびステータスレジスタ [オルタネート] (LPTIMx_ISR) (x = 1~3、5、6)

レジスタのこの説明は、出力比較モードに対してのみ使用できます。入力キャプチャモードについては次のセクションを参照してください。

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROK	Res.	Res.	Res.	Res.	CMP2OK	Res.	Res.	Res.
							r					r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC2IF	REPOK	UE	DOWN	UP	ARROK	CMP1OK	EXTTRIG	ARRM	CC1IF
						r	r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROK** : 割込みイネーブルレジスタ更新 OK

DIEROK は、LPTIM_DIER レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。DIEROK フラグは、LPTIM_ICR レジスタの DIEROKCF ビットに 1 を書き込むことでクリアできます。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CMP2OK** : 比較レジスタ 2 更新 OK

CMP2OK は、LPTIM_CCR2 レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。CMP2OK フラグは、LPTIM_ICR レジスタの CMP2OKCF ビットに 1 を書き込むことでクリアできます。

注： LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 18:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2IF** : 比較 2 割込みフラグ

CC2 チャンネルが出力として設定されている場合：

CC2IF フラグは、LPTIM_CNT レジスタの値が比較レジスタの値に一致したことをアプリケーションに知らせるために、ハードウェアによってセットされます。

CC2IF フラグは、LPTIM_ICR レジスタの CC2CF ビットに 1 を書き込むことでクリアできます。

0 : 一致していません。

1 : カウンタ LPTIM_CNT レジスタの値の内容が LPTIM_CCR2 レジスタの値と一致しました。

注： LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 8 **REPOK** : 繰り返しレジスタ更新 OK

REPOK は、LPTIM_RCR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。REPOK フラグは、LPTIM_ICR レジスタの REPOKCF ビットに 1 を書き込むことでクリアできます。

ビット 7 **UE** : LPTIM 更新イベントの発生

UE は、更新イベントが生成されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。対応する割込みまたは DMA リクエストが生成されます（有効な場合）。UE フラグは、LPTIM_ICR レジスタの UECF ビットに 1 を書き込むことでクリアできます。LPTIM_ARR レジスタが CPU や DMA などのバスマスタによって書き込まれると、UE フラグがハードウェアによって自動的にクリアされます。

ビット 6 **DOWN** : カウンタの方向をアップからダウンへ変更

エンコーダモードでは、DOWN ビットは、カウンタの方向がアップからダウンに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。DOWN フラグは、LPTIM_ICR レジスタの DOWNCF ビットに 1 を書き込むことでクリアできます。

注： LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 5 **UP** : カウンタの方向をダウンからアップへ変更

エンコーダモードでは、UP ビットは、カウンタの方向がダウンからアップに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。UP フラグは、LPTIM_ICR レジスタの UPCF ビットに 1 を書き込むことでクリアできます。

注： LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 4 **ARROK** : 自動再ロードレジスタ更新 OK

ARROK は、LPTIM_ARR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARROK フラグは、LPTIM_ICR レジスタの ARROKCF ビットに 1 を書き込むことでクリアできます。

ビット 3 **CMP1OK** : 比較レジスタ 1 更新 OK

CMP1OK は、LPTIM_CCR1 レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。CMP1OK フラグは、LPTIM_ICR レジスタの CMP1OKCF ビットに 1 を書き込むことでクリアできます。

ビット 2 **EXTTRIG** : 外部トリガエッジイベント

EXTTRIG は、選択された外部トリガ入力で有効なエッジが発生したことをアプリケーションに知らせるために、ハードウェアによってセットされます。タイマがすでに開始していたためにトリガが無視された場合、このフラグはセットされません。EXTTRIG フラグは、LPTIM_ICR レジスタの EXTTRIGCF ビットに 1 を書き込むことでクリアできます。

ビット 1 **ARRM** : 自動再ロード一致

ARRM は、LPTIM_CNT レジスタの値が LPTIM_ARR レジスタの値に達したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARRM フラグは、LPTIM_ICR レジスタの ARRMCF ビットに 1 を書き込むことでクリアできます。

ビット 0 **CC1IF** : 比較 1 割込みフラグ

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグは、LPTIM_CNT レジスタの値が比較レジスタの値に一致したことをアプリケーションに知らせるために、ハードウェアによってセットされます。CC1IF フラグは、LPTIM_ICR レジスタの CC1CF ビットに 1 を書き込むことでクリアできます。

0 : 一致していません。

1 : カウンタ LPTIM_CNT レジスタの値の内容が LPTIM_CCR1 レジスタの値と一致しました。

43.7.3 LPTIMx 割込みおよびステータスレジスタ [オルタネート] (LPTIMx_ISR) (x = 1~3、5、6)

レジスタのこの説明は、入力キャプチャモードに対してのみ使用できます。出力比較モードについては前のセクションを参照してください。

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIER OK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC2 OF	CC1 OF	Res.	Res.	CC2IF	REP OK	UE	DOWN	UP	ARR OK	Res.	EXT TRIG	ARRM	CC1IF
		r	r			r	r	r	r	r	r		r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROK** : 割込みイネーブルレジスタ更新 OK

DIEROK は、LPTIM_DIER レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。DIEROK フラグは、LPTIM_ICR レジスタの DIEROKCF ビットに 1 を書き込むことでクリアできます。

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC2OF** : キャプチャ 2 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。LPTIM_ICR レジスタの CC2OCF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC2IF フラグがすでにセットされているときに、カウンタの値が LPTIM_CCR2 レジスタにキャプチャされました。

LPTIM が少なくとも 2 つのチャンネルを実装していない場合、このビットは予約済みです。

セクション 43.3 を参照してください。

ビット 12 **CC1OF** : キャプチャ 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。LPTIM_ICR レジスタの CC1OCF ビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が LPTIM_CCR1 レジスタにキャプチャされました。

注 : LPTIM が少なくとも 1 つのチャンネルを実装していない場合、このビットは予約済みです。

セクション 43.3 を参照してください。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2IF** : キャプチャ 2 割込みフラグ

CC2 チャンネルが入力として設定されている場合 :

CC2IF は、カウンタの現在値が LPTIM_CCR2 レジスタにキャプチャされたことをアプリケーションに知らせるために、ハードウェアによってセットされます。対応する割込みまたは DMA リクエストが生成されます (有効な場合)。CC2IF フラグがすでにハイの場合、CC2OF フラグがセットされます。

0 : 入力キャプチャは発生していません。

1 : カウンタ値が LPTIM_CCR2 レジスタにキャプチャされました。(選択された極性に一致するエッジが IC2 で検出されました)。キャプチャされた値が (CPU または DMA によって) 読み出されると、CC2IF フラグはハードウェアによって自動的にクリアされます。CC2IF フラグは、LPTIM_ICR レジスタの CC2CF ビットに 1 を書き込むことでクリアできます。

注 : LPTIM が少なくとも 2 つのチャンネルを実装していない場合、このビットは予約済みです。

セクション 43.3 を参照してください。

ビット 8 **REPOK** : 繰り返しレジスタ更新 OK

REPOK は、LPTIM_RCR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。REPOK フラグは、LPTIM_ICR レジスタの REPOKCF ビットに 1 を書き込むことでクリアできます。

ビット 7 **UE** : LPTIM 更新イベントの発生

UE は、更新イベントが生成されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。UE フラグは、LPTIM_ICR レジスタの UECF ビットに 1 を書き込むことでクリアできます。

ビット 6 **DOWN** : カウンタの方向をアップからダウンへ変更

エンコーダモードでは、DOWN ビットは、カウンタの方向がアップからダウンに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。DOWN フラグは、LPTIM_ICR レジスタの DOWNCF ビットに 1 を書き込むことでクリアできます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。

セクション 43.3 を参照してください。

ビット 5 **UP** : カウンタの方向をダウンからアップへ変更

エンコーダモードでは、UP ビットは、カウンタの方向がダウンからアップに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。UP フラグは、LPTIM_ICR レジスタの UPCF ビットに 1 を書き込むことでクリアできます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 4 **ARROK** : 自動再ロードレジスタ更新 OK

ARROK は、LPTIM_ARR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARROK フラグは、LPTIM_ICR レジスタの ARROKCF ビットに 1 を書き込むことでクリアできます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **EXTTRIG** : 外部トリガエッジイベント

EXTTRIG は、選択された外部トリガ入力で有効なエッジが発生したことをアプリケーションに知らせるために、ハードウェアによってセットされます。タイマがすでに開始していたためにトリガが無視された場合、このフラグはセットされません。EXTTRIG フラグは、LPTIM_ICR レジスタの EXTTRIGCF ビットに 1 を書き込むことでクリアできます。

ビット 1 **ARRM** : 自動再ロード一致

ARRM は、LPTIM_CNT レジスタの値が LPTIM_ARR レジスタの値に達したことをアプリケーションに知らせるために、ハードウェアによってセットされます。ARRM フラグは、LPTIM_ICR レジスタの ARRMCF ビットに 1 を書き込むことでクリアできます。

ビット 0 **CC1IF** : キャプチャ 1 割込みフラグ

CC1 チャンネルが入力として設定されている場合 :

CC1IF は、カウンタの現在値が LPTIM_CCR1 レジスタにキャプチャされたことをアプリケーションに知らせるために、ハードウェアによってセットされます。対応する割込みまたは DMA リクエストが生成されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

0 : 入力キャプチャは発生していません。

1 : カウンタ値が LPTIM_CCR1 レジスタにキャプチャされました。(選択された極性に一致するエッジが IC1 で検出されました)。キャプチャされた値が (CPU または DMA によって) 読み出されると、CC1IF フラグはハードウェアによって自動的にクリアされます。CC1IF フラグは、LPTIM_ICR レジスタの CC1CF ビットに 1 を書き込むことでクリアできます。

43.7.4 LPTIM4 割込みクリアレジスタ (LPTIM4_ICR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOKCF	UECF	DOWNCF	UPCF	ARROKCF	CMP1OKCF	EXTTRIGCF	ARRMCF	CC1CF
							w	w	w	w	w	w	w	w	w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROKCF** : 割込みイネーブルレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DIEROK フラグがクリアされます。

ビット 23:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **REPOKCF** : 繰り返しレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの REPOK フラグがクリアされます。

ビット 7 **UECF** : 更新イベントクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UE フラグがクリアされます。

ビット 6 **DOWNCF** : ダウンへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DOWN フラグがクリアされます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 5 **UPCF** : アップへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UP フラグがクリアされます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 4 **ARROKCF** : 自動再ロードレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARROK フラグがクリアされます。

ビット 3 **CMP1OKCF** : 比較レジスタ 1 更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CMP1OK フラグがクリアされます。

ビット 2 **EXTTRIGCF** : 外部トリガ有効エッジクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの EXTTRIG フラグがクリアされます。

ビット 1 **ARRMCF** : 自動再ロード一致クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARRM フラグがクリアされます。

ビット 0 **CC1CF** : キャプチャ/比較 1 クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC1IF フラグがクリアされます。

43.7.5 LPTIMx 割込みクリアレジスタ [オルタネート] (LPTIMx_ICR) (x = 1~3、5、6)

レジスタのこの説明は、出力比較モードに対してのみ使用できます。入力キャプチャ比較モードについては次のセクションを参照してください。

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	CMP2OKCF	Res.	Res.	Res.
							w					w			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC2CF	REPOKCF	UECF	DOWNCF	UPCF	ARROKCF	CMP1OKCF	EXTTRIGCF	ARRMCF	CC1CF
						w	w	w	w	w	w	w	w	w	w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROKCF** : 割込みイネーブルレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DIEROK フラグがクリアされます。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CMP2OKCF** : 比較レジスタ 2 更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CMP2OK フラグがクリアされます。

注: LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 18:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2CF** : キャプチャ/比較 2 クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC2IF フラグがクリアされます。

注: LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 8 **REPOKCF** : 繰り返しレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの REPOK フラグがクリアされます。

ビット 7 **UECF** : 更新イベントクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UE フラグがクリアされます。

ビット 6 **DOWNCf** : ダウンへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DOWN フラグがクリアされます。

注: LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 5 **UPCF** : アップへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UP フラグがクリアされます。

注: LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 4 **ARROKCF** : 自動再ロードレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARROK フラグがクリアされます。

ビット 3 **CMP1OKCF** : 比較レジスタ 1 更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CMP1OK フラグがクリアされます。

ビット 2 **EXTTRIGCF** : 外部トリガ有効エッジクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの EXTTRIG フラグがクリアされます。

ビット 1 **ARRMCF** : 自動再ロード一致クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARRM フラグがクリアされます。

ビット 0 **CC1CF** : キャプチャ/比較 1 クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC1IF フラグがクリアされます。

43.7.6 LPTIMx 割込みクリアレジスタ [オルタネート] (LPTIMx_ICR) (x = 1~3、5、6)

レジスタのこの説明は、入力キャプチャモードに対してのみ使用できます。出力比較モードについては前のセクションを参照してください。

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC2OCF	CC1OCF	Res.	Res.	CC2OCF	REPOKCF	UECF	DOWNCF	UPCF	ARROKCF	Res.	EXTTRIGCF	ARRMCF	CC1CF
		w	w			w	w	w	w	w	w		w	w	w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DIEROKCF** : 割込みイネーブルレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DIEROK フラグがクリアされます。

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC2OCF** : キャプチャ/比較 2 オーバーキャプチャクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC2OF フラグがクリアされます。

注 : LPTIM が少なくとも 2 つのチャンネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 12 **CC1OCF** : キャプチャ/比較 1 オーバーキャプチャクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC1OF フラグがクリアされます。

注 : LPTIM が少なくとも 1 つのチャンネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2CF** : キャプチャ/比較 2 クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC2IF フラグがクリアされます。

注 : LPTIM が少なくとも 2 つのチャンネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 8 **REPOKCF** : 繰り返しレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの REPOK フラグがクリアされます。

ビット 7 **UECF** : 更新イベントクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UE フラグがクリアされます。

ビット 6 **DOWNCF** : ダウンへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの DOWN フラグがクリアされます。

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 5 **UPCF** : アップへの方向変更フラグクリア

このビットに 1 を書き込むと、LPTIM_ISR レジスタの UP フラグがクリアされます。

注： LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 4 **ARROKCF** : 自動再ロードレジスタ更新 OK クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARROK フラグがクリアされます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **EXTTRIGCF** : 外部トリガ有効エッジクリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの EXTTRIG フラグがクリアされます。

ビット 1 **ARRMCF** : 自動再ロード一致クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの ARRM フラグがクリアされます。

ビット 0 **CC1CF** : キャプチャ/比較 1 クリアフラグ

このビットに 1 を書き込むと、LPTIM_ISR レジスタの CC1IF フラグがクリアされます。

43.7.7 LPTIM4 割込み有効レジスタ (LPTIM4_DIER)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOK IE	UEIE	DOWNI E	UPIE	ARRO KIE	CMP1 OKIE	EXT TRIGIE	ARRM IE	CC1IE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **REPOKIE** : 繰り返しレジスタ更新 OK 割込みイネーブル

0 : 繰り返しレジスタ更新 OK 割込みは無効です。

1 : 繰り返しレジスタ更新 OK 割込みは有効です。

ビット 7 **UEIE** : 更新イベント割込みイネーブル

0 : 更新イベント割込みは無効です。

1 : 更新イベント割込みは有効です。

ビット 6 **DOWNIE** : ダウンへの方向変更割込み有効化

0 : DOWN 割込み無効

1 : DOWN 割込み有効

注： LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 5 **UPIE** : アップへの方向変更割込み有効化

0 : UP 割込み無効

1 : UP 割込み有効

注： LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 4 **ARROKIE** : 自動再ロードレジスタ更新 OK 割込み有効化

0 : ARROK 割込み無効

1 : ARROK 割込み有効

ビット 3 **CMP1OKIE** : 比較レジスタ 1 更新 OK 割込みイネーブル

- 0 : CMP1OK レジスタ 1 割込みは無効です。
- 1 : CMP1OK レジスタ 1 割込みは有効です。

ビット 2 **EXTTRIGIE** : 外部トリガ有効エッジ割込み有効化

- 0 : EXTTRIG 割込み無効
- 1 : EXTTRIG 割込み有効

ビット 1 **ARRMIE** : 自動再ロード一致割込み有効化

- 0 : ARRM 割込み無効
- 1 : ARRM 割込み有効

ビット 0 **CC1IE** : キャプチャ/比較 1 割込みイネーブル

- 0 : キャプチャ/比較 1 割込みは無効です。
- 1 : キャプチャ/比較 1 割込みは有効です。

注意 : LPTIMx_DIER レジスタの変更は、LPTIM が有効な (ENABLE ビットが 1 にセットされている) と
きにのみ行う必要があります。LPTIMx_DIER レジスタへの書き込みの後、同じレジスタへの新しい書
込み操作は、前の書き込み操作が完了してからでなければ実行できません。DIEROK フラグがセットさ
れる前に続けて書き込みを行うと、予測不能な結果になります。

43.7.8 LPTIMx 割込み有効レジスタ [オルタネート] (LPTIMx_DIER) (x = 1~3、5、6)

レジスタのこの説明は、出力比較モードに対してのみ使用できます。入力キャプチャ比較モードにつ
いては次のセクションを参照してください。

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEDE	Res.	Res.	Res.	CMP2 OKIE	Res.	Res.	Res.
								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC2IE	REPOK IE	UEIE	DOWNI E	UPIE	ARRO KIE	CMP1 OKIE	EXT TRIGIE	ARRM IE	CC1IE
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **UEDE** : 更新イベント DMA リクエストイネーブル

- 0 : UE DMA リクエストは無効です。この UEDE ビットに "0" を書き込むと、関連する ue_dma_req 信号がリ
セットされます。
- 1 : UE DMA リクエストは有効です。

注 : LPTIM が少なくとも 1 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 22 予約済みであり、リセット値に保持する必要があります。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CMP2OKIE** : 比較レジスタ 2 更新 OK 割込みイネーブル

0 : CMPOK レジスタ 2 割込みは無効です。

1 : CMPOK レジスタ 2 割込みは有効です。

注 : LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 18:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2IE** : キャプチャ/比較 2 割込みイネーブル

0 : キャプチャ/比較 2 割込みは無効です。

1 : キャプチャ/比較 2 割込みは有効です。

注 : LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 8 **REPOKIE** : 繰り返しレジスタ更新 OK 割込みイネーブル

0 : 繰り返しレジスタ更新 OK 割込みは無効です。

1 : 繰り返しレジスタ更新 OK 割込みは有効です。

ビット 7 **UEIE** : 更新イベント割込みイネーブル

0 : 更新イベント割込みは無効です。

1 : 更新イベント割込みは有効です。

ビット 6 **DOWNIE** : ダウンへの方向変更割込み有効化

0 : DOWN 割込み無効

1 : DOWN 割込み有効

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 5 **UPIE** : アップへの方向変更割込み有効化

0 : UP 割込み無効

1 : UP 割込み有効

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 4 **ARROKIE** : 自動再ロードレジスタ更新 OK 割込み有効化

0 : ARROK 割込み無効

1 : ARROK 割込み有効

ビット 3 **CMP1OKIE** : 比較レジスタ 1 更新 OK 割込みイネーブル

0 : CMPOK レジスタ 1 割込みは無効です。

1 : CMPOK レジスタ 1 割込みは有効です。

ビット 2 **EXTTRIGIE** : 外部トリガ有効エッジ割込み有効化

0 : EXTTRIG 割込み無効

1 : EXTTRIG 割込み有効

ビット 1 **ARRMIE** : 自動再ロード一致割込み有効化

0 : ARRM 割込み無効

1 : ARRM 割込み有効

ビット 0 **CC1IE** : キャプチャ/比較 1 割込みイネーブル

0 : キャプチャ/比較 1 割込みは無効です。

1 : キャプチャ/比較 1 割込みは有効です。

注意 : LPTIMx_DIER レジスタの変更は、LPTIM が有効な (ENABLE ビットが 1 にセットされている) と
きにのみ行う必要があります。LPTIMx_DIER レジスタへの書込みの後、同じレジスタへの新しい書
込み操作は、前の書込み操作が完了してからでなければ実行できません。DIEROK フラグがセットさ
れる前に続けて書込みを行うと、予測不能な結果になります。

43.7.9 LPTIMx 割込み有効レジスタ [オルタネート] (LPTIMx_DIER) (x = 1~3、5、6)

レジスタのこの説明は、入力キャプチャモードに対してのみ使用できます。出力比較モードについては前のセクションを参照してください。

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	CC2DE	Res.	UEDE	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE
						rw		rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC2OIE	CC1OIE	Res.	Res.	CC2IE	REPOKIE	UEIE	DOWNIE	UPIE	ARROKIE	Res.	EXTTRIGIE	ARRMIE	CC1IE
		rw	rw			rw	rw	rw	rw	rw	rw		rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **CC2DE** : キャプチャ/比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。この CC2DE ビットに "0" を書き込むと、関連する ic2_dma_req 信号がリセットされます。

1 : CC2 DMA リクエストは有効です。

注 : LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **UEDE** : 更新イベント DMA リクエストイネーブル

0 : UE DMA リクエストは無効です。この UEDE ビットに "0" を書き込むと、関連する ue_dma_req 信号がリセットされます。

1 : UE DMA リクエストは有効です。

注 : LPTIM が少なくとも 1 つのチャネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 22:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **CC1DE** : キャプチャ/比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。この CC1DE ビットに "0" を書き込むと、関連する ic1_dma_req 信号がリセットされます。

1 : CC1 DMA リクエストは有効です。

注 : LPTIM が少なくとも 1 つのチャネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC2OIE** : キャプチャ/比較 2 オーバーキャプチャ割込みイネーブル

0 : CC2 オーバーキャプチャ割込みは無効です。

1 : CC2 オーバーキャプチャ割込みは有効です。

注 : LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 12 **CC1OIE** : キャプチャ／比較 1 オーバーキャプチャ割込みイネーブル

0 : CC1 オーバーキャプチャ割込みは無効です。

1 : CC1 オーバーキャプチャ割込みは有効です。

注 : LPTIM が少なくとも 1 つのチャンネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC2IE** : キャプチャ／比較 2 割込みイネーブル

0 : キャプチャ／比較 2 割込みは無効です。

1 : キャプチャ／比較 2 割込みは有効です。

注 : LPTIM が少なくとも 2 つのチャンネルを実装していない場合、このビットは予約済みです。
[セクション 43.3](#) を参照してください。

ビット 8 **REPOKIE** : 繰り返しレジスタ更新 OK 割込みイネーブル

0 : 繰り返しレジスタ更新 OK 割込みは無効です。

1 : 繰り返しレジスタ更新 OK 割込みは有効です。

ビット 7 **UEIE** : 更新イベント割込みイネーブル

0 : 更新イベント割込みは無効です。

1 : 更新イベント割込みは有効です。

ビット 6 **DOWNIE** : ダウンへの方向変更割込み有効化

0 : DOWN 割込み無効

1 : DOWN 割込み有効

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 5 **UPIE** : アップへの方向変更割込み有効化

0 : UP 割込み無効

1 : UP 割込み有効

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
[セクション 43.3](#) を参照してください。

ビット 4 **ARROKIE** : 自動再ロードレジスタ更新 OK 割込み有効化

0 : ARROK 割込み無効

1 : ARROK 割込み有効

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **EXTTRIGIE** : 外部トリガ有効エッジ割込み有効化

0 : EXTTRIG 割込み無効

1 : EXTTRIG 割込み有効

ビット 1 **ARRMIE** : 自動再ロード一致割込み有効化

0 : ARRM 割込み無効

1 : ARRM 割込み有効

ビット 0 **CC1IE** : キャプチャ／比較 1 割込みイネーブル

0 : キャプチャ／比較 1 割込みは無効です。

1 : キャプチャ／比較 1 割込みは有効です。

注意 : LPTIMx_DIER レジスタの変更は、LPTIM が有効な (ENABLE ビットが 1 にセットされている) ときにのみ行う必要があります。LPTIMx_DIER レジスタへの書き込みの後、同じレジスタへの新しい書き込み操作は、前の書き込み操作が完了してからでなければ実行できません。DIEROK フラグがセットされる前に続けて書き込みを行うと、予測不能な結果になります。

43.7.10 LPTIM 設定レジスタ (LPTIM_CFGR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRE LOAD	WAV POL	WAVE	TIMOUT	TRIGEN[1:0]		Res.
							rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ENC** : エンコーダモード有効化

ENC ビットは、エンコーダモードを制御します。

0 : エンコーダモード無効

1 : エンコーダモード有効

注 : LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みとなります。
セクション 43.3 を参照してください。

ビット 23 **COUNTMODE** : カウンタモード有効化

COUNTMODE ビットは、LPTIM がカウンタのクロックに使用するクロックソースを選択します。

0 : カウンタは各内部クロックのパルスに従ってインクリメントされます。

1 : カウンタは LPTIM 外部 Input1 の各有効なクロックパルスに従ってインクリメントされます。

ビット 22 **PRELOAD** : レジスタ更新モード

PRELOAD ビットは、LPTIM_ARR、LPTIM_RCR、および LPTIM_CCRx レジスタの更新方法を制御します。

0 : レジスタは、各 APB バス書き込みアクセス後に更新されます。

1 : レジスタは、現在の LPTIM 周期の終了時に更新されます。

ビット 21 **WAVPOL** : 波形極性

WAVPOL ビットは、出力の極性を制御します。

0 : LPTIM 出力は LPTIM_CNT レジスタと LPTIM_CCRx レジスタの比較結果を反映します。

1 : LPTIM 出力は LPTIM_CNT レジスタと LPTIM_CCRx レジスタの比較結果の反転を反映します。

注 : LPTIM が少なくとも 1 つのキャプチャ/比較チャネルを実装している場合、このビットは予約済みです。
セクション 43.3 を参照してください。

ビット 20 **WAVE** : 波形

WAVE ビットは、出力波形を制御します。

0 : セットワンスモードを非アクティブにします。

1 : セットワンスモードをアクティブにします。

ビット 19 **TIMOUT** : タイムアウト有効化

TIMOUT ビットは、タイムアウト機能を制御します。

0 : タイマがすでに開始しているときに着信したトリガイイベントは無視されます。

1 : タイマがすでに開始しているときにトリガイイベントが着信すると、LPTIM カウンタと繰り返しカウンタがリセットされて、再開始されます。

ビット 18:17 **TRIGEN[1:0]** : トリガ有効化および極性

TRIGEN ビットは、LPTIM カウンタが外部トリガによって開始されるかどうかを制御します。外部トリガオプションが選択された場合、トリガのアクティブエッジについて 3 つの構成が可能です。

00 : ソフトウェアトリガ (カウンタの開始はソフトウェアによって行われます)。

01 : 立上がりエッジがアクティブエッジです。

10 : 立下がりエッジがアクティブエッジです。

11 : 両方のエッジがアクティブエッジです。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **TRIGSEL[2:0]** : トリガセレクタ

TRIGSEL ビットは、次の 8 つの利用可能なソースの中から LPTIM のトリガイベントの役目を果たすトリガソースを選択します。

000 : lptim_ext_trig0

001 : lptim_ext_trig1

010 : lptim_ext_trig2

011 : lptim_ext_trig3

100 : lptim_ext_trig4

101 : lptim_ext_trig5

110 : lptim_ext_trig6

111 : lptim_ext_trig7

詳細については、[セクション 43.4.3 : LPTIM 入力およびトリガマッピング](#)を参照してください。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11:9 **PRESC[2:0]** : クロックプリスケアラ

PRESC ビットは、プリスケアラ分周比を設定します。次の分周比から選択できます。

000 : /1

001 : /2

010 : /4

011 : /8

100 : /16

101 : /32

110 : /64

111 : /128

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **TRGFLT[1:0]** : 設定可能なトリガ用デジタルフィルタ

TRGFLT 値は、内部トリガ上でレベルの変化が発生した時に、有効なレベル遷移とみなすまで検出する同じサンプルの連続数を設定します。この機能を使用するには、内部クロックソースが必要です。

00 : トリガのアクティブレベル変更は、有効なトリガとみなされます。

01 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。

10 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。

11 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:3 **CKFLT[1:0]** : 設定可能な外部クロック用デジタルフィルタ

CKFLT 値は、外部クロック信号上でレベル変化が発生した時に、有効なレベル遷移とみなすまで検出する同じサンプルの連続数を設定します。この機能を使用するには、内部クロックソースが必要です。

00 : 外部クロック信号のレベル変更は、有効な遷移とみなされます。

01 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。

10 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。

11 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 2:1 **CKPOL[1:0]** : クロック極性

LPTIM のクロックが外部クロックソースによって供給されるとき、CKPOL ビットは、カウンタによって使用されるアクティブエッジを設定するために使用されます。

00 : 立上がりエッジが、カウンタに使用されるアクティブエッジです。

LPTIM がエンコーダモードで設定されている (ENC ビットがセットされている) 場合、エンコーダサブモード 1 がアクティブです。

01 : 立下がりエッジが、カウンタに使用されるアクティブエッジです。

LPTIM がエンコーダモードで設定されている (ENC ビットがセットされている) 場合、エンコーダサブモード 2 がアクティブです。

10 : 両方のエッジがアクティブエッジです。外部クロック信号の両方のエッジがアクティブエッジとみなされるときには、LPTIM のクロックは内部クロックソースからも供給される必要があります、その周波数は外部クロック周波数の 4 倍以上である必要があります。

LPTIM がエンコーダモードで設定されている (ENC ビットがセットされている) 場合、エンコーダサブモード 3 がアクティブです。

11 : 使用できません。

エンコーダモードのサブモードの詳細については、[セクション 43.4.15 : エンコーダモード](#)を参照してください。

ビット 0 **CKSEL** : クロックセレクタ

CKSEL ビットは、LPTIM が使用するクロックソースを選択します。

0 : LPTIM のクロックは内部クロックソースによって供給されます (APB クロックまたは埋め込みオシレータ)。

1 : LPTIM のクロックは、LPTIM 外部 Input1 を通じて外部クロックソースによって供給されます。

注意 : LPTIM_CFGR レジスタの変更は、LPTIM が無効 (イネーブルビットが 0 にリセットされている) のときにのみ行う必要があります。

43.7.11 LPTIM 制御レジスタ (LPTIM_CR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST ARE	COUN TRST	CNT STRT	SNG STRT	ENA BLE
											rw	rs	rw	rw	rw

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **RSTARE** : 読出し後リセット有効化

このビットは、ソフトウェアによってセット/クリアされます。RSTARE が“1”にセットされると、LPTIM_CNT レジスタに対するあらゆる読出しアクセスによって、LPTIM_CNT レジスタの内容が非同期的にリセットされます。このビットをセットできるのは、LPTIM が有効なときだけです。

ビット 3 **COUNTRST** : カウンタリセット

このビットは、ソフトウェアによってセットされ、ハードウェアによってクリアされます。“1”にセットされると、このビットは LPTIM_CNT カウンタレジスタの同期リセットをトリガします。このリセットの同期特性のため、3 LP タイマコアクロックサイクルの同期遅延の後にはのみ、この状況が発生します (LP タイマコアクロックは APB クロックと異なる場合があります)。

このビットをセットできるのは、LPTIM が有効なときだけです。これはハードウェアによって自動的にリセットされます。

注意 : **COUNTRST** は、ハードウェアによって“0”にクリアされる前に、ソフトウェアによって“1”にセットしてはなりません。したがって、ソフトウェアで **COUNTRST** ビットを“1”にセットしようとする前に、それがすでに“0”にクリアされていることを確認する必要があります。

ビット 2 **CNTSTRT** : 連続モードでタイマ開始

このビットは、ソフトウェアによってセットされ、ハードウェアによってクリアされます。

ソフトウェア開始 (TRIGEN[1:0] = 00) の場合、このビットをセットすると、LPTIM は連続モードで開始します。ソフトウェア開始が無効 (TRIGEN[1:0] が 00 以外) の場合、このビットをセットすると、外部トリガが検出されるとすぐに、タイマは連続モードで開始します。

シングルパルスモードでのカウント中にこのビットがセットされた場合、LPTIM_ARR レジスタと LPTIM_CNT レジスタが次に一致したときにタイマは停止せず、LPTIM カウンタは連続モードでのカウントを続行します。このビットをセットできるのは、LPTIM が有効なときだけです。これはハードウェアによって自動的にリセットされます。

ビット 1 **SNGSTRT** : シングルモードで LPTIM 開始

このビットは、ソフトウェアによってセットされ、ハードウェアによってクリアされます。

ソフトウェア開始 (TRIGEN[1:0] = 00) の場合、このビットをセットすると、LPTIM はシングルパルスモードで開始します。

ソフトウェア開始が無効 (TRIGEN[1:0] が 00 以外) の場合、このビットをセットすると、外部トリガが検出されるとすぐに、LPTIM はシングルパルスモードで開始します。

LPTIM が連続カウントモードのときにこのビットがセットされた場合、LPTIM は LPTIM_ARR レジスタと LPTIM_CNT レジスタが次に一致したときに停止します。

このビットをセットできるのは、LPTIM が有効なときだけです。これはハードウェアによって自動的にリセットされます。

ビット 0 **ENABLE** : LPTIM 有効化

イネーブルビットは、ソフトウェアによってセット／クリアされます。

0 : LPTIM は無効です。ENABLE ビットに "0" を書き込むと、すべての DMA リクエスト信号（入力キャプチャおよび更新イベント DMA リクエスト）がリセットされます。

1 : LPTIM は有効です。

43.7.12 LPTIM 比較レジスタ 1 (LPTIM_CCR1)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CCR1[15:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、キャプチャ／比較 1 レジスタにロードされる値です。

PRELOAD オプションに応じて、CCR1 レジスタは PRELOAD ビットがリセットされた場合は即座に更新され、PREOAD ビットがリセットされた場合は次の LPTIM 更新イベント時に更新されます。

キャプチャ／比較レジスタ 1 は、カウンタ LPTIM_CNT と比較されて、OC1 出力に送信される値を含みます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は読出し専用となり、最後の入力キャプチャ 1 イベントによって転送されたカウンタ値を含みます。LPTIM_CCR1 レジスタは読出し専用レジスタで、プログラムできません。

LPTIM がチャンネルを実装していない場合 :

比較レジスタ 1 は、カウンタの LPTIM_CNT と比較されて、LPTIM 出力に送信される値を含みます。

注意 : LPTIM_CCR1 レジスタの変更は、LPTIM が有効な (ENABLE ビットが 1 にセットされている) ときにのみ行う必要があります。

43.7.13 LPTIM 自動再ロードレジスタ（LPTIM_ARR）

アドレスオフセット：0x018

リセット値：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ARR[15:0]**：自動再ロード値

ARR は、LPTIM の自動再ロード値です。

この値は、CCRx[15:0] 値より大きくなければなりません。

注意： LPTIM_ARR レジスタの変更は、LPTIM が有効（イネーブルビットが 1 にセットされている）のときにのみ行う必要があります。

43.7.14 LPTIM カウンタレジスタ（LPTIM_CNT）

アドレスオフセット：0x01C

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]**：カウンタ値

LPTIM が非同期クロックで実行しているとき、LPTIM_CNT レジスタを読み出すと、信頼できない値が返されることがあります。したがって、この場合、2 つの連続した読出しアクセスを実行して、返された 2 つの値が同じかどうかを確認する必要があります。

43.7.15 LPTIM 設定レジスタ 2 (LPTIM_CFGR2)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2SEL [1:0]		Res.	Res.	IC1SEL [1:0]	
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL [1:0]		Res.	Res.	IN1SEL [1:0]	
										rw	rw			rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **IC2SEL[1:0]** : LPTIM 入力 キャプチャ 2 選択

IC2SEL ビットは、LPTIM 入力キャプチャ 2 を利用可能な入力のうちの 1 つに接続する LPTIM 入力キャプチャ 2 マルチプレクサを制御します。

00 : lptim_ic2_mux0

01 : lptim_ic2_mux1

10 : lptim_ic2_mux2

11 : lptim_ic2_mux3

接続の詳細については、[セクション 43.4.3 : LPTIM 入力およびトリガマッピング](#)を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **IC1SEL[1:0]** : LPTIM 入力 キャプチャ 1 選択

IC1SEL ビットは、LPTIM 入力キャプチャ 1 を利用可能な入力のうちの 1 つに接続する LPTIM 入力キャプチャ 1 マルチプレクサを制御します。

00 : lptim_ic1_mux0

01 : lptim_ic1_mux1

10 : lptim_ic1_mux2

11 : lptim_ic1_mux3

接続の詳細については、[セクション 43.4.3 : LPTIM 入力およびトリガマッピング](#)を参照してください。

ビット 15:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **IN2SEL[1:0]** : LPTIM 入力 2 選択

IN2SEL ビットは、LPTIM 入力 2 を利用可能な入力のうちの 1 つに接続する LPTIM 入力 2 マルチプレクサを制御します。

00 : lptim_in2_mux0

01 : lptim_in2_mux1

10 : lptim_in2_mux2

11 : lptim_in2_mux3

接続の詳細については、[セクション 43.4.3 : LPTIM 入力およびトリガマッピング](#)を参照してください。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **IN1SEL[1:0]** : LPTIM 入力 1 選択

IN1SEL ビットは、LPTIM 入力 1 を利用可能な入力のうちの 1 つに接続する LPTIM 入力 1 マルチプレクサを制御します。

00 : lptim_in1_mux0

01 : lptim_in1_mux1

10 : lptim_in1_mux2

11 : lptim_in1_mux3

接続の詳細については、[セクション 43.4.3 : LPTIM 入力およびトリガマッピング](#)を参照してください。

43.7.16 LPTIM 繰返しレジスタ (LPTIM_RCR)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰返しレジスタ値

REP は LPTIM の繰返し値です。

注意 : LPTIM_RCR レジスタの変更は、LPTIM が有効（イネーブルビットが 1 にセットされている）のときにのみ行う必要があります。PRELOAD = 0 で繰返しカウンタを使用する場合、自動再ロード一致イベントの少なくとも 5 カウンタサイクル前に LPTIM_RCR レジスタを変更する必要があります。そうしないと、予測不能な動作が発生することがあります。

43.7.17 LPTIM キャプチャ／比較モードレジスタ 1 (LPTIM_CCMR1)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

チャンネルは、入力（キャプチャモード）または出力（PWM モード）で使用できます。チャンネルの方向は対応する CCxSEL ビットを設定することで定義されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	IC2F [1:0]		Res.	Res.	IC2PSC [1:0]		Res.	Res.	Res.	Res.	CC2P [1:0]		CC2E	CC2 SEL
		rW	rW			rW	rW					rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	IC1F [1:0]		Res.	Res.	IC1PSC [1:0]		Res.	Res.	Res.	Res.	CC1P [1:0]		CC1E	CC1 SEL
		rW	rW			rW	rW					rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **IC2F[1:0]** : 入力キャプチャ 2 フィルタ

このビットフィールドにより、外部入力キャプチャ信号上でレベル変化が発生した時に、有効なレベル遷移とみなすまで検出する同じサンプルの連続数を定義します。この機能を使用するには、内部クロックソースが必要です。

00 : 外部入力キャプチャ信号のいかなるレベル変化も、有効な遷移とみなされます。

01 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。

10 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。

11 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 27:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

このビットフィールドは、CC2 入力 (IC2) に作用するプリスケアラの比を定義します。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 **CC2P[1:0]** : キャプチャ / 比較 2 出力極性。

条件 : 出力としての CC2

出力モードが有効になっている場合、ビット 2 のみを使用して極性を設定します。ビット 3 は無視されます。

0 : OC2 はアクティブハイです。

1 : OC2 はアクティブローです。

条件 : 入力としての CC2

このフィールドは、キャプチャ動作での IC2 の極性を選択するために使用されます。

00 : 立上がりエッジ。回路は IC2 の立上がりエッジに反応します。

01 : 立下がりエッジ。回路は IC2 の立下がりエッジに反応します。

10 : 予約済み。この設定は使用しないでください。

11 : 両方のエッジ。回路は IC2 の立上がりエッジと立下がりエッジの両方に反応します。

ビット 17 **CC2E** : キャプチャ / 比較 2 出力イネーブル。

条件 : 出力としての CC2

0 : オフ - OC2 はアクティブではありません。CC2E ビットに "0" を書き込むと、他のすべての LPTIM チャンネルが無効になっている場合にのみ、ue_dma_req 信号がリセットされます。

1 : オン - OC2 信号は、対応する出力ピンに出力されます。

条件 : 入力としての CC2

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 2 (LPTIM_CCR2) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。この CC2E ビットに "0" を書き込むと、関連する ic2_dma_req 信号がリセットされます。

1 : キャプチャは有効です。

ビット 16 **CC2SEL** : キャプチャ / 比較 2 選択

このビットフィールドは、チャンネルの方向、すなわち、入力 (キャプチャ) モードか、出力モードかを定義します。

0 : CC2 チャンネルは出力 PWM モードに設定されます。

1 : CC2 チャンネルは入力キャプチャモードに設定されます。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 **IC1F[1:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドにより、外部入力キャプチャ信号上でレベル変化が発生した時に、有効なレベル遷移とみなすまで検出する同じサンプルの連続数を定義します。この機能を使用するには、内部クロックソースが必要です。

00 : 外部入力キャプチャ信号のいかなるレベル変化も、有効な遷移とみなされます。

01 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。

10 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。

11 : 外部入力キャプチャ信号のレベル変化が有効な遷移とみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 11:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **IC1PSC[1:0]** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの比を定義します。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:2 **CC1P[1:0]** : キャプチャ / 比較 1 出力極性。

条件 : 出力としての CC1

出力モードが有効になっている場合、ビット 2 のみを使用して極性を設定します。ビット 3 は無視されます。

0 : OC1 はアクティブハイです。LPTIM 出力は LPTIM_ARR レジスタと LPTIM_CCRx レジスタの比較結果を反映します。

1 : OC1 はアクティブローです。LPTIM 出力は LPTIM_ARR レジスタと LPTIM_CCRx レジスタの比較結果の反転を反映します。

条件 : 入力としての CC1

このフィールドは、キャプチャ動作での IC1 の極性を選択するために使用されます。

00 : 立上がりエッジ。回路は IC1 の立上がりエッジに反応します。

01 : 立下がりエッジ。回路は IC1 の立下がりエッジに反応します。

10 : 予約済み。この設定は使用しないでください。

11 : 両方のエッジ。回路は IC1 の立上がりエッジと立下がりエッジの両方に反応します。

ビット 1 **CC1E** : キャプチャ / 比較 1 出力イネーブル。

条件 : 出力としての CC1

0 : オフ - OC1 はアクティブではありません。CC1E ビットに "0" を書き込むと、他のすべての LPTIM チャンネルが無効になっている場合にのみ、ue_dma_req 信号がリセットされます。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

条件 : 入力としての CC1

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (LPTIM_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。この CC1E ビットに "0" を書き込むと、関連する ic1_dma_req 信号がリセットされます。

1 : キャプチャは有効です。

ビット 0 **CC1SEL** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向、すなわち、入力 (キャプチャ) モードか、出力モードかを定義します。

0 : CC1 チャンネルは出力 PWM モードに設定されます。

1 : CC1 チャンネルは入力キャプチャモードに設定されます。

注意 : LPTIM_CCMRx レジスタへの書き込み後、同じレジスタへの新しい書き込み操作は、(PRESC × 3) カーネルクロックサイクルの値以上の遅延の後にのみ実行できます。PRESC[2:0] は、クロックの 10 進数の分周比 (1、2、4、..128) です。この遅延を守らずに続けて書き込みを行うと、予測不能な結果になります。

注意 : CCxSEL、ICxF[1:0]、CCxP[1:0]、および ICxPSC[1:0] フィールドは、チャンネル x が無効になっている (CCxE ビットが 0 にリセットされている) 場合にのみ変更する必要があります。
LPTIM がチャンネルを実装していない場合、このレジスタは予約済みです。[セクション 43.3](#) を参照してください。

43.7.18 LPTIM 比較レジスタ 2 (LPTIM_CCR2)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 CCR2[15:0] : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、キャプチャ／比較 2 レジスタにロードされる値です。

PRELOAD オプションに応じて、CCR2 レジスタは PRELOAD ビットがリセットされた場合は即座に更新され、PRELOAD ビットがリセットされた場合は次の LPTIM 更新イベント時に更新されます。

キャプチャ／比較レジスタ 2 は、カウンタ LPTIM_CNT と比較されて、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は読出し専用となり、最後の入力キャプチャ 2 イベントによって転送されたカウンタ値を含みます。LPTIM_CCR2 レジスタは読出し専用レジスタで、プログラムできません。

注意 : LPTIM_CCR2 レジスタの変更は、LPTIM が有効な (ENABLE ビットが 1 にセットされている) ときにのみ行う必要があります。

注 : LPTIM が実装しているチャンネルが 2 未満の場合、このレジスタは予約済みです。[セクション 43.3 : LPTIM の実装](#)を参照してください。

43.7.19 LPTIM レジスタマップ

次の表に LPTIM レジスタの一覧を示します。

表 466. LPTIM レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	LPTIM4_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOK	UE	DOWN	UP	ARROK	CMP1OK	EXTTRIG	ARRM	CC1IF
	リセット値								0																0	0	0	0	0	0	0	0	0
0x000	LPTIMx_ISR (x = 1~3、5、6) 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROK	Res.	Res.	Res.	Res.	CMP2OK ⁽¹⁾	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2IF ⁽¹⁾	REPOK	UE	DOWN	UP	ARROK	CMP1OK	EXTTRIG	ARRM	CC1IF
	リセット値								0					0										0	0	0	0	0	0	0	0	0	0
	LPTIMx_ISR (x = 1~3、5、6) 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF ⁽¹⁾	CC1OF	Res.	Res.	Res.	REPOK	UE	DOWN	UP	ARROK	Res.	EXTTRIG	ARRM	CC1IF
	リセット値								0												0	0			0	0	0	0	0	0		0	0
0x004	LPTIM4_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOKCF	UECF	DOWN	UP	ARROKCF	CMP1OKCF	EXTTRIGCF	ARRMCF	CC1CF
	リセット値								0																0	0	0	0	0	0	0	0	0
0x004	LPTIMx_ICR (x = 1~3、5、6) 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	CMP2OKCF ⁽¹⁾	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOKCF	UECF	DOWN	UP	ARROKCF	CMP1OKCF	EXTTRIGCF	ARRMCF	CC1CF
	リセット値								0					0										0	0	0	0	0	0	0	0	0	0
	LPTIMx_ICR (x = 1~3、5、6) 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIEROKCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OCF ⁽¹⁾	CC1OCF	Res.	Res.	Res.	REPOKCF	UECF	DOWN	UP	ARROKCF	Res.	EXTTRIGCF	ARRMCF	CC1CF
	リセット値								0												0	0			0	0	0	0	0	0		0	0
0x008	LPTIM4_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOKIE	UEIE	DOWN	UP	ARROKIE	CMP1OKIE	EXTTRIGIE	ARRMIE	CC1IE
	リセット値																								0	0	0	0	0	0	0	0	0
0x008	LPTIMx_DIER (x = 1~3、5、6) 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEDE	Res.	Res.	Res.	Res.	CMP2OKIE ⁽¹⁾	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPOKIE	UEIE	DOWN	UP	ARROKIE	Res.	EXTTRIGIE	ARRMIE	CC1IE
	リセット値								0					0										0	0	0	0	0	0	0	0	0	0
	LPTIMx_DIER (x = 1~3、5、6) 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OIE ⁽¹⁾	CC1OIE	Res.	Res.	Res.	REPOKIE	UEIE	DOWN	UP	ARROKIE	Res.	EXTTRIGIE	ARRMIE	CC1IE
	リセット値							0	0												0	0			0	0	0	0	0	0		0	0

表 466. LPTIM レジスタのマッピングとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00C	LPTIM_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENCOD	COUNTMODE	PRELOAD	WAVPOL ⁽³⁾	WAVE	TIMOUT	TRIGEN	Res.	Res.	TRIGSEL[2:0]	Res.	Res.	PRESC	Res.	Res.	TRGFLT	Res.	Res.	Res.	Res.	CKFLT	CKPOL	CKSEL		
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	LPTIM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE	
	リセット値																											0	0	0	0	0	0
0x014	LPTIM_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	LPTIM_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x01C	LPTIM_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	LPTIM_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2SEL [1:0]	Res.	Res.	Res.	IC1SEL [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL [1:0]	Res.	Res.	Res.	IN1SEL [1:0]	
	リセット値											0	0			0	0										0	0			0	0	
0x028	LPTIM_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]															
	リセット値																									0	0	0	0	0	0	0	0
0x02C	LPTIM_CCMR1 ⁽⁴⁾	Res.	Res.	IC2F [1:0]	Res.	Res.	IC2PSC [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	CC2P [1:0]	CC2E	CC2SEL	Res.	Res.	IC1F [1:0]	Res.	Res.	IC1PSC [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	CC1P [1:0]	CC1E	CC1SEL		
	リセット値			0	0		0	0						0	0	0	0		0	0		0	0					0	0	0	0	0	0
0x034	LPTIM_CCR2 ⁽⁵⁾	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2 [15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. LPTIM が少なくとも 2 つのチャネルを実装していない場合、このビットは予約済みです。セクション 43.3 : LPTIM の実装を参照してください。
2. LPTIM がエンコーダモード機能をサポートしない場合、このビットは予約済みです。セクション 43.3 : LPTIM の実装を参照してください。
3. LPTIM が少なくとも 1 つのキャプチャ/比較チャネルを実装している場合、このビットは予約済みです。セクション 43.3 : LPTIM の実装を参照してください。
4. LPTIM がチャネルを実装していない場合、このレジスタは予約済みです。セクション 43.3 : LPTIM の実装を参照してください。
5. LPTIM が実装しているチャネルが 2 未満の場合、このレジスタは予約済みです。セクション 43.3 : LPTIM の実装を参照してください。

レジスタ境界アドレスについては、110 ページのセクション 2.3 を参照してください。

44 独立型ウォッチドッグ (IWDG)

44.1 概要

独立型ウォッチドッグ (IWDG) ペリフェラルは、ソフトウェアまたはハードウェア障害による不具合を検出する機能により、高い安全レベルを実現しています。

IWDG は独立型クロックによりクロック供給されているため、メインクロックの障害時も機能に影響が出ません。

また、ウォッチドッグ機能は V_{DD} 電圧ドメインで実行され、低電力モードでも IWDG が機能します。この製品の IWDG の機能を確認するには、[セクション 44.3](#) を参照してください。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、予期しない行動の検知が確実に行えます。

44.2 IWDG の主な機能

- 12 ビットダウンカウンタ
- デュアル電圧ドメイン、そのため低電力モードでの動作が可能
- 独立クロック
- 早期ウェイクアップ割込みの生成
- 生成のリセット
 - タイムアウトの場合
 - 想定したウィンドウの外でのリフレッシュの場合

44.3 IWDG の実装

表 467. IWDG の機能⁽¹⁾

IWDG のモード/機能	IWDG
IWDG カーネルクロックとして使用される LSI (iwdg_ker_ck)	X
ウィンドウ関数	X
早期ウェイクアップ割込みの生成	X
システムリセット生成 ⁽²⁾	X
システム STOP で有効な機能	X
システム STANDBY で有効な機能	X
システム STOP で割込みを生成する機能	X
システム STANDBY で割込みを生成する機能	-
マイクロコントローラがデバッグモードに切り替わったときにフリーズする機能 ⁽³⁾	X
STOP モードでアクティビティを制御するためのオプションバイト ⁽⁴⁾	X
STANDBY モードでアクティビティを制御するためのオプションバイト ⁽⁵⁾	X
ハードウェアモードを制御するためのオプションバイト ⁽⁶⁾	X

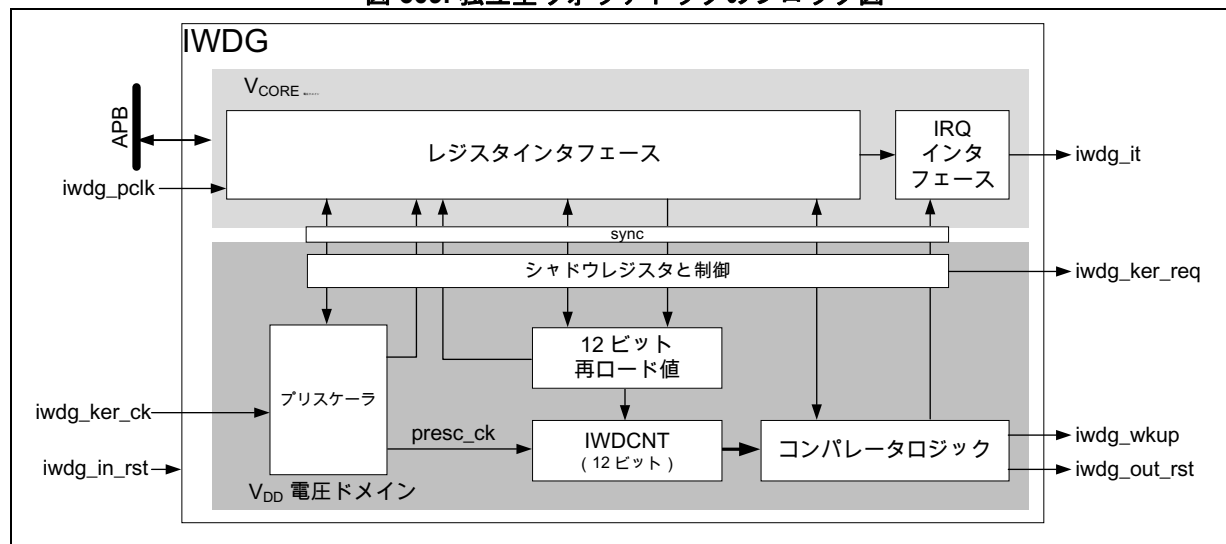
1. “X” はサポートされています。“-” はサポートされていません。
2. 詳細については、RCCのセクションを参照してください。
3. DBG セクションの DBG_IWDG_STOP により制御されます。
4. FLASH セクションのオプションバイト IWDG_STOP により制御されます。
5. FLASH セクションのオプションバイト IWDG_STDBY により制御されます。
6. FLASH セクションのオプションバイト IWDG_SW により制御されます。

44.4 IWDG の機能説明

44.4.1 IWDG ブロック図

図 609 に、独立型ウォッチドッグモジュールの機能ブロックを示します。

図 609. 独立型ウォッチドッグのブロック図



レジスタと IRQ インタフェースは V_{CORE} 電圧ドメインに配置されています。ウォッチドッグ機能自体は、低電力モードで機能できるように、 V_{DD} 電圧ドメインに配置されています。IWDG の機能については、[セクション 44.3](#) を参照してください。

レジスタと IRQ インタフェースは主に APB クロック (iwdg_pclk) によってクロック供給されますが、ウォッチドッグ機能は専用のカーネルクロック (iwdg_ker_ck) によってクロック供給されます。2つのドメイン間のデータ交換は、同期メカニズムにより行われます。レジスタインタフェースに配置されたレジスタの大部分は、V_{DD} 電圧ドメインにシャドウイングされます。

IWDG ダウンカウンタ (IWDGNT) はプリスケラクロック (presck) によってクロック供給されます。プリスケラクロックは、PR[3:0] ビットフィールドに従って、カーネルクロック iwdg_ker_ck をプリスケラで割って生成されます。

44.4.2 IWDG 内部信号

IWDG 内部信号のリストは表 468 を参照してください。

表 468. IWDG 内部输入/出力信号

信号名	信号タイプ	説明
iwdg_ker_ck	入力	IWDG カーネルクロック
iwdg_ker_req	入力	IDG カーネルクロックリクエスト
iwdg_pclk	入力	IWDG APB クロック
iwdg_out_rst	出力	IWDG リセット出力
iwdg_in_rst	入力	IWDG リセット入力

表 468. IWDG 内部入力／出力信号 (続き)

信号名	信号タイプ	説明
iwdg_wkup	出力	IWDG ウェイクアップ・イベント
iwdg_it	出力	IWDG 早期ウェイクアップ割込み

44.4.3 ソフトウェアおよびハードウェアウォッチドッグモード

ウォッチドッグモードを使用して、IWDG を有効化する方法としてソフトウェアコマンド（ソフトウェアウォッチドッグモード）または自動（ハードウェアウォッチドッグモード）を選択できます。それ以外の機能はすべて、ソフトウェアモードでもハードウェアモードでも同様に動作します。

ソフトウェアウォッチドッグモードはデフォルトの動作モードです。独立型ウォッチドッグは [IWDG キーレジスタ \(IWDG_KR\)](#) に値 0x0000 CCCC が書き込まれることによって開始され、IWDG_CNT はリセット値 (0xFFFF) からカウントダウンを開始します。

ハードウェアウォッチドッグモードでは、独立型ウォッチドッグは起動時に自動的に開始されるか、リセットのたびに (iwdg_in_rst 経由で) 開始されます。IWDG_CNT ダウンカウンタはリセット値 0xFFFF からカウントダウンを開始します。ハードウェアウォッチドッグモードの機能は、デバイスオプションビットを通じて有効化されます。詳細は[セクション 44.3](#)を参照してください。

IWDG が有効である場合、ONF フラグが 1 にセットされます。

IWDG_CNT が 0x000 に達すると、リセット信号が生成されます (iwdg_out_rst をアサート)。

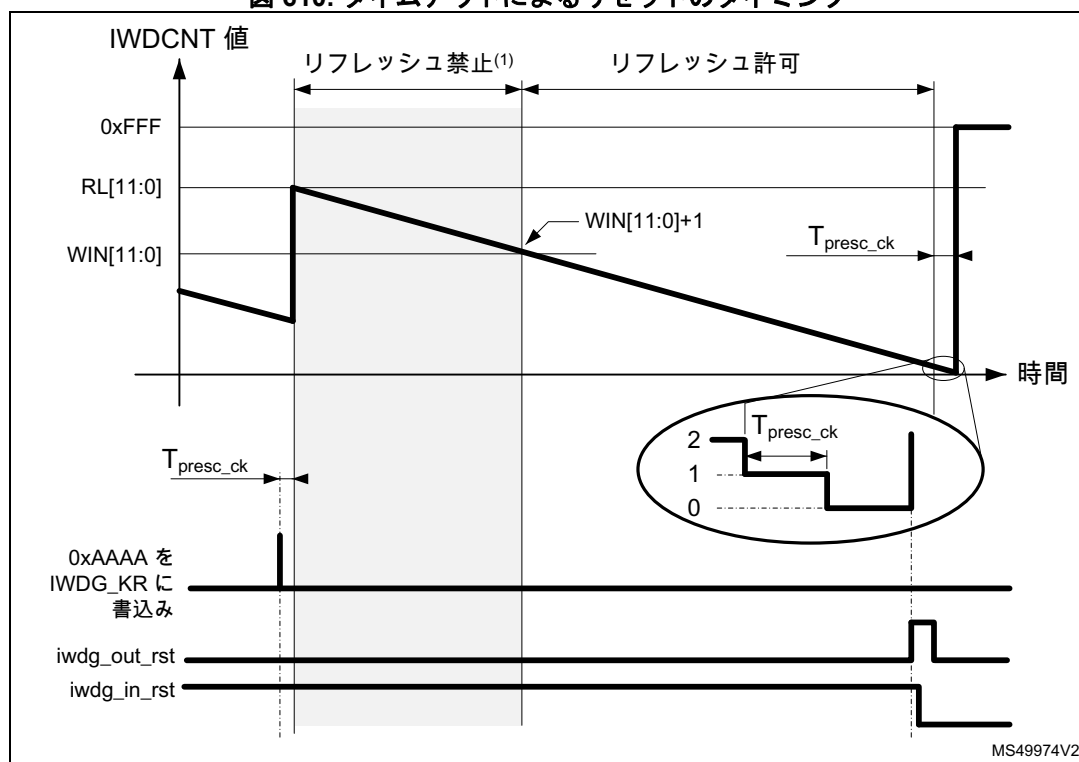
[IWDG キーレジスタ \(IWDG_KR\)](#) にキー値 0x0000 AAAA が書き込まれると、IWDG_RLR の値が IWDG_CNT に再ロードされ、ウォッチドッグのリセットが防止されます。

再同期化の遅延のため、IWDG_CNT ダウンカウンタが 1 に達する前に IWDG をリフレッシュする必要があります。

IWDG が開始された後、停止できるのはリセット時のみです (iwdg_in_rst をアサート)。

図 610 に示すように、リフレッシュコマンドが実行されると、presc_ck の 1 周期後で、RL[11:0] の内容により IWD CNT が再ロードされます。

図 610. タイムアウトによるリセットのタイミング



1. ウィンドウオプションが有効化されている場合

IWD CNT が 1 に達する前に IWDG がリフレッシュされない場合、IWDG によりリセットが生成されます (iwdg_out_rst をアサート)。それに対して、RCC により IWDG がリセットされ (iwdg_in_rst をアサート)、リセットソースがクリアされます。

44.4.4 ウィンドウオプション

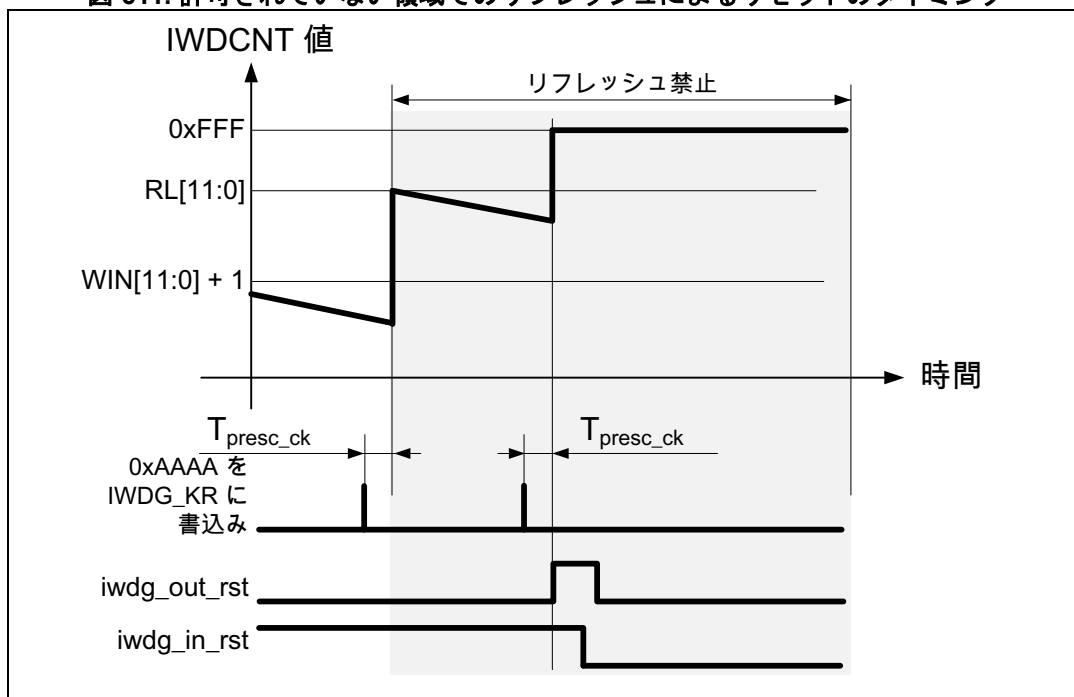
IWDG は、**IWDG ウィンドウレジスタ (IWDG_WINR)** に適切なウィンドウをセットすることによって、ウィンドウ型ウォッチドッグとしても機能します。

カウンタが WIN[11:0] + 1 より大きい間に再ロード操作が行われると、リセットが生成されます。WIN[11:0] は **IWDG ウィンドウレジスタ (IWDG_WINR)** に配置されています。図 611 に示すように、想定しないリフレッシュコマンドから presc_ck の 1 周期後にリセットが生成されます。

IWDG ウィンドウレジスタ (IWDG_WINR) のデフォルト値は 0x0000 0FFF です。そのため、この値が更新されない場合は、ウィンドウオプションは無効化されます。

ウィンドウ値が変更されるとすぐに、次のリフレッシュが必要な箇所を容易に予測できるように、ダウンカウンタ (IWD CNT) に RL[11:0] 値が再ロードされます。

図 611. 許可されていない領域でのリフレッシュによるリセットのタイミング



ウィンドウオプションが有効な場合の IWDG の設定

1. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. **IWDG プリスケアラレジスタ (IWDG_PR)** をプログラムすることによって、IWDG のプリスケアラに書き込みを行います。
4. **IWDG 再ロードレジスタ (IWDG_RLR)** に書き込みます。
5. 必要に応じて、早期ウェイクアップ割込みを有効化し、**IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)** に適切な値を書き込んで早期ウェイクアップ・コンパレータをプログラムします。
6. **IWDG ウィンドウレジスタ (IWDG_WINR)** に書き込みます。これにより、自動的に IWDG_CNT ダウンカウンタに RL[11:0] 値が再ロードされます。
7. レジスタが更新されるのを待ちます (IWDG_SR = 0x0000 0000)。
8. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 0000 を書き込んで、レジスタを書込み保護します。

注： このシーケンスの完了後にアプリケーションで APB クロックを無効化するつもりがない場合、手順 7 はスキップできます。

ウィンドウオプションが無効な場合の IWDG の設定

ウィンドウオプションが使用されていない場合、IWDG は以下のように設定することができます。

1. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. **IWDG プリスケアラレジスタ (IWDG_PR)** をプログラムすることによって、プリスケアラに書き込みを行います。
4. **IWDG 再ロードレジスタ (IWDG_RLR)** に書き込みます。
5. 必要に応じて、早期ウェイクアップ割込みを有効化し、**IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)** に適切な値を書き込んで早期ウェイクアップ・コンパレータをプログラムします。
6. レジスタが更新されるのを待ちます (IWDG_SR = 0x0000 0000)。
7. RL[11:0] 値によりカウンタをリフレッシュし、**IWDG キーレジスタ (IWDG_KR)** に 0x0000 AAAA を書き込むことによってレジスタを書込み保護します。

ウィンドウコンパレータの更新

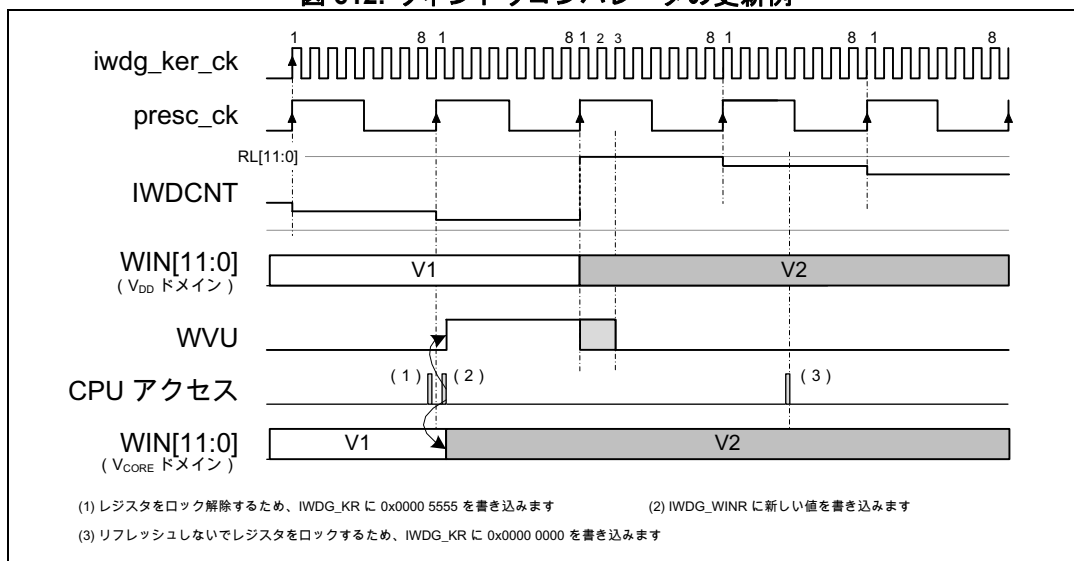
IWDG の動作中は、ウィンドウコンパレータを更新できます。IWDCNT も再ロードされます。ウィンドウコンパレータを更新するためのシーケンスは次のとおりです。

1. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 5555 を書き込むことにより、レジスタアクセスを有効化します。
2. **IWDG ウィンドウレジスタ (IWDG_WINR)** に書き込みます。これにより、自動的に IWDCNT ダウンカウンタに RL[11:0] 値が再ロードされます。
3. WVU = 0 を待ちます。
4. **IWDG_KR** に 0x0000 0000 を書き込むことにより、レジスタをロックします。

このシーケンスの完了後にアプリケーションで APB クロックを無効化するつもりがない場合、手順 3 はスキップできます。

図 612 にこのシーケンスを示します。IWDG_WINR レジスタが書き込まれるとすぐに WVU フラグがハイになります。新しいウィンドウ値と IWDCNT への RL[11:0] の再ロードは、presc_ck の次の立ち上がりエッジで有効になります。WVU フラグは、最悪のケースではカーネルクロック 2 周期後に 0 に戻ります。そのため、WVU は最大で presc_ck の 1 周期分とカーネルクロック 2 周期分、ハイのままとなります。

図 612. ウィンドウコンパレータの更新例



44.4.5 デバッグ

プロセッサがデバッグモードになると（コアは停止状態）、IWDG_CNT ダウンカウンタは、製品のデバッグ機能に応じて、通常どおりに動作を続けるか、または停止します。この製品の機能の詳細については、[セクション 44.3](#) を参照してください。

44.4.6 レジスタのアクセス保護

IWDG プリスケアラレジスタ (IWDG_PR)、IWDG 再ロードレジスタ (IWDG_RLR)、IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)、および IWDG ウィンドウレジスタ (IWDG_WINR) への書き込みアクセスは保護されます。これを変更するには、まず IWDG キーレジスタ (IWDG_KR) に 0x0000 5555 を書き込みます。このレジスタに別の値で書き込みアクセスすると、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0x0000 AAAA の書き込み) の場合に相当します。

ステータスレジスタは、プリスケアラ、あるいはダウンカウンタ再ロード値やウィンドウ値の更新が行われていることを示すために使用されます。

44.5 IWDG 低電力モード

オプション・バイトの設定によって、IWDG では低電力モード時にカウントを継続することも、継続しないこともできます。詳細については、[セクション 44.3](#) を参照してください。

表 469. 低消費電力モードが IWDG に与える影響

モード	説明
SLEEP	影響はありません。IWDG 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	オプションバイトの設定によって、IWDG は有効または無効になります。 詳細については、 セクション 44.3 を参照してください。 IWDG 割込みによって、デバイスは STOP モードを終了します。
STANDBY	オプションバイトの設定によって、IWDG は有効または無効になります。 詳細については、 セクション 44.3 を参照してください。 IWDG 割込みにより、デバイスが STANDBY モードを終了することはありません。

44.6 IWDG 割込み

IWDG には、ダウンカウンタの値に応じて早期割込みを生成する機能があります。早期割込みを有効にするには、[IWDG 早期ウェイクアップ割込みレジスタ \(IWDG_EWCR\)](#) の EWIE ビットを 1 に設定します。

コンパレータ値 (EWIT[11:0]) を使用すると、早期割込みを生成する箇所をアプリケーションで定義できます。

IWDCNT ダウンカウンタが EWIT[11:0] - 1 の値に達すると、iwdg_wkup が有効化され、必要に応じて低電力モードを終了することが可能になります。

APB クロックが利用できる場合、iwdg_it も有効化されます。

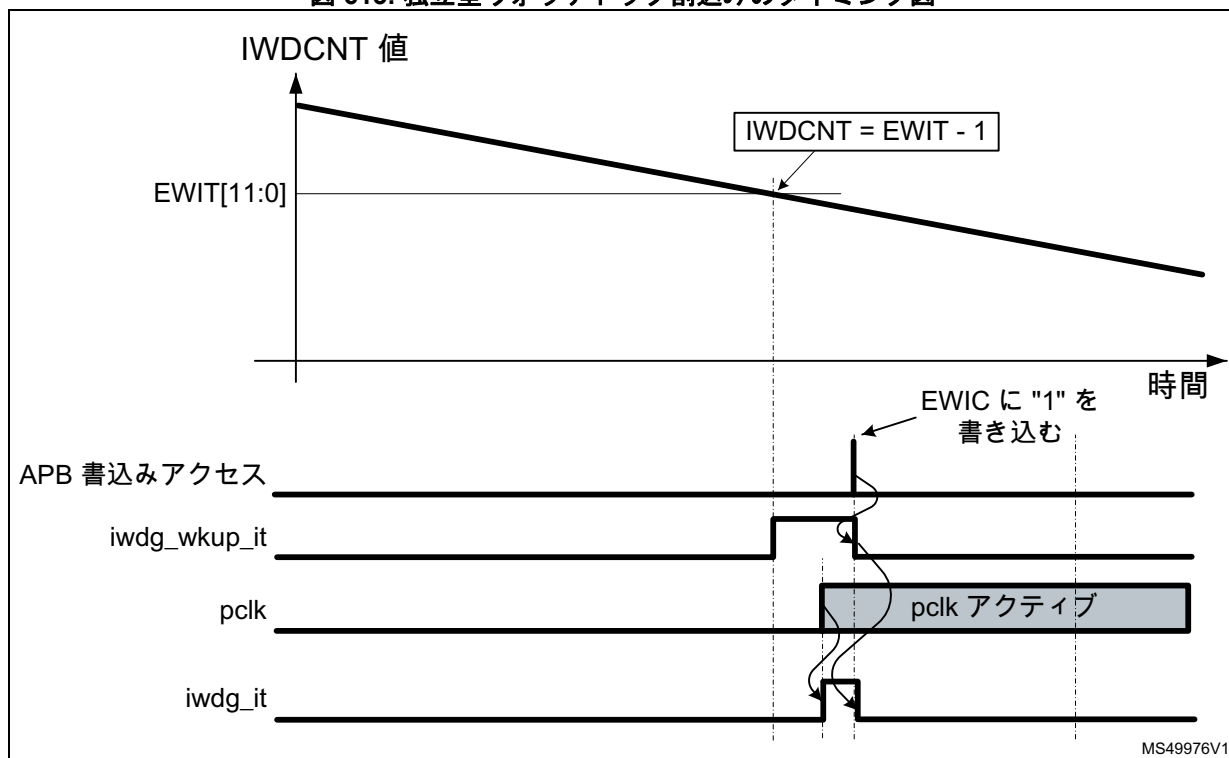
また、[IWDG ステータスレジスタ \(IWDG_SR\)](#) のフラグ EWIF が 1 に設定されます。

EWI 割込みを確認応答するには、[IWDG 早期ウェイクアップ割込みレジスタ \(IWDG_EWCR\)](#) の EWIC ビットに "1" を書き込みます。

また、IWDG_EWCR レジスタに書き込むと、ダウンカウンタ (IWDCNT) が再ロード値 RL[11:0] でリフレッシュされます。



図 613. 独立型ウォッチドッグ割込みのタイミング図



ウォッチドッグのリセットが生成される前に特定の安全処理やデータロギングを実施する必要がある場合は、早期ウェイクアップ割込み (EWI) が使用できます。

早期ウェイクアップ・コンパレータ値の変更

次のシーケンスにより、いつでも早期ウェイクアップ・コンパレータ値を変更したり、割込み生成を有効化/無効化したりできます。

1. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
2. 早期ウェイクアップ割込みを有効化/無効化したり、**IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)** に適切な値を書き込んで早期ウェイクアップ・コンパレータをプログラムしたりします。
3. EWU = '0' を待ちます。EWU は**IWDG ステータスレジスタ (IWDG_SR)** に配置されています。
4. **IWDG キーレジスタ (IWDG_KR)** に 0x0000 0000 を書き込むことにより、レジスタを書込み保護します。

このシーケンスの完了後にアプリケーションで APB クロックを無効化するつもりがない場合、手順 3 はスキップできます。

図 613 にこのシーケンスを示します。IWDG_EWCR レジスタが書き込まれるとすぐに EWU フラグがハイになります。新しいコンパレータ値と IWDGNT への RL[11:0] の再ロードは、presc_ck の次の立ち上がりエッジで有効になります。EWU フラグは、最悪のケースではカーネルクロック 2 周期後に 0 に戻ります。そのため、EWU は最大で presc_ck の 1 周期分とカーネルクロック 2 周期分、ハイのままとなります。

図 614. 早期ウェイクアップ・コンパレータの更新例

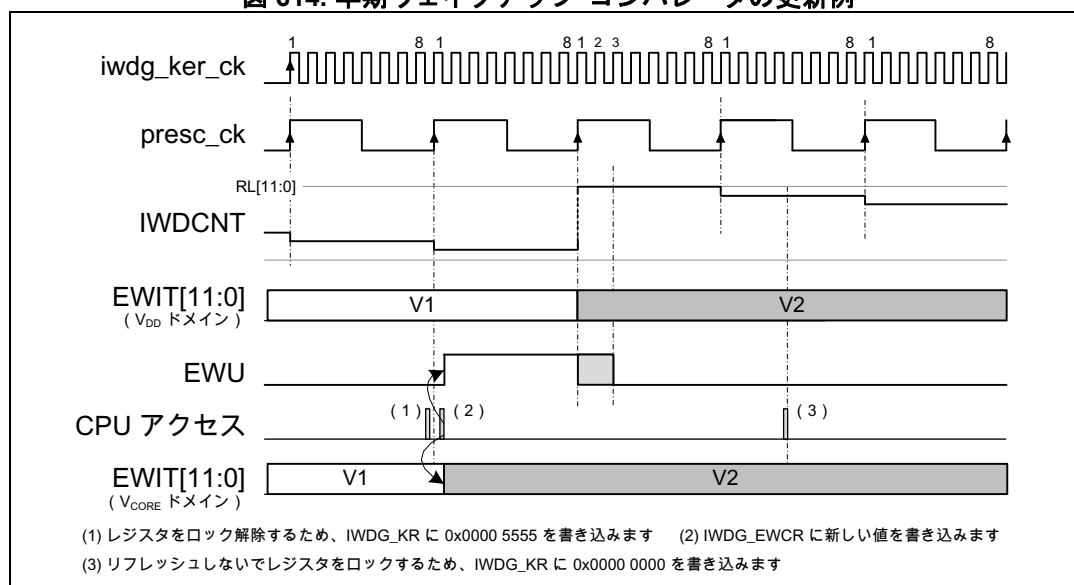


表 470 に IWDG 割込みリクエストをまとめます。

表 470. IWDG 割込みリクエスト

割込みイベント	イベントフラグ	割込みのクリア方法	割込み有効制御ビット	有効な割込み	
				iwdg_it	iwdg_wkup_it
IWDCNT が EWIT 値に到達	EWIF	EWIC に '1' を書き込む	EWIE	Y ⁽¹⁾	Y ⁽²⁾

1. iwdg_pclk 入力にクロックがある場合に生成されます。
2. iwdg_ker_ck 入力にクロックがある場合に生成されます。

44.7 IWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

レジスタインタフェースに配置されたレジスタの大部分は、V_{DD} 電圧ドメインにシャドウイングされます。iwdg_in_rst がアサートされると、V_{DD} 電圧ドメインに配置されたウォッチドッグのロジックとシャドウレジスタがリセットされます。

アプリケーションでウォッチドッグレジスタが読み戻されると、対応するシャドウレジスタの値がハードウェアによりレジスタインタフェースに転送されます。

アプリケーションでウォッチドッグレジスタが書き込まれると、対応するシャドウレジスタがハードウェアで更新されます。

44.7.1 IWDG キーレジスタ (IWDG_KR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **KEY[15:0]** : キー値 (書込み専用、読出しでは 0x0000)

これらのビットは、アプリケーションによって書き込まれた値に応じて、いくつかの機能に使用できます。

- 0xAAAA : IWDGNT ダウンカウンタに RL[11:0] 値を再ロードし (ウォッチドッグのリフレッシュ)、レジスタを書込み保護します。この値は、ソフトウェアによって一定間隔で書き込まれなければなりません。そうしないと、カウンタが 0 に達したときにウォッチドッグによりリセットが生成されます。

- 0x5555 : レジスタへの書込みアクセスを有効にします。

- 0xCCCC : レジスタのウォッチドッグ (ハードウェアウォッチドッグオプションが選択されている場合を除く) と書込み保護を有効にします。

- 0x5555 以外の値 : レジスタを書込み保護します。

書込み保護メカニズムを備えているのは、IWDG_PR、IWDG_RLR、IWDG_EWCR、および IWDG_WINR レジスタのみです。

44.7.2 IWDG プリスケーラレジスタ (IWDG_PR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[3:0]			
													rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PR[3:0]** : プリスケーラ分周回路

これらのビットは、書込みアクセス保護されています。[セクション 44.4.6](#) を参照してください。カウンタクロックを供給するプリスケーラ分周回路を選択するようにソフトウェアで書き込まれます。プリスケーラ分周回路を変更できるようにするには、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の PVU ビットがリセットされる必要があります。

0000 : 4 分周

0001 : 8 分周

0010 : 16 分周

0011 : 32 分周

0100 : 64 分周

0101 : 128 分周

0110 : 256 分周

0111 : 512 分周

その他 : 1024 分周

注 : このレジスタを読み出すと、V_{DD} 電圧ドメインからプリスケーラ値が返されます。このレジスタへの書込み操作が進行中の場合には、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の PVU ビットがリセットされているときのみとなります。

44.7.3 IWDG 再ロードレジスタ (IWDG_RLR)

アドレスオフセット : 0x08

リセット値 : 0x0000 : 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RL[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **RL[11:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書込みアクセス保護されています。[セクション 44.4.6](#) を参照してください。[IWDG キーレジスタ \(IWDG_KR\)](#) に値 0xAAAA が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とプリスケールクロックによって決まります。RL[11:0] に 2 未満の値を設定することは推奨しません。

再ロード値を変更できるようにするには、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の RVU ビットがリセットされる必要があります。

注 : このレジスタを読み出すと、V_{DD} 電圧ドメインから再ロード値が返されます。このレジスタへの書込み操作が進行中の場合には、この値は最新でないか、有効でないことがあるため、このレジスタから読み出された値が有効なのは、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の RVU ビットがリセットされているときのみとなります。

44.7.4 IWDG ステータスレジスタ (IWDG_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (0xFFFF FEFF)

このレジスタは各種ステータスフラグを備えています。カッコ内のマスク値は ONF ビットのリセット値が定義されていないことを示します。IWDG がソフトウェアモードで設定されている場合、ONF ビットのリセット値は 0 となり、IWDG がハードウェアモードで設定されている場合、ONF ビットのリセット値は 1 となります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EWIF	Res.	Res.	Res.	Res.	Res.	ONF	Res.	Res.	Res.	Res.	EWU	WVU	RVU	PVU
	r						r					r	r	r	r

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **EWIF** : ウォッチドッグの早期割込みフラグ

このビットは、早期割込みが保留中であることを示すため、ハードウェアによって "1" にセットされます。このビットをソフトウェアでクリアするには、IWDG_EWCR レジスタのビット EWIC に "1" を書き込む必要があります。

ビット 13:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 ONF : ウォッチドッグイネーブルステータスビット

IWDG が始動するとすぐにハードウェアにより "1" にセットされます。ソフトウェアモードでは、IWDG がリセットされるまで "1" のままです。ハードウェアモードでは、このビットは常に "1" にセットされます。

0 : IWDG は有効化されていません。

1 : IWDG が有効化されています。アプリケーションによって定期的にリフレッシュされる必要があります。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 EWU : ウォッチドッグ割込みコンパレータ値の更新

このビットは、割込みコンパレータ値 (EWIT[11:0]) の更新または EWID の更新が進行中であることを示すために、ハードウェアによってセットされます。V_{DD} 電圧ドメインで更新処理が完了するとハードウェアによってリセットされます (最大で presc_ck の 1 周期分と IWDG カーネルクロック iwdg_ker_ck の 2 周期分かかります)。

EWIT[11:0] および EWIE フィールドは、EWU ビットがリセットされているときのみ更新できます。

ビット 2 WVU : ウォッチドッグカウンタウィンドウ値の更新

このビットは、ウィンドウ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V_{DD} 電圧ドメインで再ロード値の更新処理が完了するとハードウェアによってリセットされます (最大で presc_ck の 1 周期分と IWDG カーネルクロック iwdg_ker_ck の 2 周期分かかります)。

ウィンドウ値は、WVU ビットがリセットされているときのみ更新できます。

このビットは、一般のウィンドウが 1 の場合のみ生成されます。

ビット 1 RVU : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。V_{DD} 電圧ドメインで再ロード値の更新処理が完了するとハードウェアによってリセットされます (最大 6 IWDG カーネルクロック iwdg_ker_ck の 6 周期分かかります)。

再ロード値は、RVU ビットがリセットされているときのみ更新できます。

ビット 0 PVU : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V_{DD} 電圧ドメインでプリスケアラ更新処理が完了するとハードウェアによってリセットされます (最大 IWDG カーネルクロック iwdg_ker_ck の 6 周期分かかります)。

プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。

注 : 複数の再ロード値、プリスケアラ値、早期割込み位置、またはウィンドウ値がアプリケーションで 사용되는場合は、それぞれ、再ロード値を変更する前に RVU ビットがリセットされるまで待つか、プリスケアラ値を変更する前に PVU ビットがリセットされるまで待つか、ウィンドウ値を変更する前に WVU ビットがリセットされるまで待つか、早期割込み位置値を変更する前に EWU ビットがリセットされるまで待つ必要があります。プリスケアラ値、再ロード値、ウィンドウ値、早期割込み値を更新した後は、RVU、PVU、WVU、EWU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードが開始された場合を除く)。

44.7.5 IWDG ウィンドウレジスタ (IWDG_WINR)

アドレスオフセット : 0x10

リセット値 : 0x0000 : 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WIN[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **WIN[11:0]** : ウォッチドッグカウンタウィンドウ値

これらのビットは書込みアクセス保護されており (セクション 44.4.6 を参照)、ダウンカウンタと比較されるウィンドウ値の上限を格納しています。

リセットを防ぐには、カウンタの値が WIN[11:0] + 1 よりも小さく、1 よりも大きい IWDG_CNT ダウンカウンタを再ロードする必要があります。

再ロード値を変更できるようにするには、**IWDG ステータスレジスタ (IWDG_SR)** の WVU ビットがリセットされる必要があります。

注 : このレジスタを読み出すと、V_{DD} 電圧ドメインから再ロード値が返されます。このレジスタへの書込み操作が進行中の場合には、この値は有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、**IWDG ステータスレジスタ (IWDG_SR)** の WVU ビットがリセットされているときのみとなります。

44.7.6 IWDG 早期ウェイクアップ割込みレジスタ (IWDG_EWCR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EWIE	EWIC	Res.	Res.	EWIT[11:0]											
rW	W			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EWIE** : ウォッチドッグの早期割込みイネーブル

ソフトウェアでセット/クリアされます。

0 : 早期割込みインタフェースは無効です。

1 : 早期割込みインタフェースは有効です。

このビットの値を変更できるようにするには、**IWDG ステータスレジスタ (IWDG_SR)** の EWU ビットがリセットされる必要があります。

ビット 14 **EWIC** : ウォッチドッグの早期割込み確認応答

早期ウェイクアップ割込みを確認応答し、EWIF フラグをクリアするには、ソフトウェアでこのビットに 1 を書き込む必要があります。0 を書き込んだ場合は無効で、このフラグを読み出すと 0 が返されます。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **EWIT[11:0]** : ウォッチドッグカウンタウィンドウ値

これらのビットは、書き込みアクセス保護されています ([セクション 44.4.6](#) を参照)。早期ウェイクアップ割込みを生成する IWDGNT ダウンカウンタの位置を定義するために、ソフトウェアによって書き込まれます。早期割込みは、IWDGNT が EWIT[11:0] - 1 以下の場合に生成されます。

EWIT[11:0] は 1 より大きくなければなりません。

EWIE = 1 の場合のみ割込みが発生します。

再ロード値を変更できるようにするには、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の EWU ビットがリセットされる必要があります。

注： このレジスタを読み出すと、早期ウェイクアップのコンパレータ値と V_{DD} 電圧ドメインの割込み有効ビットが返されます。このレジスタへの書き込み操作が進行中の場合には、この値は最新でないか、有効でないことがあるため、このレジスタから読み出された値が有効なのは、[IWDG ステータスレジスタ \(IWDG_SR\)](#) の EWU ビットがリセットされているときのみとなります。

44.7.7 IWDG レジスタマップ

表 471. IWDG レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
0x00	IWDG_KR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY[15:0]																									
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
0x04	IWDG_PR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR[3:0]													
	リセット値																														0	0	0	0									
0x08	IWDG_RLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RL[11:0]																				
	リセット値																					1	1	1	1	1	1	1	1	1	1	1	1	1									
0x0C	IWDG_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EWIF	Res	Res	Res	Res	Res	Res	ONF	Res	1	1	1	1	1	1	1	1								
	リセット値																		0							x						0	0	0	0								
0x10	IWDG_WINR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WIN[11:0]																				
	リセット値																					1	1	1	1	1	1	1	1	1	1	1	1	1									
0x14	IWDG_EWCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EWIE	EWIC	Res	Res	Res	EWIT[11:0]																				
	リセット値																	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0								

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

45 システムウィンドウ型ウォッチドッグ (WWDG)

45.1 概要

システムウィンドウ型ウォッチドッグ (WWDG) は、通常、外部の影響や予期しない論理条件などによって発生し、アプリケーションプログラムを正常なシーケンスから逸脱させるソフトウェア障害の発生を検出するために使用されます。

ウォッチドッグ回路は、T6 ビットがクリアされる前にプログラムがダウンカウンタの内容をリフレッシュしない限り、プログラムされた時間の経過後にリセットを生成します。リセットは、ダウンカウンタがウィンドウレジスタ値に達する前に 7 ビットのダウンカウンタの値 (制御レジスタ内) がリフレッシュされた場合にも生成されます。このことは、限られた時間枠 (time-window) の間にカウンタがリフレッシュされなければならないことを意味します。

WWDG クロックは、APB クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。

45.2 WWDG の主な機能

- プログラム可能なフリーランニングダウンカウンタ
- 条件付きリセット
 - ダウンカウンタの値が 0x40 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
 - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合) (図 616 を参照)。
- 早期ウェイクアップ割込み (EWI) : ダウンカウンタが 0x40 になったときにトリガ (有効であり、ウォッチドッグがアクティブな場合)

45.3 WWDG の実装

表 472. WWDG の機能⁽¹⁾

WWDG のモード/機能	WWDG
ウィンドウ関数	X
早期ウェイクアップ割込みの生成	X
システムリセット生成 ⁽²⁾	X
システム STOP で有効な機能	-
システム STANDBY で有効な機能	-
マイクロコントローラがデバッグモードに切り替わったときにフリーズする機能 ⁽³⁾	X
ハードウェアモードを制御するためのオプションバイト ⁽⁴⁾	X

1. "X" はサポートされています。"-" はサポートされていません。

2. 詳細については、RCCのセクションを参照してください。

3. DBG ブロックの DBG_WWDG_STOP により制御されます。

4. オプションバイト WWDG_SW によって制御されます。WWDG_SW が HW モードで設定されている場合、CPU が RUN モードまたは SLEEP モードに切り替わるとすぐに WWDG は動作します。

45.4 WWDG の機能説明

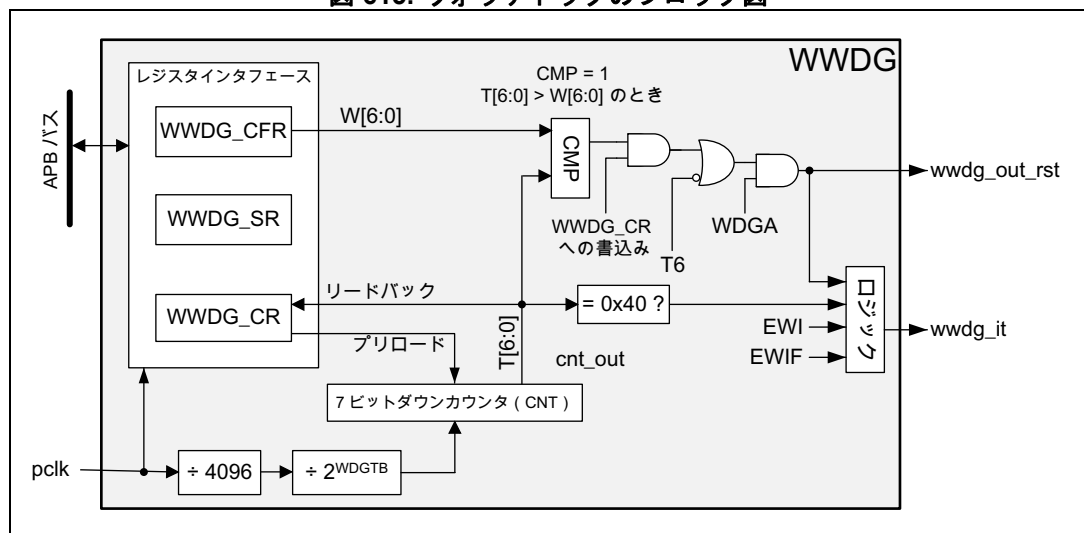
ウォッチドッグが有効な (WWDG_CR レジスタの WDGA ビットがセットされている) 場合、7 ビットのダウンカウンタ (T[6:0] ビット) が 0x40 に達して 0x3F にデクリメントされた (T6 がクリアされた) 時点で、リセットを開始します。カウンタがウィンドウレジスタに格納された値より大きい間にソフトウェアがカウンタを再ロードした場合にも、リセットが生成されます。

アプリケーションプログラムは、通常動作時には定期的に WWDG_CR レジスタへの書き込みを行って、リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値以下のとき、かつ 0x3F より高いときに行えます。WWDG_CR レジスタに格納される値は、0xFF から 0xC0 の間でなければなりません。

WWDG のブロック図については、図 615 を参照してください。

45.4.1 WWDG ブロック図

図 615. ウォッチドッグのブロック図



45.4.2 WWDG 内部信号

表 473 に WWDG の内部信号のリストを示します。

表 473. WWDG 内部入力／出力信号

信号名	信号タイプ	説明
pclk	デジタル入力	APB バスクロック
wwdg_out_rst	デジタル出力	WWDG リセット信号出力
wwdg_it	デジタル出力	WWDG 早期割込み出力

45.4.3 ウォッチドッグの有効化

ユーザオプション WWDG_SW で“ソフトウェアウィンドウウォッチドッグ”を選択すると、リセット後、ウォッチドッグは常に無効です。これを有効にするには、WWDG_CR レジスタの WDGA ビットをセットします。この後は、リセット以外の方法でウォッチドッグを無効にすることはできません。

ユーザオプション WWDG_SW で“ハードウェアウィンドウウォッチドッグ”を選択すると、リセット後、ウォッチドッグは常に有効であり、無効にできません。

45.4.4 ダウンカウンタの制御

このダウンカウンタはフリーランニングであり、ウォッチドッグが無効状態であってもカウントダウンを続けます。ウォッチドッグを有効にするときには、T6 ビットをセットして、ただちにリセットが生成されるのを防ぐ必要があります。

T[5:0] ビットは、ウォッチドッグがリセットを生成するまでの時間遅延を表すインクリメント数を含みます。このタイミングは、WWDG_CR レジスタへの書き込み時のプリスケアラの状態が不明なので最小値から最大値の間で変化します (図 616 参照)。WWDG 設定レジスタ (WWDG_CFR) はウィンドウの上限値を含みます。リセットを防ぐには、カウンタの値がウィンドウレジスタの値以下で、0x3F よりも大きい間にダウンカウンタを再ロードする必要があります。図 616 に、ウィンドウ型ウォッチドッグのプロセスを示します。

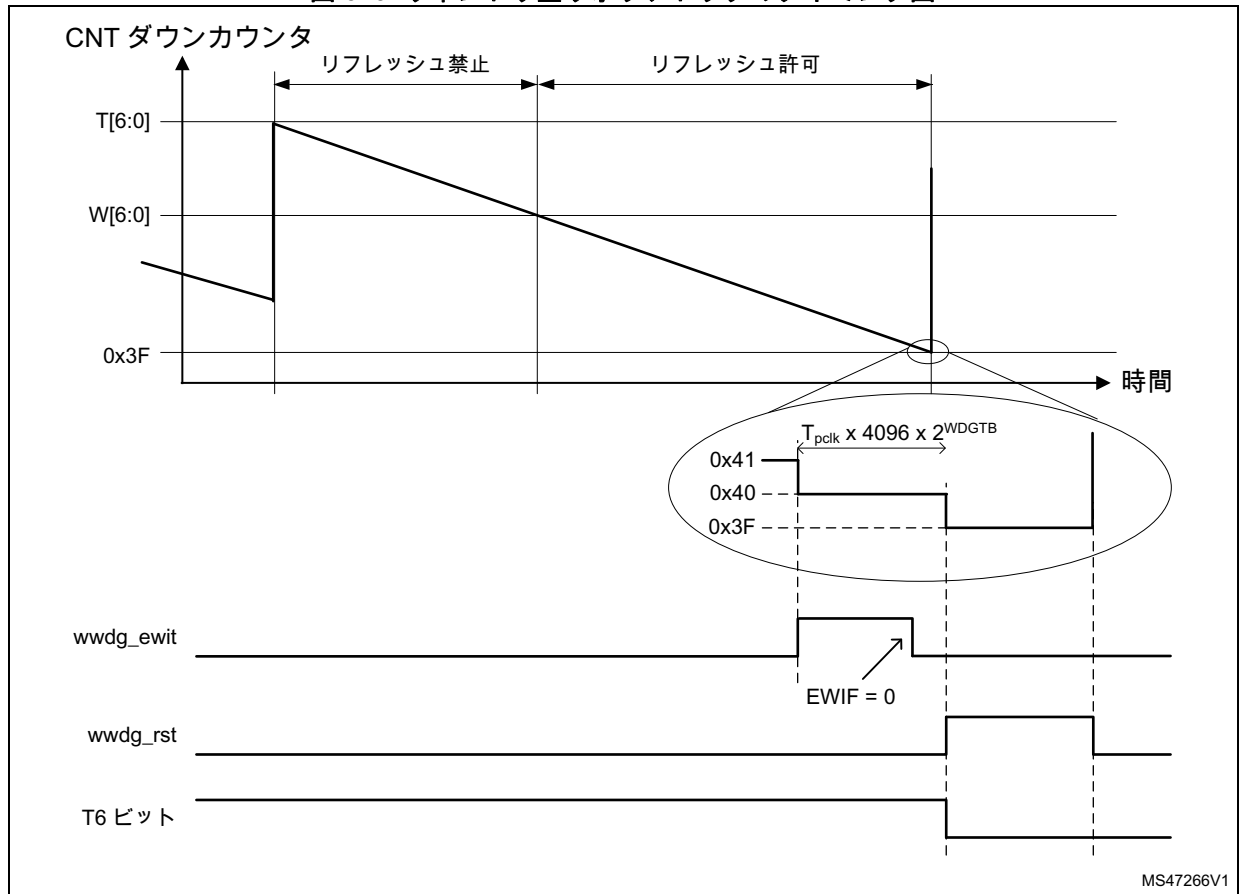
注： T6 ビットを使用して、ソフトウェアリセットを生成することができます (WDGA ビットはセット、T6 ビットはクリアされる)。

45.4.5 ウォッチドッグタイムアウトをプログラムする方法

図 616 の式を使用して、WWDG のタイムアウトを計算します。

警告： WWDG_CR レジスタに書き込むときには、ただちにリセットされるのを防ぐために、常に T6 ビットに 1 を書き込んでください。

図 616. ウィンドウ型ウォッチドッグのタイミング図



タイムアウト値は次の式で算出されます。

$$t_{\text{WWDG}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}[2:0]} \times (T[5:0] + 1) \quad (\text{ms})$$

ここで、

t_{WWDG} : WWDG タイムアウト

t_{PCLK} : APB クロック周期の測定値 (ms)

4096 : 内部分周器に対応する値

たとえば、APB 周波数が 48 MHz の場合、WDGTB[2:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = (1/48000) \times 4096 \times 2^3 \times (63 + 1) = 43.69\text{ms}$$

t_{WWDG} の最小値と最大値については、データシートを参照してください。

45.4.6 デバッグモード

デバイスがデバッグモードになると（プロセッサは停止状態）、WWDG カウンタは、DBG モジュールの設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 58 : デバッグサポート \(DBG\)](#) を参照してください。

45.5 WWDG 割込み

リセットが生成される前に特定の安全処理やデータロギングを実施する必要がある場合は、早期ウェイクアップ割込み（EWI）が使用できます。早期ウェイクアップ割込みを有効にするには、アプリケーションで以下の操作を行う必要があります。

- WWDG_SR レジスタの EWIF ビットに 0 を書き込み、望ましくない保留中の割込みをクリアする
- WWDG_CFR レジスタの EWI ビットに 1 を書き込み、割込みを有効にする

ダウンカウンタ値が 0x40 に到達すると、ウォッチドッグ割込みが生成され、対応する割込みサービスルーチン（ISR）を使用して、デバイスをリセットする前に特定の処理（通信やデータロギングなど）をトリガすることができます。

アプリケーションによっては、EWI 割込みを使用して、WWDG リセットを生成せずにソフトウェアのシステムチェックやシステム復旧／グレースフルデグラデーションを管理することができます。この場合、対応する ISR で WWDG カウンタを再ロードし、WWDG リセットを回避してから必要な操作をトリガしてください。

ウォッチドッグ割込みは、WWDG_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされます。

注：優先順位の高いタスクにおけるシステムロックなどによってウォッチドッグ割込みが使用できない場合、最終的には WWDG リセットが生成されます。

表 474. WWDG 割込みリクエスト

割込み		イベント フラグ	有効制御 ビット	割込みのクリア方法	モードの終了		
ベクタ	イベント				SLEEP	STOP (1)	STANDBY (1)
WWDG ⁽²⁾	早期ウェイクアップ割込み	EWIF	EWI	EWIF フラグに 0 を書き込む	可能	不可	不可

1. WWDG 割込みでは追加の機能が利用できます。詳細は、[セクション 45.3](#) を参照してください。
2. WWDG ベクタは wwdg_it 信号のアサートに相当します。

45.6 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスすることができます。

45.6.1 WWDG 制御レジスタ (WWDG_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
								rs	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA** : 有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。
WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。

0 : ウォッチドッグは無効です。

1 : ウォッチドッグは有効です。

ビット 6:0 **T[6:0]** : 7 ビットカウンタ (MSB から LSB まで)

これらのビットは、ウォッチドッグカウンタの値を含みます。

($4096 \times 2^{WDGTB[2:0]}$) PCLK サイクルごとにデクリメントされます。0x40 に達して 0x3F にデクリメントされると (T6 がクリアされると)、リセットが生成されます。

45.6.2 WWDG 設定レジスタ (WWDG_CFR)

アドレスオフセット : 0x004

リセット値 : 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	WDGTB[2:0]			Res.	EWI	Res.	Res.	W[6:0]						
		rw	rw	rw		rs			rw	rw	rw	rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:11 **WDGTB[2:0]** : タイムベース

プリスケアラのタイムベースは、次のように変更できます。

000 : CK カウンタクロック (PCLK/4096) 1 分周

001 : CK カウンタクロック (PCLK/4096) 2 分周

010 : CK カウンタクロック (PCLK/4096) 4 分周

011 : CK カウンタクロック (PCLK/4096) 8 分周

100 : CK カウンタクロック (PCLK/4096) 16 分周

101 : CK カウンタクロック (PCLK/4096) 32 分周

110 : CK カウンタクロック (PCLK/4096) 64 分周

111 : CK カウンタクロック (PCLK/4096) 128 分周

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割込み有効化

ソフトウェアでセットされ、リセット後はハードウェアによってクリアされます。このビットがセットされているときには、カウンタの値が 0x40 に達したときに割込みが発生します。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

45.6.3 WWDG ステータスレジスタ (WWDG_SR)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EWIF** : 早期ウェイクアップ割込みフラグ

このビットは、カウンタの値が 0x40 に達したときにハードウェアによってセットされます。0 を書き込んでソフトウェアでクリアする必要があります。1 を書き込んでも、ビットの値は変化しません。このビットは、割込みが有効でない場合にもセットされます。

45.6.4 WWDG レジスタマップ

次の表に、WWDG レジスタマップとリセット値を示します。

表 475. WWDG レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x000	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]									
	リセット値																									0	1	1	1	1	1	1	1	1		
0x004	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGTB [2:0]		Res.		EWI	Res.	Res.	W[6:0]									
	リセット値																			0	0	0		0			1	1	1	1	1	1	1	1		
0x008	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF			
	リセット値																																0			

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

46 リアルタイムクロック (RTC)

46.1 概要

RTC は、あらゆる低消費電力モードを管理する自動ウェイクアップ機能を提供します。

本リアルタイムクロック (RTC) は、独立した BCD タイマ／カウンタです。RTC は、プログラム可能なアラーム割込みを備えた時刻クロック／カレンダーを搭載しています。

供給電圧が動作範囲内にある間は、デバイスのステータス（実行モード、低消費電力モード、またはリセット中）に関係なく、RTC が停止することはありません。

RTC は V_{BAT} モードで機能します。

46.2 RTC の主な機能

RTC は、次の機能をサポートしています（図 617 : RTC ブロック図を参照）。

- サブセカンド、秒、分、時（12 または 24 時間形式）、曜日、日、月、年に BCD（2 進化 10 進数）形式で対応するカレンダー。
- 32 ビットのフリーランニングカウンタを使用するバイナリモード。
- 月の 28 日、29 日（うるう年）、30 日、31 日の自動補正
- 2 つのプログラム可能なアラーム
- 1~32767 の RTC クロックパルスの動作中の補正。マスタクロックとの同期に使用可能
- リファレンスクロック検出：より正確な秒のクロックソース（50 または 60 Hz）の使用で、カレンダーの精度を向上。
- 0.95 ppm の分解能を持つデジタル較正回路で、クォーツクリスタルの精度を補償
- カレンダーの内容の保存に使用できるタイムスタンプ機能。この機能は、タイムスタンプピンでのイベント、またはタンパイイベント、あるいは V_{BAT} モードへの切り替えでトリガ可能
- 分解能と周期がプログラム可能な周期的なイベント用 17 ビット自動再ロードウェイクアップタイマ（WUT）
- TrustZone サポート：
 - RTC の完全なセキュリティ
 - アラーム A、アラーム B、ウェイクアップタイマおよびタイムスタンプの個別のセキュア/非セキュア設定
- アラーム A、アラーム B、ウェイクアップタイマおよびタイムスタンプの個別の特権保護

RTC の電源は、 V_{DD} 供給（ある場合）または V_{BAT} ピンから取得するスイッチ経由で供給されます。

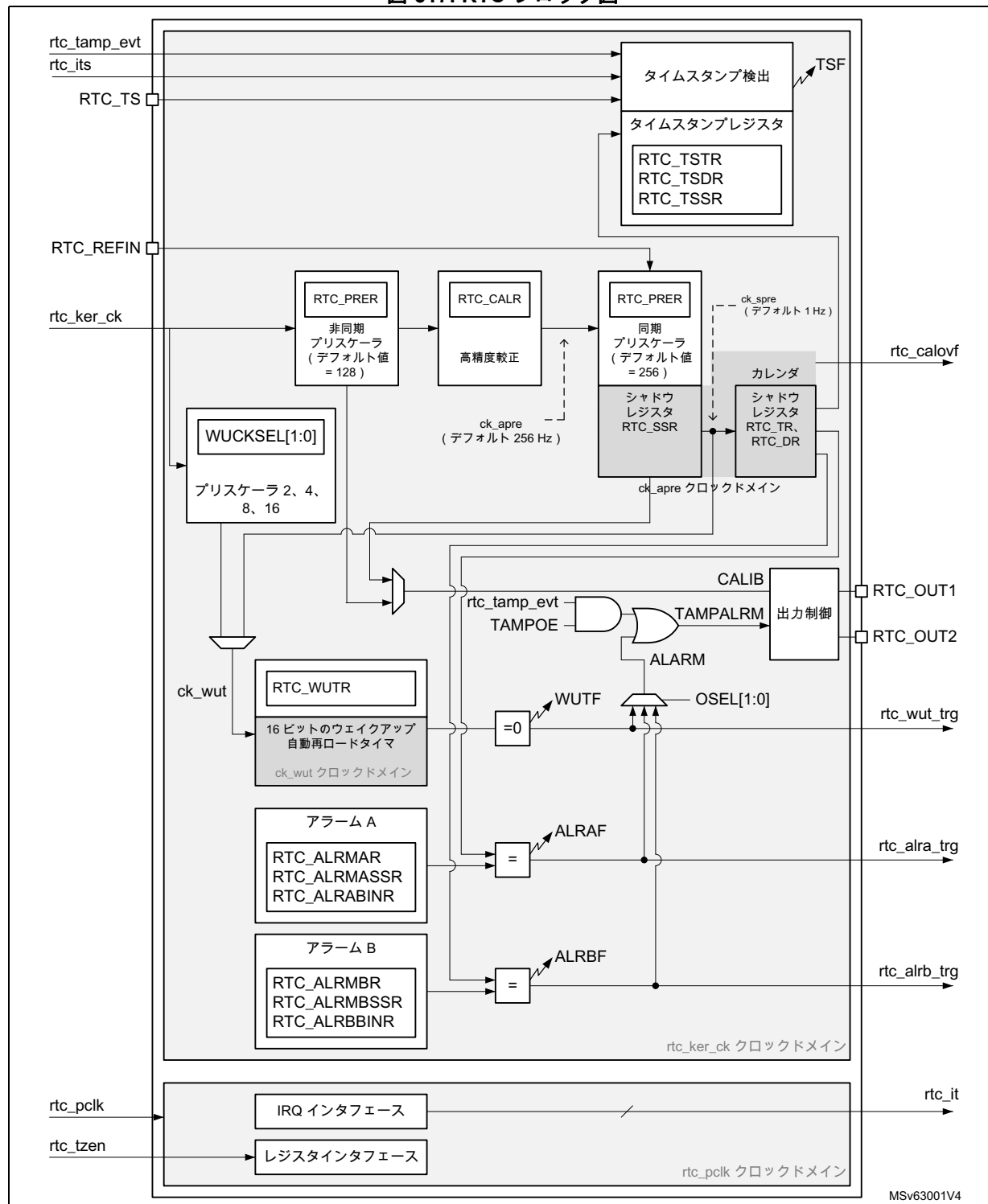
RTC は LSE によってクロック供給されている場合、 V_{BAT} モードおよびすべての低消費電力モードで機能します。

すべての RTC イベント（アラーム、ウェイクアップタイマ、タイムスタンプ）で割込みを生成して、デバイスを低消費電力モードからウェイクアップさせることができます。

46.3 RTC の機能説明

46.3.1 RTC ブロック図

図 617. RTC ブロック図



MSv63001V4

46.3.2 RTC ピンおよび内部信号

表 476. RTC の入出力ピン

ピン名	信号タイプ	説明
RTC_TS	入力	RTC タイムスタンプ入力
RTC_REFIN	入力	RTC 50 または 60 Hz リファレンスクロック入力
RTC_OUT1	出力	RTC 出力 1
RTC_OUT2	出力	RTC 出力 2

RTC_OUT1 および RTC_OUT2 は、次の 2 つの出力のうち 1 つを選択します。

- CALIB : 512 Hz または 1 Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。この出力は、RTC_CR レジスタの COE ビットをセットして有効にします。
- TAMPALRM : この出力は、rtc_tamp_evt 信号と ALARM 信号の論理和 (OR) です。

ALARM を有効化するには、アラーム A、アラーム B、またはウェイクアップ出力を選択する RTC_CR レジスタの OSEL[1:0] ビットを設定します。rtc_tamp_evt を有効化するには、タンパイイベント出力を選択する RTC_CR レジスタの TAMPOE ビットをセットします。

表 477. RTC 内部入力／出力信号

内部信号名	信号タイプ	説明
rtc_ker_ck	入力	RTC カーネルクロック (本書では RTCCLK とも言います)
rtc_pclk	入力	RTC APB クロック
rtc_its	入力	RTC 内部タイムスタンプイベント
rtc_tamp_evt	入力	TAMP ペリフェラルで検出されたタンパイイベント (内部または外部)
rtc_tzen	入力	RTC TrustZone 有効
rtc_it	出力	RTC 割込み (詳細については、 セクション 46.5 : RTC 割込み を参照してください)
rtc_alra_trg	出力	RTC アラーム A イベント検出トリガ
rtc_alrb_trg	出力	RTC アラーム B イベント検出トリガ
rtc_wut_trg	出力	RTC ウェイクアップタイマイイベント検出トリガ
rtc_calovf	出力	RTC カレンダーオーバーフロー : この信号は、RTC カレンダーが 2099 年 12 月 31 日 23:59:59 に最大値に達すると生成されます。このとき、カレンダーは停止し、オーバーフローしません。

通常、RTC カーネルクロックは、RCC で他のクロックソースを選択できますが、32.768 kHz での LSE です (詳細は RCC を参照してください)。選択したクロックが LSE ではない場合、一部の低消費電力モードや V_{BAT} で使用できない機能があります。詳細については、[セクション 46.4 : RTC 低消費電力モード](#)を参照してください。

表 478. RTC 相互接続

信号名	転送元／転送先
rtc_its	電源制御 (PWR) から : V_{BAT} 検出出力の主電源の損失／切り替え
rtc_tamp_evt	TAMP ペリフェラルから : tamp_evt
rtc_tzen	Flash オプションバイトから : TZEN
rtc_calovf	TAMP ペリフェラルへ : tamp_itamp5

TZEN オプションビットは、デバイスで TrustZone をアクティブにするために使用します。

TZEN = 1 : TrustZone は有効です。

TZEN = 0 : TrustZone は無効です。

TrustZone が無効である場合、RTC レジスタへの APB アクセスは非セキュアとなります。

トリガ出力は、その他のペリフェラルのトリガとして使用できます。

46.3.3 RTC および TAMP によって制御される GPIO

バッテリーバックアップドメイン (V_{BAT}) に含まれる GPIO は、どの GPIO 設定であっても、これらの I/O での機能を備えたペリフェラルで直接制御されます。

RTC_OUT1、RTC_TS、TAMP_IN1、TAMP_OUT2 および TAMP_OUT3 は、同じピン (PC13) に配置されます。PC13 に配置された RTC および TAMP の機能は、すべての低消費電力モードと V_{BAT} モードで使用できます。

出力形式は 表 479 に示す優先順位に従います。

表 479. RTC ピン PC13 の設定⁽¹⁾

PC13 ピンの機能		OSEL[1:0] (ALARM 出カインネーブル)	TAMPOE (TAMPER 出カインネーブル)	COE (CALIB 出カインネーブル)	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	ATOSHARE=0 で TAMP2E=TAMP2AM=1、または ATOSHARE=1 および ATOSELx=1 で TAMPxE=TAMPxAM=1	ATOSHARE=0 および OUT3_RMP=00 で TAMP3E=TAMP3AM=1、または ATOSHARE=1、 ATOSELx=2 および OUT3_RMP=00 で TAMPxE=TAMPxAM=1	TAMP1E (TAMP_IN1 入カインネーブル)	TSE (RTC_TS 入カインネーブル)
TAMPALRM 出力 プッシュプル		01、10、 または 11	0	無視	無視	0	0	無視	無視	無視	無視
		00	1								
		01、10、 または 11	1								
TAMPALRM 出力オープン ドレイン ⁽²⁾	プルアップ・ プルダウン なし	01、10、 または 11	0	無視	無視	1	0	無視	無視	無視	無視
		00	1								
		01、10、 または 11	1								
	内部プル アップ	01、10、 または 11	0	無視	無視	1	1	無視	無視	無視	無視
		00	1								
		01、10、 または 11	1								
CALIB 出力 PP		00	0	1	0	無視	無視	無視	無視	無視	無視
TAMP_OUT2 出力 PP		00	0	0	0	無視	無視	1	0	無視	無視
TAMP_OUT3 出力 PP		00	0	0	0	無視	無視	0	1	無視	無視
TAMP_IN1 入力フローティング		00	0	0	無視	無視	無視	0	0	1	0
		00	0	1	1						
		無視	無視	0							

表 479. RTC ピン PC13 の設定⁽¹⁾ (続き)

PC13 ピンの機能	OSEL[1:0] (ALARM 出カインネーブル)	TAMPOE (TAMPER 出カインネーブル)	COE (CALIB 出カインネーブル)	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	ATOSHARE=0 で TAMP2E=TAMP2AM=1、または ATOSHARE=1 および ATOSELx=1 で TAMPxE=TAMPxAM=1	ATOSHARE=0 および OUT3_RMP=00 で TAMP3E=TAMP3AM=1、または ATOSHARE=1、 ATOSELx=2 および OUT3_RMP=00 で TAMPxE=TAMPxAM=1	TAMP1E (TAMP_IN1 入カインネーブル)	TSE (RTC_TS 入カインネーブル)
RTC_TS および TAMP_IN1 入カフローティング	00	0	0	無視	無視	無視	0	0	1	1
	00	0	1	1			0	0		
	無視	無視	0				0	0		
RTC_TS 入カフローティング	00	0	0	無視	無視	無視	0	0	0	1
	00	0	1	1			0	0		
	無視	無視	0				0	0		
ウェイクアップピン	00	0	0	無視	無視	無視	0	0	0	0
	00	0	1	1			0	0		
	無視	無視	0				0	0		
標準 GPIO	00	0	0	無視	無視	無視	0	0	0	0
	00	0	1	1			0	0		
	無視	無視	0				0	0		

1. OD : オープンドレイン、PP : プッシュプル

2. この設定では、GPIO を入力で設定する必要があります。

RTC_OUT2、TAMP_IN2、および TAMP_OUT3 は同一ピン (PI8) 上に配置されます。PI8 の設定は、PI8 GPIO の設定にかかわらず、RTC によって制御されます。PI8に配置された RTC または TAMP の機能は、すべての低消費電力モードと VBAT モードで使用できます。

出力形式は 表 480 に示す優先順位に従います。

表 480. PI8 設定

PI1 ピンの機能		OSEL[1:0] (ALARM 出カインネーブル)	TAMPOE (TAMPER 出カインネーブル)	COE (CALIB 出カインネーブル)	OUT2EN と OUT2_RMP	TAMPALRM_TYPE	TAMPALRM_PU	ATOSHARE=0 および OUT3_RMP=01 で TAMP3E=TAMP3AM=1、または ATOSHARE=1、 ATOSELx=2 および OUT3_RMP=01 で TAMPxE=TAMPxAM=1	TAMP2E と IN2_RMP
TAMPALRM 出力 プッシュプル		01、10、 または 11	0	無視	1	0	0	無視	無視
		00	1						
		01、10、 または 11	1						
TAMPALRM 出力オープン ドレイン	プルアップ・ プルダウンなし	01、10、 または 11	0	無視	1	1	0	無視	無視
		00	1						
		01、10、 または 11	1						
	内部プルアップ	01、10、 または 11	0	無視	1	1	1	無視	無視
		00	1						
		01、10、 または 11	1						
CALIB 出力 PP		00	0	1	1	無視	無視	無視	無視
TAMP_OUT3 出力 PP		00	0	0	0	無視	無視	1	0
TAMP_IN2 入力フローティング		00	0	0	0	無視	無視	0	1
ウェイクアップピン		00	0	0	0	無視	無視	0	0
標準 GPIO		00	0	0	0	無視	無視	0	0

さらに、OUT2EN および OUT2_RMP ビットにより、RTC_OUT2 を PB2 または PI8 ピンに出力することも可能です。表 481 の表に示すとおり、OSEL、COE および OUT2EN の設定に応じて、異なる機能が RTC_OUT1 や RTC_OUT2 に配置されます。

表 481. RTC_OUT の配置

OSEL[1:0] ビット (ALARM 出力 イネーブル)	COE ビット (CALIB 出力 イネーブル)	OUT2EN ビット	PC13 での RTC_OUT1	PB2 または PI8での RTC_OUT2
00	0	0	-	-
00	1		CALIB	-
01、10、または 11	無視		TAMPALRM	-
00	0	1	-	-
00	1		-	CALIB
01、10、または 11	0		-	TAMPALRM
01、10、または 11	1		TAMPALRM	CALIB

46.3.4 RTC セキュア保護モード

デフォルトでは、バックアップドメインのパワーオンリセット後、セキュアモードでのみ書き込み可能な RTC セキュア設定レジスタ (RTC_SECCFGR) を除いて、すべての RTC レジスタがセキュアモードと非セキュアモードの両方で読出しまたは書き込みできます。RTC 保護設定は、システムリセットの影響を受けません。

RTC_SECCFGR レジスタの SEC ビットがセットされている場合：

- RTC レジスタへの書き込みはセキュアモードでのみ可能です。
- RTC_SECCFGR、RTC_PRIVCFGR、RTC_MISR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER および RTC_CALR の読出しは、セキュアモードと非セキュアモードで常に可能です。その他のすべての RTC レジスタは、セキュアモードでのみ読出しできます。

SEC ビットがクリアされている場合でも、専用の INITSEC、CALSEC、TSSEC、WUTSEC、ALRASEC または ALRBSEC の制御ビットをセットすることで、いくつかのレジスタを保護できます。これらすべてのビットもセットされている場合、すべての RTC レジスタをセキュアモードおよび非セキュアモードで読み書きできます。

- INITSEC がセットされている場合：
 - RTC_TR、RTC_DR、RTC_PRER のレジスタ、INIT、BIN および BCDU (RTC_ICSR)、FMT 制御ビット (RTC_CR)、INITPRIV (RTC_PRIVCFGR) は、セキュアモードでのみ書き込みできます。
 - これらのレジスタと制御ビットは、セキュアモードおよび非セキュアモードで読出しできます。
- CALSEC がセットされている場合：
 - RTC_SHIFTR と RTC_CALR のレジスタ、および ADD1H、SUB1H、REFCKON の制御ビット (RTC_CR)、CALPRIV (RTC_PRIVCFGR) は、セキュアモードでのみ書き込みできます。
 - これらのレジスタと制御ビットは、セキュアモードおよび非セキュアモードで読出しできます。
- ALRASEC がセットされている場合：
 - RTC_ALRMAR、RTC_ALRMASR および RTC_ALRABINR レジスタ、ALRAE、ALRAFCLR、ALRAIE および SSRUIE (RTC_CR)、CALRAF および CSSRUF (RTC_SCR)、ALRAF および SSRUF (RTC_SR)、ALRAMF および SSRUMF (RTC_SMISR) はセキュアモードでのみ読み書きできます。
 - ALRAPRIV (RTC_PRIVCFGR) はセキュアモードでのみ書き込みできます。

- ALRBSEC がセットされている場合 :
 - RTC_ALRMBR、RTC_ALRMBSSR、RTC_ALRBBINR のレジスタ、および ALRBE ALRBFCLR、ALRBIE (RTC_CR)、CALRBF (RTC_SCR)、ALRBF (RTC_SR)、ALRBMF (RTC_SMISR) は、セキュアモードでのみ読み書きできます。
 - ALRBPRIV (RTC_PRIVCFGR) は、セキュアモードでのみ書込みできます。
- WUTSEC がセットされている場合 :
 - RTC_WUTR レジスタ、および WUTE、WUTIE、WUCKSEL の制御ビット (RTC_CR)、CWUTF (RTC_SCR)、WUTF (RTC_SR)、WUTMF (RTC_SMISR) は、セキュアモードでのみ読み書きできます。
 - WUTPRIV (RTC_PRIVCFGR) は、セキュアモードでのみ書込みできます。
- TSSEC がセットされている場合 :
 - RTC_TSTR、RTC_TSDR および RTC_TSSSR のレジスタ、および TAMPTS、ITSE、TSE、TSIE、TSEDGE の制御ビット (RTC_CR)、CITSF、CTSOVF、CTSF のビット (RTC_SCR)、TSF、TSOVF、ITSF (RTC_SR)、TSMF、TSOVMF、ITSMF (RTC_SMISR) は、セキュアモードでのみ読み書きできます。
 - TSPRIV (RTC_PRIVCFGR) は、セキュアモードでのみ書込みできます。

セキュア保護されたレジスタへの非セキュアアクセスは拒否されます。

- 生成されるバスエラーはありません。
- レジスタがグローバル保護されている場合 : TZIC (TrustZone 不正アクセスコントローラ)でのフラグ/割込みを通じて通知が生成されます。レジスタの数ビットだけが保護されている場合 (RTC_CR など混合機能を持ったレジスタ)、通知は生成されません。
- 書込み保護されている場合、そのビットは書き込まれません。
- 読出し保護されている場合、そのビットは 0 として読み出されます。

1 つ以上の機能がセキュアに設定されると、RTC リセットおよびクロック制御も RCC でセキュアになります。

46.3.5 RTC 特権保護モード

デフォルトでは、バックアップドメインのパワーオンリセット後、特権モードのみで書込み可能な RTC 特権モード制御レジスタ (RTC_PRIVCFGR) を除いて、すべての RTC レジスタが特権と非特権モードの両方で読出しまたは書込みできます。RTC 保護設定は、システムリセットの影響を受けません。

RTC_PRIVCFGR レジスタの PRIV ビットがセットされている場合 :

- RTC レジスタへの書込みは特権モードでのみ可能です。
- RTC_SECCFGR、RTC_PRIVCFGR、RTC_TR、RTC_DR、RTC_SSR、RTC_PRER および RTC_CALR の読出しは、特権と非特権モードで常に可能です。その他のすべての RTC レジスタは、特権モードでのみ読出しできます。

PRIV ビットがクリアされている場合、専用の INITPRIV、CALPRIV、TSPRIV、WUTPRIV、ALRAPRV または ALRBPRIV の制御ビットをセットすることで、いくつかのレジスタを保護できます。これら

すべてのビットもクリアされている場合、すべての RTC レジスタを特権および非特権モードで読み書きできます。

- INITPRIV がセットされている場合 :
 - RTC_TR、RTC_DR、RTC_PRER のレジスタ、INIT、BIN および BCDU (RTC_ICSR)、FMT 制御ビット (RTC_CR)、INITSEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。
 - これらのレジスタと制御ビットは、特権および非特権モードで読出しできます。
- CALPRIV がセットされている場合 :
 - RTC_SHIFTR と RTC_CALR のレジスタ、および ADD1H、REFCKON の制御ビット (RTC_CR)、CALDSEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。
 - これらのレジスタと制御ビットは、特権および非特権モードで読出しできます。
- ALRAPRIV がセットされている場合 :
 - RTC_ALRMAR、RTC_ALRMASR および RTC_ALRABINR レジスタ、ALRAE ALRAFCLR、ALRAIE および SSRUIE (RTC_CR)、および CALRAF および CSSRUF (RTC_SCR)、ALRAF および SSRUF (RTC_SR)、ALRAMF および SSRUMF (RTC_SMISR)、RTC_SMISR は特権モードでのみ読み書きできます。
 - ALRASEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。
- ALRBPRIV がセットされている場合 :
 - RTC_ALRMBR、RTC_ALRMBSSR、RTC_ALRBBINR のレジスタ、ALRBE ALRBFCLR、ALRBIE (RTC_CR)、CALRBF (RTC_SCR)、ALRBF (RTC_SR)、ALRBMF (RTC_MISR および RTC_SMISR) は、特権モードでのみ読み書きできます。
 - ALRBSEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。
- WUTPRIV がセットされている場合 :
 - RTC_WUTR レジスタ、および WUTE、WUTIE、WUCKSEL の制御ビット (RTC_CR)、CWUTF (RTC_SCR)、WUTF (RTC_SR)、WUTMF (RTC_MISR および RTC_SMISR) は、特権モードでのみ読み書きできます。
 - WUTSEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。
- TSPRIV がセットされている場合 :
 - RTC_TSTR、RTC_TSDR および RTC_TSSSR のレジスタ、および TAMPTS、ITSE、TSE、TSIE、TSEDGE の制御ビット (RTC_CR)、CITSF、CTSOVF、CTSF のビット (RTC_SCR)、TSF、TSOVF、ITSF (RTC_SR)、TSMF、TSOVMF、ITSMF (RTC_MISR)、および RTC_SMISR は、特権モードでのみ読み書きできます。
 - TSSEC (RTC_SECCFGR) は、特権モードでのみ書き込みできます。

特権保護されたレジスタへの非特権アクセスは拒否されます。

- 生成されるバスエラーはありません。
- 書き込み保護されている場合、そのビットは書き込まれません。
- 読出し保護されている場合、そのビットは 0 として読み出されます。

46.3.6 クロックとプリスケアラ

RTC クロック (RTCCLK) ソース設定の詳細については、「リセットおよびクロック制御 (RCC)」を参照してください。

BCD モード (BIN=00)

プログラム可能なプリスケアラステージで、カレンダーの更新に使用する 1 Hz のクロックを生成します。消費電力を最少に抑えるため、プリスケアラは以下に示す 2 つのプログラム可能なプリスケアラに分割されます (図 617 : RTC ブロック図を参照)。

- RTC_PRER レジスタの PREDIV_A ビットで設定される 7 ビットの非同期プリスケアラ
- RTC_PRER レジスタの PREDIV_S ビットで設定される 15 ビットの同期プリスケアラ

注： 両方のプリスケアラを使用する場合は、非同期プリスケアラを高い値に設定して消費を最低限に抑えることをお勧めします。

LSE 周波数 32.768 kHz で 1 Hz (ck_spre) の内部クロック周波数を得るため、非同期プリスケアラ分周比は 128、同期プリスケアラの分周比は 256 に設定されます。

最低分周比は 1、最大分周比は 2^{22} です。

これは、約 4 MHz の最大入力周波数に相当します。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1}$$

ck_apre クロックは、サブセカンドダウンカウンタであるバイナリ RTC_SSR にクロックを供給するために使用されます。値がゼロになると、RTC_SSR は、PREDIV_S の内容で再ロードされます。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)}$$

ck_spre クロックは、カレンダーの更新に、または 16 ビットウェイクアップ自動再ロードタイマのタイムベースとして使用できます。短いタイムアウト期間を得るため、16 ビットウェイクアップ自動再ロードタイマを、プログラム可能な 4 ビット非同期プリスケアラで分周した RTCCLK で動作させることもできます (詳細は [セクション 46.3.10 : 周期的自動ウェイクアップ](#) を参照)。

バイナリモード (BIN=01)

SSR バイナリダウンカウンタは 32 ビット長に拡張され、フリーランニングです。日付と時刻のカレンダーである BCD レジスタは機能しません。

このダウンカウンタは、ck_apre によってクロック供給され、RTC_PRER レジスタの PREDIV_A ビットを通して設定される 7 ビット非同期プリスケアラの出力です。

PREDIV_S の値は無視されます。

混合モード (BIN=10 または 11)

SSR バイナリダウンカウンタは 32 ビット長に拡張され、フリーランニングです。日付と時刻のカレンダーである BCD レジスタも利用できます。

このダウンカウンタは、ck_apre によってクロック供給され、RTC_PRER レジスタの PREDIV_A ビットを通して設定される 7 ビット非同期プリスケアラの出力です。ビット BCDU[2:0] は、SSR 最下位ビットを使用して、カレンダーを 1 ずつインクリメントするタイミングを定義するのに使用します。

46.3.7 リアルタイムクロックとカレンダー

RTC カレンダーの時刻および日付レジスタには、PCLK (APB クロック) と同期するシャドウレジスタからアクセスします。同期するための待ち時間を避けるため、これらのレジスタに直接アクセスすることもできます。

- サブセカンド用 RTC_SSR
- 時刻用 RTC_TR
- 日付用 RTC_DR

RTCCLK サイクルごとに現在のカレンダー値がシャドウレジスタにコピーされ、RTC_ICSR レジスタの RSF ビットがセットされます ([セクション 46.6.12 : RTC シフト制御レジスタ \(RTC_SHIFTR\)](#) を参照)。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 4 サイクル以内にシャドウレジスタが更新されます。

アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC_SSR、RTC_TR または RTC_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 (f_{APB}) は、RTC クロック (f_{RTCCLK}) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

46.3.8 カレンダー超低消費電力モード

RTC_CALR レジスタの LPCAL ビットをセットすることによって、RTC 電力消費を大幅に削減できます。この設定では、RTC 全体が RTCCLK と ck_apre の両方ではなく、ck_apre のみによってクロック供給されます。その結果、一部のフラグ遅延が長くなり、較正時間範囲も長くなります ([セクション : RTC 超低消費電力モード](#) を参照)。

非同期プリスケアラ分周比 ($PREDIV_A+1$) が 2 の累乗の数ではない場合、LPCAL ビットは無視されます (0 とみなされます)。

LPCAL=0 から LPCAL=1 へ、または LPCAL=1 から LPCAL=0 への切り替えは即時ではなく、完了に ck_apre 数周期かかります。

46.3.9 プログラム可能なアラーム

RTC ユニットにはプログラム可能なアラーム (アラーム A とアラーム B) が搭載されています。以下にアラーム A の説明を示していますが、アラーム B にも転用できます。

プログラム可能なアラーム機能は、RTC_CR レジスタの ALRAE ビットを通じて有効にします。

ALRAF は、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC_ALRMASR および RTC_ALRMAR にプログラムされている値と一致する場合は 1 にセットされます。各カレンダー項目は、RTC_ALRMAR レジスタの MSKx ビットおよび RTC_ALRMASR レジスタの MASKSSx ビットで個別に選択できます。

バイナリモードを使用する場合、アラームバイナリレジスタ RTC_ALRABINR でサブセカンドフィールドをプログラムできます。

アラームの割込みは、RTC_CR レジスタの ALRAIE ビットを通じて有効にします。

アラームを使用して別のペリフェラル用にトリガーイベントを生成する場合、RTC_CR レジスタで ALRAFCLR ビットに 1 を設定すると、ハードウェアにより自動的に ALRAF が作成されます。この設定では、ALRAF フラグをクリアするためにソフトウェアによる介入は必要はありません。

注意： 秒の項目が選択されている場合 (RTC_ALRMAR で MSK1 ビットがリセットされている場合)、正しい動作を保証するため、RTC_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上である必要があります。

アラーム A およびアラーム B (RTC_CR レジスタの OSEL[1:0] ビットで有効になっている場合) は、TAMPALRM 出力に送ることができます。TAMPALRM 出力の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

46.3.10 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力 ck_wut には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)
RTCCLK が LSE (32.768 kHz) である場合、最小分解能 61 μ s で、ウェイクアップ割込み周期を 122 μ s から 32 秒の範囲で設定できます。
- BCD モードの ck_spre (通常は 1 Hz の内部クロック)、またはバイナリモードまたは混合モード (BCD-binary) で BCDU での定義に従ってカレンダーを更新するのに使用されるクロック。
ck_spre の周波数が 1 Hz の場合、1 秒の分解能により、1 秒から約 36 時間までのウェイクアップタイムが得られます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。
 - WUCKSEL[2:1] = 10 の場合は 1 秒から 18 時間です。
 - WUCKSEL[2:1] = 11 の場合は約 18 時間から 36 時間です。この最後のケースでは、 2^{16} が 16 ビットカウンタの現在値に追加されます。初期化シーケンスが完了すると (1939 ページのウェイクアップタイマのプログラミングを参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効である場合、カウントダウンは低消費電力モードでアクティブのままとなります。さらに、カウンタが 0 に到達すると、RTC_SR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC_WUTR レジスタ値) で自動的に再ロードされます。

RTC_WUTR レジスタの WUTOCLR に応じて、WUTF フラグはソフトウェアでクリアする必要があります (WUTOCLR = 0x0000)、そうしない場合は自動再ロードダウンカウンタが WUTOCLR の値に達したときに (0x0000 < WUTOCLR \leq WUT)、WUTF がハードウェアによって自動でクリアされます。

ウェイクアップフラグは、その他のペリフェラルで使用できる内部信号 rtc_wut で出力されます (セクション 46.3.1 : RTC ブロック図のセクションを参照)。

RTC_CR レジスタの WUTIE ビットをセットして周期的ウェイクアップ割込みを有効にすると、デバイスは低消費電力モードを終了できます。

周期的ウェイクアップフラグは、RTC_CR レジスタの OSEL[1:0] ビットを通じて有効になっている場合に、TAMPALRM 出力に送ることができます。TAMPALRM 出力の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

低消費電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイマには影響しません。

46.3.11 RTC の初期化と設定

RTC バイナリ、BCD、または混合モード

デフォルトでは RTC は BCD モード (RTC_ICSR レジスタで BIN = 00) : RTC_SSR レジスタにはサブセカンドフィールド SS[15:0] があり、ck_apre によってクロック供給され、1 Hz のクロックを生成して BCD 形式でカレンダーのレジスタ (RTC_TR および RTC_DR) を更新することができます。

RTC がバイナリモードに設定されている場合 (RTC_ICSR レジスタで BIN = 01) : RTC_SSR レジスタにはバイナリカウンタ SS[31:0] があり、ck_apre によってクロック供給されます。BCD 形式のカレンダーのレジスタ (RTC_TR および RTC_DR) は使用されません。

RTC がバイナリモードに設定されている場合 (RTC_ICSR レジスタで BIN = 10 または 11) : RTC_SSR レジスタにはバイナリカウンタ SS[31:0] があり、ck_apre によってクロック供給されます。カレンダーは、SSR[BCDU+7:0] が 0 に達するたびに更新されます (1 秒のインクリメント)。

RTC レジスタ書込み保護

システムリセット後、電源制御ペリフェラルの DBP ビットによって、RTC レジスタは不要な書込みアクセスから保護されます (PWR 電源制御のセクションを参照)。RTC レジスタ書込みアクセスを可能にするには、DBP ビットをセットする必要があります。

バックアップドメインリセット後、次の一部の RTC レジスタは書込み保護されます。RTC_TR、RTC_DR、RTC_PRER、RTC_CALR、RTC_SHIFTR、RTC_ICSR の INIT、BIN、BCDU のビット、RTC_CR の FMT、SUB1H、ADD1H、REFCKON のビットです。

保護された RTC レジスタの書込み保護を解除するには、次のステップが必要です。

1. RTC_WPR レジスタに 0xCA を書き込みます。
2. RTC_WPR レジスタに 0x53 を書き込みます。

誤ったキーを書き込むと、書込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

INITPRIV によって保護されたレジスタは、INIT KEY で書込み保護されます。

CALDPRIV によって保護されたレジスタは、CAL KEY で書込み保護されます。

PRIV または INITPRIV が RTC_PRIVCFGR でセットされている場合、SEC または INITSEC が RTC_SECCFGR でセットされている場合 : RTC_WPR レジスタへの書込みアクセスが特権モードおよびセキュリティモード (PRIV、INITPRIV、SEC、INITSEC の設定で定義) で実行される場合のみ、INIT KEY はロック解除またはロックされます。

PRIV または CALPRIV が RTC_PRIVCFGR でセットされている場合、または SEC や CALSEC が RTC_SECCFGR でセットされている場合 : RTC_WPR レジスタへの書込みアクセスが特権モードおよびセキュリティモード (PRIV、CALPRIV、SEC、CALSEC の設定で定義) で実行される場合のみ、CAL KEY はロック解除またはロックされます。

カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC_ICSR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC_ICSR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。
 - LPCAL=0 の場合：INITF は、INIT ビットがセットされた後、およそ 2 RTCCLK サイクルでセットされます。
 - LPCAL=1 の場合：INITF は、INIT ビットがセットされた後、最大 2 ck_apre サイクルでセットされます。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、RTC_PRER レジスタで両方のプリスケアラ分周比をプログラムし、RTC_ICSR レジスタの BIN および BCDU を設定します。
4. シャドウレジスタ (RTC_TR および RTC_DR) に時刻と日付の初期値をロードし、RTC_CR レジスタの FMT ビットを介して時間形式 (12 時間または 24 時間) を設定します。
5. INIT ビットをクリアして初期化モードを終了します。その後、カレンダーカウンタの実際の値が自動的にロードされます。
 - LPCAL=0 の場合：4 RTCCLK クロックサイクル後にカウントが再開します。
 - LPCAL=1 の場合：最大 2 RTCCLK + 1 ck_apre 後にカウントが再開します。

初期化シーケンスが完了すると、カレンダーがカウントを開始します。RTC_SSR の内容は次のように初期化します。

- PREDIV_S は BCD モード (BIN=00)
- バイナリで 0xFFFF FFFF または混合 (BCD-binary) モード (BIN=01、10 または 11)

BCD モードでは、RTC_SSR には、同期プリスケアラのカウントの値が入っています。これにより、RTC によって維持されている正確な時刻を $1/(PREDIV_S + 1)$ 秒の分解能まで計算することができます。その結果、同期プリスケアラ値 (PREDIV_S[14:0]) を増加させることにより分解能を改善できます。許可されている最大分解能 (32768 Hz クロックで 30.52 μ s) は、PREDIV_S を 0x7FFF にセットすることにより得られます。

ただし、PREDIV_S を増加させるということは、同期プリスケアラの出力を 1 Hz に維持するため PREDIV_A を減らす必要があることを意味します。このように、非同期プリスケアラの出力周波数が増加すると、RTC の動的消費電力が増加する場合があります。RTC の動的消費電力は、2 の累乗の数になる PREDIV_A+1 に最適化されます。

注： システムリセット後、アプリケーションは RTC_ICSR レジスタの INITF フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 であれば、カレンダーの年の項目がバックアップドメインリセットのデフォルト値 (0x00) にセットされているため、初期化されていません。

注： 初期化後にカレンダーを読み出すには、まずソフトウェアで RTC_ICSR レジスタの RSF フラグがセットされていることを確認する必要があります。

サマータイム

サマータイム管理は、RTC_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

アラームのプログラミング

プログラム可能なアラームをプログラムまたは更新するには、同様な手順を踏む必要があります。以下にアラーム A の手順を示していますが、アラーム B にも転用できます。

1. RTC_CR の ALRAE をクリアしてアラーム A を無効にします。
2. アラーム A レジスタ (RTC_ALRMASR/RTC_ALRMAR または RTC_ALRABINR) をプログラムします。
3. RTC_CR レジスタで ALRAE をセットしてアラーム A を再び有効にします。

注： RTC_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 サイクル後に有効になります。

ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値 (RTC_WUTR の WUT[15:0]) の設定または変更には、次の手順が必要です。

1. RTC_CR の WUTE をクリアしてウェイクアップタイマを無効にします。
2. RTC_ICSR の WUTWF がセットされ、ウェイクアップ自動再ロードカウンタおよび WUCKSEL[2:0] ビットへのアクセスが許可されていることが確認されるまで WUTWF をポーリングします。このステップは、カレンダー初期化モードではスキップする必要があります。
 - WUCKSEL[2] = 0 : WUTWF は、WUTE ビットがクリアされた後、およそ $1 \text{ ck_wut} + 1 \text{ RTCCLK}$ サイクルでセットされます。
 - WUCKSEL[2] = 1 : WUTWF は、WUTE ビットがクリアされた後、最大 $1 \text{ ck_apre} + 1 \text{ RTCCLK}$ サイクルでセットされます。
3. ウェイクアップ自動再ロード値 WUT[15:0]、WUTOCLR[15:0] およびウェイクアップクロック選択 (RTC_CR の WUCKSEL[2:0] ビット) をプログラムします。RTC_CR で WUTE をセットしてタイマを再び有効にします。ウェイクアップタイマがカウントダウンを再開します。
 - WUCKSEL[2] = 0 : WUTWF は、WUTE ビットがセットされた後、およそ $1 \text{ ck_wut} + 1 \text{ RTCCLK}$ サイクルでクリアされます。
 - WUCKSEL[2] = 1 : WUTWF は、WUTE ビットがセットされた後、最大 $1 \text{ ck_apre} + 1 \text{ RTCCLK}$ サイクルでクリアされます。

46.3.12 カレンダーの読出し

RTC_CR レジスタの BYPSHAD 制御ビットがクリアされている場合

RTC カレンダーレジスタ (RTC_SSR、RTC_TR、および RTC_DR) を正しく読み出すには、APB クロック周波数 (f_{PCLK}) が RTC クロック周波数 (f_{RTCCLK}) の 7 倍以上でなければなりません。これにより、同期メカニズムの安全な動作が保証されます。

APB クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR の 2 回目の読出しが 1 回目の読出しと同じ結果であれば、データが正しいことが保証されます。同じでない場合は、3 回目の読出しアクセスを行う必要があります。どの場合も、APB クロック周波数は必ず RTC クロック周波数以上でなければなりません。

RTC_ICSR レジスタの RSF ビットは、カレンダーレジスタが RTC_SSR、RTC_TR、および RTC_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、RTCCLK サイクルごとに行われます。3 つの値における一貫性を保証するため、RTC_SSR または RTC_TR のどちらかを読み出すと、高次カレンダーシャドウレジスタの値は RTC_DR が読み出されるまでロックされます。ソフトウェアが 1 RTCCLK サイクル未満の間隔でカレンダーの読出しアクセスを行う場合、最初のカレンダー読出し後

に RSF をソフトウェアでクリアする必要がある、その後ソフトウェアは、RSF ビットがセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再度読み出す必要があります。

低消費電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いまだ RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再度読み出す必要があります。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低消費電力モードに入る前には、その必要はありません。

システムリセット後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。実際、システムリセットがかかると、シャドウレジスタはデフォルト値にリセットされます。

初期化 (1938 ページの [カレンダーの初期化と設定](#) を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

同期 ([セクション 46.3.14 : RTC の同期](#) を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

RTC_CR レジスタ (バイパスシャドウレジスタ) の BYPSHAD 制御ビットがセットされている場合

カレンダーレジスタを読み出すと、カレンダーカウンタの値が直接与えられるため、RSF ビットがセットされるのを待つ必要はありません。シャドウレジスタは低消費電力モード (STOP または STANDBY) では更新されないため、低消費電力モード終了後に特にこのような読出しが有効です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

注 : BYPSHAD = 1 の間、カレンダーレジスタの読出し命令が完了するには 1 APB サイクルだけ余計に必要となります。

46.3.13 RTC のリセット

カレンダーシャドウレジスタ (RTC_SSR、RTC_TR、RTC_DR) および RTC ステータスレジスタ (RTC_ICSR) の一部のビットは、利用可能なすべてのシステムリセットリソースによってデフォルト値にリセットされます。

逆に、次のレジスタはバックアップドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響は受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC_CR)、プリスケールレジスタ (RTC_PRER)、RTC 較正レジスタ (RTC_CALR)、RTC シフトレジスタ (RTC_SHIFTR)、RTC タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、および RTC_TSDR)、ウェイクアップタイマレジスタ (RTC_WUTR)、アラーム A およびアラーム B レジスタ (RTC_ALRMASR/RTC_ALRMAR/RTC_ALRABINR および RTC_ALRMBSSR/RTC_ALRMBR/RTC_ALRBBINR)。

さらに、LSE クロックによって駆動されている際にリセットソースがバックアップドメインリセットと異なる場合、システムリセットがかかっても RTC は動作を続けます (システムリセットの影響を受けない RTC クロックソースの詳細については、RCC を参照してください)。バックアップドメインリセットが発生すると、RTC は停止し、すべての RTC レジスタがリセット値にセットされます。

46.3.14 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC_SSR または RTC_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC は、RTC シフト制御レジスタ (RTC_SHIFTR) を使って微調整できます。RTC_SHIFTR に書き込むことにより、1 ck_{apre} 周期の分解能でシフト (遅れ/進み) させることができます。

このシフト操作の本質は、同期プリスケアラのカウンタ SS[15:0] に SUBFS[14:0] 値を加算することであり、この操作はクロックを遅らせることになります。

同時に BCD または混合モードで ADD1S ビットがセットされた場合、1 秒追加すると同時に秒の小数部を差し引くことになるため、クロックを進めることになります。バイナリモードでは ADD1S は影響を及ぼしません。

RTC_SHIFTR レジスタへの書き込みによってシフト操作が始まるとすぐに、シフト操作が保留中であることを示す SHPF フラグがハードウェアによってセットされます。このビットは、シフト操作が完了するとすぐに、ハードウェアによってクリアされます。

注意： 混合モードでは (BIN=10 または 11)、SUBFS[14:BCDU+8] には 0 を書き込む必要があります。

注意： BCD モードでシフト操作を始める前に、ユーザは SS[15] = 0 であることを確認し、オーバーフローが発生しないようにする必要があります。混合モードでは、SS[BCDU+8] のビットが 0 であることを確認する必要があります。

注意： この同期機能はリファレンスクロック検出機能とは両立できません。具体的には、REFCKON = 1 のときにファームウェアから RTC_SHIFTR への書き込みはできません。

46.3.15 RTC リファレンスクロック検出

この機能は、BCD モード (BIN=00) でのみ利用できます。

RTC カレンダの更新は、リファレンスクロックである RTC_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC_REFIN リファレンスクロックには、32.768 kHz LSE クロックより高い精度が必要です。RTC_REFIN 検出を有効にした際 (RTC_CR の REFCKON ビットが 1 にセット)、カレンダは引き続き LSE クロックによって駆動され、RTC_REFIN はカレンダ更新周波数 (1 Hz) の誤差の補正に使用されます。

各 1 Hz クロックエッジは、一番近い RTC_REFIN クロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、その後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck_{apre}) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダ更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck_{apre} 7 周期に等しくなります。その後のカレンダ更新では、 ck_{apre} 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、 ck_{spre} クロックを出力する非同期プリスケアラは強制的に再ロードされます。プリスケアラは同時に再ロードされるので、リファレンスクロックおよび 1 Hz のクロックが整列するタイミングには影響しません。クロックが整列していない場合、後の 1 Hz クロックエッジは、リファレンスクロックと整列するように再ロードによって少しシフトされます。

リファレンスクロックが停止した (ck_apre 3 周期の枠内でリファレンスクロックエッジが発生しない) 場合、カレンダーは LSE クロックのみを基準にして更新が継続されます。その後 RTC は ck_spre エッジを中心として ck_apre 7 周期という広い検出時間枠でリファレンスクロックを待ちます。

RTC_REFIN検出を有効にした場合、PREDIV_A および PREDIV_S を以下に示すそれぞれのデフォルト値にセットする必要があります。

- PREDIV_A = 0x007F
- PREDIV_S = 0x00FF

注： RTC_REFIN クロック検出は、STANDBY モードでは利用できません。

46.3.16 RTC の高精度デジタル校正

RTC 周波数の精度は、-487.1 ppm から +488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に校正できます。周波数の修正は、一連の微調整 (個々の ck_cal パルスの追加や削除) によって行われます。

LPCAL=0 の場合：ck_cal = RTCCLK

LPCAL=1 の場合：ck_cal = ck_apre

このような調整は、短い期間で観測された場合でも RTC が十分に校正されるように、かなり広範囲に分散して行われます。

RTC 超低消費電力モード

RTC の消費電力は、RTC 校正レジスタ (RTC_CALR) の LPCAL ビットをセットすることで低減できます。この場合、校正メカニズムは RTCCLK ではなく ck_apre で適用されます。結果となる精度は同じですが、LPCAL=0 の場合は 2^{20} RTCCLK パルスではなく、およそ $2^{20} \times \text{PREDIV_A} \times \text{RTCCLK}$ パルスの校正サイクルの間に校正が実行されます。

高精度校正メカニズム

高精度校正レジスタ (RTC_CALR) によって校正サイクル中にマスクされる ck_cal クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると校正サイクルの中でちょうど 1 パルスがマスクされます。
- CALM[1] ビットを 1 にセットすると、さらに 2 サイクルがマスクされます。
- CALM[2] ビットを 1 にセットすると、さらに 4 サイクルがマスクされます。
- CALM[8] ビットを 1 にセットするまで続けると、256 クロックがマスクされます。

注： CALM[8:0] (RTC_CALR) によって校正サイクル中にマスクされる ck_cal パルス数を指定します。CALM[0] ビットを 1 にセットすると、cal_cnt[19:0] = 0x80000 になった時点で、校正サイクル中でちょうど 1 パルスがマスクされます。CALM[1] = 1 では、さらに 2 サイクルがマスクされ (cal_cnt = 0x40000 および 0xC0000)、CALM[2] = 1 では、さらに 4 サイクルがマスクされ (cal_cnt = 0x20000/0x60000/0xA0000/ 0xE0000)、CALM[8] = 1 まで続けると、256 クロックがマスクされ (cal_cnt = 0xFF800)。

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、ck_cal 2^{11} サイクルごとに、ck_cal パルスが 1 パルス追加で挿入されます。すなわち校正サイクルごとに 512 クロックが追加されることとなります。

CALM を CALP と合わせて使用すると校正サイクルの間に ck_cal -511 から +512 ck_cal サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で校正範囲 -487.1 ~ +488.5 ppm に換算されます。

有効校正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (\text{CALP} \times 512 - \text{CALM}) / (2^{20} + \text{CALM} - \text{CALP} \times 512)]$$

注意： PREDIV_A は 3 以上でなければなりません。

PREDIV_A < 3 の場合の校正

非同期プリスケアラ値 (RTC_PRER レジスタの PREDIV_A ビット) が 3 未満の場合、CALP ビットを 1 にセットすることはできません。CALP がすでに 1 にセットされていて、PREDIV_A ビットが 3 未満の値にセットされた場合、CALP の設定値は無視され、CALP が 0 に設定された場合と同じように校正されます。

BCD モードでは PREDIV_A が 3 未満の状態では校正できませんが、各秒のカウントが 8 ck_cal クロックサイクル分早められるように同期プリスケアラ値 (PREDIV_S) を小さくする必要があります。これは、校正サイクルごとに 256 クロックサイクル追加することに相当します。結果として、CALM ビットのみを使用して校正周期の間に 255~256 クロックパルス (243.3~244.1 ppm の校正範囲に相当) を追加することができます。

公称 RTCCLK 周波数が 32768 Hz で、PREDIV_A が 1 (分周比 2) の場合、PREDIV_S を 16383 ではなく 16379 (4 少ない) にセットする必要があります。また、PREDIV_A が 0 の場合、PREDIV_S を 32767 ではなく 32759 (8 少ない) にセットする必要がありますので注意してください。

PREDIV_S をこのように減少させた場合、校正された入力クロックの有効周波数の式は次のようになります。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (256 - \text{CALM}) / (2^{20} + \text{CALM} - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

RTC 校正値の確認

32 秒の校正サイクルにするために、LPCAL=0 で RTC 校正を検証することを推奨します。

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより保証されます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。

ある時間間隔で RTC の周波数を精密に測定すると、デジタル校正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が校正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル校正の分解能による誤差となります。

- デフォルトでは、校正サイクル周期は 32 秒です。
このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (校正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。
- RTC_CALR レジスタの CALW16 ビットを 1 にセットして、校正サイクル周期を強制的に 16 秒にすることができます。
この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、校正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。
- RTC_CALR レジスタの CALW8 ビットを 1 にセットして、校正サイクル周期を強制的に 8 秒にすることができます。
この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒間で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

動作中の再較正

次の処理を実施することにより、RTC_ICSR/INITF = 0 の間でも、較正レジスタ (RTC_CALR) を動作中に更新することができます。

1. RTC_ICSR/RECALPF (再較正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC_CALR への書き込み動作後 ck_apre 3 サイクル以内に、新しい較正設定が有効になります。

46.3.17 タイムスタンプ機能

タイムスタンプは、RTC_CR レジスタの TSE ビットまたは ITSE ビットを 1 にセットすることにより有効になります。

TSE がセットされている場合：

RTC_TS ピンでタイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) にカレンダーが保存されます。

TAMPTS がセットされている場合：

内部または外部タンパイイベントが検出されると、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) にカレンダーが保存されます。[RTC 制御レジスタ \(RTC_CR\)](#) および[セクション：タンパイイベント時のタイムスタンプ](#)を参照してください。

ITSE がセットされている場合：

内部タイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) にカレンダーが保存されます。内部タイムスタンプイベントはV_{BAT} 供給に切り替えによって生成されます。

タイムスタンプイベントが発生すると、内部または外部イベントによって、RTC_SR レジスタのタイムスタンプフラグビット (TSF) がセットされます。イベントが内部の場合、RTC_SR レジスタの ITSF フラグもセットされます。

RTC_CR レジスタの TSIE ビットをセットすることにより、タイムスタンプイベントが発生したときに割込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC_TSTR および RTC_TSDR) は、その前のイベントの結果を維持します。

注： 同期プロセスにより、RTC_TS ピンまたは rtc_its 内部信号からのタイムスタンプイベントが発生してから、最大で ck_apre の 2 サイクル後に、TSF がセットされます。TSF は、タンパフラグから最大で ck_apre の 3 サイクル後にセットされます。

TSOVF は、イベントが発生してから最大で ck_apre の 1 サイクル後にセットされます。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。

注意： TSF ビットのクリア処理を行った直後にタイムスタンプイベントが発生した場合、TSF ビットおよび TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、アプリケーションは TSF がすでに 1 と読み出されていない限り、0 クリアの処理を行ってはなりません。

46.3.18 較正クロック出力

RTC_CR レジスタで COE ビットが 1 にセットされると、CALIB デバイス出力にリファレンスクロックが供給されます。

RTC_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV_A = 0x7F である場合、CALIB 周波数は $f_{\text{RTCCLK}}/64$ です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。立下りエッジには軽いジッタがあるため、CALIB のデューティサイクルは不規則になります。したがって、立上りエッジの使用が推奨されます。

COSEL がセットされ、かつ “PREDIV_S+1” がゼロ以外の 256 の倍数である場合（すなわち、PREDIV_S[7:0] = 0xFF）、CALIB 周波数は $f_{\text{RTCCLK}}/(256 * (\text{PREDIV_A}+1))$ となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケアラデフォルト値（PREDIV_A = 0x7F、PREDIV_S = 0xFF）に対する 1 Hz の較正出力に相当します。

注： COSEL がクリアされると、CALIB 出力は非同期プリスケアラの 6 番目のステージの出力になります。LPCAL を 0 から 1 に変更する場合、LPCAL 切り替え中は出力が不規則（グリッチ）になる可能性があります。LPCAL = 1 の場合、この出力は常に使用できます。LPCAL = 0 の場合、PREDIV_A が 0x20 未満のとき出力はありません。

COSEL がセットされると、CALIB 出力は同期プリスケアラの 8 番目のステージの出力になります。

46.3.19 タンパおよびアラーム出力

RTC_CR レジスタの OSEL[1:0] 制御ビットを使用してアラーム出力 TAMPALRM を有効にし、出力となる機能を選択します。これらの機能は、RTC_SR レジスタの該当するフラグの内容を反映します。

TAMPOE 制御ビットが RTC_CR でセットされている場合、すべての外部および内部のタンパフラグは論理和がとられ、TAMPALRM 出力に送られます。OSEL = 00 の場合、TAMPALRM 出力はタンパフラグのみ反映します。OSEL ≠ 00 の場合、TAMPALRM の信号でタンパフラグと、アラーム A、B、ウェイクアップフラグのいずれかがともに供給されます。

RTC_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、TAMPALRM 出力の極性を決定します。

TAMPALRM 出力

TAMPALRM ピンは、制御ビット TAMPALRM_TYPE（RTC_CR レジスタ）を使用して、出力オープンドレインまたは出力プッシュプルに設定できます。TAMPALRM_PU（RTC_CR）によって、出力モードで内部プルアップを適用できます。

注： TAMPALRM 出力が有効になると、RTC_OUT1 で CALIB より優先順位が高くなります。

TAMPALRM が RTC でオープンドレインに設定されている場合、RTC_OUT1 GPIO を入力として設定する必要があります。

46.4 RTC 低消費電力モード

表 482. 低消費電力モードが RTC に与える影響

モード	説明
SLEEP	影響なし。 RTC 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC 割込みによって、デバイスは STOP モードを終了します。
STANDBY	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC 割込みによって、デバイスは STANDBY モードを終了します。

次の表は、すべてのモードでの RTC ピンと機能の一覧です。

表 483. RTC ピン機能 (モード共通)

機能	すべての低消費電力モードで 使用可能な機能 (STANDBY モードを除く)	STANDBY モードで 使用可能な機能	V _{BAT} モードで 使用可能な機能
RTC_TS	可能	はい	はい
RTC_REFIN	可能	不可	不可
RTC_OUT1	可能	はい	はい
RTC_OUT2	可能	はい	はい

46.5 RTC 割込み

割込みチャネルは、マスク済み割込みステータスレジスタ、またはセキュリティモード設定に応じたセキュアマスク済み割込みステータスレジスタで設定されます。また、非セキュア割込み出力やセキュア割込み出力も有効化されます。

表 484. 非セキュア割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ ⁽¹⁾	有効制御ビット ⁽²⁾	割込みのクリア方法	低消費電力モードの終了
RTC	アラーム A	ALRAF	ALRAIE および (ALRASEC=0 かつ SEC=0)	CALRAF に 1 を書き込む	可能 ⁽³⁾
	アラーム B	ALRBF	ALRBIE および (ALRBSEC=0 かつ SEC=0)	CALRBF に 1 を書き込む	あり ⁽³⁾
	タイムスタンプ	TSF	TSIE および (TSSEC=0 かつ SEC=0)	CTSF に 1 を書き込む	あり ⁽³⁾
	ウェイクアップ・タイマ	WUTF	WUTIE および (WUTSEC=0 かつ SEC=0)	CWUTF に 1 を書き込む	あり ⁽³⁾
	SSR アンダーフロー	SSRUF	SSRUIE および (ALRASEC=0 かつ SEC=0)	CSSRUF に 1 を書き込む	あり ⁽³⁾

- イベントフラグは、RTC_SR レジスタ内です。
- 割込みマスク済みフラグ (イベントフラグおよびイネーブル制御ビットからの結果) は、RTC_MISR レジスタ内です。
- RTC が低消費電力モードでオシレータ機能によってクロック供給される場合。

表 485. セキュア割込みリクエスト

項目 (割込みの略称)	割込みイベント	イベントフラグ ⁽¹⁾	有効制御ビット ⁽²⁾	割込みのクリア方法	低消費電力モードの終了
RTC_S	アラーム A	ALRAF	ALRAIE および (ALRASEC=1 または SEC=1)	CALRAF に 1 を書き込む	可能 ⁽³⁾
	アラーム B	ALRBF	ALRBIE および (ALRBSEC=1 または SEC=1)	CALRBF に 1 を書き込む	あり ⁽³⁾
	タイムスタンプ	TSF	TSIE および (TSSEC=1 または SEC=1)	CTSF に 1 を書き込む	あり ⁽³⁾
	ウェイクアップ・タイマ	WUTF	WUTIE および (WUTSEC=1 または SEC=1)	CWUTF に 1 を書き込む	あり ⁽³⁾
	SSR アンダーフロー	SSRUF	SSRUIE および (ALRASEC=1 または SEC=1)	CSSRUF に 1 を書き込む	あり ⁽³⁾

- イベントフラグは、RTC_SR レジスタ内です。
- 割込みマスク済みフラグ (イベントフラグおよびイネーブル制御ビットからの結果) は、RTC_SMISR レジスタ内です。
- RTC が低消費電力モードでオシレータ機能によってクロック供給される場合。

46.6 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの[セクション 1.2](#)を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

46.6.1 RTC 時刻レジスタ (RTC_TR)

RTC_TR は、カレンダー時刻シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。[1938 ページのカレンダーの初期化と設定](#) および [1939 ページのカレンダーの読出し](#)を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[1937 ページのRTC レジスタ書き込み保護](#)を参照してください。

このレジスタは、非セキュアアクセスから書き込み保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから書き込み保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x00

バックアップドメインリセット値 : 0x0000 0000

システムリセット値 : 0x0000 0000 (BYPHAD = 0 の場合。BYPHAD = 1 の場合は、影響を受けません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

46.6.2 RTC 日付レジスタ (RTC_DR)

RTC_DR は、カレンダー日付シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。1938 ページの [カレンダーの初期化と設定](#) および 1939 ページの [カレンダーの読出し](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、1937 ページの [RTC レジスタ書き込み保護](#) を参照してください。

このレジスタは、非セキュアアクセスから書き込み保護できます。セクション 46.3.4: [RTC セキュア保護モード](#) を参照してください。

このレジスタは、非特権アクセスから書き込み保護できます。セクション 46.3.5: [RTC 特権保護モード](#) を参照してください。

アドレスオフセット : 0x04

バックアップドメインリセット値 : 0x0000 2101

システムリセット値 : 0x0000 2101 (BYP SHAD = 0 の場合。BYP SHAD = 1 の場合は、影響を受けません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

000 : 禁止
001 : 月曜日
.....
111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

注 : カレンダーは最大値に達すると停止し、ロールオーバーできません。

46.6.3 RTC サブセカンドレジスタ (RTC_SSR)

アドレスオフセット : 0x08

バックアップドメインリセット値 : 0x0000 0000

システムリセット値 : 0x0000 0000 (BYP SHAD = 0 の場合。BYP SHAD = 1 の場合は、影響を受けません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **SS[31:0]** : 同期バイナリカウンタ

SS[31:16] : 同期バイナリカウンタの MSB 値

バイナリモードまたは混合モードが選択されている場合 (BIN = 01、10、11 のいずれか) :

SS[31:16] は、SS[31:0] フリーランニングダウンカウンタの 16 MSB です。

BCD モードが選択されている場合 (BIN=00) :

SS[31:16] はハードウェアによって 0x0000 に固定されています。

SS[15:0] : サブセカンド値/同期バイナリカウンタの LSB 値

バイナリモードが選択されている場合 (BIN = 01、10、11 のいずれか) :

SS[15:0] は、SS[31:0] フリーランニングダウンカウンタの 16 MSB です。

BCD モードが選択されている場合 (BIN=00) :

SS[15:0] は、同期プリスケアラのカウンタ内の値です。秒の小数部は、下の式によって与えられます。

秒の小数部 = $(\text{PREDIV_S} - \text{SS}) / (\text{PREDIV_S} + 1)$

SS は、シフト操作後に限り、PREDIV_S より大きな値となる場合があります。この場合、正確な時刻 / 日付は、RTC_TR/RTC_DR で示される値よりも 1 秒少ない値となります。

46.6.4 RTC 初期化制御およびステータスレジスタ (RTC_ICSR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[1937 ページの RTC レジスタ書き込み保護](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非セキュアアクセスから個別に保護することもできます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x0C

バックアップドメインリセット値 : 0x0000 0007

システムリセット : 0 にクリアされる INIT、INITF、RSF ビット以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	BCDU[2:0]			BIN[1:0]		INIT	INITF	RSF	INITS	SHPF	WUTWF	Res.	Res.
			rw	rw	rw	rw	rw	rw	r	rc_w0	r	r	r		

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **RECALPF** : 再校正保留フラグ

ソフトウェアによって RTC_CALR レジスタに書き込みが行われると、RECALPF ステータスフラグが自動的に 1 にセットされ、RTC_CALR レジスタがブロックされたことを示します。新たな校正設定が認識されると、このビットは 0 に戻ります。[動作中の再校正](#)を参照してください。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:10 **BCDU[2:0]** : BCD 更新 (BIN = 10 または 11)

BCD カレンダーおよびバイナリ拡張カウンタを使用した混合モード (BIN = 10 または 11) では、カレンダーの秒は SSR 最下位ビットでインクリメントされます。

0x0 : SS[7:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x1 : SS[8:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x2 : SS[9:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x3 : SS[10:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x4 : SS[11:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x5 : SS[12:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x6 : SS[13:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

0x7 : SS[14:0] = 0 のたびに 1 秒のカレンダーインクリメントが生成されます。

ビット 9:8 **BIN[1:0]** : バイナリモード

00 : フリーランニング BCD カレンダーモード (バイナリモード無効)

01 : フリーランニングバイナリモード (BCD モード無効)

10 : フリーランニング BCD カレンダーモードおよびバイナリモード

11 : フリーランニング BCD カレンダーモードおよびバイナリモード

ビット 7 **INIT** : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC_TR と RTC_DR)、プリスケアラレジスタ (RTC_PRER)、BIN および BCDU フィールドのプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

ビット 6 INITF : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

ビット 5 RSF : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC_SSR、RTC_TR および RTC_DR) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1) の初期化モードで、またはバイパスシャドウレジスタモード (BYPSHAD = 1) で、ハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

初期化モードでソフトウェアまたはハードウェアによってクリアされます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

ビット 4 INITS : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (バックアップドメインリセット状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

ビット 3 SHPF : シフト操作保留

このフラグは、RTC_SHIFTR への書込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF ビットに書き込んでも影響はありません。

0 : 保留中のシフト操作はありません。

1 : 保留中のシフト操作があります。

ビット 2 WUTWF : ウェイクアップタイマ書込みフラグ

このビットは、RTC_CR で WUTE ビットが 0 にセットされた後、WUT 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : 初期化モード以外で、ウェイクアップタイマ設定は更新できません。

1 : ウェイクアップタイマ設定が更新できます。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

46.6.5 RTC プリスケーラレジスタ (RTC_PRER)

このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。1938 ページの**カレンダーの初期化と設定**を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、1937 ページの**RTC レジスタ書き込み保護**を参照してください。

このレジスタは、非セキュアアクセスから書き込み保護できます。**セクション 46.3.4: RTC セキュア保護モード**を参照してください。

このレジスタは、非特権アクセスから書き込み保護できます。**セクション 46.3.5: RTC 特権保護モード**を参照してください。

アドレスオフセット : 0x10

バックアップドメインリセット値 : 0x007F : 00FF

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						
									r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **PREDIV_A[6:0]** : 非同期プリスケーラ分周比

非同期分周比です。

$ck_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV_A + 1)$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV_S[14:0]** : 同期プリスケーラ分周比

同期分周比です。

$ck_spre \text{ 周波数} = ck_apre \text{ 周波数} / (PREDIV_S + 1)$

46.6.6 RTC ウェイクアップタイマレジスタ (RTC_WUTR)

このレジスタは、RTC_ICSR の WUTWF が 1 にセットされているときにのみ書き込みます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x14

バックアップドメインリセット値 : 0x0000 FFFF

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUTOCLR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 **WUTOCLR[15:0]** : ウェイクアップ自動再ロード出力クリア値

WUTOCLR[15:0] が 0x0000 ではない場合、WUTF は自動再ロードダウンカウンタが 0 に達するとハードウェアによってセットされ、自動再ロードダウンカウンタが WUTOCLR[15:0] に達するとハードウェアによってクリアされます。

WUTOCLR[15:0] = 0x0000 の場合、WUTF は WUT ダウンカウンタが 0 に達するとハードウェアによってセットされ、

ソフトウェアによってクリアされます。

ビット 15:0 **WUT[15:0]** : ウェイクアップ自動再ロード値ビット

ウェイクアップタイマが有効 (WUTE が 1 にセット) なとき、ck_wut の (WUT[15:0] + 1) サイクルごとに WUTF フラグがセットされます。ck_wut の周期は、RTC_CR レジスタの WUCKSEL[2:0] ビットで選択します。

WUCKSEL[2] = 1 のとき、ウェイクアップタイマは 17 ビットとなり、WUCKSEL[1] が事実上タイマに再ロードされる最上位ビットである WUT[16] となります。

WUTF の最初のアサートは、WUTE がセットされてから WUT と ck_wut の (WUT + 2) サイクルの間に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。

46.6.7 RTC 制御レジスタ (RTC_CR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[1937 ページのRTC レジスタ書き込み保護](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非セキュアアクセスから個別に保護することもできます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x18

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT2 EN	TAMP ALRM_ TYPE	TAMP ALRM_ PU	ALRBF CLR	ALRAF CLR	TAMP OE	TAMP TS	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRB IE	ALRA IE	TSE	WUTE	ALRBE	ALRAE	SSR UIE	FMT	BYP SHAD	REFCK ON	TS EDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **OUT2EN** : RTC_OUT2 出力は有効です。

このビットがセットされている場合、RTC_OUT2 で次のように RTC 出力を再配置できます。

OUT2EN=0 : RTC 出力 2 無効

OSEL ≠ 00 または TAMPOE = 1 の場合 : RTC_OUT1 に TAMPALRM が出力されます。

OSEL = 00 かつ TAMPOE = 0 かつ COE = 1 の場合 : RTC_OUT1 に CALIB が出力されます。

OUT2EN=1 : RTC 出力 2 有効

(OSEL ≠ 00 または TAMPOE = 1) かつ COE = 0 の場合 : RTC_OUT2 に TAMPALRM が出力されます。

OSEL = 00 かつ TAMPOE = 0 かつ COE = 1 の場合 : RTC_OUT2 に CALIB が出力されます。

(OSEL ≠ 00 または TAMPOE = 1) かつ COE = 1 の場合 : RTC_OUT2 に CALIB が、RTC_OUT1 に TAMPALRM が出力されます。

ビット 30 **TAMPALRM_TYPE** : TAMPALRM 出力形式

0 : TAMPALRM はプッシュプル出力になります。

1 : TAMPALRM はオープンドレイン出力になります。

ビット 29 **TAMPALRM_PU** : TAMPALRM プルアップイネーブル

0 : TAMPALRM 出力でプルアップは適用されません。

1 : TAMPALRM 出力でプルアップは適用されます。

ビット 28 **ALRBFCLR** : アラーム B フラグ自動クリア

0 : アラーム B イベントでトリガイイベントが生成され、次のアラームイベントを許可するために ALRBF をソフトウェアでクリアする必要があります。

1 : アラーム B イベントによってトリガイイベントが生成されます。ALRBF は ck_apre の 1 サイクル後にハードウェアにより自動的にクリアされます。

ビット 27 **ALRAFCLR** : アラーム A フラグ自動クリア

0 : アラーム A イベントでトリガイイベントが生成され、次のアラームイベントを許可するために ALRAF をソフトウェアでクリアする必要があります。

1 : アラーム A イベントによってトリガイイベントが生成されます。ALRAF は ck_apre の 1 サイクル後にハードウェアにより自動的にクリアされます。

ビット 26 **TAMPOE** : TAMPALRM でのタンパ検出出力イネーブル

0 : タンパフラグが TAMPALRM で送られません。

1 : タンパフラグが OSEL による信号および POL による極性ととともに TAMPALRM で送られます。

ビット 25 **TAMPTS** : タンパ検出イベント時のタイムスタンプの有効化

0 : タンパ検出イベントがあっても、RTC タイムスタンプは保存されません。

1 : タンパ検出イベント時、RTC タイムスタンプが保存されます。

RTC_CR レジスタで TSE = 0 であっても TAMPTS は有効です。タイムスタンプフラグ は、タンパフラグから最大で ck_apre の 3 サイクル後にセットされます。

注 : RTC 初期化モードに入る前に TAMPTS をクリアにする必要があります。

ビット 24 **ITSE** : 内部イベントでのタイムスタンプイネーブル

0 : 内部イベントでのタイムスタンプは無効です。

1 : 内部イベントでのタイムスタンプは有効です。

ビット 23 **COE** : 較正出力イネーブル

このビットは、CALIB 出力を有効にします。

0 : 較正出力は無効です。

1 : 較正出力は有効です。

ビット 22:21 **OSEL[1:0]** : 出力選択

これらのビットは、TAMPALRM 出力に送られるフラグの選択に使用します。

00 : 出力は無効です。

01 : アラーム A 出力は有効です。

10 : アラーム B 出力は有効です。

11 : ウェイクアップ出力は有効です。

ビット 20 **POL** : 出力極性

このビットは、TAMPALRM 出力の極性の設定に使用します。

0 : ALRAF/ALRBF/WUTF が (OSEL[1:0] に応じて) アサートされた場合、または TAMPxF/ITAMPxF がアサートされた場合 (TAMPOE = 1)、このピンはハイになります。

1 : ALRAF/ALRBF/WUTF が (OSEL[1:0] に応じて) アサートされた場合、または TAMPxF/ITAMPxF がアサートされた場合 (TAMPOE = 1)、このピンはローになります。

ビット 19 **COSEL** : 較正出力選択

COE = 1 のとき、このビットによって CALIB に出力される信号を選択します。

0 : 較正出力は 512 Hz です。

1 : 較正出力は 1 Hz です。

これらの周波数は、RTCCLK が 32.768 kHz で、プリスケアラがデフォルト値 (PREDIV_A = 127 および PREDIV_S = 255) の場合に有効です。[セクション 46.3.18 : 較正クロック出力](#)を参照してください。

ビット 18 **BKP** : バックアップ

このビットは、サマータイムの変更を実施したか否かを記憶しておくため、ユーザが書き込むことができます。

ビット 17 **SUB1H** : 1 時間差し引き (冬時間変更)

このビットを初期化モード以外のときにセットすると、現在時刻が 0 でない場合にカレンダー時刻から 1 時間を差し引きます。このビットは常に 0 として読み出されます。

現在時間が 0 のときにこのビットをセットしても、影響はありません。

0 : 影響なし。

1 : 現在時刻から 1 時間差し引きます。これは、冬時間変更に使えます。

ビット 16 **ADD1H** : 1 時間加算 (サマータイム変更)

このビットを初期化モード以外のときにセットすると、カレンダー時刻に 1 時間加算します。このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : 現在時刻に 1 時間加算します。これは、サマータイム変更に使えます。

ビット 15 **TSIE** : タイムスタンプ割込みは有効です。

0 : タイムスタンプ割込みは無効です。

1 : タイムスタンプ割込みは有効です。

ビット 14 **WUTIE** : ウェイクアップタイマ割込みは有効です。

0 : ウェイクアップタイマ割込みは無効です。

1 : ウェイクアップタイマ割込みは有効です。

ビット 13 **ALRBIE** : アラーム B 割込みは有効です。

0 : アラーム B 割込みは無効です。

1 : アラーム B 割込みは有効です。

ビット 12 **ALRAIE** : アラーム A 割込みイネーブル

0 : アラーム A 割込みは無効です。

1 : アラーム A 割込みは有効です。

ビット 11 **TSE** : タイムスタンプイネーブル

0 : タイムスタンプは無効です。

1 : タイムスタンプは有効です。

ビット 10 **WUTE** : ウェイクアップタイマは有効です。

0 : ウェイクアップタイマは無効です。

1 : ウェイクアップタイマは有効です。

注 : ウェイクアップタイマが無効の場合は、再度有効にする前に WUTWF=1 になるまで待ちます。

ビット 9 **ALRBE** : アラーム B イネーブル

0 : アラーム B は無効です。

1 : アラーム B は有効です。

ビット 8 **ALRAE** : アラーム A イネーブル

0 : アラーム A は無効です。

1 : アラーム A は有効です。

ビット 7 **SSRUIE** : SSR アンダーフロー割込みイネーブル

0 : SSR アンダーフロー割込みは無効です。

1 : SSR アンダーフロー割込みは有効です。

ビット 6 **FMT** : 時間形式

0 : 24 時間/日形式

1 : AM/PM 時間形式

ビット 5 **BYPHAD** : シャドウレジスタをバイパスします。

0 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。

1 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、カレンダカウンタから直接取得されます。

注 : APB クロックの周波数が RTCCLK の 7 倍未満である場合、BYPHAD は 1 にセットする必要があります。

ビット 4 **REFCKON** : RTC_REFIN リファレンスクロック検出イネーブル (50 または 60 Hz)

0 : RTC_REFIN 検出は無効です。

1 : RTC_REFIN 検出は有効です。

注 : BIN は 0x00、PREDIV_S は 0x00FF でなければなりません。

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

0 : RTC_TS 入力の立上りエッジによってタイムスタンプイベントを生成します。

1 : RTC_TS 入力の立下りエッジによってタイムスタンプイベントを生成します。

不要な TSF 設定を回避するため、TSEDGE が変化した場合には TSE をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ck_wut ウェイクアップクロックの選択

000 : RTC/16 クロックが選択されます。

001 : RTC/8 クロックが選択されます。

010 : RTC/4 クロックが選択されます。

011 : RTC/2 クロックが選択されます。

10x : BCD モードでは ck_spre (通常は 1 Hz) クロックが選択されます。バイナリまたは混合モードでは、このクロックが BCDU によって選択されます。

11x : BCD モードでは ck_spre (通常は 1 Hz) クロックが選択されます。バイナリまたは混合モードでは、このクロックが BCDU によって選択されます。さらに、 2^{16} が WUT カウンタ値に追加されます。

注 : 初期化モード (RTC_ICSR/INITF = 1) の場合のみ、このレジスタのビット 6 および 4 が書き込めます。
WUT = ウェイクアップユニットカウンタ値。 $WUT = (0x0000 \sim 0xFFFF) + 0x10000$ (WUCKSEL[2:1] = 11 の場合追加されます。)

このレジスタのビット 2~0 は、RTC_CR の WUTE ビット = 0 かつ RTC_ICSR の WUTWF ビット = 1 の場合にのみ書き込めます。

カレンダーの時間項目のインクリメント中は時間を変更しないことが推奨されます。カレンダーの時間項目のインクリメントがマスクされる可能性があるためです。

ADD1H および SUB1H の変更は、次の秒から有効になります。

46.6.8 RTC 特権モード制御レジスタ (RTC_PRIVCFGR)

このレジスタに書き込めるのは、APB アクセスが特権のときだけです。このレジスタは書き込み保護することができ、このレジスタの各ビットは、RTC_SECCFGR の設定に応じて、非セキュアアクセスから個別に書き込み保護することもできます ([セクション 46.3.5 : RTC 特権保護モード](#)を参照)。

アドレスオフセット : 0x1C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV	INIT PRIV	CAL PRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS PRIV	WUT PRIV	ALRB PRIV	ALRA PRIV
rw	rw	rw										rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **PRIV** : RTC 特権保護

0 : APB アクセスが特権でも非特権でも、その他の特権保護ビットで保護されているレジスタ以外、すべての RTC レジスタに書き込めます。

1 : APB アクセスが特権の場合のみ、すべての RTC レジスタに書き込めます。

ビット 14 **INITPRIV** : 初期化の特権保護

0 : APB アクセスが特権でも非特権でも、RTC 初期化モード、カレンダー、プリスケアラのレジスタに書き込めます。

1 : APB アクセスが特権の場合のみ、RTC 初期化モード、カレンダー、プリスケアラのレジスタに書き込めます。

ビット 13 **CALPRIV** : シフトレジスタ、サマータイム、較正およびリファレンスクロックの特権保護

0 : APB アクセスが特権でも非特権でも、シフトレジスタ、サマータイム、較正およびリファレンスクロックを書き込みます。

1 : APB アクセスが特権の場合のみ、シフトレジスタ、サマータイム、較正およびリファレンスクロックを書き込みます。

ビット 12:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TSPRIV** : タイムスタンプの特権保護

0 : APB アクセスが特権でも非特権でも、RTC タイムスタンプの設定と割込みクリアを書き込みます。

1 : APB アクセスが特権の場合のみ、RTC タイムスタンプの設定と割込みクリアを書き込みます。

ビット 2 **WUTPRIV** : ウェイクアップタイマ特権保護

0 : APB アクセスが特権でも非特権でも、RTC ウェイクアップタイマの設定と割込みクリアを書き込みます。

1 : APB アクセスが特権の場合のみ、RTC ウェイクアップタイマの設定と割込みクリアを書き込みます。

ビット 1 **ALRBPRIV** : アラーム B の特権保護

0 : APB アクセスが特権でも非特権でも、RTC アラーム B の設定と割込みクリアを書き込みます。

1 : APB アクセスが特権の場合のみ、RTC アラーム B の設定と割込みクリアを書き込みます。

ビット 0 **ALRAPRIV** : アラーム A および SSR アンダーフロー特権保護

0 : APB アクセスが特権でも非特権でも、RTC アラーム A および SSR アンダーフローの設定と割込みクリアを書き込みます。

1 : APB アクセスが特権の場合のみ、RTC アラーム A および SSR アンダーフローの設定と割込みクリアを書き込みます。

注 : 読出し保護についての詳細は、[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

46.6.9 RTC セキュア設定レジスタ (RCC_SECCFGR)

このレジスタに書き込めるのは、APB アクセスがセキュアなときだけです。

このレジスタは、全体的に書き込み保護することができ、このレジスタの各ビットは、RTC_PRIVCFGR の設定に応じて、非特権アクセスから個別に書き込み保護することもできます ([セクション 46.3.5 : RTC 特権保護モード](#)を参照)。

アドレスオフセット : 0x20

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC	INIT SEC	CAL SEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS SEC	WUT SEC	ALRB SEC	ALRA SEC
rw	rw	rw										rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **SEC** : RTC グローバル保護

0 : APB アクセスがセキュアでも非セキュアでも、その他のセキュア保護ビットで保護されているレジスタ以外、すべての RTC レジスタに書き込みます。

1 : APB アクセスがセキュアな場合のみ、すべての RTC レジスタに書き込みます。

ビット 14 **INITSEC** : 初期化の保護

0 : APB アクセスがセキュアでも非セキュアでも、RTC 初期化モード、カレンダー、プリスケアラのレジスタに書き込めます。

1 : APB アクセスがセキュアな場合のみ、RTC 初期化モード、カレンダー、プリスケアラのレジスタに書き込めます。

ビット 13 **CALSEC** : シフトレジスタ、サマータイム、較正およびリファレンスクロックの保護

0 : APB アクセスがセキュアでも非セキュアでも、シフトレジスタ、サマータイム、較正およびリファレンスクロックを書き込めます。

1 : APB アクセスがセキュアな場合のみ、シフトレジスタ、サマータイム、較正およびリファレンスクロックを書き込めます。

ビット 12:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TSSEC** : タイムスタンプの保護

0 : APB アクセスがセキュアでも非セキュアでも、RTC タイムスタンプの設定と割込みクリアを書き込めます。

1 : APB アクセスがセキュアな場合のみ、RTC タイムスタンプの設定と割込みクリアを書き込めます。

ビット 2 **WUTSEC** : ウェイクアップタイマ保護

0 : APB アクセスがセキュアでも非セキュアでも、RTC ウェイクアップタイマの設定と割込みクリアを書き込めます。

1 : APB アクセスがセキュアな場合のみ、RTC ウェイクアップタイマの設定と割込みクリアを書き込めます。

ビット 1 **ALRBSEC** : アラーム B の保護

0 : APB アクセスがセキュアでも非セキュアでも、RTC アラーム B の設定と割込みクリアを書き込めます。

1 : APB アクセスがセキュアな場合のみ、RTC アラーム B の設定と割込みクリアを書き込めます。

ビット 0 **ALRASEC** : アラーム A および SSR アンダーフロー保護

0 : APB アクセスがセキュアでも非セキュアでも、RTC アラーム A および SSR アンダーフローの設定と割込みクリアを書き込めます。

1 : APB アクセスがセキュアな場合のみ、RTC アラーム A および SSR アンダーフローの設定と割込みクリアを書き込めます。

注 : 読出し保護についての詳細は、[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

46.6.10 RTC 書込み保護レジスタ (RTC_WPR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	KEY[7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **KEY[7:0]** : 書込み保護キー

このバイトはソフトウェアで書き込まれます。

このバイトを読み出すと常に 0x00 が返されます。

RTC レジスタの書込み保護解除方法については、[RTC レジスタ書込み保護](#)を参照してください。

46.6.11 RTC 較正レジスタ (RTC_CALR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[1937 ページのRTC レジスタ書き込み保護](#)を参照してください。

このレジスタは、非セキュアアクセスから書き込み保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから書き込み保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x28

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	LPCAL	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **CALP** : RTC 周波数を 488.5 ppm 増加

0 : RTCCLK パルスは加えられません。

1 : RTCCLK の 2^{11} パルスごとに 1 パルス効果的に挿入されます (周波数が 488.5 ppm 増加)。

この機能は、CALM と共に使用されることを想定しており、カレンダーの周波数を高分解能で下げることができます。入力周波数が 32768 Hz の場合、32 秒枠の間に追加される RTCCLK パルスの数は次のように算出されます : $(512 * CALP) - CALM$

[セクション 46.3.16 : RTC の高精度デジタル較正](#)を参照してください。

ビット 14 **CALW8** : 8 秒較正サイクル周期の使用

CALW8 が 1 にセットされると、8 秒較正サイクル周期が選択されます。

注 : **CALW8 = 1 の場合、CALM[1:0] は 00 に固定されます。**[セクション 46.3.16 : RTC の高精度デジタル較正](#)を参照してください。

ビット 13 **CALW16** : 16 秒較正サイクル周期の使用

CALW16 が 1 にセットされると、16 秒較正サイクル周期が選択されます。CALW8=1 の場合、このビットを 1 にセットすることはできません。

注 : **CALW16 = 1 の場合、CALM[0] は 0 に固定されます。**[セクション 46.3.16 : RTC の高精度デジタル較正](#)を参照してください。

ビット 12 **LPCAL** : RTC 低消費電力モード

0 : 較正時間範囲が 2^{20} RTCCLK となり、高消費電力モードとなります。このモードは、32 秒未満の較正時間範囲が必要な場合のみセットする必要があります。

1 : 較正時間範囲が 2^{20} ck_apre となり、超低消費電力モードで必要な設定となります。

ビット 11:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **CALM[8:0]** : 較正マイナス

RTCCLK 2^{20} パルス (入力周波数が 32768 Hz の場合 32 秒) 内の CALM をマスクすることによって、カレンダーの周波数が下げられます。この方法により、カレンダーの周波数を 0.9537 ppm の分解能で下げることができます。

カレンダーの周波数を上げるには、この機能を CALP と共に使用する必要があります。[セクション 46.3.16 : 1942 ページのRTC の高精度デジタル較正](#)を参照してください。

46.6.12 RTC シフト制御レジスタ (RTC_SHIFTR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[1937 ページの RTC レジスタ書き込み保護](#)を参照してください。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x2C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **ADD1S** : 1 秒加算

0 : 影響なし。

1 : 時計／カレンダーに 1 秒加算します。

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ICSR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

この機能は、SUBFS (下記説明を参照) と共に使用されることを想定しており、不可分操作で、効果的に時計に秒の小数部を加算することを目的としています。

ビット 30:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **SUBFS[14:0]** : 秒の小数部差し引き

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ICSR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

SUBFS に書き込まれた値は、同期プリスケラのカウンタに加算されます。このカウンタはカウントダウンしていくので、この操作によって、次の式で求める値が効果的にクロックから差し引き (遅延) されます。

遅れ (秒) = $SUBFS / (PREDIV_S + 1)$

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。

進み (秒) = $(1 - (SUBFS / (PREDIV_S + 1)))$

混合 BCD-binay モードでは (BIN=10 または 11)、SUBFS[14:BCDU+8] には 0 を書き込む必要があります。

注 : SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実にあります。

46.6.13 RTC タイムスタンプ時刻レジスタ (RTC_TSTR)

このレジスタの内容は、RTC_SR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x30

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

46.6.14 RTC タイムスタンプ日付レジスタ (RTC_TSDR)

このレジスタの内容は、RTC_SR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x34

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **WDU[2:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

46.6.15 RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)

このレジスタの内容は、RTC_SR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x38

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 SS[31:0] : サブセカンド値/同期バイナリカウンタ値

SS[31:0] は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。

46.6.16 RTC アラーム A レジスタ (RTC_ALRMAR)

このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x40

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **MSK4** : アラーム A 日付マスク

0 : 日付/曜日が一致すると、アラーム A がセットされます。

1 : アラーム A の比較では日付/曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

0 : DU[3:0] は日付の一の位を表します。

1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム A 時マスク

0 : 時が一致すると、アラーム A がセットされます。

1 : アラーム A の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム A 分マスク

0 : 分が一致すると、アラーム A がセットされます。

1 : アラーム A の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム A 秒マスク

0 : 秒が一致すると、アラーム A がセットされます。

1 : アラーム A の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

46.6.17 RTC アラーム A サブセカンドレジスタ (RTC_ALRMASSR)

このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット: 0x44

バックアップドメインリセット値: 0x0000 0000

システムリセット: 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCLR	Res.	MASKSS[5:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw		rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

ビット 31 **SSCLR**: アラームでの同期カウンタのクリア (バイナリモードのみ)

0: 同期バイナリカウンタ (RTC_SSR の SS[31:0]) はフリーランニングです。

1: 同期バイナリカウンタ (RTC_SSR の SS[31:0]) は、0xFFFF FFFF から RTC_ALRABINR.SS[31:0] まで動作し、RTC_ALRABINR.SS[31:0] に達してから ck_apre の 1 サイクル後に自動的に 0xFFFF FFFF で再ロードされます。

注: BCD または混合モードが使用されている場合 (BIN = 00、10、11 のいずれか)、SSCLR は 0 のままにしておく必要があります。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:24 **MASKSS[5:0]**: このビットから始まる最上位ビットのマスク

0: アラーム A に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

1: アラーム A の比較では SS[31:1] を無視します。SS[0] のみ比較されます。

2: アラーム A の比較では SS[31:2] を無視します。SS[1:0] のみ比較されます。

.....

31: アラーム A の比較では SS[31] を無視します。SS[30:0] のみ比較されます。

32~63: アラームをアクティブにするには、32 の全ての SS ビットを比較し一致する必要があります。

注: BCD モード (BIN=00) では、同期カウンタのオーバーフロービット (ビット 31:15) が比較されることはありません。これらのビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **SS[14:0]**: サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

このフィールドは、RTC_ALRABINR の SS[14:0] のミラーであり、RTC_ALRABINR でも読出しまたは書き込みできます。

注: TAMP_ATCR1 で SSCLR に ATCKSEL[3] = 1 がセットされている場合、SS[3:0] は 0000 でなければなりません。

46.6.18 RTC アラーム B レジスタ (RTC_ALRMBR)

このレジスタは、RTC_CR レジスタの ALRBE がリセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x48

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WD SEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **MSK4** : アラーム B 日付マスク

0 : 日付／曜日が一致すると、アラーム B がセットされます。

1 : アラーム B の比較では日付／曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

0 : DU[3:0] は日付の一の位を表します。

1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム B 時マスク

0 : 時が一致すると、アラーム B がセットされます。

1 : アラーム B の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム B 分マスク

0 : 分が一致すると、アラーム B がセットされます。

1 : アラーム B の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム 秒マスク

0 : 秒が一致すると、アラーム B がセットされます。

1 : アラーム B の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

46.6.19 RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)

このレジスタは、RTC_CR レジスタの ALRBE がリセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x4C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCLR	Res.	MASKSS[5:4]		MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw		rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

ビット 31 **SSCLR** : アラームでの同期カウンタのクリア (バイナリモードのみ)

0 : 同期バイナリカウンタ (RTC_SSR の SS[31:0]) はフリーランニングです。

1 : 同期バイナリカウンタ (RTC_SSR の SS[31:0]) は、0xFFFF FFFF から RTC_ALRBBINR.SS[31:0] まで動作し、RTC_ALRBBINR.SS[31:0] に達してから ck_apre の 1 サイクル後に自動的に 0xFFFF FFFF で再ロードされます。

注 : BCD または混合モードが使用されている場合 (BIN = 00、10、11 のいずれか)、SSCLR は 0 のままにしておく必要があります。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:24 **MASKSS[5:0]** : このビットから始まる最上位ビットのマスク

0 : アラーム B に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

1 : アラーム B の比較では SS[31:1] を無視します。SS[0] のみ比較されます。

2 : アラーム B の比較では SS[31:2] を無視します。SS[1:0] のみ比較されます。

.....

31 : アラーム B の比較では SS[31] を無視します。SS[30:0] のみ比較されます。

32~63 : アラームをアクティブにするには、32 の全ての SS ビットを比較し一致する必要があります。

注 : BCD モード (BIN=00) では、同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **SS[14:0]** : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

このフィールドは、RTC_ALRBBINR の SS[14:0] のミラーであり、RTC_ALRBBINR でも読出しまたは書込みできます。

注 : TAMP_ATCR1 で SSCLR に ATCKSEL[3] = 1 がセットされている場合、SS[3:0] は 0000 でなければなりません。

46.6.20 RTC ステータスレジスタ (RTC_SR)

このレジスタは全体的に保護することができ、このレジスタの各ビットは非セキュアアクセスから個別に保護することもできます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x50

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSRUF	ITSF	TSOVF	TSF	WUTF	ALRBF	ALRAF
									r	r	r	r	r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SSRUF** : SSR アンダーフローフラグ

このフラグは、SSR が 0 を下回る場合にハードウェアによってセットされます。SSCLR=1 の場合、SSRUF はセットされません。

ビット 5 **ITSF** : 内部タイムスタンプフラグ

このフラグは、内部イベントでタイムスタンプが発生したときに、ハードウェアによってセットされます。

ビット 4 **TSOVF** : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 3 **TSF** : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

ITSF フラグがセットされている場合、ITSF とともに TSF をクリアする必要があります。

注 : **TAMPTS = 1** で、タンパイイベントに続く **ck_apre** の 3 サイクルの間にタンパフラグが読み出される場合、TSF はセットされません。詳細については、[タンパイイベント時のタイムスタンプ](#)を参照してください。

ビット 2 **WUTF** : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。

WUTOCLR[15:0] が 0x0000 でない場合、ウェイクアップ自動再ロードカウンタが WUTOCLR 値に達すると、ハードウェアによってWUTF がクリアされます。

WUTOCLR[15:0] が 0x0000 である場合、WUTF はソフトウェアでクリアする必要があります。

このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 1 **ALRBF** : アラーム B フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム B レジスタ (RTC_ALRMBR) と一致したときにハードウェアによってセットされます。

ビット 0 **ALRAF** : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム A レジスタ (RTC_ALRMAR) と一致したときにハードウェアによってセットされます。

注 : このレジスタのビットは、RTC_SCR レジスタで対応するクリアビットをセットしてから 2 APB クロックサイクル後にクリアされます。

46.6.21 RTC 非セキュアマスク済み割込みステータスレジスタ (RTC_MISR)

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x54

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSR UMF	ITS MF	TSOVM F	TS MF	WUT MF	ALRB MF	ALRA MF
									r	r	r	r	r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SSRUMF** : SSR アンダーフロー非セキュアマスク済みフラグ

このフラグは、SSR アンダーフロー非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 5 **ITSMF** : 内部タイムスタンプ非セキュアマスク済みフラグ

このフラグは、内部イベントでタイムスタンプが発生して、タイムスタンプ非セキュア割込みが発生したときに、ハードウェアによってセットされます。

ビット 4 **TSOVMF** : タイムスタンプオーバーフロー非セキュアマスク済みフラグ

このフラグは、TSMF がすでにセットされている間にタイムスタンプ割込みが発生した場合に、ハードウェアによってセットされます。

TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 3 **TSMF** : タイムスタンプ非セキュアマスク済みフラグ

このフラグは、タイムスタンプ非セキュア割込みが発生した場合にハードウェアによってセットされます。ITSF フラグがセットされている場合、ITSF とともに TSF をクリアする必要があります。

ビット 2 **WUTMF** : ウェイクアップタイマ非セキュアマスク済みフラグ

このフラグは、ウェイクアップタイマ非セキュア割込みが発生した場合にハードウェアによってセットされます。

このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 1 **ALRBMF** : アラーム B 非セキュアマスク済みフラグ

このフラグは、アラーム B 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 0 **ALRAMF** : アラーム A マスク済みフラグ

このフラグは、アラーム A 非セキュア割込みが発生した場合にハードウェアによってセットされます。

46.6.22 RTC セキュアマスク済み割込みステータスレジスタ (RTC_SMISR)

このレジスタは全体的に保護することができ、このレジスタの各ビットは非セキュアアクセスから個別に保護することもできます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x58

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSR UMF	ITS MF	TSOV MF	TS MF	WUT MF	ALRB MF	ALRA MF
									r	r	r	r	r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **SSRUMF** : SSR アンダーフローセキュアマスク済みフラグ

このフラグは、SSR アンダーフローセキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 5 **ITSMF** : 内部タイムスタンプ割込みセキュアマスク済みフラグ

このフラグは、内部イベントでタイムスタンプが発生して、タイムスタンプセキュア割込みが発生したときに、ハードウェアによってセットされます。

ビット 4 **TSOVMF** : タイムスタンプオーバーフロー割込みセキュアマスク済みフラグ

このフラグは、TSMF がすでにセットされている間にタイムスタンプセキュア割込みが発生した場合に、ハードウェアによってセットされます。

TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 3 **TSMF** : タイムスタンプ割込みセキュアマスク済みフラグ

このフラグは、タイムスタンプセキュア割込みが発生した場合にハードウェアによってセットされます。ITSF フラグがセットされている場合、ITSF とともに TSF をクリアする必要があります。

ビット 2 **WUTMF** : ウェイクアップタイマ割込みセキュアマスク済みフラグ

このフラグは、ウェイクアップタイマセキュア割込みが発生した場合にハードウェアによってセットされます。

このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 1 **ALRBMF** : アラーム B 割込みセキュアマスク済みフラグ

このフラグは、アラーム B セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 0 **ALRAMF** : アラーム A 割込みセキュアマスク済みフラグ

このフラグは、アラーム A セキュア割込みが発生した場合にハードウェアによってセットされます。

46.6.23 RTC ステータスクリアレジスタ (RTC_SCR)

このレジスタは全体的に保護することができ、このレジスタの各ビットは非セキュアアクセスから個別に保護することもできます。セクション 46.3.4 : RTC セキュア保護モードを参照してください。

このレジスタは全体的に保護することができ、このレジスタの各ビットは非特権アクセスから個別に保護することもできます。セクション 46.3.5 : RTC 特権保護モードを参照してください。

アドレスオフセット : 0x5C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSR UF	CITS F	CTSOV F	CTS F	CWUT F	CALRB F	CALRA F
									W	W	W	W	W	W	W

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CSSRUF** : SSR アンダーフローフラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの SSRUF がクリアされます。

ビット 5 **CITSF** : 内部タイムスタンプフラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの ITSF ビットがクリアされます。

ビット 4 **CTSOVF** : タイムスタンプオーバーフローフラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの TSOVF ビットがクリアされます。

TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 3 **CTS F** : タイムスタンプフラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの TSF ビットがクリアされます。

ITSF フラグがセットされている場合、CRSF と CITSF をセットして、ITSF とともに TSF をクリアする必要があります。

ビット 2 **CWUTF** : ウェイクアップタイマフラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの WUTF ビットがクリアされます。

ビット 1 **CALRBF** : アラーム B フラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの ALRBF ビットがクリアされます。

ビット 0 **CALRAF** : アラーム A フラグのクリア

このビットに 1 を書き込むと、RTC_SR レジスタの ALRAF ビットがクリアされます。

46.6.24 RTC オプションレジスタ (RTC_OR)

アドレスオフセット : 0x60

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUT2_RMP
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **OUT2_RMP** : RTC_OUT2 マッピング

0 : RTC_OUT2 は PI8 に配置されます。

1 : RTC_OUT2 は PB2 に配置されます。

46.6.25 RTC アラーム A バイナリモードレジスタ (RTC_ALRABINR)

このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4 : RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5 : RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x70

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **SS[31:0]** : バイナリモードでの同期カウンタアラーム値

この値が同期カウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0～MASKSS-1 のビットだけが比較されます。

SS[14:0] は RTC_ALRMASRR の SS[14:0] のミラーであり、RTC_ALRMASRR でも読出しまたは書き込みできます。

注: **TAMP_ATCR1** で **SSCLR** に **ATCKSEL[3] = 1** がセットされている場合、**SS[3:0]** は **0000** でなければなりません。

46.6.26 RTC アラーム B バイナリモードレジスタ (RTC_ALRBBINR)

このレジスタは、RTC_CR レジスタの ALRBE がリセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは、非セキュアアクセスから保護できます。[セクション 46.3.4: RTC セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 46.3.5: RTC 特権保護モード](#)を参照してください。

アドレスオフセット : 0x74

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SS[31:16]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 31:0 **SS[31:0]** : バイナリモードでの同期カウンタアラーム値

この値が同期カウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0～MASKSS-1 のビットだけが比較されます。

SS[14:0] は RTC_ALRMBSSRR の SS[14:0] のミラーであり、RTC_ALRMBSSR でも読みまたは書き込みできます。

注 : TAMP_ATCR1 で SSCLR に ATCKSEL[3] = 1 がセットされている場合、SS[3:0] は 0000 でなければなりません。

46.6.27 RTC レジスタマップ

表 486. RTC レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x00	RTC_TR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]	HU[3:0]			Res.	MNT[2:0]			MNU[3:0]			Res.	ST[2:0]			SU[3:0]											
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0					
0x04	RTC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]			YU[3:0]			WDU[2:0]			MT	MU[3:0]			Res.	Res.	DT [1:0]		DU[3:0]										
	リセット値										0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1		0	0	0	0	0	0	1				
0x08	RTC_SSR	SS[31:16]															SS[15:0]																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0C	RTC_ICSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF	Res.	Res.	Res.	BCDU [2:0]			BIN [1:0]		INIT	INITF	RSF	INTS	SHPF	WUT WF	Res.	Res.						
	リセット値															0				0	0	0	0	0	0	0	0	0	0	0	1							
0x10	RTC_PRER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						PREDIV_S[14:0]																					
	リセット値										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1				
0x14	RTC_WUTR	WUTOCLR[15:0]															WUT[15:0]																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x18	RTC_CR	OUT2EN	TAMPALRM_TYPE	TAMPALRM_PU	ALRBFCLR	ALRAFCLR	TAMPOE	TAMPTS	ITSE	COE	OSEL [1:0]		POL	COSEL	BKP	SUBH	ADDH	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	SSRUIE	FMT	BYPHAD	REFCKON	TSEDGE	WUCK SEL[2:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x1C	RTC_PRIVCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIV	INITPRIV	CALPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSPRIV	WUTPRIV	ALRBPRIV	ALRAPRIV					
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x20	RTC_SECCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC	INITSEC	CALSEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSSEC	WUTSEC	ALRBSEC	ALRASEC					
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x24	RTC_WPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]														
	リセット値																							0	0	0	0	0	0	0	0	0	0					
0x28	RTC_CALR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	LPCAL	Res.	Res.	Res.	CALM[8:0]														
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x2C	RTC_SHIFTR	ADDIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]																				
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x30	RTC_TSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]	HU[3:0]			Res.	MNT[2:0]			MNU[3:0]			Res.	ST[2:0]			SU[3:0]											
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

表 486. RTC レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	RTC_TSDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	リセット値																	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0
0x38	RTC_TSSSR	SS[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	RTC_ALRMAR	MSK4	WDSEL	DT [1:0]		DU[3:0]			MSK3	PM	HT [1:0]		HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK1	ST[2:0]		SU[3:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	RTC_ ALRMASR	SSCLR	Res.	MASKSS [5:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]																
	リセット値	0		0	0	0	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	RTC_ALRMBR	MSK4	WDSEL	DT [1:0]		DU[3:0]			MSK3	PM	HT [1:0]		HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK1	ST[2:0]		SU[3:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	RTC_ALRMBSSR	SSCLR	Res.	MASKSS [5:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]																
	リセット値	0		0	0	0	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	RTC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSRUF	ITSF	TSOVF	TSF	WUTF	ALRBF	ALRAF
	リセット値																										0	0	0	0	0	0	0
0x54	RTC_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSRUMF	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
	リセット値																										0	0	0	0	0	0	0
0x58	RTC_SMISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSRUMF	ITSMF	TSOVMF	TSMF	WUTMF	ALRBMF	ALRAMF
	リセット値																										0	0	0	0	0	0	0
0x5C	RTC_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSRUF	CITSF	CTSOVF	CTSF	CWUTF	CALRBF	CALRAF
	リセット値																										0	0	0	0	0	0	0
0x60	RTC_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUT2_RMP
	リセット値																																0
0x70	RTC_ ALRABINR	SS[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	RTC_ ALRBBINR	SS[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

47 タンパおよびバックアップレジスタ (TAMP)

47.1 概要

耐タンパ検出回路は、機密データを外部攻撃から保護するために使用されます。32 個の 32 ビットバックアップレジスタは、すべての低消費電力モードと V_{BAT} モードで保持されます。バックアップレジスタとデバイスのその他の機密情報は、11 タンパピンと 13 内部タンパを備えたこの耐タンパ検出回路によって保護されます。外部のタンパピンは、エッジ検出、フィルタあり／なしのレベル検出、タンパピンが外部にオープンまたは短絡したことを自動チェックすることで、セキュリティレベルを向上させるアクティブタンパに対して設定できます。

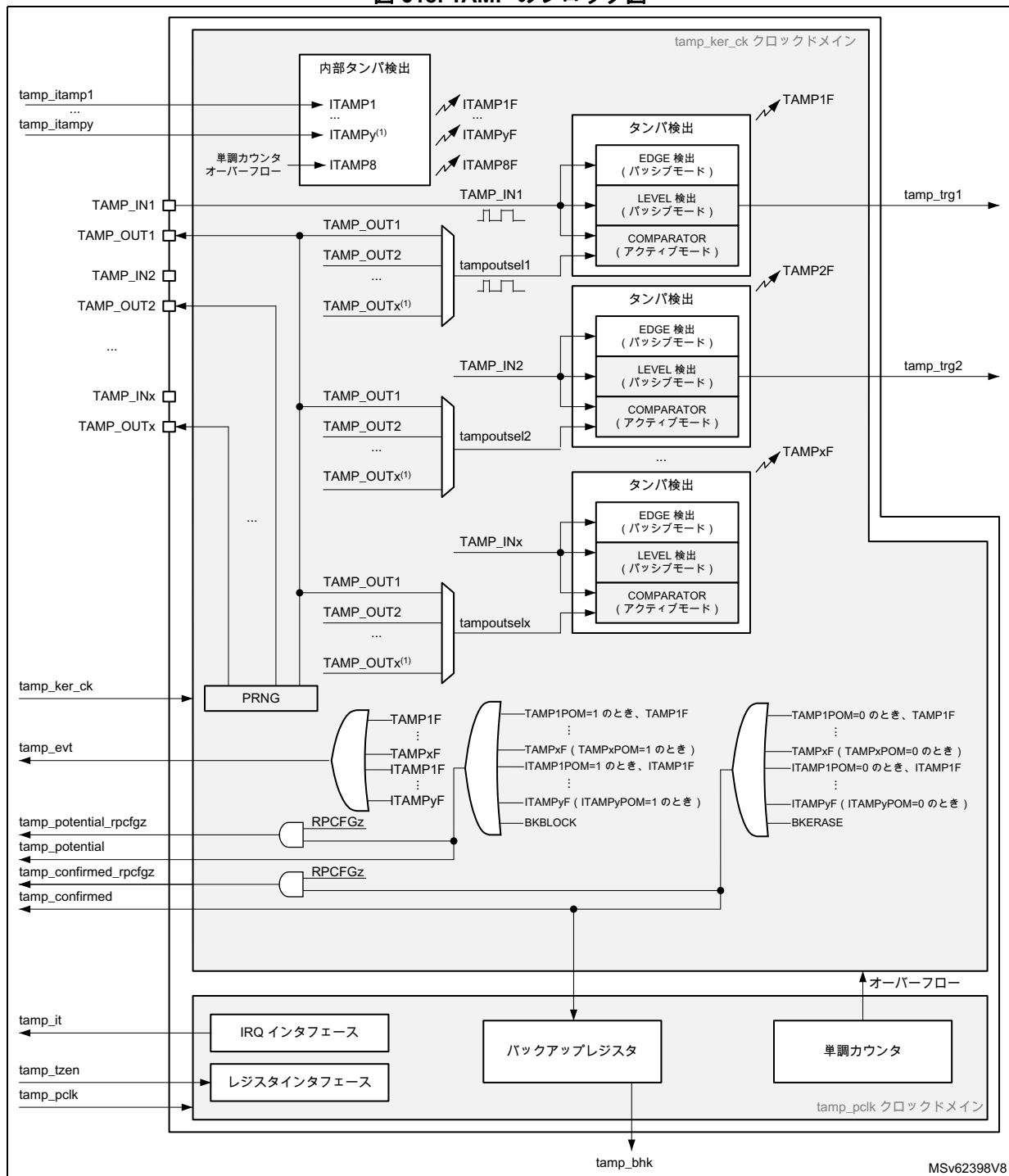
47.2 TAMP の主な機能

- タンパ検出によってバックアップレジスタ、バックアップ SRAM、SRAM2、キャッシュ、および暗号化ペリフェラルを消去することもできます。タンパによって保護されるデバイスリソースは、「デバイス機密情報」と呼ばれます。
- 32 の 32 ビットバックアップレジスタ：
 - バックアップレジスタ (TAMP_BKPxR) は、 V_{DD} 電源がオフに切り替わったときに V_{BAT} による電源がオンの状態になるバックアップドメインに実装されます。
- 8 の外部タンパ検出イベント用の最大 11 個のタンパピン：
 - アクティブタンパモード：タンパ出力とタンパ入力を連続的に比較して、物理的なオープン/ショート攻撃から保護します。
 - フレキシブルアクティブタンパ I/O 管理：メッシュ 5 枚（各入力専用出力に関連付けられる）からメッシュ 8 枚（1 つの出力を最大 8 つのタンパ入力で共有）
 - パッシブタンパ：内部プルアップハードウェア管理により超低消費電力エッジまたはレベル検出を行います。
 - 設定可能なデジタルフィルタ。
- 過渡または環境変動攻撃に対する保護のための 13 の内部タンパイベント
- 各タンパは次の 2 つのモードで設定できます。
 - 確認モード：タンパ検出時に、バックアップレジスタを含めて、機密情報がすぐに消去されます。
 - 潜在モード：タンパ検出の後に、大部分の機密情報の消去がソフトウェアによって起動されます。
- あらゆるタンパ検出によって RTC タイムスタンプイベントを生成可能
- TrustZone サポート：
 - タンパのセキュアまたは非セキュア設定
 - 3 種類の設定可能なサイズを持つエリアでのバックアップレジスタ設定
 - 読出し／書込みがセキュアなエリア 1 個
 - 書込みがセキュア／読出しが非セキュアのエリア 1 個
 - 読出し／書込みが非セキュアなエリア 1 個
 - バックアップレジスタに格納され、読出し／書込みアクセスに対して保護されたセキュアな AES 用のブートハードウェアキー
- タンパ設定とバックアップレジスタ特権保護
- 単調カウンタ

47.3 TAMP の機能説明

47.3.1 TAMP ブロック図

図 618. TAMP のブロック図



1. 外部タンパおよび内部タンパの数は製品によります。

47.3.2 TAMP ピンおよび内部信号

表 487. TAMP の入出力ピン

ピン名	信号タイプ	説明
TAMP_INx (x = ピンインデックス)	入力	タンパ入力ピン
TAMP_OUTx (x = ピンインデックス)	出力	タンパ出力ピン (アクティブモードのみ)

表 488. TAMP 内部入力／出力信号

内部信号名	信号タイプ	説明
tamp_ker_ck	入力	TAMP カーネルクロック (rtc_ker_ck に接続、本書では RTCCLK とも言います)
tamp_pclk	入力	TAMP APB クロック (rtc_pclk に接続)
tamp_itamp[y] (y = 信号インデックス)	入力	内部タンパイベントソース
tamp_tzen	入力	TAMP TrustZone 有効
tamp_evt	出力	タンパイベント検出フラグ (内部または外部タンパ)。確認済みモード設定と潜在モード設定の両方。
tamp_potential	出力	潜在タンパ検出信号。デバイス機密情報 ⁽¹⁾ の保護のために使用されます。 この信号は次の場合にアクティブです。 – タンパイベント検出フラグ (内部または外部タンパ) が潜在モードで生成されている場合 – BKBLOCK に 1 を書き込んでソフトウェアリクエストを行った場合
tamp_confirmed	出力	確認済みタンパ検出信号。デバイス機密情報 ⁽¹⁾ の保護のために使用されます。 この信号は次の場合にアクティブです。 – タンパイベント検出フラグ (内部または外部タンパ) が確認済みモードで生成されている場合 – BKERASE に 1 を書き込んでソフトウェアリクエストを行った場合
tamp_potential_rpcfgz (z = 信号インデックス)	出力	潜在タンパ検出信号は RPCFGz = 1 の場合のみ生成されます。 この信号は次の場合にアクティブです。 – タンパイベント検出フラグ (内部または外部タンパ) が潜在モードで生成されている場合 – BKBLOCK に 1 を書き込んでソフトウェアリクエストを行った場合
tamp_confirmed_rpcfgz (z = 信号インデックス)	出力	確認済みタンパ検出信号は RPCFGz = 1 の場合のみ生成されます。 この信号は次の場合にアクティブです。 – タンパイベント検出フラグ (内部または外部タンパ) が確認済みモードで生成されている場合 – BKERASE に 1 を書き込んでソフトウェアリクエストを行った場合

表 488. TAMP 内部入力／出力信号 (続き)

内部信号名	信号タイプ	説明
tamp_it	出力	TAMP 割込み (詳細については、 セクション 47.5 : TAMP 割込み を参照してください)
tamp_trg[x] (x = 信号インデックス)	出力	タンパ検出トリガ
tamp_bhk	出力	タンパブートハードウェアキーバス

1. [表 489 : TAMP 相互接続](#)を参照してください。

通常、TAMP カーネルクロックは、RCC で他のクロックソースを選択できますが、32.768 kHz での LSE です (詳細は RCC を参照してください)。選択したクロックによって、一部の低消費電力モードや V_{BAT} で使用できない検出モードがあります (詳細は、[セクション 47.4 : TAMP 低消費電力モード](#)を参照してください)。

表 489. TAMP 相互接続

信号名	転送元／転送先
tamp_tzen	Flash オプションバイトから : TZEN
tamp_evt	タイムスタンプイベントの生成に使用される rtc_tamp_evt
tamp_potential	<p>tamp_potential 信号は、以下に示すデバイス機密情報への読み出し/書き込みアクセスをブロックするために使用されます。</p> <ul style="list-style-type: none"> バックアップレジスタ SRAM2 <p>システム Flash メモリの RHUK (ルート・ハードウェア・ユニーク・キー) と SAES への BHK (ブート・ハードウェア・キー) ハードウェアバスはブロックされます。</p> <p>tamp_potential 信号は、以下に示すデバイス機密情報を消去するために使用されます。</p> <ul style="list-style-type: none"> ICACHE の内容 SAES、AES、HASH ペリフェラル PKA SRAM <p>消去の実行中は、デバイス機密情報のアクセスがブロックされます。</p>
tamp_confirmed	<p>tamp_confirmed 信号は、以下に示すデバイス機密情報を消去するために使用されます。</p> <ul style="list-style-type: none"> バックアップレジスタ SRAM2 ICACHE/DCACHE の内容 OTFDEC キーおよび CRC レジスタ SAES、AES、HASH ペリフェラル PKA SRAM <p>消去の実行中は、デバイス機密情報のアクセスがブロックされます。</p> <p>SAES へのシステム Flash メモリの RHUK (ルート・ハードウェア・ユニーク・キー) ハードウェアバスはブロックされます。</p>
tamp_potential_rpcfg0	<p>TAMP_RPCFGR でビット RPCFG0 がセットされている場合、以下に示すデバイス機密情報への読み出し/書き込みアクセスをブロックするために、tamp_potential_rpcfg0 信号が使用されます。</p> <ul style="list-style-type: none"> バックアップ SRAM

表 489. TAMP 相互接続 (続き)

信号名	転送元／転送先
tamp_confirmed_rpcfg0	TAMP_RPCFGR でビット RPCFG0 がセットされている場合、以下に示すデバイス機密情報を消去するために、tamp_confirmed_rpcfg0 信号が使用されます。 – バックアップ SRAM 消去の実行中は、デバイス機密情報のアクセスがブロックされます。
tamp_itamp1	バックアップドメイン電圧閾値監視 ⁽¹⁾
tamp_itamp2	温度監視 ⁽¹⁾
tamp_itamp3	LSE 監視 (LSECSS) ⁽²⁾
tamp_itamp4	HSE 監視 (rcc_hsecss_fail)
tamp_itamp5	RTC カレンダーオーバーフロー (rtc_calovf)
tamp_itamp6	NVSTATE ≠ OPEN の場合の JTAG/SWD アクセス
tamp_itamp7	ADC2 ウォッチドッグ監視 1
tamp_itamp8 ⁽³⁾	単調カウンタ 1 オーバーフロー
tamp_itamp9	暗号化ペリフェラル・フォールト (SAES または AES または PKA または TRNG)
tamp_itamp11	タンバフラグ (潜在タンバタイムアウト) のセット時、IWDG のリセット
tamp_itamp12	ADC2 アナログウォッチドッグ監視 2
tamp_itamp13	ADC2 アナログウォッチドッグ監視 3
tamp_itamp15	システム・フォールト
tamp_bhk	saes_bhk。このバスは、セキュアな AES コプロセッサにブート・ハードウェア・キーをロードするために使用されます。

1. この監視を有効にするには、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) で MONEN をセットする必要があります。
2. この監視を有効にするには、[PWR バックアップドメイン制御レジスタ \(PWR_BDCR\)](#) で LSEON をセットする必要があります。
3. この信号は TAMP ペリフェラルで生成されます。

TZEN オプションビットは、デバイスで TrustZone をアクティブにするために使用します。

TZEN = 1 : TrustZone は有効です。

TZEN = 0 : TrustZone は無効です。

TrustZone が無効である場合、TAMP レジスタへの APB アクセスは非セキュアとなります。

47.3.3 RTC および TAMP によって制御される GPIO

[セクション 46.3.3 : RTC および TAMP によって制御される GPIO](#) を参照してください。

47.3.4 TAMP レジスタ書込み保護

システムリセット後、電源制御ペリフェラルの DBP ビットによって、TAMP レジスタ (バックアップレジスタを含む) は不要な書込みアクセスから保護されます (PWR 電源制御のセクションを参照)。TAMP レジスタ書込みアクセスを可能にするには、DBP ビットをセットする必要があります。

47.3.5 TAMP セキュア保護モード

デフォルトでは、バックアップドメインのパワーオンリセット後、セキュアモードでのみ書き込み可能な TAMP セキュア設定レジスタ (TAMP_SECCFGR) を除いて、すべての TAMP レジスタがセキュアモードと非セキュアモードの両方で読出しまたは書き込みできます。TAMP 保護設定は、システムリセットの影響を受けません。

- TAMP_SECCFGR レジスタの TAMPSEC ビットがセットされている場合：
 - TAMP レジスタの書き込みは、独自の保護設定を持つバックアップレジスタを除き、セキュアモードでのみ可能です。
 - TAMP_SECCFGR、TAMP_PRIVCFGR および TAMP_MISR の読出しは、セキュアモードと非セキュアモードで常に可能です。その他のすべての TAMP レジスタは、独自の保護設定を持つバックアップレジスタと単調カウンタを除き、セキュアモードでのみ読出しできます。
- TAMP_SECCFGR レジスタに CNT1SEC ビットがセットされている場合：TAMP_COUNT1R はセキュアモードでのみ読み書きできます。

セキュア保護されたレジスタへの非セキュアアクセスは拒否されます。

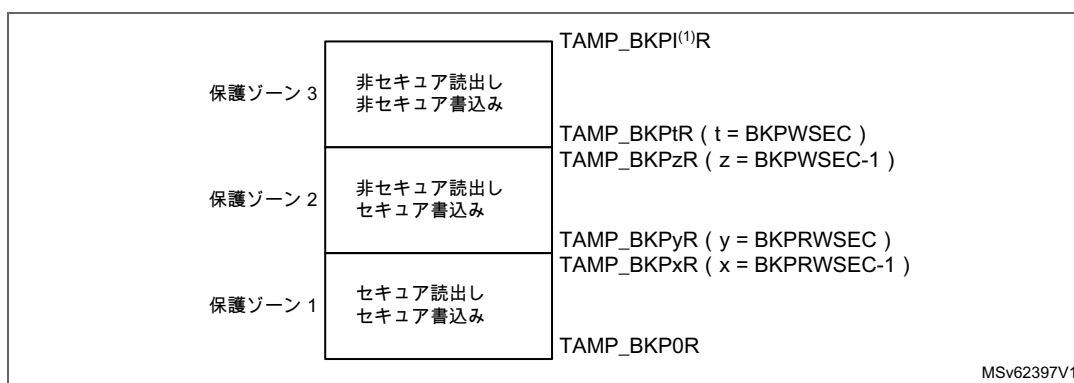
- 生成されるバスエラーはありません。
- TZIC (TrustZone 不正アクセスコントローラ)でのフラグ/割込みを通じて通知が生成されます。
- 書き込み保護されている場合、そのビットは書き込まれません。
- 読出し保護されている場合、そのビットは 0 として読み出されます。

1 つ以上の機能がセキュアに設定されると、TAMP リセットおよびクロック制御も RCC でセキュアになります。

47.3.6 バックアップレジスタ保護ゾーン

バックアップレジスタの保護は、BKPRWSEC[7:0] および BKPWSEC[7:0] によって設定されます (下図を参照してください)。

図 619. バックアップレジスタ保護ゾーン



1. l = 最後のバックアップレジスタインデックス

TZEN =1 の場合、TAMP_PRIVCFGR の BKPWPRIV および BKPRWPRIV のビットは、セキュアモードでのみ書き込みできます。

47.3.7 TAMP 特権保護モード

デフォルトでは、バックアップドメインのパワーオンリセット後、特権モードのみで書き込み可能な TAMP 特権設定レジスタ (TAMP_PRIVCFGR) を除いて、すべての TAMP レジスタが特権と非特権モードの両方で読出しまたは書き込みできます。TAMP 保護設定は、システムリセットの影響を受けません。

TAMP_PRIVCFGR レジスタの TAMPPRIV ビットがセットされている場合：

- TAMP レジスタの書き込みは、独自の保護設定を持つバックアップレジスタと単調カウンタを除き、特権モードでのみ可能です。
- TAMP_PRIVCFGR レジスタに CNT1PRIV ビットがセットされている場合：TAMP_COUNT1R は特権モードでのみ読み書きできます。
- TAMP_SECCFGR、TAMP_PRIVCFGR の読出しは、特権と非特権モードで常に可能です。その他のすべての TAMP レジスタは、独自の保護設定を持つバックアップレジスタと単調カウンタを除き、特権モードでのみ読出しできます。

バックアップレジスタの保護は、保護ゾーン 1 では BKPRWSEC[7:0] および BKPRWPRIV、保護ゾーン 2 では BKPRWSEC[7:0]、BKPWSEC[7:0] および BKPWPRIV によって設定されます (図 619 を参照してください)。BHKLOCK ビットは、BKPRWPRIV ビットがセットされている場合に、特権モードでのみ書き込みできます。

特権保護されたレジスタへの非特権アクセスは拒否されます。

- 生成されるバスエラーはありません。
- 書き込み保護されている場合、そのビットは書き込まれません。
- 読出し保護されている場合、そのビットは 0 として読み出されます。

47.3.8 ブート・ハードウェア・キー (BHK)

TAMP_BKP0R から TAMP_BKP7R までの最初の 8 個のバックアップレジスタを使用して、セキュアな AES のためのブート・ハードウェア・キーを格納できます。

そのためには、これらのレジスタは保護ゾーン 1 に属していなければなりません。BKPRWSEC は 8 以上でなければなりません。

ブート・ハードウェア・キーによるバックアップレジスタの書き込みが完了したら、TAMP_SECCFGR レジスタで BHKLOCK ビットをセットする必要があります。BHKLOCK のセットが完了すると、ソフトウェアから 8 個のバックアップレジスタへのアクセスはできなくなります。0 として読み出され、これらのレジスタへの書き込みは無視されます。BHKLOCK はソフトウェアではクリアできず、タンパイベントの後で、または読出し保護 (RDP) の無効化時にハードウェアによってクリアされます。また、BKERASE コマンドでもクリアされます (いずれの場合も、バックアップレジスタも消去されます)。

セキュアな AES コプロセッサ (SAES) にブート・ハードウェア・キーをダウンロードする手順については、SAES のセクションを参照してください。

47.3.9 タンパ検出

タンパ検出の主な目的は、デバイスに対する外部攻撃からデバイス機密情報を保護することです。検出は、TAMP_INx (x = ピンインデックス) I/O、または範囲外のデバイス状態を検出する内部モニターでのイベントに対して行います。

タンパ検出は次の目的で設定できます。

- バックアップレジスタ、および表 489 : TAMP 相互接続に示した SRAM またはペリフェラルに格納されたその他のデバイス機密情報を消去。デバイス機密情報のリストは、TAMP リソース保護設定レジスタ (TAMP_RPCFGR) によって設定できます。
- バックアップレジスタ、および表 489 : TAMP 相互接続に示した SRAM またはペリフェラルに格納されたその他のデバイス機密情報への読み出し/書き込みアクセスをブロック。デバイス機密情報のリストは、TAMP_RPCFGR によって設定できます。
- 低消費電力モードからのウェイクアップのための割込みを生成
- 低消費電力タイマのためのハードウェアトリガ、または RTC タイムスタンプイベントを生成

外部 I/O タンパ検出サポートは次の 2 種類のメイン設定に対応しています。

- パッシブモード : TAMP_INx I/O が監視され、タンパはエッジまたはレベルで検出されます。
- アクティブモード : TAMP_INx (x = ピンインデックス) が TAMP_OUTy (y = ピンインデックス) と継続的に比較され、オープン/ショートが検出されます。

誤検出を防止するため、外部タンパ検出にデジタルフィルタを適用できます。また、潜在モードで各タンパソースを設定し、タンパ検出時にハードウェアによって機密情報の消去が開始されないようにすることができます。機密情報の消去は、ソフトウェアチェック後に、ソフトウェアによって開始することができます。

47.3.10 TAMP バックアップレジスタとその他のデバイス機密情報の消去

バックアップレジスタ (TAMP_BKPxR) は、システムリセットや STANDBY モードからのウェイクアップではリセットされません。

バックアップレジスタとその他のデバイス機密情報は、対応するマスクが設定されている (TAMP_CR2 レジスタで TAMPxMSK=1) 場合はリセットされません。

注 : バックアップレジスタは、Flash の読み出し保護がレベル 1 からレベル 0 に変わったときにも消去されます。

タンパ検出 – 確認済みモード

確認済みモードは、TAMP_CR2 レジスタで TAMPxPOM = 0 がセットされているときに、TAMPx (外部タンパ x) のために選択されます。確認済みモードは、TAMP_CR3 レジスタで TAMPxPOM = 0 がセットされているときに、ITAMPx (内部タンパ x) のために選択されます。確認済みモードでのタンパ検出の影響は、表 489 : TAMP 相互接続の tamp_confirmed および tamp_confirmed_rpcfgx 信号の項を参照してください。

このモードは、タンパの検出時にデバイス機密情報を自動的に消去するために選択されます。

タンパ検出 – 潜在モード

潜在モードは、TAMP_CR2 レジスタで TAMPxPOM = 1 がセットされているときに、TAMPx (外部タンパ x) のために選択されます。潜在タンパモードは、TAMP_CR3 レジスタで ITAMPxPOM = 1 がセットされているときに、ITAMPx (内部タンパ x) のために選択されます。潜在モードでのタンパ検出の影響は、表 489 : TAMP 相互接続の tamp_potential および tamp_potential_rpcfgx 信号の項を参照してください。

このモードを選択すると、タンパの検出時に一部のデバイス機密情報が完全に消去されて復旧できなくなるのを防止できます。このモードでは、該当するタンパイベントの検出時に、一部のデバイス機密情報は消去されません。また、潜在モードでタンパ検出フラグがセットさせるとすぐに、TAMP_SCR レジスタで該当するクリアフラグを設定することでこのフラグがクリアされるまで、これらのデバイス機密情報への読み出し/書き込みアクセスはブロックされます。そのため、真のタンパと誤検出を区別するためのチェックが行われ、潜在タンパが真のタンパであると確認された場合に、機密情報の消去が

開始されます。TAMP_CR2 の BKERASE ビットをセットすると、ソフトウェアによりデバイス機密情報が消去されます。

潜在タンパから確認済みタンパのタイムアウトまで

一部の内部タンパでは、別のタンパフラグがセットされているときに独立したウォッチドッグリセットが発生すると、タンパイベントが生成されます（表 489 : TAMP 相互接続を参照）。ITAMPxPOM = 0 により、IWDG タンパを設定する必要があります。これにより、以前のタンパイベントが潜在モードであった場合に、タイムアウト後にハードウェアによって強制的にデバイス機密情報が消去されます。潜在タンパイベントに続いてソフトウェア決定の前にウォッチドッグリセットが発生する場合、これは「潜在タンパ」から「確認済みタンパ」への移行に相当します。

デバイスリソース保護設定

デバイスリソースを設定して、タンパ検出によって保護されるデバイス機密事項のリストに組み込むことができます。

TAMP_RPCFGR に RPCFGz = 0 がセットされている場合、RPCFGz に関連付けられたデバイスリソースは TAMP ペリフェラルによって保護されません。

- タンパ検出の影響を受けません（確認済みモードでも潜在モードでも）。
- BKERASE ソフトウェアコマンドの影響を受けません。
- BKBLOCK ソフトウェアコマンドの影響を受けません。

TAMP_RPCFGR に RPCFGz = 1 がセットされている場合、RPCFGz に関連付けられたデバイスリソースは TAMP ペリフェラルによって保護されます。

- 表 489 : TAMP 相互接続の tamp_confirmed_rpcfgz 信号の項に示すように、確認済みタンパ検出と BKERASE ソフトウェアコマンドの影響を受けます。
- 表 489 : TAMP 相互接続の tamp_potential_rpcfgz 信号の項に示すように、潜在タンパ検出と BKBLOCK ソフトウェアコマンドの影響を受けます。

表 490. デバイスリソース x のタンパ保護

-	潜在タンパまたは BKBLOCK	確認済みタンパまたは BKERASE
RPCFGx = 0	デバイスリソース x への影響なし	デバイスリソース x への影響なし
RPCFGx = 1	tamp_potential_rpcfgx の項に示すように、デバイス機密情報 x を保護 ⁽¹⁾	tamp_confirmed_rpcfgx ⁽¹⁾ の項に示すように、デバイス機密情報 x を保護

1. 表 489 : TAMP 相互接続を参照してください。

ソフトウェアによりデバイス機密情報へのアクセスをブロック

デフォルトでは、タンパイベントフラグが検出されなければ、アプリケーションによってデバイス機密情報にアクセスできます。タンパフラグがセットされている場合は、デバイス機密情報にはアクセスできません。

ソフトウェアによりデバイス機密情報へのアクセスをブロックするには、TAMP_CR2 レジスタの BKBLOCK ビットをセットします。デバイス機密情報にアクセスできるのは、BKBLOCK = 0 で、タンパフラグがセットされていない場合のみです。

47.3.11 タンパ検出設定と初期化

各入力は、TAMP_CR レジスタにおいて該当する TAMPxE ビットを 1 にセットすることにより有効にできます。

各 TAMP_INx タンパ検出入力、TAMP_SR レジスタのフラグ TAMPxF に関連付けられています。

TAMP_IER レジスタの TAMPxIE ビットをセットすることにより、タンパ検出イベント発生時に割込みが生成されます (TAMPxF がセットされている場合)。該当する TAMPxMSK がセットされている場合、TAMPxIE をセットすることはできません。

タンパイイベント時のトリガ出力生成

タンパイイベント検出は、低消費電力タイマによるトリガ入力として使用できます。

TAMP_CR レジスタで TAMPxMSK ビットがクリアされている場合、同じピンで新しいタンパ検出ができるようにするには、ソフトウェアによって TAMPxF フラグをクリアする必要があります。

TAMPxMSK ビットがセットされている場合、TAMPxF フラグはマスクされ、TAMP_SR レジスタでクリアされたままとなります。この設定を使用することで、システムのウェイクアップにより TAMPxF をクリアしなくても、STOP モードで低消費電力タイマを自動的にトリガできます。この場合、バックアップレジスタはクリアされません。

この機能は、タンパが **タンパ入力でのフィルタを使ったレベル検出 (パッシブモード)** モードで設定された場合にのみ利用可能です (TAMPFLT ≠ 00、およびアクティブモードが非選択の場合)。

タンパイイベント時のタイムスタンプ

RTC_CR で TAMPTS を 1 にセットすると、すべての内部または外部タンパイイベントがタイムスタンプを発生させるようになります。タンパイイベントによりタイムスタンプが発生する場合、通常のタイムスタンプイベント発生時と同様に TSF ビットまたは TSOVF ビットが RTC_SR でセットされます。

注： TSF は、TAMP x F フラグから最大で ck_apre の 3 サイクル後にセットされます。RTCCLK が停止した場合、TSF はセットされません (RTCCLK の再起動時にセットされます)。

注： TSF の立上りの前に TAMPxF がクリアされた場合、TSF はセットされません。結果的に、TAMPTS = 1 の場合、ソフトウェアではタンパフラグをクリアする前にタイムスタンプフラグを待つか、TAMP 割込みルーチンで RTC カウンタ値を読み出す必要があります。

タンパ入力でのエッジ検出 (パッシブモード)

TAMPFLT ビットが 00 の場合、該当する TAMPxTRG ビットに応じて立上りエッジまたは立下りエッジが観測されると、TAMP_INx ピンがタンパ検出イベントを生成します。エッジ検出を選択すると、TAMP_INx 入力の内部プルアップ抵抗が無効になります。

注意： TAMPFLT = 00 および TAMPxTRG = 0 (立上りエッジ検出) のとき、タンパ入力タンパ検出を有効にする前にすでにハイレベルになっている場合、タンパイイベントがハードウェアによって検出される場合があります。

タンパイイベントが検出されクリアされた後に、バックアップレジスタ (TAMP_BKPxR) を再プログラムする場合には、事前に、TAMP_INx を無効にしてから再度有効にする (TAMPxE を 1 にセット) 必要があります。これによって、TAMP_INx の入力値がタンパ検出を示している期間中に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、TAMP_INx の入力でのレベル検出に相当します。

注： タンパ検出は、V_{DD} 電源がオフのときでも有効です。バックアップレジスタの不必要なリセットを避けるには、TAMPx が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。

タンパ入力でのフィルタを使ったレベル検出 (パッシブモード)

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、(TAMPFLT に応じて) 2、4 または 8 回のいずれかの連続したサンプルが TAMPxTRG ビットで指定するレベルで観測されたときに生成されます。

TAMP_INx 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、TAMP_INx 入力により大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

注： プルアップ抵抗の電気的特性については、マイクロコントローラデータシートを参照してください。

アクティブタンパの検出

TAMPxAM ビットが TAMP_ATCR でセットされている場合、タンパイベントはアクティブモードに設定され、これは TAMP_OUTy ピンと TAMP_INx ピンの比較に基づきます。デフォルト (ATOSHARE = 0) では、TAMP_INx と TAMP_OUTx ($y = x$) が比較されます。ATOSHARE ビットがセットされている場合、同じ出力を複数のタンパ入力に使用できます。TAMP_OUTy 機能は、ATOSHARE および ATOSELx ビットにより、アクティブなタンパ入力 TAMP_INx ($\text{TAMPxEN} = \text{TAMPxAM} = 1$) との比較のために選択されるとすぐに、I/O で有効になります。TAMP_ATCRx ($x = 1, 2$) レジスタの ATOSHARE ビットおよび ATOSEL ビットの説明を参照してください。

2 CK_ATPER サイクルごと ($\text{CK_ATPER} = 2^{\text{ATPER}} \times \text{CK_ATPRE}$) に、TAMP_OUTy 出力ピンは疑似乱数発生器 (PRNG) によって生成される値を出力します。この値を出力した後、TAMP_OUTy ピンは、1 CK_ATPER サイクル後に逆の値を出力します。

表 491. アクティブタンパ出力の変更周期

ATCKSEL[3:0]	CK_ATPRE 周波数	ATPER[2:0]	タンパ出力の変更 (CK_ATPER) 周波数	タンパ出力の変更周期 ⁽¹⁾ (ms)
0x0	f_{RTCCLK}	0x0	f_{RTCCLK}	0.030
		0x1	$f_{\text{RTCCLK}}/2$	0.061
		0x2	$f_{\text{RTCCLK}}/4$	0.122
		0x3	$f_{\text{RTCCLK}}/8$	0.244
		0x4	$f_{\text{RTCCLK}}/16$	0.488
		0x5	$f_{\text{RTCCLK}}/32$	0.977
		0x6	$f_{\text{RTCCLK}}/64$	1.953
		0x7	$f_{\text{RTCCLK}}/128$	3.906
...
0x7	$f_{\text{RTCCLK}}/128$	0x0	$f_{\text{RTCCLK}}/128$	3.906
		0x1	$f_{\text{RTCCLK}}/256$	7.8125
		0x2	$f_{\text{RTCCLK}}/512$	15.625
		0x3	$f_{\text{RTCCLK}}/1024$	31.250
		0x4	$f_{\text{RTCCLK}}/2048$	62.5
		0x5	$f_{\text{RTCCLK}}/4096$	125
		0x6	$f_{\text{RTCCLK}}/8192$	250
		0x7	$f_{\text{RTCCLK}}/16384$	500

表 491. アクティブタンパ出力の変更周期 (続き)

ATCKSEL[3:0]	CK_ATPRE 周波数	ATPER[2:0]	タンパ出力の変更 (CK_ATPER) 周波数	タンパ出力の変更周期 ⁽¹⁾ (ms)
0xB	$f_{\text{RTCCLK}}/2048^{(2)}$	0x0	$f_{\text{RTCCLK}}/2048$	62.5
		0x1	$f_{\text{RTCCLK}}/4096$	125
		0x2	$f_{\text{RTCCLK}}/8192$	250
		0x3	$f_{\text{RTCCLK}}/16384$	500
		0x4	$f_{\text{RTCCLK}}/32768$	1000
		0x5	$f_{\text{RTCCLK}}/65536$	2000
		0x6	$f_{\text{RTCCLK}}/131072$	4000
		0x7	$f_{\text{RTCCLK}}/262144$	8000

1. $f_{\text{RTCCLK}} = 32768$ Hz と仮定します。

2. この設定では、 $(\text{PREDIV_A}+1) = 128$ かつ $(\text{PREDIV_S}+1)$ が 16 の倍数であることが前提となります。

PRNG は、選択したタンパ出力によってその数に応じた異なる周波数で使用されます。選択した出力の数は、TAMPxAM、TAMPxE、ATOSEL、ATOSHARE によります。

- 出力が 1 つだけ選択されている場合：PRNG は CK_ATPER 16 周期ごとに使用されます。
- 出力が 2 つ選択されている場合：PRNG は CK_ATPER 8 周期ごとに使用されます。
- 出力が 3 つまたは 4 つ選択されている場合：PRNG は CK_ATPER 4 周期ごとに使用されます。
- 出力が 5 つ以上選択されている場合：PRNG は CK_ATPER 2 周期ごとに使用されます。

PRNG は新しい値の出力に最低 9 CK_ATPRE サイクルを必要とします。そのため、正しく機能するための最低 ATPER 値を以下の表に示します。

表 492. 最低 ATPER 値

選択した出力数	最低 ATPER
1	0
2	1
3 または 4	2
5 以上	3

TAMP_INx ピンは TAMP_OUTy ピンに外部接続されます。CK_ATPRE サイクルごとに、TAMP_OUTy 出力値と TAMP_INx 受信値が比較されます。比較して不一致となった場合、TAMPxF ビットが TAMP_SR レジスタでセットされます。

たとえば、アクティブモードで TAMP1 および TAMP2 の両方を設定して有効にすることで、TAMP_OUT1 を TAMP_IN1 と TAMP_IN2 との比較に使用できます (ATOSHARE = 1、ATOSEL1 = 000、ATOSEL2 = 000)。

アクティブタンパは、FLTEN = 1 の場合に入力フィルタと組み合わせることができます。この場合、連続する 4 つの比較サンプルで 2 つの比較が偽となった場合のみタンパが検出されます。

図 620. アクティブタンパフィルタ

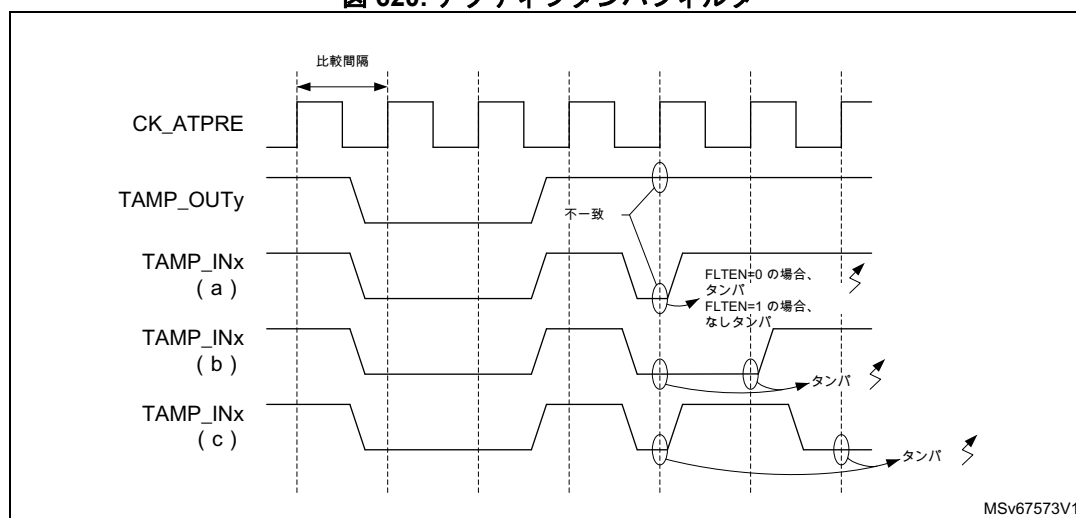


図 620 に示すように、FLTEN = 0 の場合、TAMP_OUTy 出力と関連する TAMP_INx 入力間に不一致があると、後者のサンプリング時にタンパが生成されます。これが 3 つの例 (a)、(b)、(c) に示した状態です。

FLTEN = 1 の場合、例 (a) では連続する 4 回の比較で不一致が 1 回しか検出されないため、タンパが生成されません。例 (b) では、連続する 2 回の不一致が検出されているため、タンパが生成されます。例 (c) では、連続するサンプルで不一致は発生していなくても、連続する 4 回の比較で 2 回の不一致が発生しているため、ここでもタンパが生成されます。

FLTEN = 1 をセットすると、TAMP_ATCR1 レジスタの ATCKSEL フィールドでプログラムされる CK_ATPRE の 1 周期よりも短い単一パルスを見逃すことにより、TAMP_INx 入力でのグリッチ、バウンス、または一時的な状態による望ましくないタンパの検出を回避できます。 $f_{\text{RTCCLK}} = 32.768 \text{ kHz}$ と仮定して、設定可能な各 ATCKSEL について、最小フィルタパルス幅を表 493 に示します。

表 493. アクティブタンパのフィルタパルス時間

ATCKSEL[3:0]	CK_ATPRE 周波数	最小フィルタパルス幅 (ms)
0x0	f_{RTCCLK}	0.030
0x1	$f_{\text{RTCCLK}}/2$	0.061
0x2	$f_{\text{RTCCLK}}/4$	0.122
0x3	$f_{\text{RTCCLK}}/8$	0.244
0x4	$f_{\text{RTCCLK}}/16$	0.488
0x5	$f_{\text{RTCCLK}}/32$	0.977
0x6	$f_{\text{RTCCLK}}/64$	1.953
0x7	$f_{\text{RTCCLK}}/128$	3.906
0xB	$f_{\text{RTCCLK}}/2048$	62.500 ⁽¹⁾

1. この設定では、 $(\text{PREDIV_A}+1) = 128$ かつ $(\text{PREDIV_S}+1)$ が 16 の倍数であることが前提となります。

注： 複数のパルスが CK_ATPRE の 1 周期分より短い場合でも、連続する 4 回の比較で 2 回の不一致が発生した場合、タンパが生じることがあります。

注意 : ATCKSEL[3] = 1 の場合、RTC 初期化モードが開始されると、CK_ATPRE および CK_ATPER クロックが停止します。そのため、INIT モードが終了するまで、TAMP_OUTy ピンは停止します。

セクション : カレンダの初期化と設定を参照してください。

ATCKSEL[3] = 1 と一緒に RTC バイナリモードを使用する場合は、RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)、RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)、RTC アラーム A バイナリモードレジスタ (RTC_ALRABINR)、RTC アラーム B バイナリモードレジスタ (RTC_ALRBBINR) も参照してください。

注意 : 注 : ATCKSEL[3] = 1 の場合、カレンダーオーバーフローではアクティブタンパ検出は機能しません。タンパ保護を行うには、カレンダーオーバーフローで内部タンパ 5 を有効化する必要があります。

疑似乱数発生器は、シードで初期化する必要があります。これは TAMP_ATSEEDR レジスタに連続して 4 つの 32 ビット乱数値を書き込むことによって行います。シードをプログラムすると、自動的に PRNG に送信されます。新しいシードが転送され、PRNG によって処理されると、SEEDF ビットが TAMP_ATOR でセットされ、TAMP APB クロックをオフにできません。処理時間は、4 つ目のシードが書き込まれてから最大 184 APB クロックサイクルです。そのため、ユーザは新しいシードを書き込んでから、SEEDF がクリアされるのを待ち、低消費電力モードに移行する必要があります。

アクティブタンパ出力は、1 つ目のシードが書き込まれて処理を完了した後でのみアクティブになります。その後、アクティブタンパ動作中は、新しいシードを書き込んで処理できます。

アクティブタンパの初期化

ここに、システムリセット後にソフトウェアでアクティブタンパを初期化する手順を示します。

TAMP_ATOR レジスタの INITS を読み出します。

- INITS = 0x0 (初期化されていない場合) :
 - a) TAMP_ATCR を書き込んでアクティブタンパクロック、フィルタ、出力共有 (ある場合)、アクティブモードを設定します。
 - b) TAMP_CR1 を書き込んでタンパを有効にします (すべての必要なタンパを同じ書き込み処理で有効にする必要があります)。
 - c) TAMP_ATSEEDR に SEED を 4 回書き込みます。
 - d) TAMP_ATOR で SEEDF = 0 になるまで待ちます。これでバックアップレジスタがアクティブタンパで保護されます。
- INITS = 0x1 (すでに初期化されている場合) :

初期化しません。ランダム性を高めるには、新しい SEED を定期的に供給する必要があります。新しい SEED が供給されると、SEEDF = 0 になるまで待ってから、TAMP APB クロックをオフに切り替える低消費電力モードに移行します。
- タンパをソフトウェアで無効にして、後で再び有効にする場合、タンパを有効にした後で SEED を書き込む必要があります。
 - a) TAMP_CR1 を書き込んでタンパを有効にします (すべての必要なタンパを同じ書き込み処理で有効にする必要があります)。
 - b) TAMP_ATSEEDR に SEED を 4 回書き込みます。
 - c) TAMP_ATOR で SEEDF = 0 になるまで待ちます。これでバックアップレジスタがアクティブタンパで保護されます。

47.4 TAMP 低消費電力モード

表 494. 低消費電力モードが TAMP に与える影響

モード	説明
SLEEP	影響はありません。 TAMP 割込みによって、デバイスは SLEEP モードから復帰します。
STOP	クロックソースが LSE または LSI である場合のみアクティブなままとなるフィルタを使ったレベル検出とアクティブタンパモードを除き、すべての機能に影響はありません。 タンパイイベントによって、デバイスは STOP モードを終了します。
STANDBY	クロックソースが LSE または LSI である場合のみアクティブなままとなるフィルタを使ったレベル検出とアクティブタンパモードを除き、すべての機能に影響はありません。 タンパイイベントによって、デバイスは STANDBY モードを終了します。

表 495. TAMP ピン機能 (モード共通)

ピン名	すべての低消費電力モードで機能するか	V _{BAT} モードで使用可能な機能
TAMP_IN[8:1]	可能 ⁽¹⁾	あり ⁽¹⁾
TAMP_OUT[8:1]	可能 ⁽²⁾	可能 ⁽³⁾

1. STANDBY および V_{BAT} モードでは PC13、PI8、PA0、PA1、PA2 のみが機能します。
2. STANDBY モードでは PC13、PA1、PI8 のみが機能します。
3. V_{BAT} モードでは PC13、PA1、PI8 のみが機能します。

47.5 TAMP 割込み

割込みチャネルは、マスク済み割込みステータスレジスタ、またはセキュリティモード設定に応じたセキュアマスク済み割込みステータスレジスタで設定されます（TAMPSEC）。

表 496. 割込みリクエスト

項目（割込みの略称）	割込みイベント	イベントフラグ ⁽¹⁾	有効制御ビット	割込みのクリア方法	低消費電力モードの終了
TAMP	タンパ x ⁽²⁾	TAMPxF	TAMPxIE	CTAMPxF に 1 を書き込む	可能 ⁽³⁾
	内部タンパ y ⁽²⁾	ITAMPyF	ITAMPyIE	CITAMPyF に 1 を書き込む	あり ⁽³⁾

- イベントフラグは、TAMP_SR レジスタ内です。
- タンパおよび内部タンパイベントの数は製品により異なります。
- 低消費電力モードで利用可能な機能については、表 494：低消費電力モードが TAMP に与える影響を参照してください。

47.6 TAMP レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの[セクション 1.2](#)を参照してください。ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスすることができます。

47.6.1 TAMP 制御レジスタ 1（TAMP_CR1）

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5：TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7：TAMP 特権保護モード](#)を参照してください。

アドレスオフセット：0x00

バックアップドメインリセット値：0x0000 0000

システムリセット：影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ITAMP15E	Res.	ITAMP13E	ITAMP12E	ITAMP11E	Res.	ITAMP9E	ITAMP8E	ITAMP7E	ITAMP6E	ITAMP5E	ITAMP4E	ITAMP3E	ITAMP2E	ITAMP1E
	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8E	TAMP7E	TAMP6E	TAMP5E	TAMP4E	TAMP3E	TAMP2E	TAMP1E
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ITAMP15E**：内部タンパ 15 イネーブル

0：内部タンパ 15 は無効です。

1：内部タンパ 15 は有効です。

ビット 29 予約済みであり、リセット値に保持する必要があります。

- ビット 28 **ITAMP13E** : 内部タンパ 13 イネーブル
0 : 内部タンパ 13 は無効です。
1 : 内部タンパ 13 は有効です。
- ビット 27 **ITAMP12E** : 内部タンパ 12 イネーブル
0 : 内部タンパ 12 は無効です。
1 : 内部タンパ 12 は有効です。
- ビット 26 **ITAMP11E** : 内部タンパ 11 イネーブル
0 : 内部タンパ 11 は無効です。
1 : 内部タンパ 11 は有効です。
- ビット 25 予約済みであり、リセット値に保持する必要があります。
- ビット 24 **ITAMP9E** : 内部タンパ 9 イネーブル
0 : 内部タンパ 9 は無効です。
1 : 内部タンパ 9 は有効です。
- ビット 23 **ITAMP8E** : 内部タンパ 8 イネーブル
0 : 内部タンパ 8 は無効です。
1 : 内部タンパ 8 は有効です。
- ビット 22 **ITAMP7E** : 内部タンパ 7 イネーブル
0 : 内部タンパ 7 は無効です。
1 : 内部タンパ 7 は有効です。
- ビット 21 **ITAMP6E** : 内部タンパ 6 イネーブル
0 : 内部タンパ 6 は無効です。
1 : 内部タンパ 6 は有効です。
- ビット 20 **ITAMP5E** : 内部タンパ 5 イネーブル
0 : 内部タンパ 5 は無効です。
1 : 内部タンパ 5 は有効です。
- ビット 19 **ITAMP4E** : 内部タンパ 4 イネーブル
0 : 内部タンパ 4 は無効です。
1 : 内部タンパ 4 は有効です。
- ビット 18 **ITAMP3E** : 内部タンパ 3 イネーブル
0 : 内部タンパ 3 は無効です。
1 : 内部タンパ 3 は有効です。
- ビット 17 **ITAMP2E** : 内部タンパ 2 イネーブル
0 : 内部タンパ 2 は無効です。
1 : 内部タンパ 2 は有効です。
- ビット 16 **ITAMP1E** : 内部タンパ 1 イネーブル
0 : 内部タンパ 1 は無効です。
1 : 内部タンパ 1 は有効です。
- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TAMP8E** : TAMP_IN8 のタンパ検出イネーブル ⁽¹⁾
0 : TAMP_IN8 のタンパ検出は無効です。
1 : TAMP_IN8 のタンパ検出は有効です。
- ビット 6 **TAMP7E** : TAMP_IN7 のタンパ検出イネーブル ⁽¹⁾
0 : TAMP_IN7 のタンパ検出は無効です。
1 : TAMP_IN7 のタンパ検出は有効です。

ビット 5 **TAMP6E** : TAMP_IN6 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN6 のタンパ検出は無効です。

1 : TAMP_IN6 のタンパ検出は有効です。

ビット 4 **TAMP5E** : TAMP_IN5 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN5 のタンパ検出は無効です。

1 : TAMP_IN5 のタンパ検出は有効です。

ビット 3 **TAMP4E** : TAMP_IN4 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN4 のタンパ検出は無効です。

1 : TAMP_IN4 のタンパ検出は有効です。

ビット 2 **TAMP3E** : TAMP_IN3 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN3 のタンパ検出は無効です。

1 : TAMP_IN3 のタンパ検出は有効です。

ビット 1 **TAMP2E** : TAMP_IN2 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN2 のタンパ検出は無効です。

1 : TAMP_IN2 のタンパ検出は有効です。

ビット 0 **TAMP1E** : TAMP_IN1 のタンパ検出イネーブル ⁽¹⁾

0 : TAMP_IN1 のタンパ検出は無効です。

1 : TAMP_IN1 のタンパ検出は有効です。

1. タンパ検出モード (TAMP_FLTCR、TAMP_ATCR1、TAMP_ATCR2 レジスタおよび TAMP_CR2 の TAMPxTRG ビットで選択) は、タンパ検出を有効化する前に設定する必要があります。

47.6.2 TAMP 制御レジスタ 2 (TAMP_CR2)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x04

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMP8 TRG	TAMP7 TRG	TAMP6 TRG	TAMP5 TRG	TAMP4 TRG	TAMP3 TRG	TAMP2 TRG	TAMP1 TRG	BK ERASE	BK BLOCK	Res.	Res.	Res.	TAMP3 MSK	TAMP2 MSK	TAMP1 MSK
rw	rw	rw	rw	rw	rw	rw	rw	w	rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8 POM	TAMP7 POM	TAMP6 POM	TAMP5 POM	TAMP4 POM	TAMP3 POM	TAMP2 POM	TAMP1 POM
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **TAMP8TRG** : タンパ 8 の入力のアクティブレベル (アクティブモード無効)

0 : TAMPFLT ≠ 00 の場合、タンパ 8 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 8 の入力立下りエッジのとき、タンパ検出イベントがトリガされます。

1 : TAMPFLT ≠ 00 の場合、タンパ 8 の入力が高いままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 8 の入力立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 30 **TAMP7TRG** : タンパ 7 の入力のアクティブレベル (アクティブモード無効)

- 0 : TAMPFLT = 00 の場合、タンパ 7 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 7 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 7 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 7 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 29 **TAMP6TRG** : タンパ 6 の入力のアクティブレベル (アクティブモード無効)

- 0 : TAMPFLT ≠ 00 の場合、タンパ 6 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 6 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 6 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 6 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 28 **TAMP5TRG** : タンパ 5 の入力のアクティブレベル (アクティブモード無効)

- 0 : TAMPFLT = 00 の場合、タンパ 5 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 5 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 5 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 5 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 27 **TAMP4TRG** : タンパ 4 の入力のアクティブレベル (アクティブモード無効)

- 0 : TAMPFLT = 00 の場合、タンパ 4 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 4 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 4 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 4 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 26 **TAMP3TRG** : タンパ 3 の入力のアクティブレベル

- 0 : TAMPFLT = 00 の場合、タンパ 3 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 3 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 3 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 3 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 25 **TAMP2TRG** : タンパ 2 の入力のアクティブレベル

- 0 : TAMPFLT = 00 の場合、タンパ 2 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 2 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 2 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 2 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 24 **TAMP1TRG** : タンパ 1 の入力のアクティブレベル

- 0 : TAMPFLT = 00 の場合、タンパ 1 の入力がローのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 1 の入力が立上りエッジのとき、タンパ検出イベントがトリガされます。
- 1 : TAMPFLT ≠ 00 の場合、タンパ 1 の入力がハイのままのとき、タンパ検出イベントがトリガされます。
TAMPFLT = 00 の場合、タンパ 1 の入力が立下りエッジのとき、タンパ検出イベントがトリガされます。

ビット 23 **BKERASE** : バックアップレジスタとデバイス機密情報⁽¹⁾ の消去

- このビットに「1」を書き込むと、バックアップレジスタとデバイス機密情報⁽¹⁾がリセットされます。
0 を書き込んで、ビットの値は変化しません。このビットは常に 0 として読み出されます。

ビット 22 **BKBLOCK** : バックアップレジスタとデバイス機密情報⁽¹⁾ のアクセスのブロック

- 0 : タンパフラグがセットされていない場合、バックアップレジスタとデバイス機密情報⁽¹⁾ にアクセスできます。
- 1 : バックアップレジスタとデバイス機密情報⁽¹⁾ にアクセスできません。

ビット 21:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TAMP3MSK** : タンパ 3 マスク

- 0 : タンパ 3 イベントでトリガイイベントが生成され、次のタンパイベント検出を許可するために TAMP3F をソフトウェアでクリアする必要があります。
- 1 : タンパ 3 イベントによってトリガイイベントが生成されます。TAMP3F はマスクされ、ハードウェアで内部的にクリアされます。バックアップレジスタとデバイス機密情報⁽¹⁾ は消去されません。
TAMP3MSK がセットされている場合、タンパ 3 割込みを有効にできません。

ビット 17 **TAMP2MSK** : タンパ 2 マスク

0: タンパ 2 イベントでトリガイイベントが生成され、次のタンパイイベント検出を許可するために TAMP2F をソフトウェアでクリアする必要があります。

1: タンパ 2 イベントによってトリガイイベントが生成されます。TAMP2F はマスクされ、ハードウェアで内部的にクリアされます。バックアップレジスタとデバイス機密情報⁽¹⁾は消去されません。

TAMP2MSK がセットされている場合、タンパ 2 割込みを有効にできません。

ビット 16 **TAMP1MSK** : タンパ 1 マスク

0: タンパ 1 イベントでトリガイイベントが生成され、次のタンパイイベント検出を許可するために TAMP1F をソフトウェアでクリアする必要があります。

1: タンパ 1 イベントによってトリガイイベントが生成されます。TAMP1F はマスクされ、ハードウェアで内部的にクリアされます。バックアップレジスタとデバイス機密情報⁽¹⁾は消去されません。

TAMP1MSK がセットされている場合、タンパ 1 割込みを有効にできません。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TAMP8POM** : タンパ 8 潜在モード

0: タンパ 8 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 8 イベント検出は潜在モードです⁽²⁾。

ビット 6 **TAMP7POM** : タンパ 7 潜在モード

0: タンパ 7 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 7 イベント検出は潜在モードです⁽²⁾。

ビット 5 **TAMP6POM** : タンパ 6 潜在モード

0: タンパ 6 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 6 イベント検出は潜在モードです⁽²⁾。

ビット 4 **TAMP5POM** : タンパ 5 潜在モード

0: タンパ 5 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 5 イベント検出は潜在モードです⁽²⁾。

ビット 3 **TAMP4POM** : タンパ 4 潜在モード

0: タンパ 4 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 4 イベント検出は潜在モードです⁽²⁾。

ビット 2 **TAMP3POM** : タンパ 3 潜在モード

0: タンパ 3 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 3 イベント検出は潜在モードです⁽²⁾。

ビット 1 **TAMP2POM** : タンパ 2 潜在モード

0: タンパ 2 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 2 イベント検出は潜在モードです⁽²⁾。

ビット 0 **TAMP1POM** : タンパ 1 潜在モード

0: タンパ 1 イベント検出は確認済みモードです⁽¹⁾。

1: タンパ 1 イベント検出は潜在モードです⁽²⁾。

1. 確認済みモードでのタンパ検出の影響は、表 489 : **TAMP 相互接続**の tamp_confirmed および tamp_confirmed_rpcfgx 信号の項を参照してください。
2. 潜在モードでのタンパ検出の影響は、表 489 : **TAMP 相互接続**の tamp_potential および tamp_potential_rpcfgx 信号の項を参照してください。

47.6.3 TAMP 制御レジスタ 3 (TAMP_CR3)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x08

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ITAMP 15POM	Res.	ITAMP 13POM	ITAMP 12POM	ITAMP 11POM	Res.	ITAMP9 POM	ITAMP8 POM	ITAMP7 POM	ITAMP6 POM	ITAMP5 POM	ITAMP4 POM	ITAMP3 POM	ITAMP2 POM	ITAMP1 POM
	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **ITAMP15POM** : 内部タンパ 15潜在モード

0 : 内部タンパ 15 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 15 イベント検出は潜在モードです ⁽²⁾。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **ITAMP13POM** : 内部タンパ 13潜在モード

0 : 内部タンパ 13 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 13 イベント検出は潜在モードです ⁽²⁾。

ビット 11 **ITAMP12POM** : 内部タンパ 12潜在モード

0 : 内部タンパ 12 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 12 イベント検出は潜在モードです ⁽²⁾。

ビット 10 **ITAMP11POM** : 内部タンパ 11潜在モード

0 : 内部タンパ 11 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 11 イベント検出は潜在モードです ⁽²⁾。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ITAMP9POM** : 内部タンパ 9潜在モード

0 : 内部タンパ 9 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 9 イベント検出は潜在モードです ⁽²⁾。

ビット 7 **ITAMP8POM** : 内部タンパ 8潜在モード

0 : 内部タンパ 8 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 8 イベント検出は潜在モードです ⁽²⁾。

ビット 6 **ITAMP7POM** : 内部タンパ 7潜在モード

0 : 内部タンパ 7 イベント検出は確認済みモードです ⁽¹⁾。

1 : 内部タンパ 7 イベント検出は潜在モードです ⁽²⁾。

ビット 5 **ITAMP6POM** : 内部タンパ 6 潜在モード

- 0 : 内部タンパ 6 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 6 イベント検出は潜在モードです⁽²⁾。

ビット 4 **ITAMP5POM** : 内部タンパ 5 潜在モード

- 0 : 内部タンパ 5 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 5 イベント検出は潜在モードです⁽²⁾。

ビット 3 **ITAMP4POM** : 内部タンパ 4 潜在モード

- 0 : 内部タンパ 4 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 4 イベント検出は潜在モードです⁽²⁾。

ビット 2 **ITAMP3POM** : 内部タンパ 3 潜在モード

- 0 : 内部タンパ 3 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 3 イベント検出は潜在モードです⁽²⁾。

ビット 1 **ITAMP2POM** : 内部タンパ 2 潜在モード

- 0 : 内部タンパ 2 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 2 イベント検出は潜在モードです⁽²⁾。

ビット 0 **ITAMP1POM** : 内部タンパ 1 潜在モード

- 0 : 内部タンパ 1 イベント検出は確認済みモードです⁽¹⁾。
- 1 : 内部タンパ 1 イベント検出は潜在モードです⁽²⁾。

- 確認済みモードでの内部タンパ検出の影響は、表 489 : TAMP 相互接続の tamp_confirmed および tamp_confirmed_rpcfgx 信号の項を参照してください。
- 潜在モードでの内部タンパ検出の影響は、表 489 : TAMP 相互接続の tamp_potential および tamp_potential_rpcfgx 信号の項を参照してください。

47.6.4 TAMP フィルタ制御レジスタ (TAMP_FLTCR)

このレジスタは、非セキュアアクセスから保護できます。セクション 47.3.5 : TAMP セキュア保護モードを参照してください。

このレジスタは、非特権アクセスから保護できます。セクション 47.3.7 : TAMP 特権保護モードを参照してください。

アドレスオフセット : 0x0C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP PUDIS	TAMPPRCH [1:0]	TAMPFLT [1:0]	TAMPFREQ [2:0]				
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TAMPPUDIS** : TAMP_INx プルアップディセーブル

- このビットにより、毎回のサンプリング前に各 TAMPx ピンをプリチャージするかどうか決定します。
- 0 : サンプリング前に TAMP_INx ピンをプリチャージします (内部プルアップを有効化)。
- 1 : TAMP_INx のプリチャージを無効化します。

ビット 6:5 **TAMPPRCH[1:0]** : TAMP_INx プリチャージ持続時間

これらのビットにより、各サンプリングの前にプルアップを有効化している時間を決定します。
TAMPPRCH は、各 TAMP_INx 入力に対して有効です。

0x0 : 1 RTCCLK サイクル

0x1 : 2 RTCCLK サイクル

0x2 : 4 RTCCLK サイクル

0x3 : 8 RTCCLK サイクル

ビット 4:3 **TAMPFLT[1:0]** : TAMP_INx フィルタカウント

これらのビットにより、タンパイVENTをアクティブにするのに必要な指定のレベル (TAMP*TRG) で
の連続サンプリングの数を決定します。TAMPFLT は、各 TAMP_INx 入力に対して有効です。

0x0 : TAMP_INx 入力 that アクティブレベル (TAMP_INx 入力における内部プルアップなし) に変化する
ときのエッジで、タンパイVENTがアクティブになります。

0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパイVENTがアクティブになります。

0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパイVENTがアクティブになります。

0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパイVENTがアクティブになります。

ビット 2:0 **TAMPFREQ[2:0]** : タンパサンプリング周波数

これらのビットにより、各 TAMP_INx 入力 that サンプリングされる周波数を決定します。

0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)

0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)

0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)

0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)

0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)

0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)

0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)

0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

注 : このレジスタは、パッシブモードでのタンパ入力のみ関係します。

47.6.5 TAMP アクティブタンパ制御レジスタ 1 (TAMP_ATCR1)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x10

バックアップドメインリセット値 : 0x0007 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTEN	ATO SHARE	Res.	Res.	Res.	ATPER[2:0]			Res.	Res.	Res.	Res.	ATCKSEL[3:0]			
rw	rw				rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATOSEL4 [1:0]		ATOSEL3 [1:0]		ATOSEL2 [1:0]		ATOSEL1 [1:0]		TAMP8 AM	TAMP7 AM	TAMP6 AM	TAMP5 AM	TAMP4 AM	TAMP3 AM	TAMP2 AM	TAMP1 AM
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **FLTEN** : アクティブタンパフィルタイネーブル

0 : アクティブタンパフィルタは無効です。

1 : アクティブタンパフィルタは有効です。連続した 4 つのサンプルのうち比較の不一致が 2 つ発生した場合にタンパイベントが検出されます。

ビット 30 **ATOSHARE** : アクティブタンパ出力の共有

0 : 各アクティブタンパ入力 TAMP_INi が、専用の出力 TAMP_OUTi と比較されます。

1 : 各アクティブタンパ入力 TAMP_INi が、ATOSELi ビットで定義された TAMPOUTSELi と比較されます。

ビット 29:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **ATPER[2:0]** : アクティブタンパ出力の変更周期

タンパ出力は、 $CK_ATPER = (2^{ATPER} \times CK_ATPRE)$ サイクルごとに変化します。[表 492 : 最低 ATPER 値](#)を参照してください。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **ATCKSEL[3:0]** : アクティブタンパ RTC 非同期プリスケラクロックの選択

これらのビットで RTC 非同期プリスケラステージ出力を選択します。選択されるクロックは CK_ATPRE です。

0000 : RTCCLK が選択されます。

0001 : RTCCLK/2 が選択されます。

0010 : RTCCLK/4 が選択されます。

0011 : RTCCLK/8 が選択されます。

0100 : RTCCLK/16 が選択されます。

0101 : RTCCLK/32 が選択されます。

0110 : RTCCLK/64 が選択されます。

0111 : RTCCLK/128 が選択されます。

1011 : $(PREDIV_A+1) = 128$ かつ $(PREDIV_S+1)$ が 16 の倍数の場合、RTCCLK/2048 が選択されます。

その他 : 予約済みです。

注 : これらのビットに書き込めるのは、アクティブタンパがすべて無効なときだけです。書き込み保護は、すべてのアクティブタンパが無効になってから最大 1.5 CK_ATPRE サイクル保持されます。

ビット 15:14 **ATOSEL4[1:0]** : アクティブタンパ共有出力 4 の選択

00 : TAMPOUTSEL4 = TAMP_OUT1
01 : TAMPOUTSEL4 = TAMP_OUT2
10 : TAMPOUTSEL4 = TAMP_OUT3
11 : TAMPOUTSEL4 = TAMP_OUT4

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 13:12 **ATOSEL3[1:0]** : アクティブタンパ共有出力 3 の選択

00 : TAMPOUTSEL3 = TAMP_OUT1
01 : TAMPOUTSEL3 = TAMP_OUT2
10 : TAMPOUTSEL3 = TAMP_OUT3
11 : TAMPOUTSEL3 = TAMP_OUT4

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 11:10 **ATOSEL2[1:0]** : アクティブタンパ共有出力 2 の選択

00 : TAMPOUTSEL2 = TAMP_OUT1
01 : TAMPOUTSEL2 = TAMP_OUT2
10 : TAMPOUTSEL2 = TAMP_OUT3
11 : TAMPOUTSEL2 = TAMP_OUT4

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 9:8 **ATOSEL1[1:0]** : アクティブタンパ共有出力 1 の選択

00 : TAMPOUTSEL1 = TAMP_OUT1
01 : TAMPOUTSEL1 = TAMP_OUT2
10 : TAMPOUTSEL1 = TAMP_OUT3
11 : TAMPOUTSEL1 = TAMP_OUT4

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 7 **TAMP8AM** : タンパ 8 アクティブモード

0 : タンパ 8 検出モードはパッシブです。
1 : タンパ 8 検出モードはアクティブです。

ビット 6 **TAMP7AM** : タンパ 7 アクティブモード

0 : タンパ 7 検出モードはパッシブです。
1 : タンパ 7 検出モードはアクティブです。

ビット 5 **TAMP6AM** : タンパ 6 アクティブモード

0 : タンパ 6 検出モードはパッシブです。
1 : タンパ 6 検出モードはアクティブです。

ビット 4 **TAMP5AM** : タンパ 5 アクティブモード

0 : タンパ 5 検出モードはパッシブです。
1 : タンパ 5 検出モードはアクティブです。

ビット 3 **TAMP4AM** : タンパ 4 アクティブモード

0 : タンパ 4 検出モードはパッシブです。
1 : タンパ 4 検出モードはアクティブです。

ビット 2 **TAMP3AM** : タンパ 3 アクティブモード

0 : タンパ 3 検出モードはパッシブです。
1 : タンパ 3 検出モードはアクティブです。

ビット 1 **TAMP2AM** : タンパ 2 アクティブモード

0 : タンパ 2 検出モードはパッシブです。

1 : タンパ 2 検出モードはアクティブです。

ビット 0 **TAMP1AM** : タンパ 1 アクティブモード

0 : タンパ 1 検出モードはパッシブです。

1 : タンパ 1 検出モードはアクティブです。

注 : **TAMPxAM** ビットがセットされている場合、対応する **TAMPxE** ビットが **TAMP_CR1** レジスタですべてクリアされていない限り、このレジスタのアクティブタンパ設定を変更できません。

アクティブモードに設定されたすべてのタンパは、(同じ **TAMP_CR1** 書き込みで関連するすべての **TAMPxE** をセットして) 同時に有効にする必要があります。

アクティブモードに設定されたすべてのタンパは、(同じ **TAMP_CR1** 書き込みで関連するすべての **TAMPxE** をクリアして) 同時に無効にする必要があります。

アクティブタンパを無効にしてから再び有効にするまで、最低 **CK_ATPRE** 1 周期を待つ必要があります。

47.6.6 TAMP アクティブタンパシードレジスタ (TAMP_ATSEEDR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x14

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEED[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEED[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **SEED[31:0]** : 疑似乱数発生器のシード値

このレジスタは、128 ビットのシードを PRNG に供給する 32 ビット値を 4 回書き込む必要があります。このレジスタに書き込むと、シード値が自動的に PRNG に送信されます。

47.6.7 TAMP アクティブタンパ出力レジスタ (TAMP_ATOM)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x18

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 0 にリセットされる SEEDF を除いて影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITS	SEEDF	Res.	Res.	Res.	Res.	Res.	Res.	PRNG[7:0]							
r	r							r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **INITS** : アクティブタンパの初期化ステータス

このフラグは、PRNG が最初の 128 ビットのシードを受け取ったときにハードウェアによってセットされ、有効にされたアクティブタンパが機能することを示します。このフラグは、アクティブタンパが無効になるとクリアされます。

ビット 14 **SEEDF** : シードランニングフラグ

このフラグは、新しいシードが TAMP_ATSEEDR に書き込まれたときに、ハードウェアによってセットされます。PRNG がこの新しいシードを受け取ったときにハードウェア、またシステムリセットによってクリアされます。SEEDF がセットされている場合、TAMP APB クロックはオフにしないでください。

ビット 13:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PRNG[7:0]** : 疑似乱数発生器の値

このフィールドは、PRNG 出力の値を提供します。同期遅延による潜在的な不一致があるため、PRNG は最低 2 回読み出す必要があります。前の読出し値と一致する場合、その読出し値が正しいものとなります。

このフィールドは、APB がセキュアモードのときのみ読み出すことができます。

47.6.8 TAMP アクティブタンパ制御レジスタ 2 (TAMP_ATCR2)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x1C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATOSEL8 [2:0]			ATOSEL7 [2:0]			ATOSEL6 [2:0]			ATOSEL5 [2:0]			ATOSEL4 [2:0]			ATOSEL3 [2]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATOSEL3 [1:0]			ATOSEL2 [2:0]			ATOSEL1 [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw							

ビット 31:29 **ATOSEL8[2:0]** : アクティブタンパ共有出力 8 の選択

000 : TAMPOUTSEL8 = TAMP_OUT1
 001 : TAMPOUTSEL8 = TAMP_OUT2
 010 : TAMPOUTSEL8 = TAMP_OUT3
 011 : TAMPOUTSEL8 = TAMP_OUT4
 100 : TAMPOUTSEL8 = TAMP_OUT5
 101 : TAMPOUTSEL8 = TAMP_OUT6
 110 : TAMPOUTSEL8 = TAMP_OUT7
 111 : TAMPOUTSEL8 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 28:26 **ATOSEL7[2:0]** : アクティブタンパ共有出力 7 の選択

000 : TAMPOUTSEL7 = TAMP_OUT1
 001 : TAMPOUTSEL7 = TAMP_OUT2
 010 : TAMPOUTSEL7 = TAMP_OUT3
 011 : TAMPOUTSEL7 = TAMP_OUT4
 100 : TAMPOUTSEL7 = TAMP_OUT5
 101 : TAMPOUTSEL7 = TAMP_OUT6
 110 : TAMPOUTSEL7 = TAMP_OUT7
 111 : TAMPOUTSEL7 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 25:23 **ATOSEL6[2:0]** : アクティブタンパ共有出力 6 の選択

000 : TAMPOUTSEL6 = TAMP_OUT1
001 : TAMPOUTSEL6 = TAMP_OUT2
010 : TAMPOUTSEL6 = TAMP_OUT3
011 : TAMPOUTSEL6 = TAMP_OUT4
100 : TAMPOUTSEL6 = TAMP_OUT5
101 : TAMPOUTSEL6 = TAMP_OUT6
110 : TAMPOUTSEL6 = TAMP_OUT7
111 : TAMPOUTSEL6 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 22:20 **ATOSEL5[2:0]** : アクティブタンパ共有出力 5 の選択

000 : TAMPOUTSEL5 = TAMP_OUT1
001 : TAMPOUTSEL5 = TAMP_OUT2
010 : TAMPOUTSEL5 = TAMP_OUT3
011 : TAMPOUTSEL5 = TAMP_OUT4
100 : TAMPOUTSEL5 = TAMP_OUT5
101 : TAMPOUTSEL5 = TAMP_OUT6
110 : TAMPOUTSEL5 = TAMP_OUT7
111 : TAMPOUTSEL5 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 19:17 **ATOSEL4[2:0]** : アクティブタンパ共有出力 4 の選択

000 : TAMPOUTSEL4 = TAMP_OUT1
001 : TAMPOUTSEL4 = TAMP_OUT2
010 : TAMPOUTSEL4 = TAMP_OUT3
011 : TAMPOUTSEL4 = TAMP_OUT4
100 : TAMPOUTSEL4 = TAMP_OUT5
101 : TAMPOUTSEL4 = TAMP_OUT6
110 : TAMPOUTSEL4 = TAMP_OUT7
111 : TAMPOUTSEL4 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 18:17 は、TAMP_ATCR1 の ATOSEL2[1:0] のミラーであり、TAMP_ATCR1 でも読出しまたは書込みできます。

ビット 16:14 **ATOSEL3[2:0]** : アクティブタンパ共有出力 3 の選択

000 : TAMPOUTSEL3 = TAMP_OUT1
001 : TAMPOUTSEL3 = TAMP_OUT2
010 : TAMPOUTSEL3 = TAMP_OUT3
011 : TAMPOUTSEL3 = TAMP_OUT4
100 : TAMPOUTSEL3 = TAMP_OUT5
101 : TAMPOUTSEL3 = TAMP_OUT6
110 : TAMPOUTSEL3 = TAMP_OUT7
111 : TAMPOUTSEL3 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 15:14 は、TAMP_ATCR1 の ATOSEL3[1:0] のミラーであり、TAMP_ATCR1 でも読出しまたは書込みできます。

ビット 13:11 **ATOSEL2[2:0]** : アクティブタンパ共有出力 2 の選択

000 : TAMPOUTSEL2 = TAMP_OUT1
001 : TAMPOUTSEL2 = TAMP_OUT2
010 : TAMPOUTSEL2 = TAMP_OUT3
011 : TAMPOUTSEL2 = TAMP_OUT4
100 : TAMPOUTSEL2 = TAMP_OUT5
101 : TAMPOUTSEL2 = TAMP_OUT6
110 : TAMPOUTSEL2 = TAMP_OUT7
111 : TAMPOUTSEL2 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 12:11 は、TAMP_ATCR1 の ATOSEL2[1:0] のミラーであり、TAMP_ATCR1 でも読出しまたは書込みできます。

ビット 10:8 **ATOSEL1[2:0]** : アクティブタンパ共有出力 1 の選択

000 : TAMPOUTSEL1 = TAMP_OUT1
001 : TAMPOUTSEL1 = TAMP_OUT2
010 : TAMPOUTSEL1 = TAMP_OUT3
011 : TAMPOUTSEL1 = TAMP_OUT4
100 : TAMPOUTSEL1 = TAMP_OUT5
101 : TAMPOUTSEL1 = TAMP_OUT6
110 : TAMPOUTSEL1 = TAMP_OUT7
111 : TAMPOUTSEL1 = TAMP_OUT8

パッケージのピン配置でTAMP_OUTx 出力が利用できない場合、出力選択値は予約済みとなり、使用できません。

ビット 9:8 は、TAMP_ATCR1 の ATOSEL1[1:0] のミラーであり、TAMP_ATCR1 でも読出しまたは書込みできます。

ビット 7:0 予約済みであり、リセット値に保持する必要があります。

注 : TAMPxAM ビットがセットされている場合、対応する TAMPxE ビットが TAMP_CR1 レジスタですべてクリアされていない限り、このレジスタのアクティブタンパ設定を変更できません。

アクティブモードに設定されたすべてのタンパは、(同じ TAMP_CR1 書込みで関連するすべての TAMPxE をセットして) 同時に有効にする必要があります。

アクティブモードに設定されたすべてのタンパは、(同じ TAMP_CR1 書込みで関連するすべての TAMPxE をクリアして) 同時に無効にする必要があります。

アクティブタンパを無効にしてから再び有効にするまで、最低 CK_ATPRE 1 周期を待つ必要があります。

47.6.9 TAMP セキュア設定レジスタ (TAMP_SECCFGR)

TZEN = 1 の場合、このレジスタに書き込めるのは、APB アクセスがセキュアなときだけです。TZEN = 0 の場合、BKPRWSEC[7:0]、BKPWSEC[7:0]、BHKLOCK は非セキュア APB アクセスにより書き込みができ、TAMPSEC、CNT1SEC は書き込みができません。

このレジスタは全体的に書き込み保護することができ、このレジスタの各ビットは、TAMP_PRIVCFGR の設定に応じて、非特権アクセスから個別に書き込み保護することもできます ([セクション 47.3.7: TAMP 特権保護モード](#)を参照)。

アドレスオフセット : 0x20

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMP SEC	BHK LOCK	Res.	Res.	Res.	Res.	Res.	Res.	BKPRWSEC[7:0]							
rw	rs							rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT1 SEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPRWSEC[7:0]							
rw								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **TAMPSEC** : タンパ保護 (単調カウンタとバックアップレジスタを除く)

0 : APB アクセスがセキュアでも非セキュアでも、タンパの設定と割込みを書き込みます。

1 : APB アクセスがセキュアな場合のみ、タンパの設定と割込みを書き込みます。

注 : 読出し保護についての詳細は、[セクション 47.3.5: TAMP セキュア保護モード](#)を参照してください。

ビット 30 **BHKLOCK** : ブート・ハードウェア・キーのロック

このビットはソフトウェアによって読出しと、1 の値の書き込みのみが行えます。タンパ検出イベントに続いて、または読出し保護 (RDP) が無効な状態で、バックアップレジスタと一緒にハードウェアによってクリアされます。

0 : TAMP_BKP0R から TAMP_BKP7R までのバックアップレジスタには、所属する保護ゾーンに従ってアクセスできます。

1 : TAMP_BKP0R から TAMP_BKP7R までのバックアップレジスタには、読出しでも書き込みでもアクセスできません (0 として読み出され、書き込みは無視されます)。

ビット 29:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **BKPWSEC[7:0]** : バックアップレジスタ書き込み保護オフセット

BKPWSEC値は 0 ~ 32 でなければなりません。

保護ゾーン 2 は TAMP_BKPyR (y = BKPWSEC) から TAMP_BKpZR (z = BKPWSEC-1 で、BKPWSEC > BKPWSEC) までのバックアップレジスタ用に定義されています。

- TZEN=1 の場合、これらのバックアップレジスタは、セキュアアクセスでのみ書き込めます。
- 読出しは、セキュアアクセスと非セキュアアクセスで可能です。

BKPWSEC = 0 または BKPWSEC ≤ BKPWSEC の場合 : 保護ゾーン 2 はありません。

保護ゾーン 3 は TAMP_BKPtR (BKPWSEC ≥ BKPWSEC の場合は t = BKPWSEC、それ以外の場合は t = BKPWSEC) からのバックアップレジスタ用に定義されています。

- 読出しと書き込みは、セキュアアクセスと非セキュアアクセスで可能です。

BKPWSEC = 32 の場合 : 保護ゾーン 3 はありません。

図 619 : バックアップレジスタ保護ゾーンを参照してください。

注 : TZEN=0 の場合 : 非セキュアアクセスで保護ゾーン 2 の読み書きが可能です。

注 : BKPWPPriv がセットされている場合、BKPWSEC[7:0] は特権モードでのみ書き込みできます。

ビット 15 **CNT1SEC** : 単調カウンタ 1 のセキュア保護

0 : 単調カウンタ 1 (TAMP_COUNT1R) は、APB アクセスがセキュアまたは非セキュアの場合に読み書きできます。

1 : 単調カウンタ 1 (TAMP_COUNT1R) は、APB アクセスがセキュアの場合のみ読み書きできます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **BKPRWSEC[7:0]** : バックアップレジスタの読出し/書き込み保護オフセット

BKPRWSEC値は 0 ~ 32 でなければなりません。

保護ゾーン 1 は TAMP_BKP0R から TAMP_BKPxR (x = BKPRWSEC-1 で、BKPRWSEC ≥ 1) までのバックアップレジスタ用に定義されています。

- TZEN=1 の場合、これらのバックアップレジスタは、セキュアアクセスでのみ読み書きできます。

BKPRWSEC = 0 の場合 : 保護ゾーン 1 はありません。

図 619 : バックアップレジスタ保護ゾーンを参照してください。

注 : TZEN=0 の場合 : 非セキュアアクセスで保護ゾーン 1 の読み書きが可能です。

注 : BKPRWPPriv がセットされている場合、BKPRWSEC[7:0] は特権モードでのみ書き込みできます。

47.6.10 TAMP 特権設定レジスタ (TAMP_PRIVCFGR)

このレジスタに書き込めるのは、APB アクセスが特権のときだけです。

TZEN = 1 の場合、このレジスタは書き込み保護することができ、このレジスタの各ビットは、TAMP_SECCFGR の設定に応じて、非セキュアアクセスから個別に書き込み保護することもできます (セクション 47.3.5 : TAMP セキュア保護モードを参照)。

アドレスオフセット : 0x24

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAMP PRIV	BKP WPRIV	BKPR WPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT1 PRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

ビット 31 **TAMPPRIV** : タンパ特権保護 (バックアップレジスタを除く)

0 : タンパ設定および割込みは、特権アクセスまたは非特権アクセスで書き込みできます。

1 : タンパの設定と割込みは、特権アクセスでのみ書き込みます。

注 : 読出し保護についての詳細は、セクション 47.3.7 : TAMP 特権保護モードを参照してください。

ビット 30 **BKPWPRIV** : バックアップレジスタゾーン 2 の特権保護

0 : バックアップレジスタゾーン 2 は、特権アクセスまたは非特権アクセスで書き込みできます。

1 : バックアップレジスタゾーン 2 は、特権アクセスでのみ書き込みできます。

ビット 29 **BKPRWPRIV** : バックアップレジスタゾーン 1 の特権保護

0 : バックアップレジスタゾーン 1 は、特権アクセスまたは非特権アクセスで読み書きできます。

1 : バックアップレジスタゾーン 1 は、特権アクセスでのみ読み書きできます

ビット 28:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **CNT1PRIV** : 単調カウンタ 1 の特権保護

0 : 単調カウンタ 1 (TAMP_COUNT1R) は、APB アクセスが特権または非特権の場合に読み書きできます。

1 : 単調カウンタ 1 (TAMP_COUNT1R) は、APB アクセスが特権の場合のみ読み書きできます。

ビット 14:0 予約済みであり、リセット値に保持する必要があります。

47.6.11 TAMP 割込み有効レジスタ (TAMP_IER)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x2C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ITAMP15 IE	Res.	ITAMP13 IE	ITAMP12 IE	ITAMP11 IE	Res.	ITAMP9 IE	ITAMP8 IE	ITAMP7 IE	ITAMP6 IE	ITAMP5 IE	ITAMP4 IE	ITAMP3 IE	ITAMP2 IE	ITAMP1 IE
	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP 8IE	TAMP 7IE	TAMP 6IE	TAMP 5IE	TAMP 4IE	TAMP 3IE	TAMP 2IE	TAMP 1IE
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ITAMP15IE** : 内部タンパ 15 割込みイネーブル

0 : 内部タンパ 15 割込みは無効です。

1 : 内部タンパ 15 割込みは有効です。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ITAMP13IE** : 内部タンパ 13 割込みイネーブル

0 : 内部タンパ 13 割込みは無効です。

1 : 内部タンパ 13 割込みは有効です。

ビット 27 **ITAMP12IE** : 内部タンパ 12 割込みイネーブル

0 : 内部タンパ 12 割込みは無効です。

1 : 内部タンパ 12 割込みは有効です。

ビット 26 **ITAMP11IE** : 内部タンパ 11 割込みイネーブル

0 : 内部タンパ 11 割込みは無効です。

1 : 内部タンパ 11 割込みは有効です。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ITAMP9IE** : 内部タンパ 9 割込みイネーブル

0 : 内部タンパ 9 割込みは無効です。

1 : 内部タンパ 9 割込みは有効です。

ビット 23 **ITAMP8IE** : 内部タンパ 8 割込みイネーブル

0 : 内部タンパ 8 割込みは無効です。

1 : 内部タンパ 8 割込みは有効です。

ビット 22 **ITAMP7IE** : 内部タンパ 7 割込みイネーブル

0 : 内部タンパ 7 割込みは無効です。

1 : 内部タンパ 7 割込みは有効です。

ビット 21 **ITAMP6IE** : 内部タンパ 6 割込みイネーブル

0 : 内部タンパ 6 割込みは無効です。

1 : 内部タンパ 6 割込みは有効です。

- ビット 20 **ITAMP5IE** : 内部タンパ 5 割込みイネーブル
0 : 内部タンパ 5 割込みは無効です。
1 : 内部タンパ 5 割込みは有効です。
- ビット 19 **ITAMP4IE** : 内部タンパ 4 割込みイネーブル
0 : 内部タンパ 4 割込みは無効です。
1 : 内部タンパ 4 割込みは有効です。
- ビット 18 **ITAMP3IE** : 内部タンパ 3 割込みイネーブル
0 : 内部タンパ 3 割込みは無効です。
1 : 内部タンパ 3 割込みは有効です。
- ビット 17 **ITAMP2IE** : 内部タンパ 2 割込みイネーブル
0 : 内部タンパ 2 割込みは無効です。
1 : 内部タンパ 2 割込みは有効です。
- ビット 16 **ITAMP1IE** : 内部タンパ 1 割込みイネーブル
0 : 内部タンパ 1 割込みは無効です。
1 : 内部タンパ 1 割込みは有効です。
- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TAMP8IE** : タンパ 8 割込みイネーブル
0 : タンパ 8 割込みは無効です。
1 : タンパ 8 割込みは有効です。
- ビット 6 **TAMP7IE** : タンパ 7 割込みイネーブル
0 : タンパ 7 割込みは無効です。
1 : タンパ 7 割込みは有効です。
- ビット 5 **TAMP6IE** : タンパ 6 割込みイネーブル
0 : タンパ 6 割込みは無効です。
1 : タンパ 6 割込みは有効です。
- ビット 4 **TAMP5IE** : タンパ 5 割込みイネーブル
0 : タンパ 5 割込みは無効です。
1 : タンパ 5 割込みは有効です。
- ビット 3 **TAMP4IE** : タンパ 4 割込みイネーブル
0 : タンパ 4 割込みは無効です。
1 : タンパ 4 割込みは有効です。
- ビット 2 **TAMP3IE** : タンパ 3 割込みイネーブル
0 : タンパ 3 割込みは無効です。
1 : タンパ 3 割込みは有効です。
- ビット 1 **TAMP2IE** : タンパ 2 割込みイネーブル
0 : タンパ 2 割込みは無効です。
1 : タンパ 2 割込みは有効です。
- ビット 0 **TAMP1IE** : タンパ 1 割込みイネーブル
0 : タンパ 1 割込みは無効です。
1 : タンパ 1 割込みは有効です。

47.6.12 TAMP ステータスレジスタ (TAMP_SR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x30

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ITAMP15 F	Res.	ITAMP13 F	ITAMP12 F	ITAMP11 F	Res.	ITAMP9 F	ITAMP8 F	ITAMP7 F	ITAMP6 F	ITAMP5 F	ITAMP4 F	ITAMP3 F	ITAMP2 F	ITAMP1 F
	rw		r	r	r		r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP 8F	TAMP 7F	TAMP 6F	TAMP 5F	TAMP 4F	TAMP 3F	TAMP 2F	TAMP 1F
								r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ITAMP15F** : 内部タンパ 15 フラグ

このフラグは、内部タンパ 15 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ITAMP13F** : 内部タンパ 13 フラグ

このフラグは、内部タンパ 13 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 27 **ITAMP12F** : 内部タンパ 12 フラグ

このフラグは、内部タンパ 12 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 26 **ITAMP11F** : 内部タンパ 11 フラグ

このフラグは、内部タンパ 11 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ITAMP9F** : 内部タンパ 9 フラグ

このフラグは、内部タンパ 9 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 23 **ITAMP8F** : 内部タンパ 8 フラグ

このフラグは、内部タンパ 8 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 22 **ITAMP7F** : 内部タンパ 7 フラグ

このフラグは、内部タンパ 7 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ビット 21 **ITAMP6F** : 内部タンパ 6 フラグ

このフラグは、内部タンパ 6 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

- ビット 20 **ITAMP5F** : 内部タンパ 5 フラグ
このフラグは、内部タンパ 5 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 19 **ITAMP4F** : 内部タンパ 4 フラグ
このフラグは、内部タンパ 4 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 18 **ITAMP3F** : 内部タンパ 3 フラグ
このフラグは、内部タンパ 3 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 17 **ITAMP2F** : 内部タンパ 2 フラグ
このフラグは、内部タンパ 2 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 16 **ITAMP1F** : 内部タンパ 1 フラグ
このフラグは、内部タンパ 1 にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TAMP8F** : TAMP8 検出フラグ
このフラグは、TAMP8 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 6 **TAMP7F** : TAMP7 検出フラグ
このフラグは、TAMP7 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 5 **TAMP6F** : TAMP6 検出フラグ
このフラグは、TAMP6 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 4 **TAMP5F** : TAMP5 検出フラグ
このフラグは、TAMP5 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 3 **TAMP4F** : TAMP4 検出フラグ
このフラグは、TAMP4 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 2 **TAMP3F** : TAMP3 検出フラグ
このフラグは、TAMP3 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 1 **TAMP2F** : TAMP2 検出フラグ
このフラグは、TAMP2 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
- ビット 0 **TAMP1F** : TAMP1 検出フラグ
このフラグは、TAMP1 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

47.6.13 TAMP 非セキュアマスク済み割込みステータスレジスタ (TAMP_MISR)

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x34

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ITAMP15 MF	Res.	ITAMP13 MF	ITAMP12 MF	ITAMP11 MF	Res.	ITAMP9 MF	ITAMP8 MF	ITAMP7 MF	ITAMP6 MF	ITAMP5 MF	ITAMP4 MF	ITAMP3 MF	ITAMP2 MF	ITAMP1 MF
	r		r	r	r		r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8 MF	TAMP7 MF	TAMP6 MF	TAMP5 MF	TAMP4 MF	TAMP3 MF	TAMP2 MF	TAMP1 MF
								r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ITAMP15MF** : 内部タンパ 15 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 15 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ITAMP13MF** : 内部タンパ 13 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 13 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 27 **ITAMP12MF** : 内部タンパ 12 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 12 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 26 **ITAMP11MF** : 内部タンパ 11 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 11 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ITAMP9MF** : 内部タンパ 9 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 9 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 23 **ITAMP8MF** : 内部タンパ 8 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 8 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 22 **ITAMP7MF** : 内部タンパ 7 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 7 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 21 **ITAMP6MF** : 内部タンパ 6 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 6 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 20 **ITAMP5MF** : 内部タンパ 5 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 5 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 19 **ITAMP4MF** : 内部タンパ 4 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 4 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 18 **ITAMP3MF** : 内部タンパ 3 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 3 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 17 **ITAMP2MF** : 内部タンパ 2 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 2 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 16 **ITAMP1MF** : 内部タンパ 1 非セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 1 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TAMP8MF** : TAMP8 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 8 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 6 **TAMP7MF** : TAMP7 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 7 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 5 **TAMP6MF** : TAMP6 セキュア割込みマスク済みフラグ

このフラグは、タンパ 6 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 4 **TAMP5MF** : TAMP5 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 5 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 3 **TAMP4MF** : TAMP4 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 4 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 2 **TAMP3MF** : TAMP3 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 3 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 1 **TAMP2MF** : TAMP2 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 2 非セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 0 **TAMP1MF** : TAMP1 非セキュア割込みマスク済みフラグ

このフラグは、タンパ 1 非セキュア割込みが発生した場合にハードウェアによってセットされます。

47.6.14 TAMP セキュアマスク済み割込みステータスレジスタ (TAMP_SMISR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x38

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ITAMP15 MF	Res.	ITAMP13 MF	ITAMP12 MF	ITAMP11 MF	Res.	ITAMP9 MF	ITAMP8 MF	ITAMP7 MF	ITAMP6 MF	ITAMP5 MF	ITAMP4 MF	ITAMP3 MF	ITAMP2 MF	ITAMP1 MF
	r		r	r	r		r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8 MF	TAMP7 MF	TAMP6 MF	TAMP5 MF	TAMP4 MF	TAMP3 MF	TAMP2 MF	TAMP1 MF
								r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **ITAMP15MF** : 内部タンパ 15 セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 15 セキュア割込みが発生した場合にハードウェアによってセットされます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ITAMP13MF** : 内部タンパ 13 セキュア割込みマスク済みフラグ

このフラグは、内部タンパ 13 セキュア割込みが発生した場合にハードウェアによってセットされます。

- ビット 27 **ITAMP12MF** : 内部タンパ 12 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 12 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 26 **ITAMP11MF** : 内部タンパ 11 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 11 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 25 予約済みであり、リセット値に保持する必要があります。
- ビット 24 **ITAMP9MF** : 内部タンパ 9 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 9 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 23 **ITAMP8MF** : 内部タンパ 8 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 8 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 22 **ITAMP7MF** : 内部タンパ 7 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 7 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 21 **ITAMP6MF** : 内部タンパ 6 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 6 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 20 **ITAMP5MF** : 内部タンパ 5 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 5 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 19 **ITAMP4MF** : 内部タンパ 4 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 4 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 18 **ITAMP3MF** : 内部タンパ 3 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 3 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 17 **ITAMP2MF** : 内部タンパ 2 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 2 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 16 **ITAMP1MF** : 内部タンパ 1 セキュア割込みマスク済みフラグ
このフラグは、内部タンパ 1 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TAMP8MF** : TAMP8 セキュア割込みマスク済みフラグ
このフラグは、タンパ 8 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 6 **TAMP7MF** : TAMP7 セキュア割込みマスク済みフラグ
このフラグは、タンパ 7 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 5 **TAMP6MF** : TAMP6 セキュア割込みマスク済みフラグ
このフラグは、タンパ 6 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 4 **TAMP5MF** : TAMP5 セキュア割込みマスク済みフラグ
このフラグは、タンパ 5 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 3 **TAMP4MF** : TAMP4 セキュア割込みマスク済みフラグ
このフラグは、タンパ 4 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 2 **TAMP3MF** : TAMP3 セキュア割込みマスク済みフラグ
このフラグは、タンパ 3 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 1 **TAMP2MF** : TAMP2 セキュア割込みマスク済みフラグ
このフラグは、タンパ 2 セキュア割込みが発生した場合にハードウェアによってセットされます。
- ビット 0 **TAMP1MF** : TAMP1 セキュア割込みマスク済みフラグ
このフラグは、タンパ 1 セキュア割込みが発生した場合にハードウェアによってセットされます。

47.6.15 TAMP ステータスクリアレジスタ (TAMP_SCR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x3C

システムリセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	C ITAMP 15F	Res.	C ITAMP 13F	C ITAMP 12F	C ITAMP 11F	Res.	C ITAMP 9F	C ITAMP 8F	C ITAMP 7F	C ITAMP 6F	C ITAMP 5F	C ITAMP 4F	C ITAMP 3F	C ITAMP 2F	C ITAMP 1F
	w		w	w	w		w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTAMP 8F	CTAMP 7F	CTAMP 6F	CTAMP 5F	CTAMP 4F	CTAMP 3F	CTAMP 2F	CTAMP 1F
								w	w	w	w	w	w	w	w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **CITAMP15F** : ITAMP15 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP15F ビットがクリアされます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **CITAMP13F** : ITAMP13 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP13F ビットがクリアされます。

ビット 27 **CITAMP12F** : ITAMP12 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP12F ビットがクリアされます。

ビット 26 **CITAMP11F** : ITAMP11 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP11F ビットがクリアされます。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **CITAMP9F** : ITAMP9 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP9F ビットがクリアされます。

ビット 23 **CITAMP8F** : ITAMP8 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP8F ビットがクリアされます。

ビット 22 **CITAMP7F** : ITAMP7 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP7F ビットがクリアされます。

ビット 21 **CITAMP6F** : ITAMP6 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP6F ビットがクリアされます。

ビット 20 **CITAMP5F** : ITAMP5 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP5F ビットがクリアされます。

ビット 19 **CITAMP4F** : ITAMP4 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP4F ビットがクリアされます。

ビット 18 **CITAMP3F** : ITAMP3 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP3F ビットがクリアされます。

ビット 17 **CITAMP2F** : ITAMP2 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP2F ビットがクリアされます。

ビット 16 **CITAMP1F** : ITAMP1 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの ITAMP1F ビットがクリアされます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CTAMP8F** : TAMP8 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP8F ビットがクリアされます。

ビット 6 **CTAMP7F** : TAMP7 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP7F ビットがクリアされます。

ビット 5 **CTAMP6F** : TAMP6 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP6F ビットがクリアされます。

ビット 4 **CTAMP5F** : TAMP5 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP5F ビットがクリアされます。

ビット 3 **CTAMP4F** : TAMP4 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP4F ビットがクリアされます。

ビット 2 **CTAMP3F** : TAMP3 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP3F ビットがクリアされます。

ビット 1 **CTAMP2F** : TAMP2 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP2F ビットがクリアされます。

ビット 0 **CTAMP1F** : TAMP1 検出フラグのクリア

このビットに 1 を書き込むと、TAMP_SR レジスタの TAMP1F ビットがクリアされます。

47.6.16 TAMP 単調カウンタ 1 レジスタ (TAMP_COUNT1R)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x040

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COUNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **COUNT[31:0]** :

このレジスタは読み出し専用で、このレジスタに対して書き込みアクセスがあると、1 ずつインクリメントされます。このレジスタは、最大値に達するとロールオーバーせずに停止します。

47.6.17 TAMP オプションレジスタ (TAMP_OR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x50

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	IN4_RMP	IN3_RMP	IN2_RMP	Res.	Res.	Res.	Res.	OUT5_RMP	OUT3_RMP [1:0]		Res.
					rw	rw	rw					rw	rw	rw	

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **IN4_RMP** : TAMP_IN4 マッピング

0 : TAMP_IN4 は PA2 上にあります

1 : TAMP_IN4 は PI11 上にあります

ビット 9 **IN3_RMP** : TAMP_IN3 マッピング

0 : TAMP_IN3 は PC1 上にあります

1 : TAMP_IN3 は PE6 上にあります

ビット 8 **IN2_RMP** : TAMP_IN2 マッピング

0 : TAMP_IN2 は PA0 上にあります

1 : TAMP_IN2 は PI8 上にあります

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OUT5_RMP** : TAMP_OUT5 マッピング

0 : TAMP_OUT5 は PI11 上にあります

1 : TAMP_OUT5 は PC1 上にあります

ビット 2:1 **OUT3_RMP[1:0]** : TAMP_OUT3 マッピング

00 : TAMP_OUT3 は PC13 上にあります

01 : TAMP_OUT3 は PI8 上にあります

10 : TAMP_OUT3 は PE3 上にあります

11 : TAMP_OUT3 は PA2 上にあります

ビット 0 予約済みであり、リセット値に保持する必要があります。

47.6.18 TAMP リソース保護設定レジスタ (TAMP_RPCFGR)

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x54

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RP CFG0
															rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 RPCFG0 : リソース 0 保護を設定可能⁽¹⁾

0 : リソース 0 は、TAMP ペリフェラルによって保護されるデバイス機密情報には含まれません

1 : リソース 0 は、TAMP ペリフェラルによって保護されるデバイス機密情報に含まれます

1. 表 487 : TAMP の入出力ピンと表 489 : TAMP 相互接続の tamp_confirmed_rpcfg0 および tamp_potential_rpcfg0 信号を参照してください。

47.6.19 TAMP バックアップ x レジスタ（TAMP_BKPxR）

このレジスタは、非セキュアアクセスから保護できます。[セクション 47.3.5 : TAMP セキュア保護モード](#)を参照してください。

このレジスタは、非特権アクセスから保護できます。[セクション 47.3.7 : TAMP 特権保護モード](#)を参照してください。

アドレスオフセット : 0x100 + 0x04 * x、(x = 0 から 31)

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

ビット 31:0 **BKP[31:0]** :

アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。

デフォルト（ERASE）設定で、このレジスタはタンパ検出イベントでリセットされます。最低 1 つの内部または外部タンパフラグがセットされている限り、強制的にリセット値となります。このレジスタは、読出し保護（RDP）が無効である場合もリセットされます。

47.6.20 TAMP レジスタマップ

表 497. TAMP レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	TAMP_CR1	Res.	ITAMP15E	Res.	ITAMP13E	ITAMP12E	ITAMP11E	Res.	ITAMP9E	ITAMP8E	ITAMP7E	ITAMP6E	ITAMP5E	ITAMP4E	ITAMP3E	ITAMP2E	ITAMP1E	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8E	TAMP7E	TAMP6E	TAMP5E	TAMP4E	TAMP3E	TAMP2E	TAMP1E					
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0					
0x04	TAMP_CR2	TAMP8TRG	TAMP7TRG	TAMP6TRG	TAMP5TRG	TAMP4TRG	TAMP3TRG	TAMP2TRG	TAMP1TRG	BKERASE	BKBLOCK	Res.	Res.	Res.	TAMP3MSK	TAMP2MSK	TAMP1MSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8POM	TAMP7POM	TAMP6POM	TAMP5POM	TAMP4POM	TAMP3POM	TAMP2POM	TAMP1POM					
	リセット値	0	0	0	0	0	0	0	0	0						0	0	0									0	0	0	0	0	0	0	0					
0x08	TAMP_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITAMP15POM	Res.	ITAMP13POM	ITAMP12POM	ITAMP11POM	Res.	ITAMP9POM	ITAMP8POM	ITAMP7POM	ITAMP6POM	ITAMP5POM	ITAMP4POM	ITAMP3POM	ITAMP2POM	ITAMP1POM						
	リセット値																		0		0	0	0		0	0	0	0	0	0	0	0	0						
0x0C	TAMP_FLTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMPPUDIS	TAMPPRCH[1:0]		TAMPFELT[1:0]		TAMPFREQ[2:0]								
	リセット値																									0	0	0	0	0	0	0	0	0					
0x10	TAMP_ATCR1	FLTEN	ATOSHARE	Res.	Res.	Res.	AT PER[2: 0]		Res.	Res.	Res.	Res.	Res.	ATCK SEL[3:0]			ATO SEL4 [1:0]		ATO SEL3 [1:0]		ATO SEL2 [1:0]		ATO SEL1 [1:0]		TAMP8AM	TAMP7AM	TAMP6AM	TAMP5AM	TAMP4AM	TAMP3AM	TAMP2AM	TAMP1AM							
	リセット値	0	0				0	0	0					0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x14	TAMP_ ATSEEDR	SEED[31:0]																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x18	TAMP_ATOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRNG[7:0]													
	リセット値																		0	0							0	0	0	0	0	0	0	0					
0x1C	TAMP_ATCR2	ATO SEL8 [2:0]		ATO SEL7 [2:0]		ATO SEL6 [2:0]		ATO SEL5 [2:0]		ATO SEL4 [2:0]		ATO SEL3 [2:0]		ATO SEL2 [2:0]		ATO SEL1 [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x20	TAMP_SEC CFGR	TAMPSEC	BKLOCK	Res.	Res.	Res.	Res.	Res.	Res.	BKPWSEC[7:0]								CNT1SEC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKPRWSEC[7:0]											
	リセット値	0	0							0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0					
0x24	TAMP_ PRIVCFGR	TAMPPRIV	BKPWPRIV	BKPRWPRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT1PRIV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.						
	リセット値	0	0	0						0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0					

表 497. TAMP レジスタのマッピングとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x2C	TAMP_IER	Res.	ITAMP15IE	Res.	ITAMP13IE	ITAMP12IE	ITAMP11IE	Res.	ITAMP9IE	ITAMP8IE	ITAMP7IF	ITAMP6IE	ITAMP5IE	ITAMP4IE	ITAMP3IE	ITAMP2IE	ITAMP1IE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8IE	TAMP7IE	TAMP6IE	TAMP5IE	TAMP4IE	TAMP3IE	TAMP2IE	TAMP1IE
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0	ITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.		0	0	0	0	0	0	0	
0x30	TAMP_SR	Res.	ITAMP15F	Res.	ITAMP13F	ITAMP12F	ITAMP11F	Res.	ITAMP9F	ITAMP8F	ITAMP7F	ITAMP6F	ITAMP5F	ITAMP4F	ITAMP3F	ITAMP2F	ITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8F	TAMP7F	TAMP6F	TAMP5F	TAMP4F	TAMP3F	TAMP2F	TAMP1F
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0	ITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.		0	0	0	0	0	0	0	
0x34	TAMP_MISR	Res.	ITAMP15MF	Res.	ITAMP13MF	ITAMP12MF	ITAMP11MF	Res.	ITAMP9MF	ITAMP8MF	ITAMP7MF	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.		0	0	0	0	0	0	0	
0x38	TAMP_SMISR	Res.	ITAMP15F	Res.	ITAMP13F	ITAMP12F	ITAMP11F	Res.	ITAMP9F	ITAMP8F	ITAMP7F	ITAMP6MF	ITAMP5MF	ITAMP4MF	ITAMP3MF	ITAMP2MF	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP8MF	TAMP7MF	TAMP6MF	TAMP5MF	TAMP4MF	TAMP3MF	TAMP2MF	TAMP1MF
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0	ITAMP1MF	Res.	Res.	Res.	Res.	Res.	Res.	Res.		0	0	0	0	0	0	0	
0x3C	TAMP_SCR	Res.	CITAMP15F	Res.	CITAMP13F	CITAMP12F	CITAMP11F	Res.	CITAMP9F	CITAMP8F	CITAMP7F	CITAMP6F	CITAMP5F	CITAMP4F	CITAMP3F	CITAMP2F	CITAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTAMP8F	CTAMP7F	CTAMP6F	CTAMP5F	CTAMP4F	CTAMP3F	CTAMP2F	CTAMP1F
	リセット値		0		0	0	0		0	0	0	0	0	0	0	0	0	CTAMP1F	Res.	Res.	Res.	Res.	Res.	Res.	Res.		0	0	0	0	0	0	0	
0x40	TAMP_COUNTR	COUNT[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50	TAMP_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN4_RMP	IN3_RMP	IN2_RMP	Res.	Res.	Res.	Res.	OUT5_RMP	OUT3_RMP	Res.		
	リセット値																					0	0	0					0	0	0			
0x54	TAMP_RPCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPCFG0		
	リセット値																																0	
0x100 + 0x04*x (x = 0 ~ 31)	TAMP_BKPxR	BKP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

48 I²C (Inter-integrated circuit) インタフェース

48.1 概要

I²C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I²C バス間の通信を処理します。マルチマスタ機能を備え、すべての I²C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準モード (Sm)、高速モード (Fm)、および高速モードプラス (Fm+) をサポートします。

また、I²C バスインタフェースは、SMBus (System Management Bus) および PMBus® (Power Management Bus) と互換性があります。

DMA を使用して、CPU の負荷を軽減できます。

48.2 I²C の主な機能

- I²C バス仕様 rev03 との互換性 :
 - スレーブおよびマスタモード
 - マルチマスタ機能
 - 標準モード (最大 100 kHz)
 - 高速モード (最大 400 kHz)
 - 高速モードプラス (最大 1 MHz)
 - 7 ビットおよび 10 ビットアドレッシングモード
 - 複数の 7 ビットスレーブアドレス (2 つのアドレス、1 つは設定可能なマスク付き)
 - すべての 7 ビットアドレス確認応答モード
 - 同報 (General call) コール
 - プログラム可能なセットアップおよびホールド時間
 - 使いやすいイベント管理
 - クロックストレッチオプション
 - ソフトウェアリセット
- DMA 機能付きの 1 バイトバッファ
- プログラム可能なアナログおよびデジタルノイズフィルタ

製品の実装によっては、次の機能も使用できます ([セクション 48.3](#) を参照) :

- SMBus 仕様 rev 3.0 との互換性 :
 - ハードウェア PEC (Packet Error Checking) の生成と ACK 制御による確認
 - コマンドおよびデータ確認応答制御
 - アドレス解決プロトコル (ARP) サポート
 - ホストおよびデバイスのサポート
 - SMBus アラート
 - タイムアウトおよびアイドル条件の検出
- PMBus rev 1.3 標準との互換性
- 独立したクロック : 独立したクロックソースの選択により、I²C の通信速度は i2c_pclk の再プログラミングから独立
- アドレス一致時に STOP モードからウェイクアップ

48.3 I²C の実装

表 498. STM32H563/H573および STM32H562I2C の実装

I2C の機能 ⁽¹⁾	I2C1	I2C2	I2C3	I2C4
7 ビットアドレスモード	X	X	X	X
10 ビットアドレスモード	X	X	X	X
標準モード (最大 100 kbit/s)	X	X	X	X
高速モード (最大 400 kbit/s)	X	X	X	X
20mA 出力駆動 I/O 搭載高速モードプラス (最大 1 Mbit/s)	X	X	X	X
独立クロック	X	X	X	X
STOP モードからのみウェイクアップ (非自律モード)	X	X	X	X
SMBus/PMBus	X	X	X	X

1. X : サポートされています。

48.4 I²C の機能詳細

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I²C バスに接続されます。標準（最大 100 kHz）、高速モード（最大 400 kHz）、または高速モードプラス（最大 1 MHz）の I²C バスで接続できます。

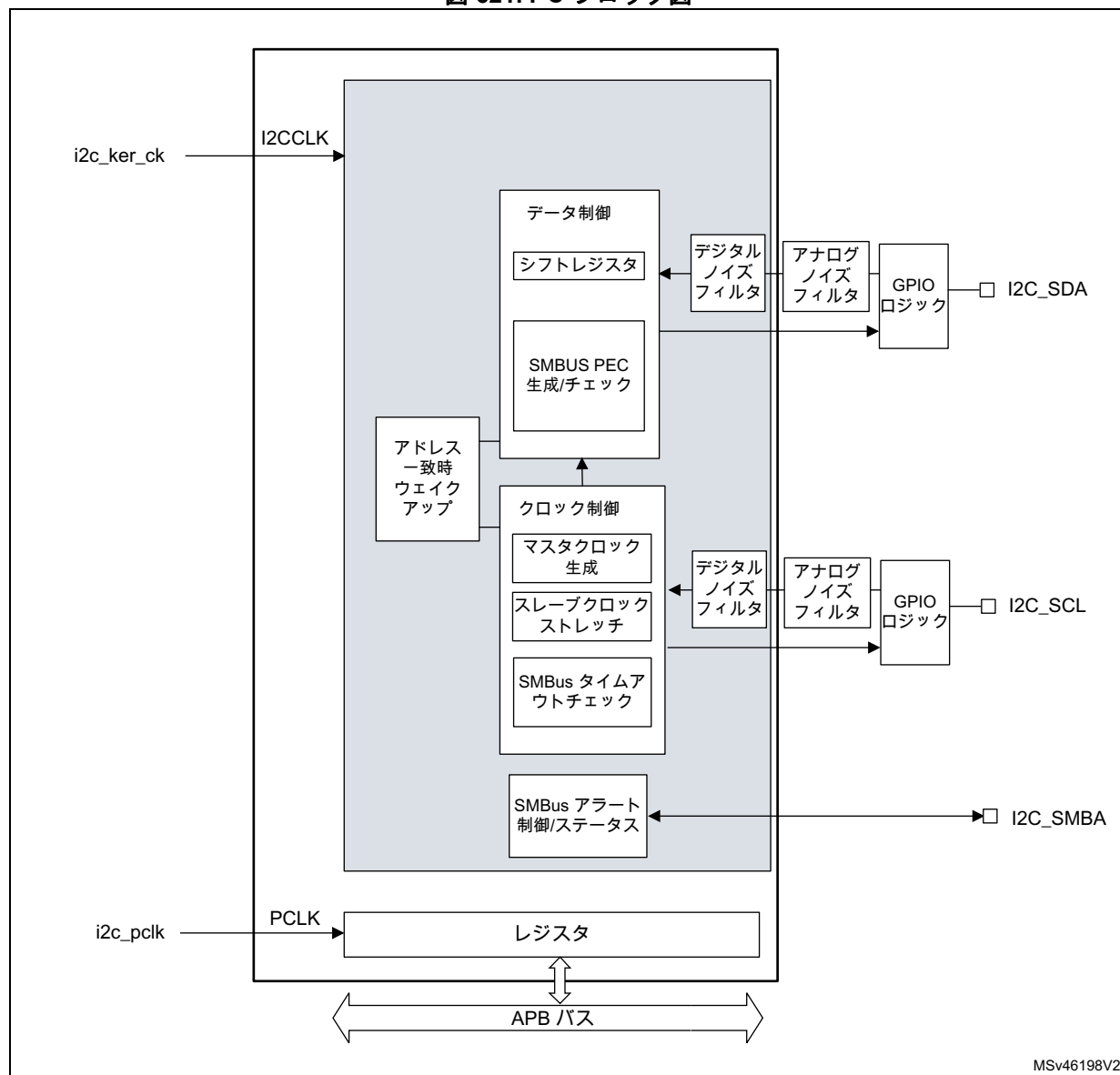
このインタフェースは、データ（SDA）ピンとクロック（SCL）ピンによって SMBus に接続することもできます。

SMBus 機能がサポートされる場合、オプションの SMBus アラートピン（SMBA）も使用できます。

48.4.1 I²C ブロック図

I²C インタフェースのブロック図を図 621 に示します。

図 621. I²C ブロック図



I²C は、独立したクロックソースによってクロック供給されるため、I²C は i2c_pclk 周波数から独立して動作できます。

48.4.2 I²C ピンおよび内部信号

表 499. I²C の入出力ピン

ピン名	信号タイプ	説明
I2C_SDA	双方向	I2C データ
I2C_SCL	双方向	I2C クロック
I2C_SMBA	双方向	SMBus アラート

表 500. I²C 内部入力／出力信号

内部信号名	信号タイプ	説明
i2c_ker_ck	入力	I ² C カーネルクロック（本書では I2CCLK とも言います）
i2c_pclk	入力	I ² C APB クロック
i2c_it	出力	I ² C 割込み。割込みソースのリストは、表 513 を参照してください。
i2c_rx_dma	出力	I ² C 受信データ DMA リクエスト (I2C_RX)
i2c_tx_dma	出力	I ² C 送信データ DMA リクエスト (I2C_TX)

48.4.3 I²C クロックの要件

I²C カーネルは i2c_ker_ck によってクロック供給されます。

i2c_ker_ck の周期 t_{I2CCLK} は、次の条件を満たす必要があります。

- $t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$
- $t_{I2CCLK} < t_{HIGH}$

ここで：

t_{LOW} ：SCL ロー時間、および t_{HIGH} ：SCL ハイ時間

$t_{filters}$ ：アナログフィルタとデジタルフィルタ（有効な場合）による遅延の合計。

デジタルフィルタの遅延は、 $DNF \times t_{I2CCLK}$ です。

i2c_pclk クロックの周期 t_{PCLK} は、次の条件を満たす必要があります。

- $t_{PCLK} < 4/3 t_{SCL}$ (t_{SCL} : SCL 周期)

注意： I²C カーネルが i2c_pclk によってクロック供給されるとき、このクロックは t_{I2CCLK} の条件を満たす必要があります。

48.4.4 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成したときにはスレーブからマスタへ、アービトレーションの喪失または STOP 生成が発生したときにはマスタからスレーブへ自動的に切り替わるため、マルチマスタ機能を使用できます。

通信の流れ

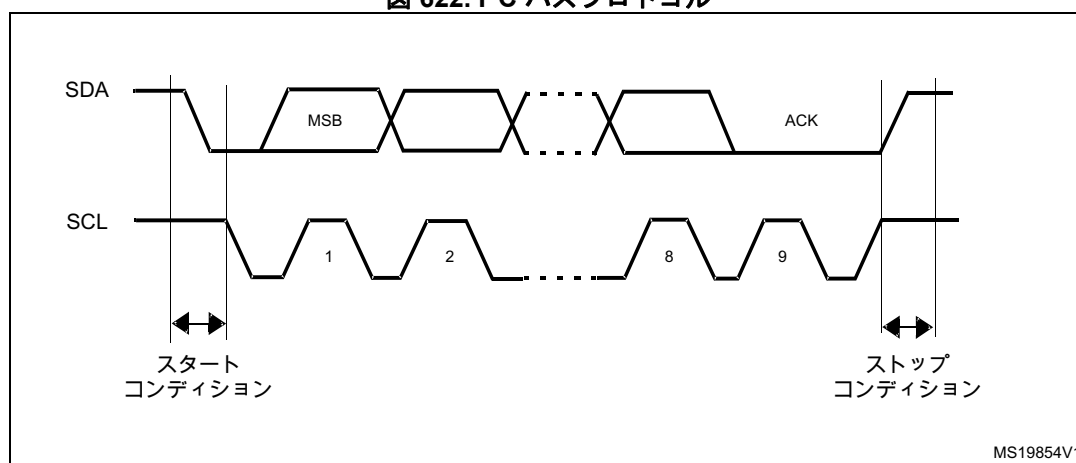
マスタモードでは、I²C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス（7 または 10 ビット）と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効または無効にできます。予約済みの SMBus アドレスもソフトウェアによって有効にできます。

データとアドレスは、MSBが先の 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト（7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト）にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 つの クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります（[図 622](#) を参照）。

図 622. I²C バスプロトコル



確認応答 (Acknowledge) は、ソフトウェアによって有効または無効にできます。I²C インタフェースのアドレスは、ソフトウェアによって選択できます。

48.4.5 I²C の初期化

ペリフェラルの有効化と無効化

クロックコントローラで I²C ペリフェラルクロックを設定して有効にする必要があります。その後 I²C は、I2C_CR1 レジスタの PE ビットをセットすることで有効にできます。

I²C が無効にされると (PE = 0)、I²C はソフトウェアリセットを実行します。詳細については、[セクション 48.4.6](#) を参照してください。

ノイズフィルタ

I2C_CR1 レジスタの PE ビットをセットすることによって I²C ペリフェラルを有効にする前に、必要な場合は、ノイズフィルタを設定する必要があります。デフォルトでは、SDA および SCL 入力にアナログノイズフィルタがあります。このフィルタは I²C 仕様に準拠しており、高速モードおよび高速モードプラスで最大 50 ns のパルス幅を持つスパイクを抑制します。ANOFF ビットをセットすることによって、このアナログフィルタを無効にし、I2C_CR1 レジスタの DNF[3:0] ビットを設定することによってデジタルフィルタを選択することができます。

デジタルフィルタが有効なときには、SCL または SDA ラインのレベルは、DNF x i2c_ker_ck 周期より長く安定していた場合のみ、内部で変更されます。これにより、プログラム可能な 1~15 i2c_ker_ck 周期の長さを持つスパイクを抑制できます。

表 501. アナログフィルタとデジタルフィルタの比較

-	アナログフィルタ	デジタルフィルタ
抑制されるスパイクの パルス幅	≥ 50 ns	長さを I2C ペリフェラルクロック 1 ~ 15 にプログラム 可能
利点	STOP モードで使用可能	– プログラム可能な長さ: 標準要件に対して追加のフィル タリング機能 – 安定した長さ
欠点	温度、電圧、プロセスの ばらつき	デジタルフィルタが有効なときには、アドレス一致時の STOP モードからのウェイクアップは使用できない

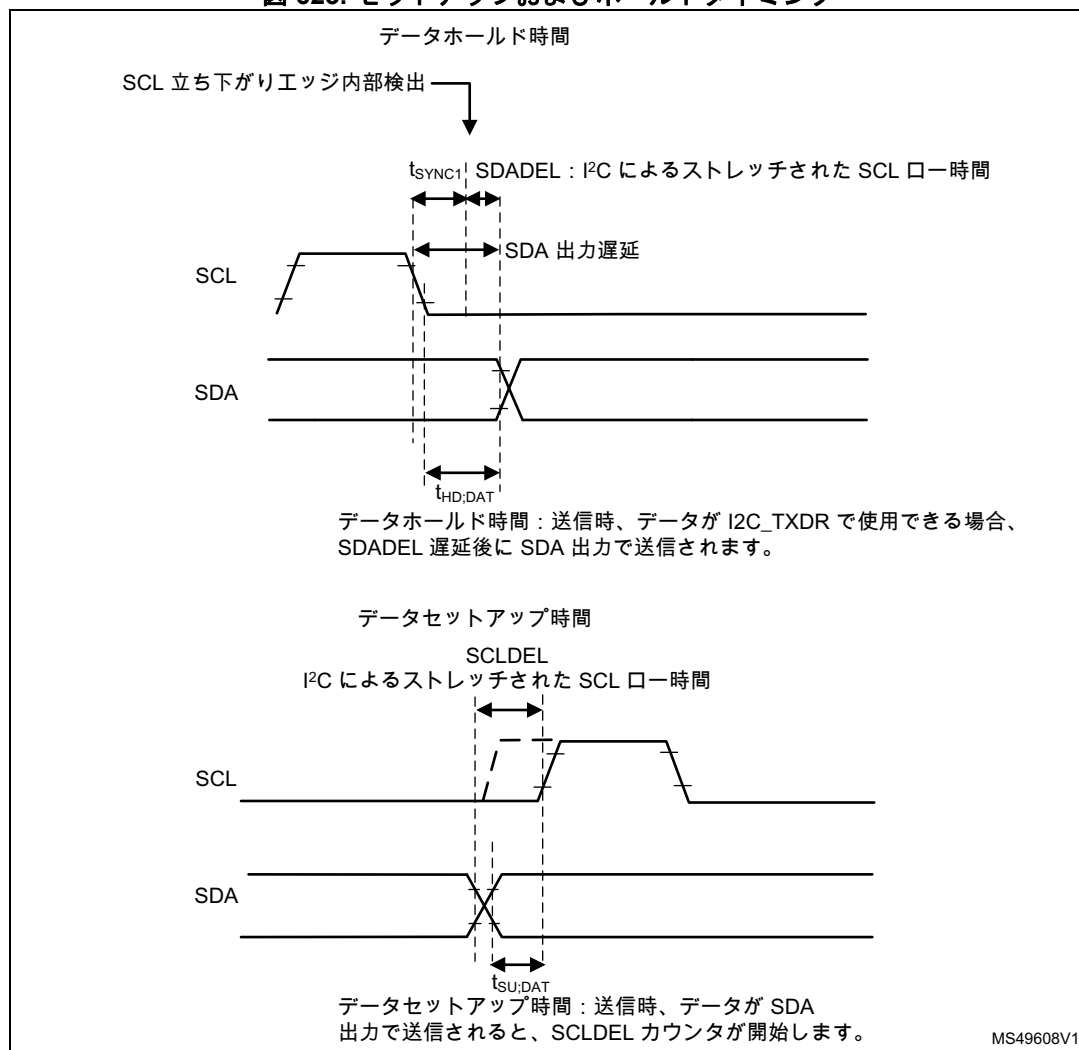
注意 : I²C が有効なときには、フィルタ構成は変更できません。

I²C のタイミング

マスタおよびスレーブモードで正しいデータホールドおよびセットアップ時間を保証するためには、タイミングを設定する必要があります。これを行うには、I2C_TIMINGR レジスタの PRESC[3:0]、SCLDEL[3:0]、および SDADEL[3:0] ビットをプログラムします。

STM32CubeMX ツールは、I²C 設定ウィンドウの I2C_TIMINGR コンテンツを計算し、提供します。

図 623. セットアップおよびホールドタイミング



SCL の立下がりエッジが内部で検出されると、遅延（ホールド時間 $t_{\text{HD,DAT}}$ へ影響を与える t_{SDADEL} ）が SDA 出力を送信する前に挿入されます。 $t_{\text{SDADEL}} = \text{SDADEL} \times t_{\text{PRESC}} + t_{\text{I2CCLK}}$ 、ここで $t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{I2CCLK}}$ 。

SDA 出力遅延の合計は、次のとおりです：

$$t_{\text{SYNC1}} + \{[\text{SDADEL} \times (\text{PRESC} + 1) + 1] \times t_{\text{I2CCLK}}\}$$

t_{SYNC1} 時間は次のものに依存します。

- － SCL 立下がり傾斜
- － アナログフィルタが有効なときの入力遅延： $t_{\text{AF(min)}} < t_{\text{AF}} < t_{\text{AF(max)}}$
- － デジタルフィルタが有効なときの入力遅延： $t_{\text{DNF}} = \text{DNF} \times t_{\text{I2CCLK}}$
- － i2c_ker_ck クロックへの SCL の同期による遅延（2 ～ 3 i2c_ker_ck 周期）

SCL 立下がりエッジの未定義の領域をブリッジするためには、SDADEL を次のようにプログラムする必要があります。

$$\{t_{\text{f(max)}} + t_{\text{HD,DAT(min)}} - t_{\text{AF(min)}} - [(\text{DNF} + 3) \times t_{\text{I2CCLK}}]\} / \{(\text{PRESC} + 1) \times t_{\text{I2CCLK}}\} \leq \text{SDADEL}$$

$$\text{SDADEL} \leq \{t_{\text{HD,DAT(max)}} - t_{\text{AF(max)}} - [(\text{DNF} + 4) \times t_{\text{I2CCLK}}]\} / \{(\text{PRESC} + 1) \times t_{\text{I2CCLK}}\}$$

注： $t_{AF(min)} / t_{AF(max)}$ は、アナログフィルタが有効なときのみ、等式に含まれます。 t_{AF} の値については、デバイスのデータシートを参照してください。

最大の $t_{HD;DAT}$ は、標準モードで 3.45 μ s、高速モードで 0.9 μ s、高速モードプラスで 0.45 μ s となる可能性があります。これは、 $t_{VD;DAT}$ の最大値よりも遷移時間だけ小さくしなければなりません。この最大値を満たす必要があるのは、デバイスが SCL 信号の LOW 周期 (t_{LOW}) をストレッチしない場合だけです。クロックが SCL をストレッチする場合、クロックをリリースする前に、データがセットアップ時間まで有効である必要があります。

通常、SDA の立上がりエッジが最悪ケースです。この場合、前の式は次のようになります。

$$SDADEL \leq \{t_{VD;DAT(max)} - t_r(max) - t_{AF(max)} - [(DNF + 4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\}.$$

注： **NOSTRETCH = 0** のときには、SCLDEL の値に従って、デバイスはセットアップ時間を保証するために SCL ローをストレッチするので、この条件に違反することがあります。

t_r 、 t_f 、 $t_{HD;DAT}$ 、および $t_{VD;DAT}$ の標準値については、表 502 を参照してください。

- t_{SDADEL} の後や、データが I2C_TXDR レジスタにまだ書き込まれていないためスレーブがクロックをストレッチしなければならなかった場合の SDA 出力送信後、SCL ラインはセットアップ時間中、ローレベルで保持されます。このセットアップ時間は、 $t_{SCLDEL} = (SCLDEL + 1) \times t_{PRESC}$ であり、ここで、 $t_{PRESC} = (PPRESC + 1) \times t_{I2CCLK}$ です。 t_{SCLDEL} はセットアップ時間 $t_{SU;DAT}$ に影響を与えます。

SDA 遷移 (立上がりエッジは通常、最悪のケース) の未定義の領域をブリッジするためには、SCLDEL を次のようにプログラムする必要があります。

$$\{[t_r(max) + t_{SU;DAT(min)}] / [(PRESC + 1) \times t_{I2CCLK}]\} - 1 \leq SCLDEL$$

t_r および $t_{SU;DAT}$ の標準値については、表 502 を参照してください。

使用する SDA および SCL 遷移時間の値は、アプリケーションでの値です。標準から最大値を使用すると、SDADEL と SCLDEL の計算の制約が増えますが、アプリケーションにかかわらず、この機能を使用できます。

注： 各クロックパルスで、SCL 立下がりエッジの検出後、I2C マスタまたはスレーブは、最低 $[(SDADEL + SCLDEL + 1) \times (PRESC + 1) + 1] \times t_{I2CCLK}$ の間、送信および受信の両モードで SCL ローをストレッチします。送信モードで、SDADEL カウンタ終了時にデータがまだ I2C_TXDR に書き込まれていない場合、I2C は次のデータが書き込まれるまで SCL ローをストレッチし続けます。その時、新しいデータ MSB が SDA 出力で送信され、SCLDEL カウンタが開始し、SCL ローのストレッチを継続して、データセットアップ時間を保証します。

スレーブモードで NOSTRETCH = 1 の場合、SCL はストレッチされません。そのため、十分なセットアップ時間が保証されるように、SDADEL をプログラムする必要があります。

表 502. I²C-SMBus 仕様のデータのセットアップおよびホールド時間

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBus		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
$t_{HD;DAT}$	データホールド時間	0	-	0	-	0	-	0.3	-	μ s
$t_{VD;DAT}$	データ有効時間	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	データセットアップ時間	250	-	100	-	50	-	250	-	ns
t_r	SDA および SCL の両信号の立上がり時間	-	1000	-	300	-	120	-	1000	
t_f	SDA および SCL の両信号の立下がり時間	-	300	-	300	-	120	-	300	

また、マスタモードでは、I2C_TIMINGR レジスタの PRESC[3:0]、SCLH[7:0]、および SCLL[7:0] ビットフィールドをプログラムすることによって、SCL クロックのハイおよびローレベルを設定する必要があります。

- SCL 立下がりエッジが内部で検出されると、SCL 出力をリリースする前に遅延が挿入されます。この遅延は、 $t_{SCLL} = (SCLL + 1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$ です。 t_{SCLL} は、SCL ロー時間 t_{LOW} に影響を与えます。
- SCL 立上がりエッジが内部で検出されると、SCL 出力を強制的にローレベルにする前に遅延が挿入されます。この遅延は、 $t_{SCLH} = (SCLH + 1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC + 1) \times t_{I2CCLK}$ です。 t_{SCLH} は SCL ハイ時間 t_{HIGH} に影響を与えます。

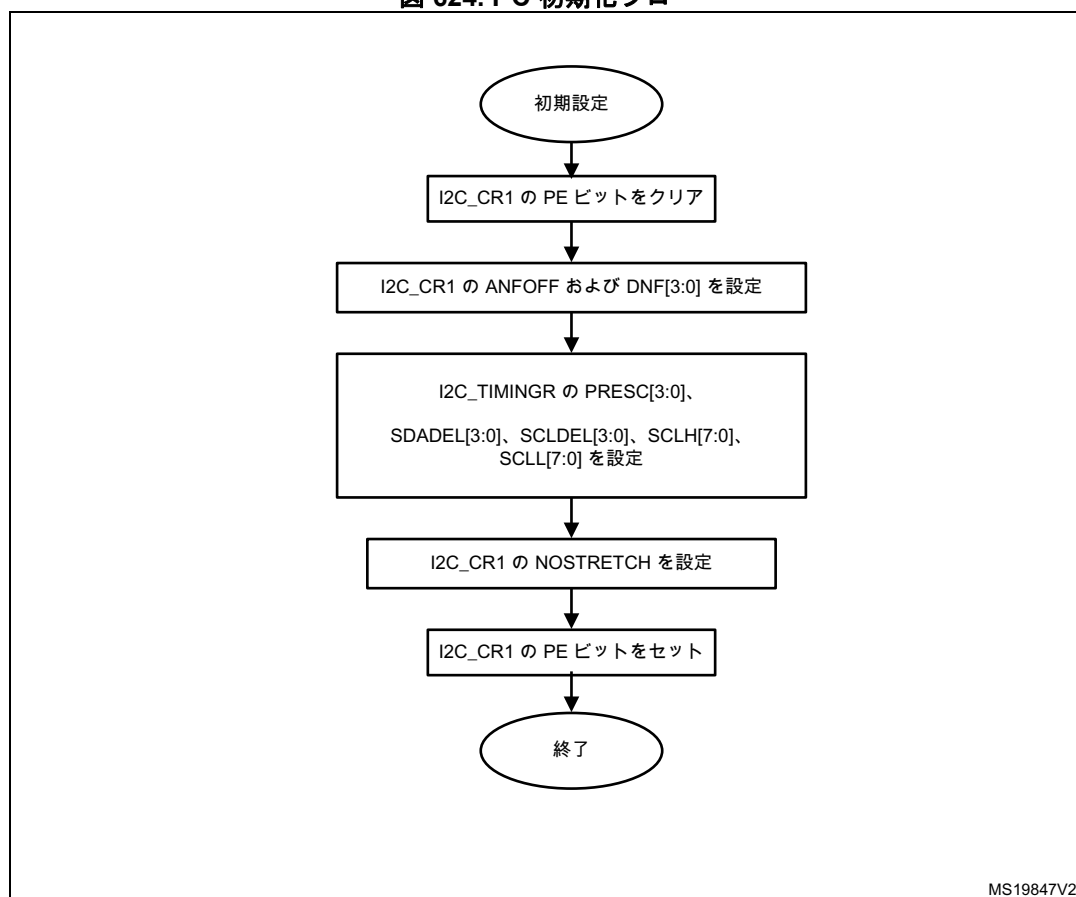
詳細については、[I2C マスタ初期化](#)を参照してください。

注意： I²C が有効なときには、タイミング構成の設定はできません。

ペリフェラルを有効にする前に、I2C スレーブ NOSTRETCH モードも設定する必要があります。詳細については、[I2C スレーブ初期化](#)を参照してください。

注意： I²C が有効なときには、NOSTRETCH 構成の変更はできません。

図 624. I²C 初期化フロー



MS19847V2

48.4.6 ソフトウェアリセット

ソフトウェアリセットを行うには、I2C_CR1 レジスタの PE ビットをクリアします。その場合、I2C のライン SCL および SDA がリリリースされます。内部ステートマシンがリセットされ、通信制御ビットとステータスビットがリセット値に戻ります。構成レジスタは影響を受けません。

影響を受けるレジスタのビット：

1. I2C_CR2 レジスタ：START、STOP、NACK
2. I2C_ISR レジスタ：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO、OVR

さらに、SMBus 機能がサポートされているときには、以下も影響を受けます：

1. I2C_CR2 レジスタ：PECBYTE
2. I2C_ISR レジスタ：PECERR、TIMEOUT、ALERT

ソフトウェアリセットを実行するためには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たなければなりません。これは、次のソフトウェア手順によって確保されます。

1. PE = 0 を書き込む。
2. PE = 0 であることを確認します。
3. PE = 1 を書き込む。

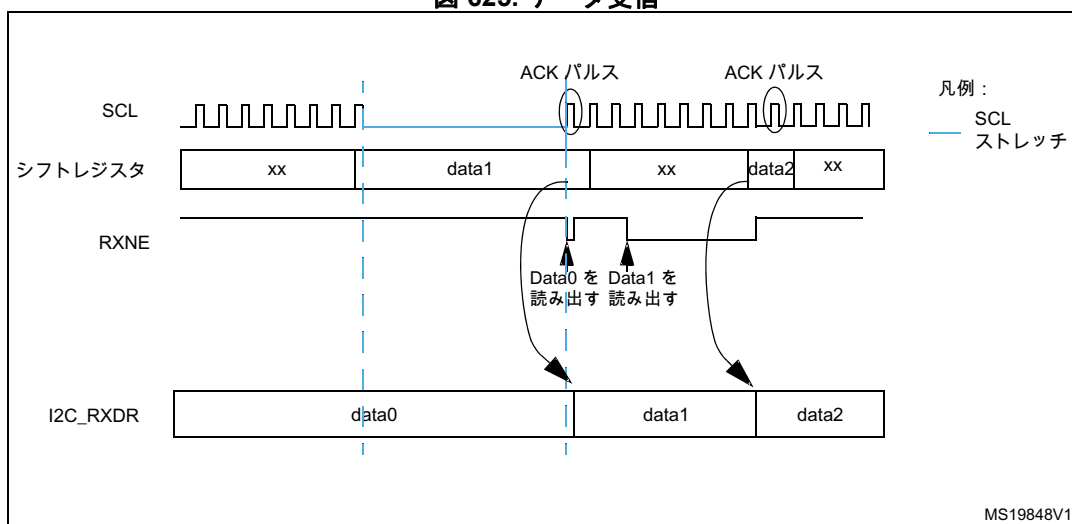
48.4.7 データ転送

データ転送は、送受信データレジスタとシフトレジスタを通じて管理されます。

受信

SDA 入力はシフトレジスタに送られます。8 番目の SCL パルスの後（完全なデータバイトの受信後）、シフトレジスタは、I2C_RXDR レジスタが空の場合（RXNE = 0）、このレジスタにコピーされます。RXNE = 1 の場合、すなわち、前に受信されたデータバイトがまだ読み出されていなかった場合、SCL ラインは I2C_RXDR が読み出されるまでストレッチされます。ストレッチは、8 番目と 9 番目の SCL パルスの間（確認応答パルスの前）に挿入されます。

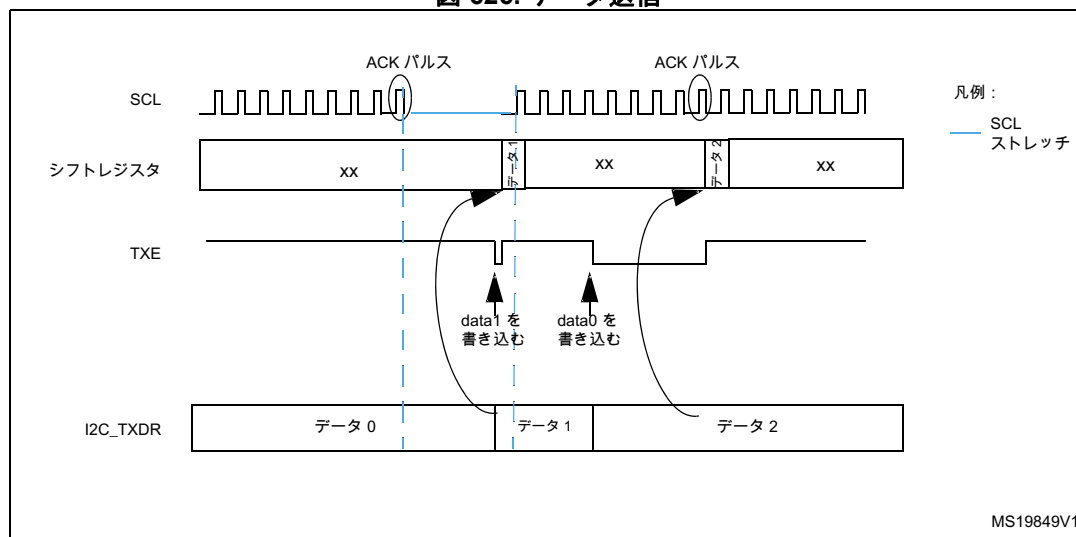
図 625. データ受信



送信

I2C_TXDR レジスタが空でない場合 (TXE = 0)、その内容が 9 番目の SCL パルス (確認応答パルス) の後、シフトレジスタにコピーされます。次に、シフトレジスタの内容が SDA ラインにシフトアウトされます。TXE = 1 の場合、すなわち、I2C_TXDR にデータがまだ書き込まれていない場合、SCL ラインは I2C_TXDR に書き込まれるまでストレッチされます。ストレッチは、9 番目の SCL パルスの後で行われます。

図 626. データ送信



ハードウェア転送管理

次のようにさまざまなモードでバイト転送を管理し、通信をクローズするために、I2C にはバイトカウンタが組み込まれています。

- マスタモードでの NACK、STOP、および ReSTART 生成
- スレーブレシーバモードでの ACK 制御
- SMBus 機能がサポートされているときの PEC 生成/確認

バイトカウンタは、マスタモードでは常に使用されます。デフォルトでは、スレーブモードでは無効になっています。I2C_CR1 レジスタの SBC (スレーブバイト制御) ビットをセットすることによって、ソフトウェアにより有効にできます。

転送されるバイト数は、I2C_CR2 レジスタの NBYTES[7:0] ビットフィールドでプログラムされます。転送バイト数 (NBYTES) が 255 より大きい場合、またはレシーバが受信データバイトの確認応答値を制御したい場合には、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。このモードでは、NBYTES でプログラムされたバイト数が転送されると、TCR フラグがセットされ、TCIE がセットされている場合は割込みが生成されます。SCL は、TCR フラグがセットされている間、ストレッチされます。TCR は、NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

NBYTES カウンタに最後のバイト数が再ロードされたときには、RELOAD ビットがクリアされる必要があります。

マスタモードで RELOAD = 0 のときには、カウンタは以下の 2 つのモードで使用できます。

- **自動終了モード** (I2C_CR2 レジスタの AUTOEND = 1)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、マスタは STOP コンディションを自動的に送信します。
- **ソフトウェア終了モード** (I2C_CR2 レジスタの AUTOEND = 0)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、ソフトウェアアクションが求められます。TC フラグがセットされ、TCIE ビットがセットされている場合は割込みが生成されます。SCL 信号は、TC フラグがセットされている間、ストレッチされます。TC フラグは、I2C_CR2 レジスタの START または STOP ビットがセットされたときに、ソフトウェアによってクリアされます。マスタが RESTART コンディションを送信したいときには、このモードを使用する必要があります。

注意： AUTOEND ビットは、RELOAD ビットがセットされているときには効果がありません。

表 503. I2C 設定

機能	SBC ビット	RELOAD ビット	AUTOEND ビット
マスタ Tx/Rx NBYTES + STOP	x	0	1
マスタ Tx/Rx + NBYTES + RESTART	x	0	0
スレーブ Tx/Rx、すべての受信バイトに ACK	0	x	x
スレーブ Rx および ACK 制御	1	1	x

48.4.8 I²C スレーブモード

I²C スレーブ初期化

スレーブモードで動作するには、少なくとも 1 つのスレーブアドレスを有効にする必要があります。レジスタ I2C_OAR1 と I2C_OAR2 を使用して、スレーブ専用アドレス OA1 および OA2 をプログラムできます。

- OA1 は、I2C_OAR1 レジスタの OA1MODE ビットをセットすることによって、7 ビットモード (デフォルト)、または 10 ビットアドレッシングモードに設定できます。

OA1 を有効にするには、I2C_OAR1 レジスタの OA1EN ビットをセットします。

- 追加のスレーブアドレスが必要な場合は、2 番目のスレーブアドレス OA2 を設定できます。I2C_OAR2 レジスタの OA2MSK[2:0] ビットを設定することによって、最大 7 つの OA2 LSB をマスクできます。したがって、OA2MSK が 1 から 6 まで設定された場合、OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]、または OA2[7] のみが受信アドレスと比較されます。OA2MSK が 0 に等しくなくなるとすぐに、OA2 のアドレスコンパレータは、確認応答されない I2C 予約済みアドレス (0000 XXX および 1111 XXX) を除外します。OA2MSK = 7 の場合、受信されたすべての 7 ビットのアドレスが確認応答されます (予約済みアドレスを除く)。OA2 は常に 7 ビットアドレスです。

これらの予約済みアドレスは、特定のイネーブルビットによって有効化された場合、I2C_OAR1 または I2C_OAR2 レジスタが OA2MSK = 0 でプログラムされた場合、確認応答できます。

OA2 を有効にするには、I2C_OAR2 レジスタの OA2EN ビットをセットします。

- 同報アドレスは、I2C_CR1 レジスタの GCEN ビットをセットすることで有効になります。

I2C が有効アドレスの 1 つによって選択されると、ADDR 割込みステータスフラグがセットされ、ADDRIE ビットがセットされている場合は割込みが生成されます。

デフォルトでは、スレーブはクロックストレッチ機能を使用します。これは、必要なときには、ソフトウェアアクションを実行するために、SCL 信号をローレベルでストレッチすることを意味します。マスタがクロックストレッチをサポートしない場合、I2C_CR1 レジスタの NOSTRETCH = 1 で I2C を設定する必要があります。

ADDR 割込みの受信後、いくつかのアドレスが有効な場合は、I2C_ISR レジスタの ADDCODE[6:0] ビットを読み出して、一致するアドレスを確認する必要があります。転送方向を知るために、DIR フラグも確認する必要があります。

スレーブクロックストレッチ (NOSTRETCH = 0)

デフォルトモードでは、I2C スレーブは次の状況で SCL クロックをストレッチします：

- ADDR フラグがセットされると：受信アドレスは有効なスレーブアドレスの 1 つと一致します。このストレッチは、ADDRCF ビットをセットすることによりソフトウェアによって ADDR フラグがクリアされたときにリリースされます。
- 送信時、前のデータ送信が完了し、新しいデータが I2C_TXDR レジスタに書き込まれなかった場合、または ADDR フラグがクリアされたときに (TXE = 1)、最初のデータバイトが書き込まれていなかった場合。このストレッチは、データが I2C_TXDR レジスタに書き込まれたときにリリースされます。
- 受信時、I2C_RXDR レジスタがまだ読み出されておらず、新しいデータ受信が完了したとき。このストレッチは、I2C_RXDR が読み出されたときにリリースされます。
- スレーブバイト制御モードおよび再ロードモード (SBC = 1 および RELOAD = 1) で TCR = 1 のとき、すなわち、最後のデータバイトが転送されたとき。このストレッチは、NBYTES[7:0] フィールドにゼロ以外の値を書き込むことによって TCR がクリアされたときにリリースされます。
- SCL 立下がりエッジの検出後、I2C は、 $[(SDADEL + SCLDEL + 1) \times (PRESC + 1) + 1] \times t_{I2CCCLK}$ の間、SCL ローをストレッチします。

クロックストレッチなしのスレーブ (NOSTRETCH = 1)

I2C_CR1 レジスタの NOSTRETCH = 1 のとき、I2C スレーブは SCL 信号をストレッチしません。

- ADDR フラグがセットされている間、SCL クロックはストレッチされません。
- 送信時、転送に対応する最初の SCL パルスが発生する前に、I2C_TXDR レジスタにデータが書き込まれる必要があります。そうでない場合、アンダーランが発生し、I2C_ISR レジスタで OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。OVR フラグは、最初のデータ送信が開始し、STOPF ビットがまだセットされている (クリアされていない) ときにもセットされます。したがって、次の転送で送信される最初のデータを書き込んだ後でのみ、前に転送の STOPF フラグをクリアすることによって、送信される最初のデータについても、OVR ステータスが提供されることを確実にできます。
- 受信時、次のデータバイトの 9 番目の SCL パルス (ACK パルス) が発生する前に、I2C_RXDR レジスタからデータが読み出される必要があります。そうでない場合、オーバーランが発生し、I2C_ISR レジスタの OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

スレーブバイト制御モード

スレーブ受信モードでバイト ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。これは、SMBus 標準に準拠する必要があります。

スレーブ受信モードでバイト ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD = 1)。各バイトの制御を得るには、ADDR 割込みサブルーチンで NBYTES を 0x1 に初期化し、各受信バイト後に 0x1 に再ロードする必要があります。バイトが受信されると、TCR ビットがセットされ、8 番目と 9 番目の SCL パルスの間で、SCL 信号ローをストレッチします。I2C_RXDR レジスタからデータを読み出すことができ、その後、I2C_CR2 レジスタの ACK ビットを設定することによって、確認応答するかどうかを決定できます。SCL ストレッチは、NBYTES をゼロ以外の値にプログラムすることによってリリースされ、確認応答または非確認応答が送信され、次のバイトを受信できます。

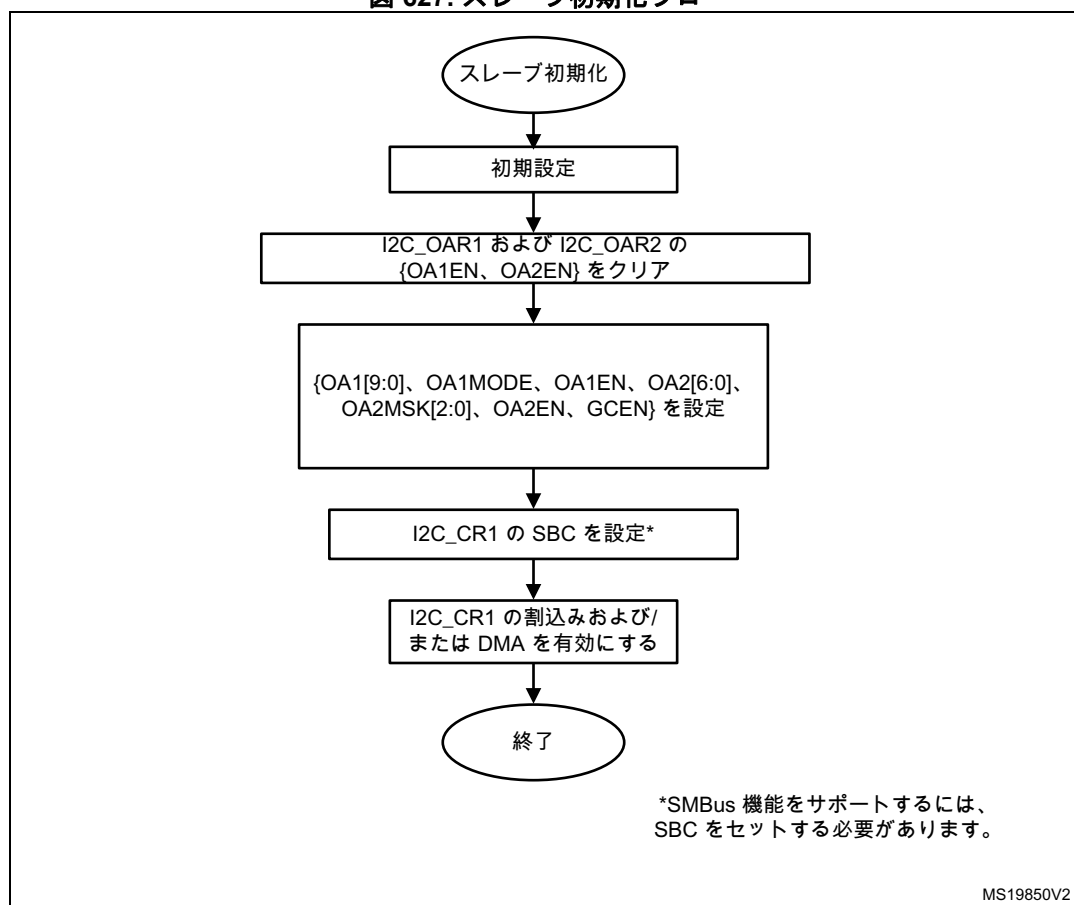
NBYTES に 0x1 より大きい値をロードでき、この場合、受信フローは NBYTES データ受信で、続きます。

注： SBC ビットは、I2C が無効なとき、またはスレーブがアドレス指定されていないとき、または ADDR = 1 のときに設定する必要があります。

RELOAD ビットの値は、ADDR = 1 のとき、または TCR = 1 のときに変更できます。

注意： スレーブバイト制御モードは、NOSTRETCH モードと互換性がありません。NOSTRETCH = 1 のときに SBC をセットすることはできません。

図 627. スレーブ初期化フロー



スレーブトランスミッタ

I2C_TXDR レジスタが空になると、送信割込みステータス (TXIS) が生成されます。I2C_CR1 レジスタの TXIE ビットがセットされている場合は、割込みが生成されます。

TXIS ビットは、I2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

NACK が受信されると、I2C_ISR レジスタの NACKF ビットがセットされ、I2C_CR1 レジスタの NACKIE ビットがセットされていた場合は割込みが生成されます。マスタが STOP または RESTART コンディションを実行できるように、スレーブは SCL および SDA ラインを自動的にリリースします。TXIS ビットは、NACK 受信時にはセットされません。

STOP が受信され、I2C_CR1 レジスタの STOPIE ビットがセットされると、I2C_ISR レジスタの STOPF フラグがセットされ、割込みが生成されます。ほとんどのアプリケーションでは、SBC ビットは通常、0 にプログラムされます。この場合、スレーブアドレスが受信されたときに (ADDR = 1)、TXE = 0 であった場合、I2C_TXDR レジスタの内容を最初のデータバイトとして送信するか、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C_TXDR レジスタを一掃するかを選択できます。

スレーブバイト制御モード (SBC = 1) では、送信バイト数をアドレス一致割込みサブルーチンの NBYTES でプログラムする必要があります (ADDR = 1)。この場合、転送中の TXIS イベントの数は、NBYTES でプログラムされた値に対応します。

注意 : NOSTRETCH = 1 のとき、SCL クロックは ADDR フラグがセットされている間はストレッチされないため、最初のデータバイトをプログラムするために ADDR サブルーチンで I2C_TXDR レジスタの内容を一掃することはできません。最初に送信されるデータバイトは、I2C_TXDR レジスタで前もってプログラムされている必要があります。

- このデータは、前の送信メッセージの最後の TXIS イベントで書き込まれたデータでもかまいません。
- このデータバイトが送信データバイトでない場合、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C_TXDR レジスタを一掃できます。アドレスの確認応答に続いて、最初のデータ送信が開始する前にこれらが実行されることを保証するためには、STOPF ビットのクリアは、これらのアクションの後でのみ行う必要があります。

最初のデータ送信が開始したときに STOPF がまだセットされていた場合、アンダーランエラーが生成されます (OVR フラグがセットされます)。

TXIS イベントが必要な場合 (送信割込みまたは送信 DMA リクエスト)、TXIS イベントを生成するためには、TXE ビットに加えて TXIS ビットもセットする必要があります。

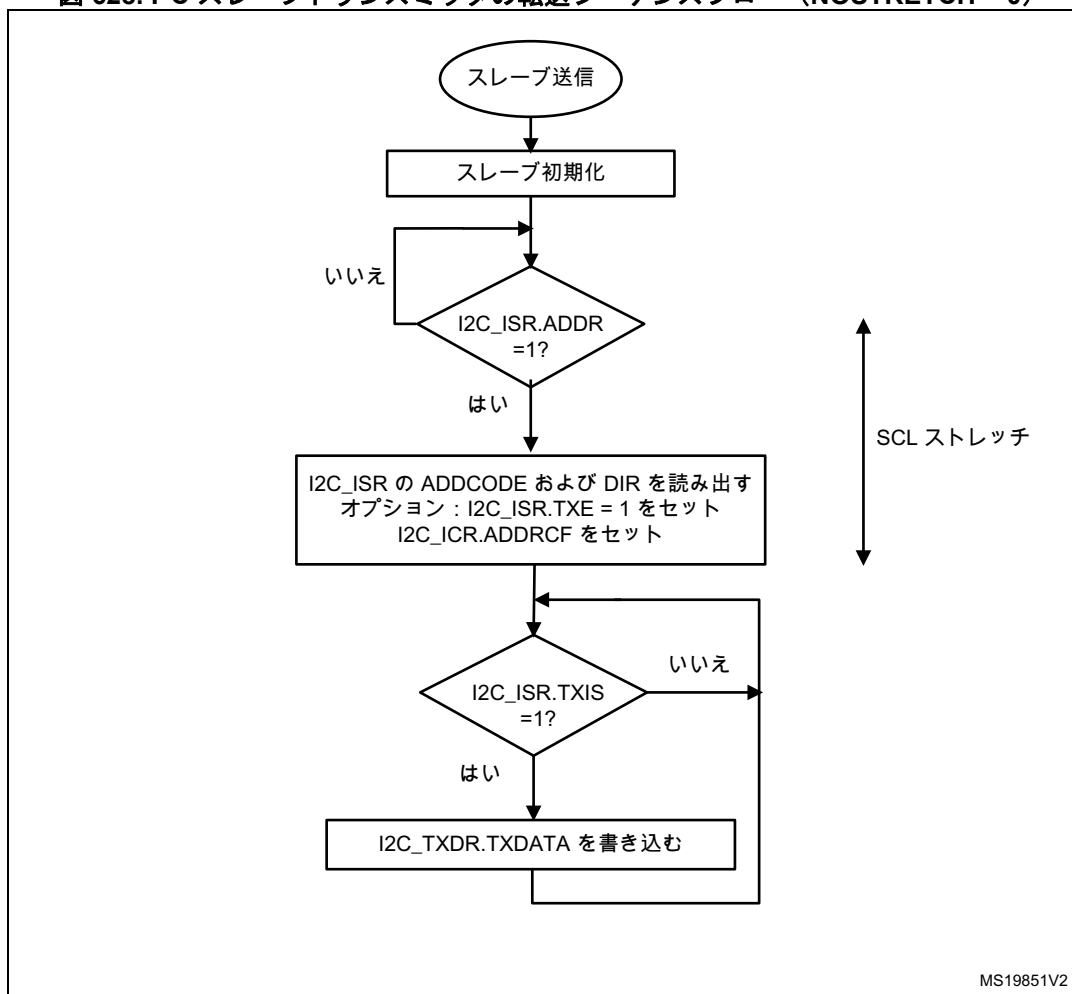
図 628. I²C スレーブトランスミッタの転送シーケンスフロー (NOSTRETCH = 0)

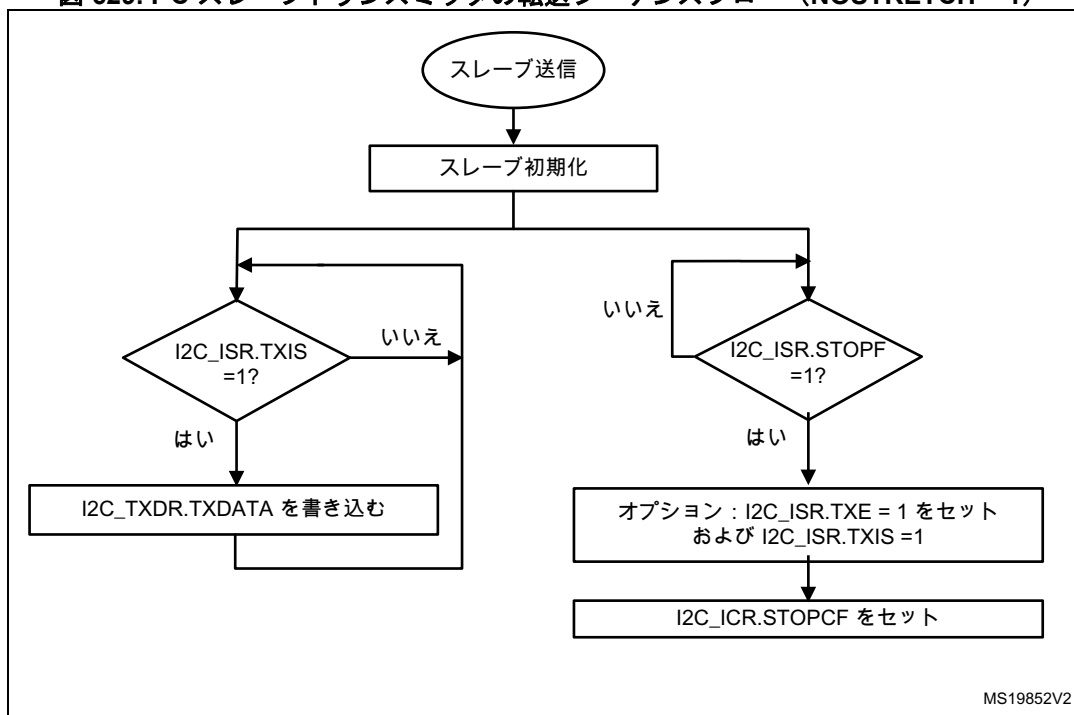
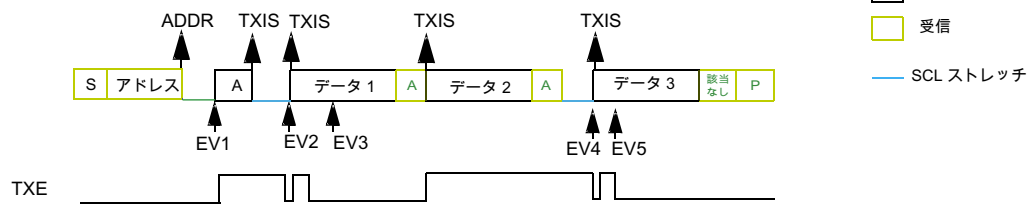
図 629. I²C スレーブトランスミッタの転送シーケンスフロー (NOSTRETCH = 1)

図 630. I²C スレーブトランスミッタの転送バス図 (必須イベントのみ)

最初のデータを一掃する I²C スレーブトランスミッタ 3 バイト、
NOSTRETCH=0 の例 :



EV1 : ADDR ISR : ADDCODE および DIR をチェック、TXE をセット、ADDRCF をセット

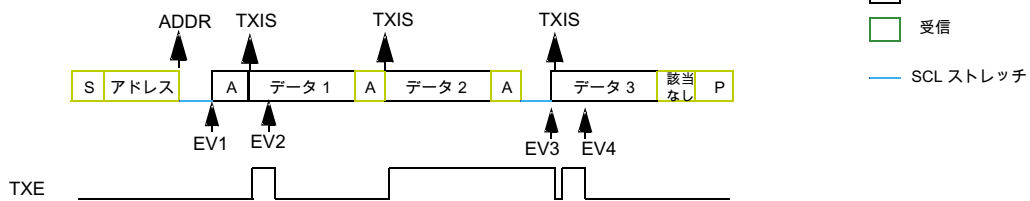
EV2 : TXIS ISR : データ 1 を書き込む

EV3 : TXIS ISR : データ 2 を書き込む

EV4 : TXIS ISR : データ 3 を書き込む

EV5 : TXIS ISR : データ 4 を書き込む (送信されない)

最初のデータをFlashしない I²C スレーブトランスミッタ 3 バイト、
NOSTRETCH=0 の例 :



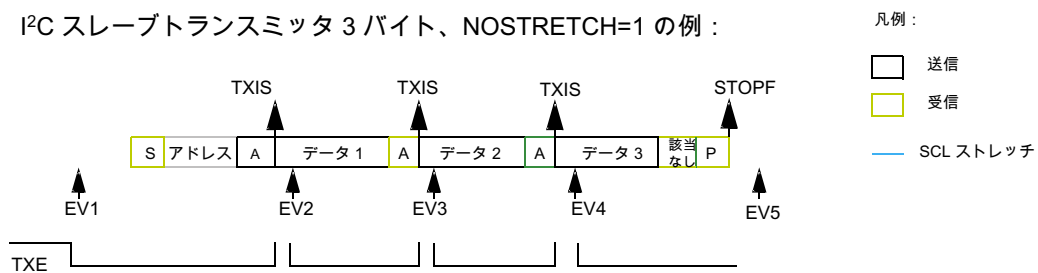
EV1 : ADDR ISR : ADDCODE および DIR をチェック、ADDRCF をセット

EV2 : TXIS ISR : データ 2 を書き込む

EV3 : TXIS ISR : データ 3 を書き込む

EV4 : TXIS ISR : データ 4 を書き込む (送信されない)

I²C スレーブトランスミッタ 3 バイト、NOSTRETCH=1 の例 :



Ev1 : データ 1 を書き込む

EV2 : TXIS ISR : データ 2 を書き込む

EV3 : TXIS ISR : データ 3 を書き込む

EV4 : TXIS ISR : データ 4 を書き込む (送信されない)

EV5 : STOPF ISR : (オプション : TXE および TXIS をセット)、STOPCF をセット

MS19853V2

スレーブレシーバ

I2C_RXDR がフルのときには、I2C_ISR の RXNE がセットされ、I2C_CR1 の RXIE がセットされている場合は割り込みが生成されます。RXNE は、I2C_RXDR が読み出されたときにクリアされます。

STOP が受信され、I2C_CR1 レジスタの STOPIE ビットがセットされると、I2C_ISR の STOPF がセットされ、割り込みが生成されます。

図 631. スレーブレシーバの転送シーケンスフロー (NOSTRETCH = 0)

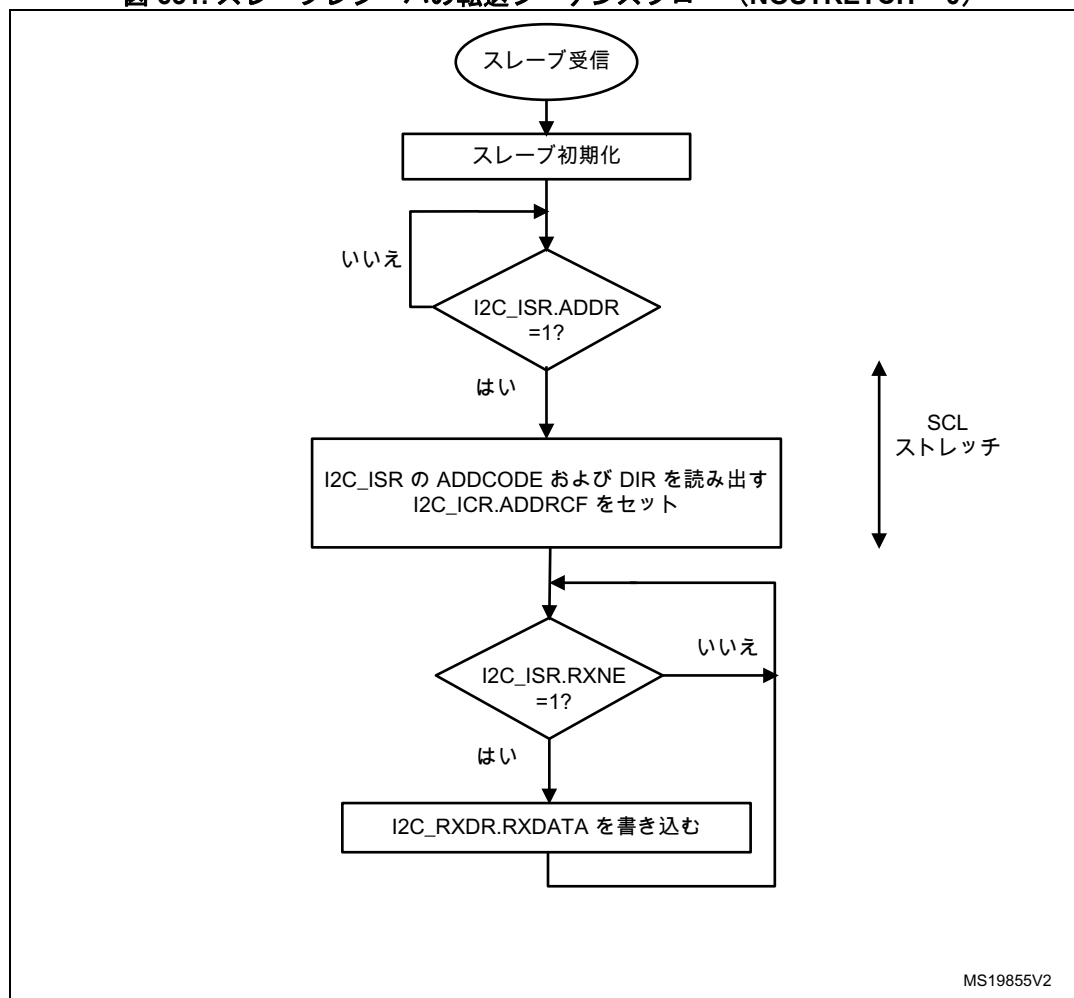


図 632. スレーブレシーバの転送シーケンスフロー (NOSTRETCH = 1)

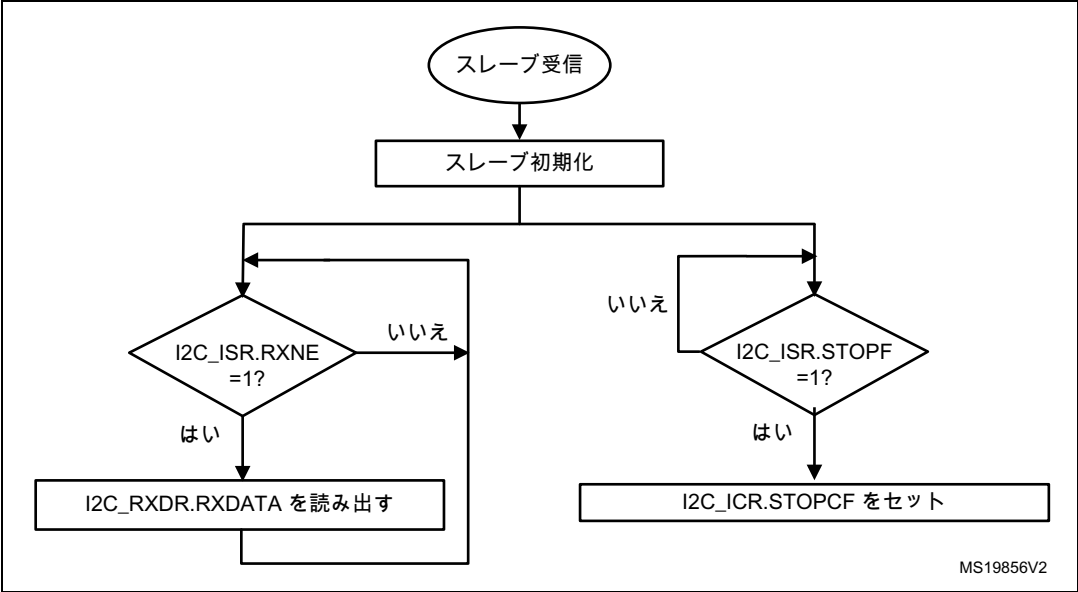
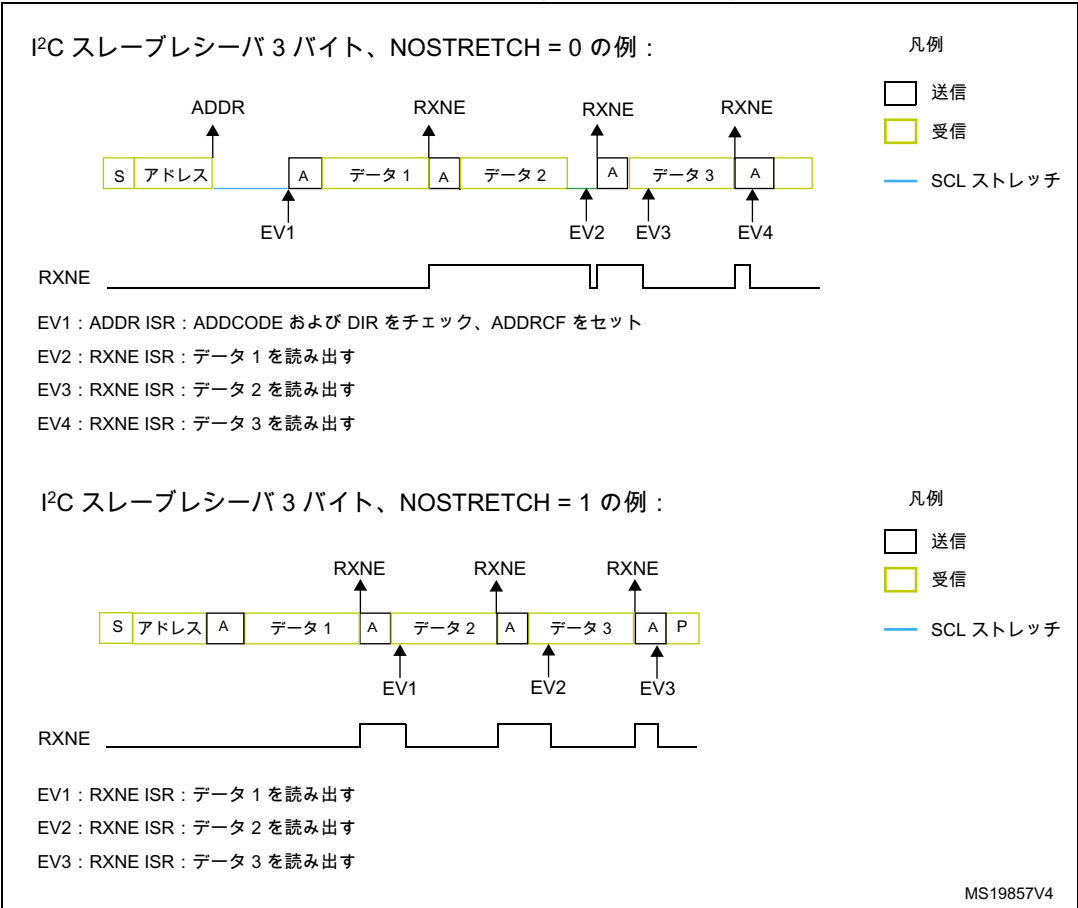


図 633. I²C スレーブレシーバの転送バス図 (必須イベントのみ)



48.4.9 I²C マスタモード

I²C マスタ初期化

ペリフェラルを有効にする前に、I2C_TIMINGR レジスタの SCLH および SCLL ビットをセットすることによって、I²C マスタクロックを設定する必要があります。

STM32CubeMX ツールは、I²C 設定ウィンドウの I2C_TIMINGR コンテンツを計算し、提供します。

マルチマスタ環境とスレーブクロックストレッチをサポートするために、クロック同期メカニズムが実装されています。

クロック同期を可能にするために：

- クロックのローレベルは SCLL カウンタを使用してカウントされ、SCL ローレベル内部検出から開始されます。
- クロックのハイレベルは SCLH カウンタを使用してカウントされ、SCL ハイレベル内部検出から開始されます。

I²C は、SCL 立下がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC1} の後に SCL ローレベルを検出します。SCLL カウンタが I2C_TIMINGR レジスタの SCLL[7:0] ビットでプログラムされた値に達すると、I²C は SCL をハイレベルにリリースします。

I²C は、SCL 立上がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC2} の後に SCL ハイレベルを検出します。SCLH カウンタが I2C_TIMINGR レジスタの SCLH[7:0] ビットでプログラムされた値に達すると、I²C は SCL をローレベルにします。

結果として、マスタクロック周期は次のとおりです：

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH} + 1) + (\text{SCLL} + 1)] \times (\text{PRESC} + 1) \times t_{\text{I2CCLK}}\}$$

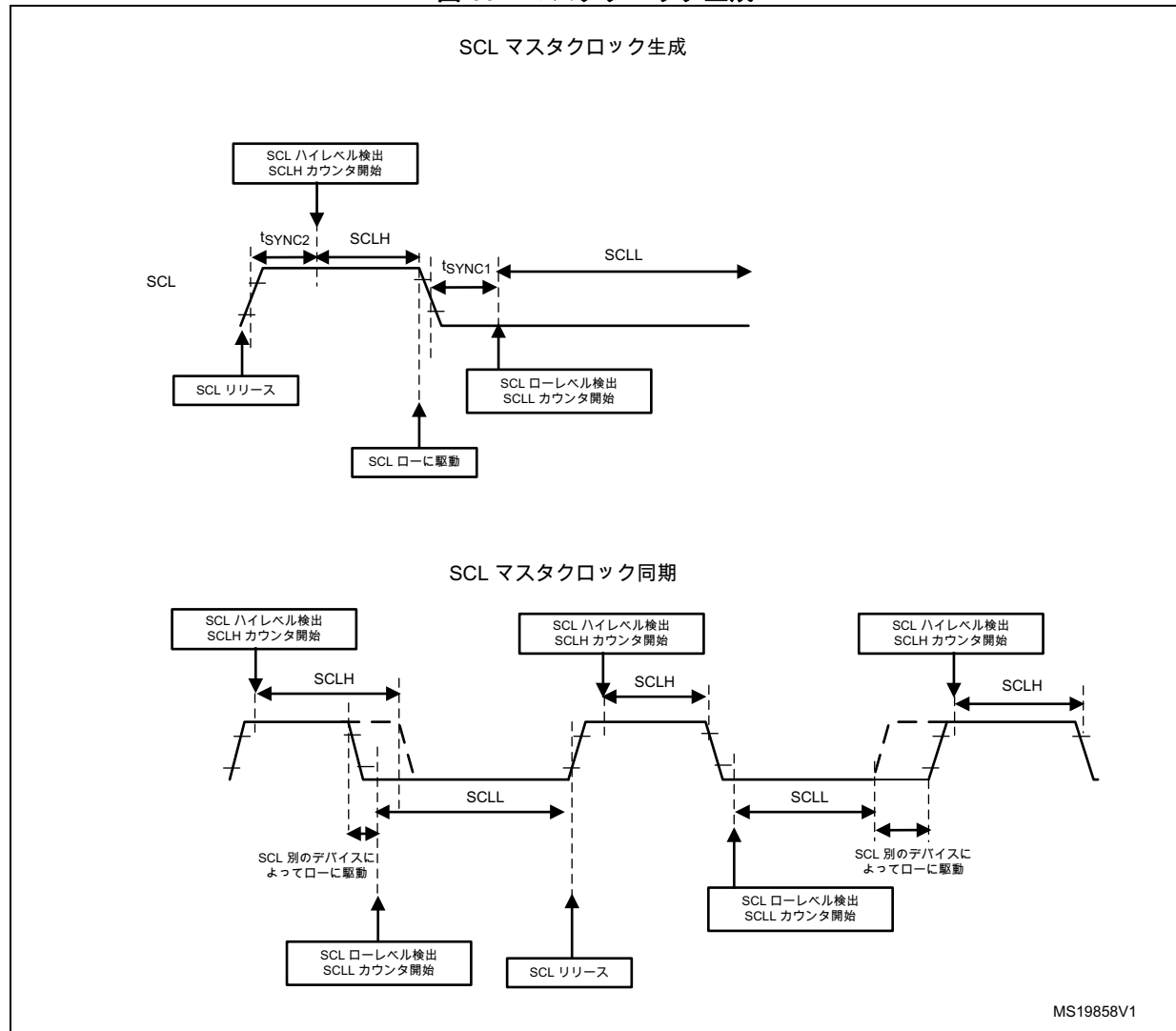
t_{SYNC1} の長さは、次のものに依存します。

- SCL 立下がり傾斜
- 有効な場合、アナログフィルタによる入力遅延：
- 有効な場合、デジタルフィルタによる入力遅延： $\text{DNF} \times t_{\text{I2CCLK}}$
- SCL と i2c_ker_ck クロックの同期による遅延（2 ～ 3 i2c_ker_ck 周期）

t_{SYNC2} の長さは、次のものに依存します。

- SCL 立上がり傾斜
- 有効な場合、アナログフィルタによる入力遅延：
- 有効な場合、デジタルフィルタによる入力遅延： $\text{DNF} \times t_{\text{I2CCLK}}$
- SCL と i2c_ker_ck クロックの同期による遅延（2 ～ 3 i2c_ker_ck 周期）

図 634. マスタクロック生成



注意： I²C または SMBus 準拠のためには、マスタクロックは次の表のタイミングを満たす必要があります。

表 504. I²C-SMBus 仕様のクロックタイミング

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBus		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
f _{SCL}	SCL クロック周波数	-	100	-	400	-	1000	-	100	kHz
t _{HD:STA}	(反復) START コンディションのホールド時間	4.0	-	0.6	-	0.26	-	4.0	-	μs
t _{SU:STA}	反復 START コンディションのセットアップ時間	4.7	-	0.6	-	0.26	-	4.7	-	
t _{SU:STO}	STOP コンディションのセットアップ時間	4.0	-	0.6	-	0.26	-	4.0	-	
t _{BUF}	STOP コンディションと START コンディションの間のバスフリー時間	4.7	-	1.3	-	0.5	-	4.7	-	
t _{LOW}	SCL クロックのロー周期	4.7	-	1.3	-	0.5	-	4.7	-	
t _{HIGH}	SCL クロックの周期	4.0	-	0.6	-	0.26	-	4.0	50	
t _r	SDA および SCL 信号の立上がり時間	-	1000	-	300	-	120	-	1000	ns
t _f	SDA および SCL 信号の立下がり時間	-	300	-	300	-	120	-	300	

注： SCLL は、t_{BUF} および t_{SU:STA} タイミングの生成にも使用され、SCLH は、t_{HD:STA} および t_{SU:STO} タイミングの生成にも使用されます。

I2C_TIMINGR 設定と i2c_ker_ck 周波数の例については、[セクション 48.4.10](#) を参照してください。

マスタ通信の初期化（アドレスフェーズ）

通信を初期化するためには、I2C_CR2 レジスタでアドレス指定されたスレーブについて次のパラメータをプログラムする必要があります。

- アドレッシングモード（7 ビットまたは 10 ビット）：ADD10
- 送信されるスレーブアドレス：SADD[9:0]
- 転送方向：RD_WRN
- 10 ビットアドレスが読み出される場合：HEAD10R ビット。HEAD10R を設定して、完全なアドレスシーケンスが送信されなければならないか、ヘッダのみ（方向の変更の場合）かを示す必要があります。
- 転送されるバイト数：NBYTES[7:0] バイト数が 255 バイト以上の場合、NBYTES[7:0] に 0xFF を書き込む必要があります。

次に、I2C_CR2 レジスタの START ビットをセットする必要があります。START ビットがセットされているとき、上記のすべてのビットを変更することはできません。

その場合、マスタは、バスがフリーである（BUSY = 0）ことを検出すると、t_{BUF} の遅延後に、自動的に START コンディションとスレーブアドレスを送信します。

アービトレーション喪失の場合、マスタはスレーブモードに自動的に切り替えて、スレーブとしてアドレス指定された場合は専用アドレスを確認応答できます。

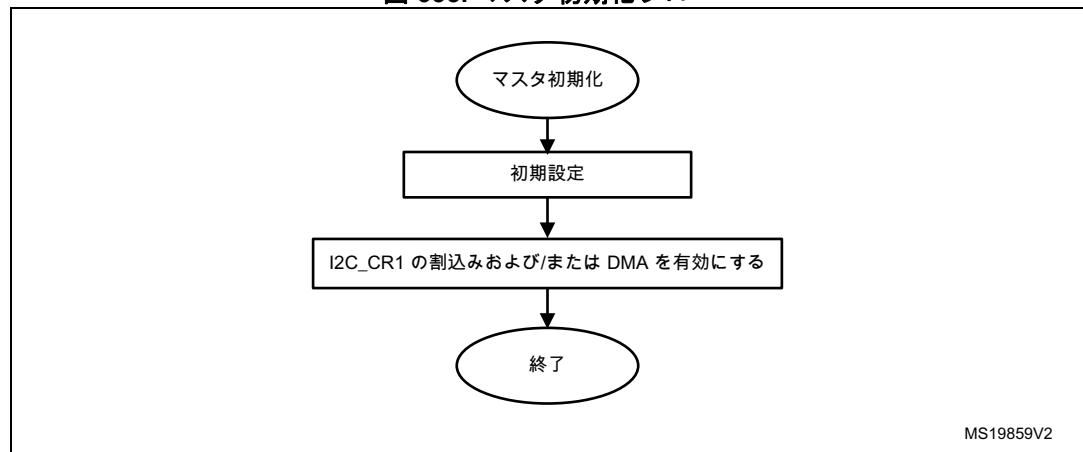
注： START ビットは、スレーブアドレスがバスに送信されたとき、受信した確認応答値にかかわらず、ハードウェアによってリセットされます。START ビットは、アービトレーション喪失が発生した場合にも、ハードウェアによってリセットされます。

10 ビットアドレッシングモードでは、スレーブアドレスの最初の 7 ビットがスレーブによって NACK されている場合、マスタは ACK が受信されるまで自動的にスレーブアドレスの送信を再開します。この場合、スレーブから NACK が受信された場合、ADDRCF をセットしてスレーブアドレスの送信を停止する必要があります。

START ビットがセットされているときに、I2C がスレーブとしてアドレス指定された場合 (ADDR = 1)、I²C はスレーブモードに切り替わり、START ビットがクリアされます。

注： 反復スタートコンディションにも同じ手順が適用されます。この場合、BUSY = 1 です。

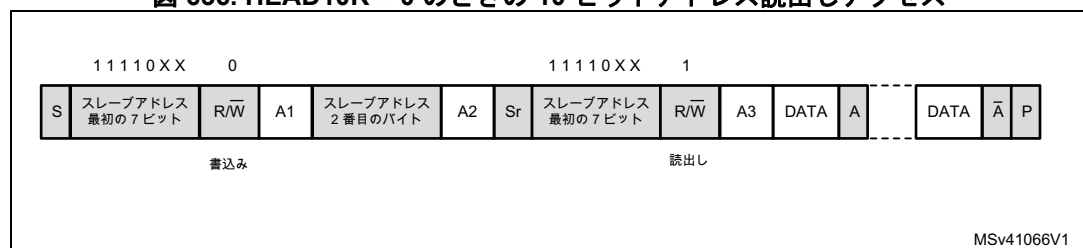
図 635. マスタ初期化フロー



10 ビットアドレススレーブをアドレス指定するマスタレシーバの初期化

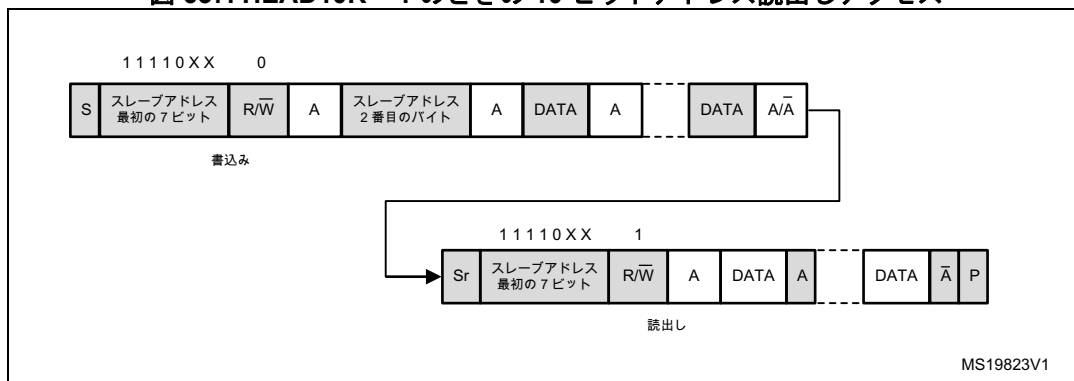
- スレーブアドレスが 10 ビット形式の場合、I2C_CR2 レジスタの HEAD10R ビットをクリアすることによって、完全な読出しシーケンスを送信することができます。この場合、マスタは、START ビットがセットされた後、次のような完全なシーケンスを自動的に送信します：(Re) START + スレーブアドレス 10 ビットヘッダ書込み + スレーブアドレスの 2 番目のバイト + REstart + スレーブアドレス 10 ビットヘッダ読出し。

図 636. HEAD10R = 0 のときの 10 ビットアドレス読出しアクセス



- マスタが 10 ビットアドレススレーブをアドレス指定して、このスレーブアドレスにデータを送信した後、同じスレーブからデータを読み出す場合には、まず、マスタ送信フローが行われる必要があります。次に、HEAD10R = 1 で設定された 10 ビットスレーブアドレスで、反復開始をセットします。この場合、マスタは次のシーケンスを送信します：ReStart + スレーブアドレス 10 ビットヘッダ読出し。

図 637. HEAD10R = 1 のときの 10 ビットアドレス読出しアクセス



マスタトランスマッタ

書き込み転送の場合、ACK が受信されたときの 9 番目の SCL パルス後、各バイトの送信後に TXIS フラグがセットされます。

I2C_CR1 レジスタの TXIE ビットがセットされている場合、TXIS イベント時に割込みが生成されます。このフラグは、I2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

転送中の TXIS イベントの数は、NBYTES[7:0] でプログラムされた値に対応します。送信されるデータバイト数の合計が 255 より大きい場合、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

TXIS フラグは、NACK 受信時にはセットされません。

- RELOAD = 0 で NBYTES データが転送されたとき：
 - 自動終了モード (AUTOEND = 1) では、STOP が自動的に送信されます。
 - ソフトウェア終了モード (AUTOEND = 0) では、TC フラグがセットされ、ソフトウェアアクションを実行するために SCL ラインがローでストレッチされます：

正しいスレーブアドレス設定と転送バイト数で I2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションがバスに送信されます。

I2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。
- NACK が受信された場合：TXIS フラグはセットされず、NACK 受信後、自動的に STOP コンディションが送信されます。I2C_ISR レジスタの NACKF フラグがセットされ、NACKIE ビットがセットされていた場合は割込みが生成されます。

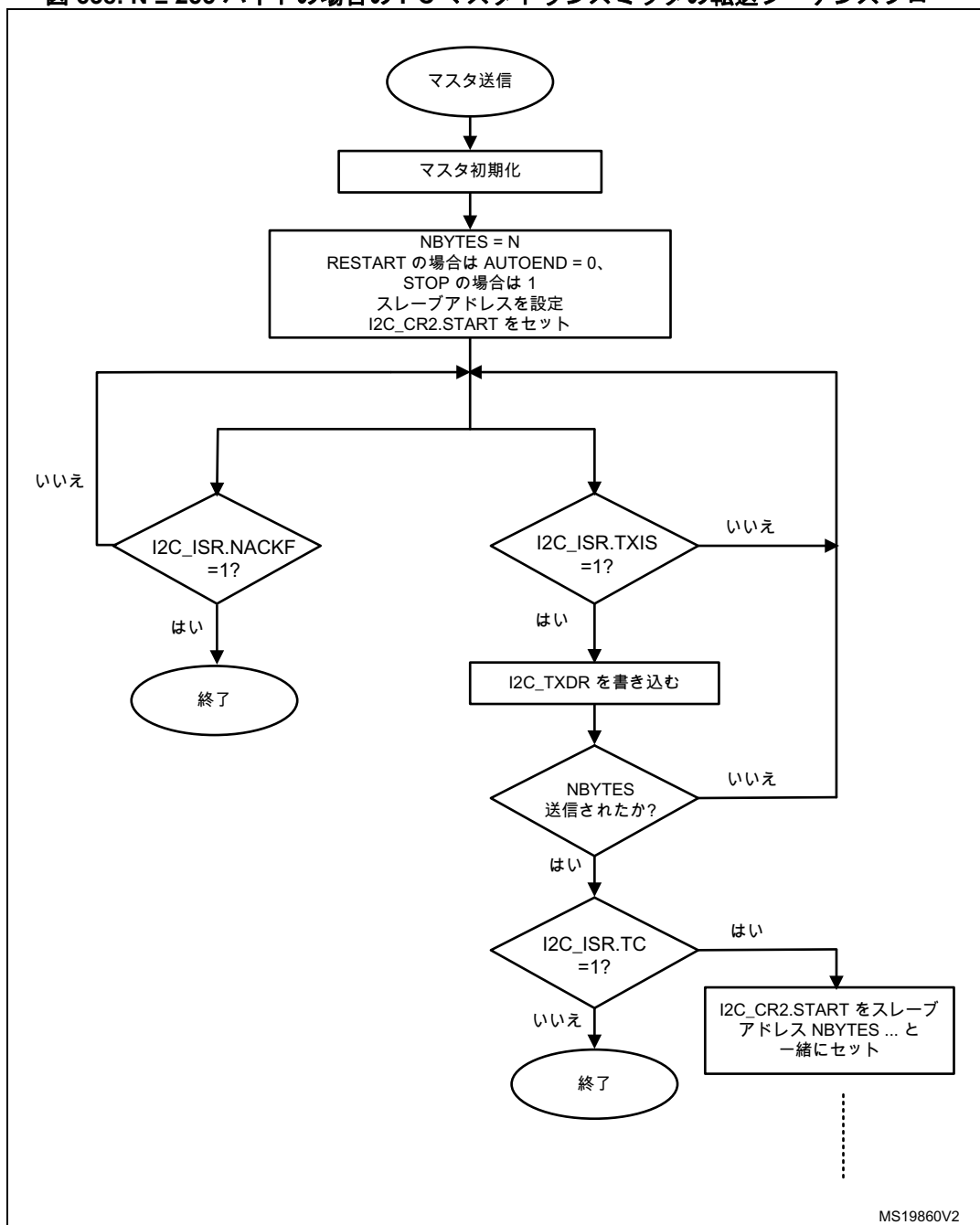
図 638. $N \leq 255$ バイトの場合の I²C マスタトランスミッタの転送シーケンスフロー

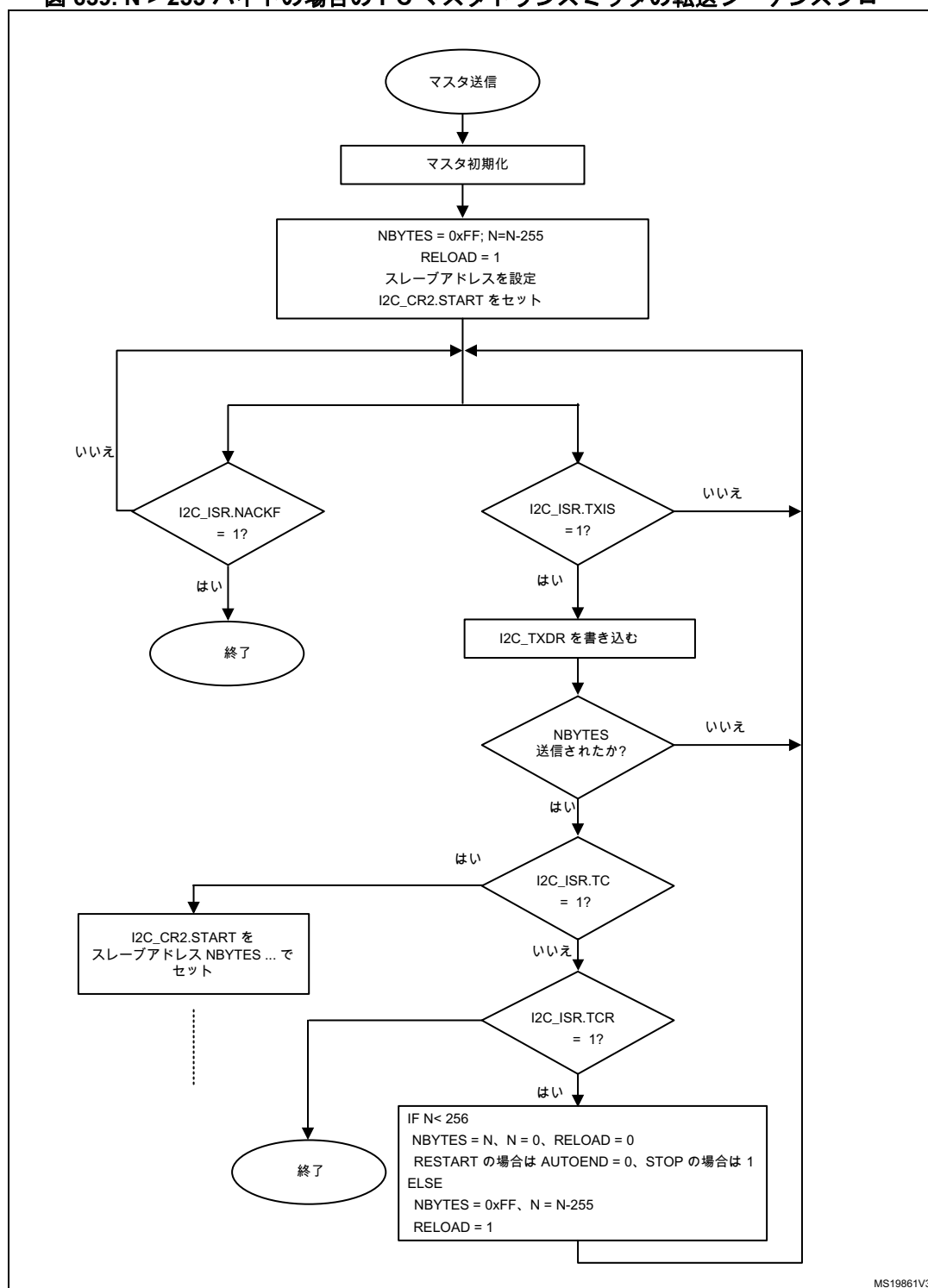
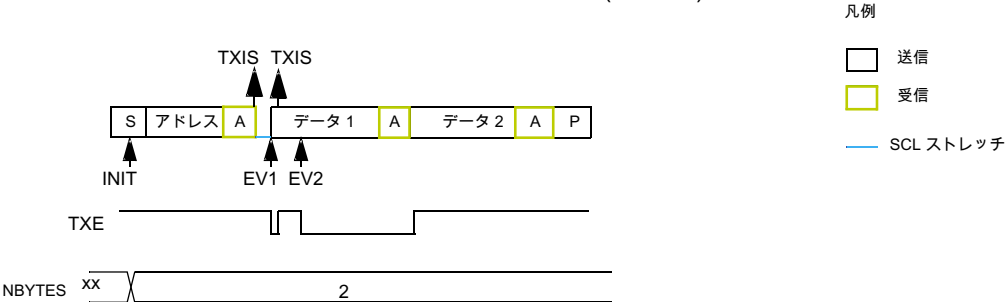
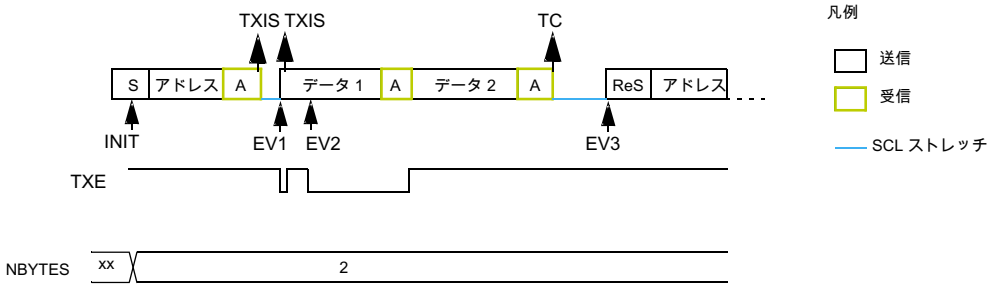
図 639. N > 255 バイトの場合の I²C マスタトランスミッタの転送シーケンスフロー

図 640. I²C マスタトランスミッタの転送バス図 (必須イベントのみ)

I²C マスタトランスミッタ 2 バイト、自動終了モード (STOP) の例



I²C マスタトランスミッタ 2 バイト、ソフトウェア終了モード (RESTART) の例



MS19862V2

マスタレシーバ

読出し転送の場合、各バイトの受信後や 8 番目の SCL パルス後に RXNE フラグがセットされます。I2C_CR1 レジスタの RXIE ビットがセットされている場合、RXNE イベント時に割込みが生成されます。このフラグは、I2C_RXDR が読み出されたときにクリアされます。

受信されるデータバイト数の合計が 255 より大きい場合、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES[7:0] データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

- RELOAD = 0 で NBYTES[7:0] データが転送されたとき：

- 自動終了モード (AUTOEND = 1) では、最後の受信バイト後に NACK および STOP が自動的に送信されます。
- ソフトウェア終了モード (AUTOEND = 0) では、最後の受信バイト後に NACK が自動的に送信され、TC フラグがセットされ、ソフトウェアアクションを実行できるように、SCL ラインがローでストレッチされます。

正しいスレーブアドレス設定と転送バイト数で I2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションとスレーブアドレスがバスに送信されます。

I2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。

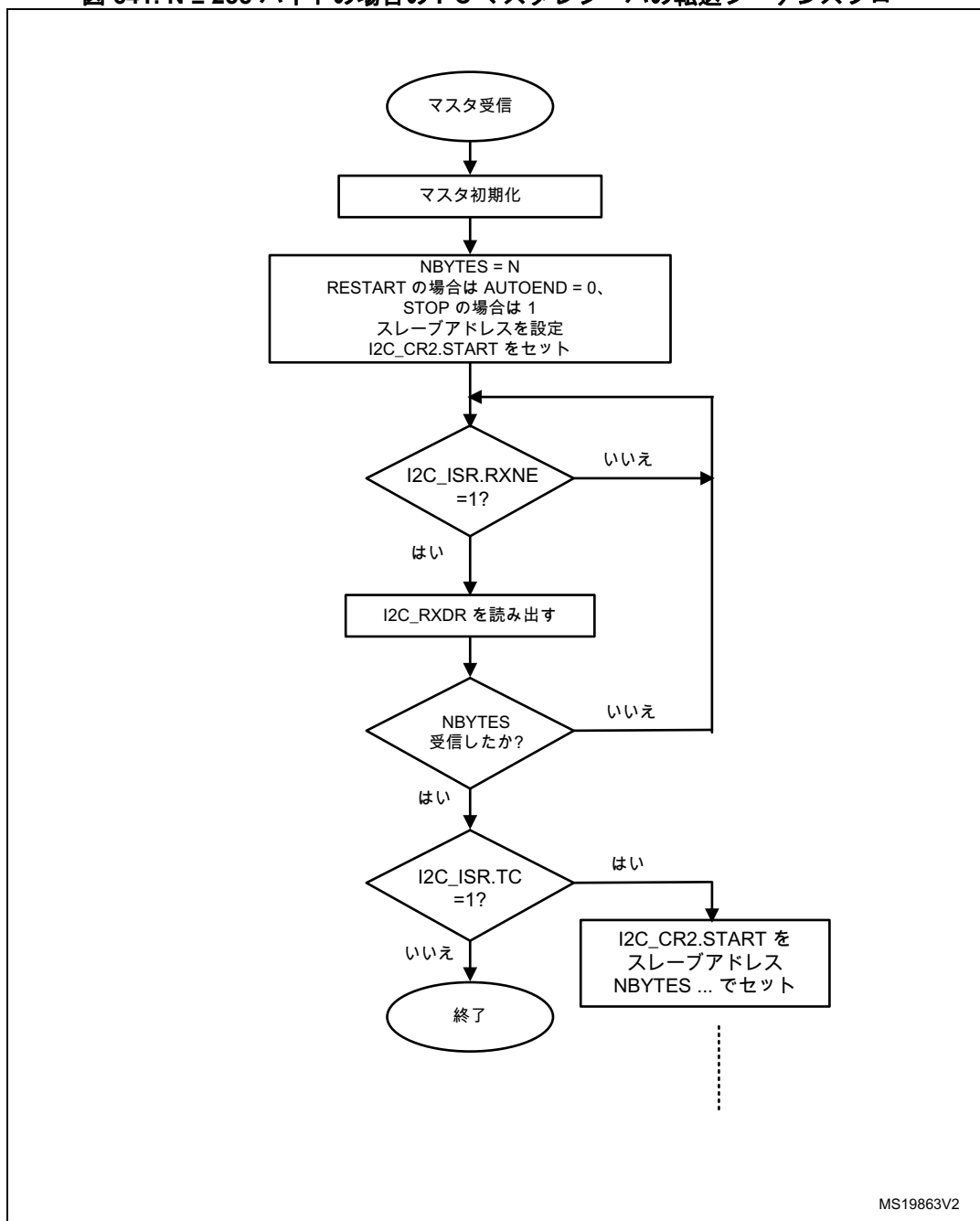
図 641. $N \leq 255$ バイトの場合の I²C マスタレシーバの転送シーケンスフロー

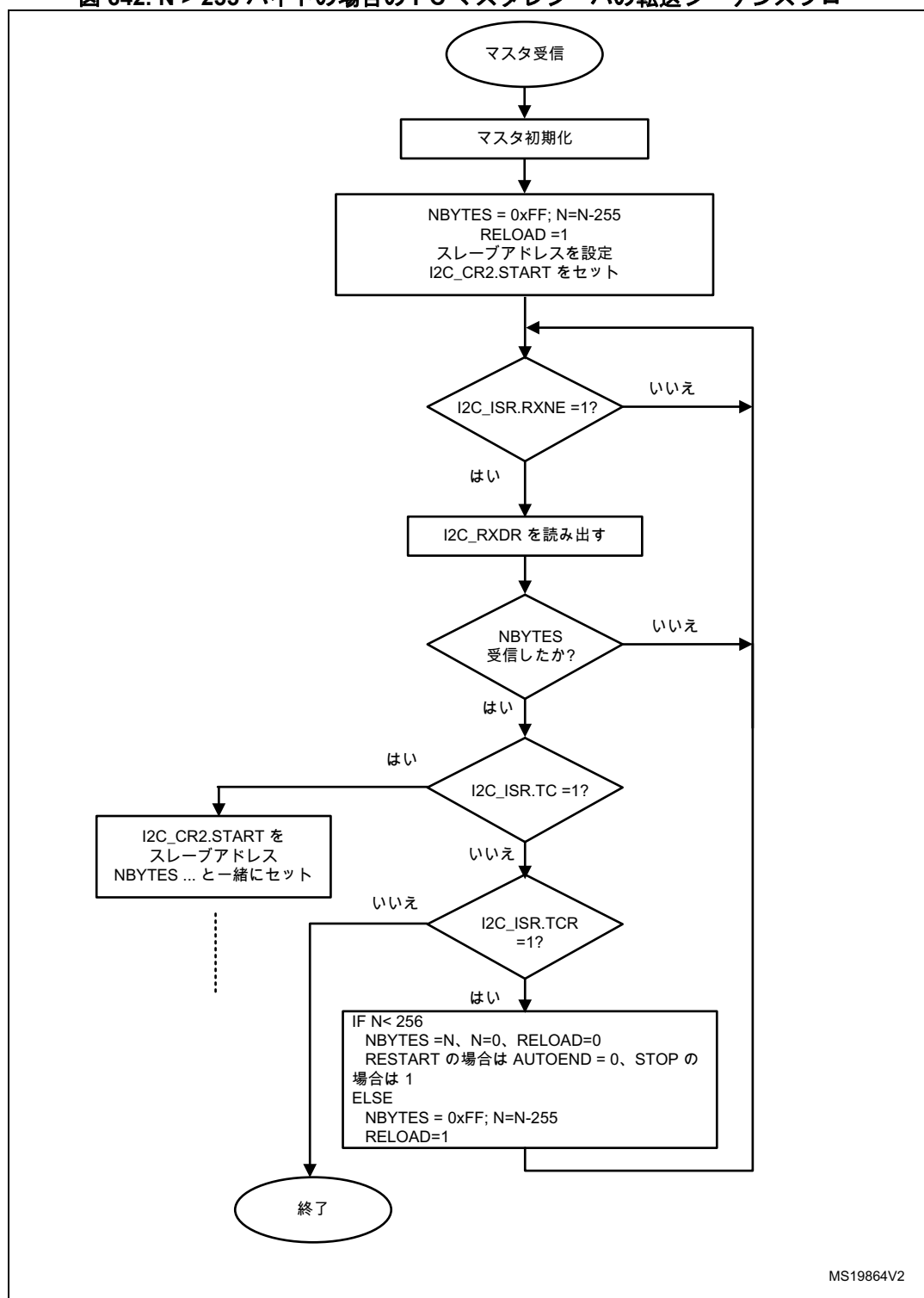
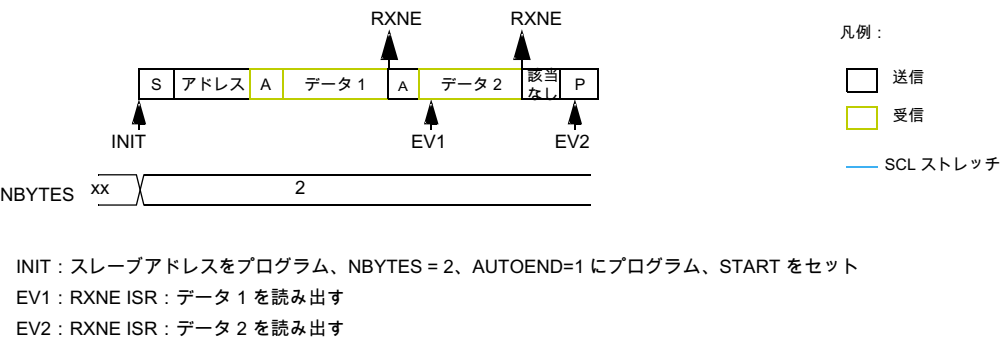
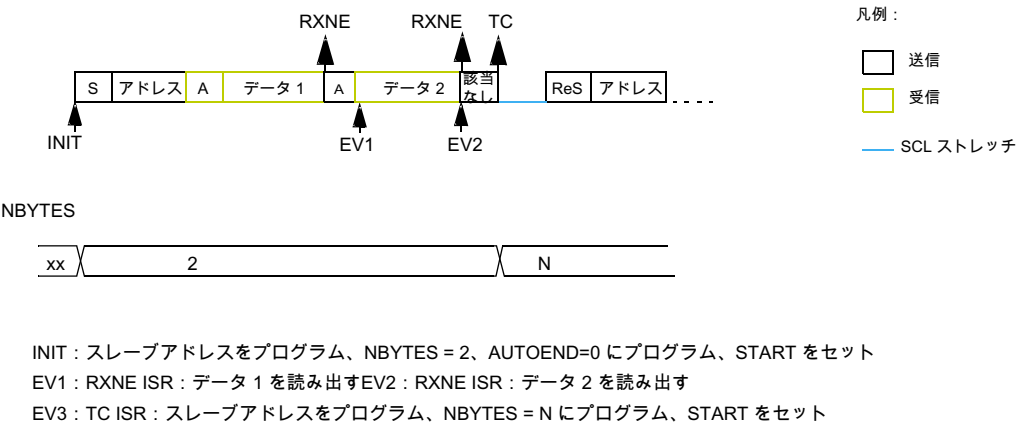
図 642. N > 255 バイトの場合の I²C マスタレシーバの転送シーケンスフロー

図 643. I²C マスタレシーバの転送バス図 (必須イベントのみ)

I²C マスタレシーバ 2 バイト、自動終了モード (STOP) の例



I²C マスタレシーバ 2 バイト、ソフトウェア終了モード (RESTART) の例



MS19865V1

48.4.10 I2C_TIMINGR レジスタの設定例

次の表に、I²C 仕様に準拠したタイミングを得るための I2C_TIMINGR をプログラムする方法の例を示します。より正確な設定値を得るには、STM32CubeMX ツール (I²C 設定ウィンドウ) を使用してください。

表 505. $f_{I2CCLK} = 8 \text{ MHz}$ でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	0x1	0x1	0x0	0x0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200 x 250 ns = 50 μ s	20 x 250 ns = 5.0 μ s	10 x 125 ns = 1250 ns	7 x 125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196 x 250 ns = 49 μ s	16 x 250 ns = 4.0 μ s	4 x 125 ns = 500 ns	4 x 125 ns = 500 ns
$t_{SCL}^{(1)}$	~100 μ s ⁽²⁾	~10 μ s ⁽²⁾	~2.5 μ s ⁽³⁾	~2.0 μ s ⁽⁴⁾
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	1 x 125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

- t_{SCL} は、SCL 内部検出遅延のため、 $t_{SCLL} + t_{SCLH}$ より大きくなります。 t_{SCL} として示されている値は例にすぎません。
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 500 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 1000 \text{ ns}$
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 500 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 750 \text{ ns}$
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 500 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 655 \text{ ns}$

表 506. $f_{I2CCLK} = 16 \text{ MHz}$ でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0x3	0x3	0x1	0x0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 μ s	20 x 250 ns = 5.0 μ s	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 μ s	16 x 250 ns = 4.0 μ s	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
$t_{SCL}^{(1)}$	~100 μ s ⁽²⁾	~10 μ s ⁽²⁾	~2.5 μ s ⁽³⁾	~1.0 μ s ⁽⁴⁾
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

- t_{SCL} は、SCL 内部検出遅延のため、 $t_{SCLL} + t_{SCLH}$ より大きくなります。 t_{SCL} として示されている値は例にすぎません。
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 250 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 1000 \text{ ns}$
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 250 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 750 \text{ ns}$
- $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 250 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 500 \text{ ns}$

48.4.11 SMBus 固有の機能

このセクションは、SMBus 機能がサポートされるときにのみ適用されます (セクション 48.3 を参照してください)。

概要

システム管理バス (SMBus) は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I²C の動作原理に基づきます。SMBus により、システムおよびパワーマネージメント関連のタスク向けの制御バスが実現できます。

このペリフェラルは、SMBus 仕様 (<http://smbus.org>) と互換性があります。

システム管理バス仕様では、3 種類のデバイスを規定しています。

- スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。
- マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。
- ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

このペリフェラルは、マスタまたはスレーブデバイスとして、また、ホストとしても設定できます。

バスプロトコル

特定のデバイスについて、11 の可能なコマンドプロトコルがあります。デバイスは、それらの一部または全部を使用して通信できます。プロトコルは、Quick Command、Send Byte、Receive Byte、Write Byte、Write Word、Read Byte、Read Word、Process Call、Block Read、Block Write、および Block Write-Block Read Process Call です。これらのプロトコルは、ユーザのソフトウェアによって実装する必要があります。

これらのプロトコルの詳細については、SMBus 仕様 (<http://smbus.org>) を参照してください。

アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス割当てを目的とする各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。128 ビットの番号がソフトウェアによって実装されます。

このペリフェラルは、アドレス解決プロトコル (ARP) をサポートします。SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。ARP コマンドは、ユーザのソフトウェアによって実装する必要があります。

ARP サポートのために、スレーブモードでアービトレーションも行われます。

SMBus アドレス解決プロトコルの詳細については、SMBus 仕様 (<http://smbus.org>) を参照してください。

受信コマンドおよびデータ確認応答制御

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、[スレーブバイト制御モード](#)を参照してください。

ホスト通知プロトコル

このペリフェラルは、I2C_CR1 レジスタの SMBHEN ビットをセットすることによって、ホスト通知プロトコルをサポートします。この場合、ホストは SMBus ホストアドレス (0b0001 000) を確認応答します。

このプロトコルが使用されると、デバイスはマスタとして動作し、ホストはスレーブとして動作します。

SMBus アラート

SMBus ALERT オプション信号がサポートされます。スレーブ専用デバイスは、通信したいホストの SMBALERT# ピンを通じてホストに信号を送信します。ホストは、割込みを処理し、アラート応答アドレス (0b0001 100) を通じて全 SMBALERT# デバイスに同時にアクセスします。SMBALERT# をローに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。

スレーブデバイスとして設定されたとき (SMBHEN = 0)、I2C_CR1 レジスタの ALERTEN ビットをセットすることによって、SMBA ピンはローに引き下げられます。同時に、アラート応答アドレスが有効になります。

ホストとして設定されたとき (SMBHEN = 1)、SMBA ピンで立下がりエッジが検出され、ALERTEN = 1 のとき、I2C_ISR レジスタの ALERT フラグがセットされます。I2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割込みが生成されます。ALERTEN = 0 のときには、外部 SMBA ピンがローの場合でも、ALERT ラインはハイとみなされます。

SMBus ALERT ピンが不要な場合には、ALERTEN = 0 の場合、SMBA ピンを標準 GPIO として使用できます。

パケットエラーチェック

信頼性と通信の堅牢性を向上させるために、SMBus 仕様にパケットエラーチェックメカニズムが導入されました。パケットエラーチェックは、各メッセージ転送の終わりにパケットエラーコード (PEC) を付加することによって実装されます。PEC は、すべてのメッセージバイト (アドレスと読出し/書き込みビットを含む) に対して $C(x) = x^8 + x^2 + x + 1$ CRC-8 多項式を使用して計算されます。

ペリフェラルにはハードウェア PEC 計算器が組み込まれており、受信バイトがハードウェアによって計算された PEC に一致しないときには自動的に非確認応答を送信できます。

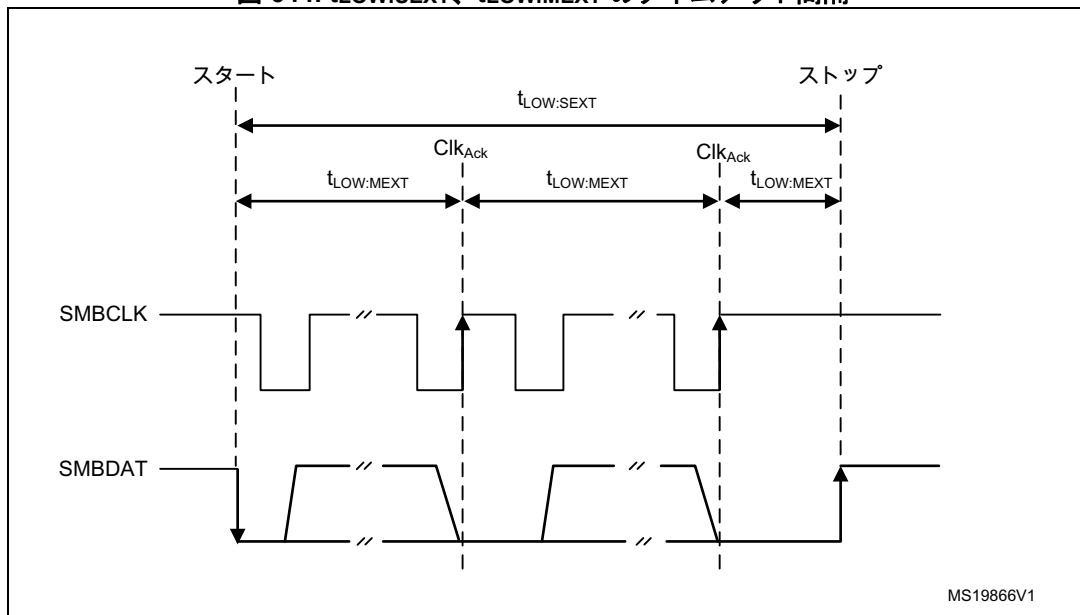
タイムアウト

このペリフェラルは、SMBus 仕様で定義された 3 つのタイムアウトに準拠するために、ハードウェアタイマが組み込まれています。

表 507. SMBus タイムアウト仕様

記号	パラメータ	リミット		単位
		最小値	最大値	
t_{TIMEOUT}	クロックロータイムアウト検出	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累積クロックロー延長時間 (スレーブデバイス)	-	25	
$t_{\text{LOW:MEXT}}^{(2)}$	累積クロックロー延長時間 (マスタデバイス)	-	10	

- $t_{\text{LOW:SEXT}}$ は、特定のスレーブデバイスが初めての START から STOP までの 1 つのメッセージのクロックサイクルを延長できる累積時間です。別のスレーブデバイスまたはマスタもクロックを延長して合計のクロックロー延長時間が $t_{\text{LOW:SEXT}}$ より大きくなる場合があります。したがって、このパラメータは、スレーブデバイスをフルスピードのマスタの単独のターゲットとして測定されます。
- $t_{\text{LOW:MEXT}}$ は、マスタがメッセージの各バイト内のクロックサイクルを START-to-ACK、ACK-to-ACK、または ACK-to-STOP から定義に従って延長できる累積時間です。スレーブデバイスまたは別のマスタも与えられたバイトでクロックを延長して、合計のクロックロー時間が $t_{\text{LOW:MEXT}}$ より大きくなる場合があります。したがって、このパラメータは、フルスピードスレーブデバイスをマスタの単独のターゲットとして測定されます。

図 644. $t_{\text{LOW:SEXT}}$ 、 $t_{\text{LOW:MEXT}}$ のタイムアウト間隔

バスアイドル検出

マスタは、クロックおよびデータ信号が $t_{\text{HIGH,MAX}}$ より大きい t_{DLE} の間ハイであった場合、バスはフリーであるとみなすことができます ([I2C のタイミング](#)を参照)。

このタイミングパラメータは、マスタがバスに動的に追加されていて、SMBCLK または SMBDAT ライン上の状態遷移を検出していなかった可能性がある状況をカバーします。この場合、マスタは十分に長い時間待って、転送が進行中でないことを確認する必要があります。このペリフェラルは、ハードウェアバスアイドル検出をサポートします。

48.4.12 SMBus 初期化

このセクションは、SMBus 機能がサポートされるときにのみ適用されます ([セクション 48.3](#)を参照)。

SMBus 通信を行うためには、I2C 初期化に加えて、他にも特定の初期化を行う必要があります。

受信コマンドおよびデータ確認応答制御 (スレーブモード)

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットによって、スレーブバイト制御モードを有効にする必要があります。詳細については、[スレーブバイト制御モード](#)を参照してください。

特定アドレス (スレーブモード)

必要な場合は、特定の SMBus アドレスを有効にしてください。詳細については、[バスアイドル検出](#)を参照してください。

- SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。
- SMBus ホストアドレス (0b0001 000) は、I2C_CR1 レジスタの SMBHEN ビットをセットすることによって有効になります。
- アラート応答アドレス (0b0001100) は、I2C_CR1 レジスタの ALERTEN ビットをセットすることによって有効になります。

パケットエラーチェック

PEC 計算を有効にするには、I2C_CR1 レジスタの PECEN ビットをセットします。その場合、PEC 転送はハードウェアバイトカウンタ (I2C_CR2 レジスタの NBYTES[7:0]) を使用して管理されます。PECEN ビットは、I²C を有効にする前に設定する必要があります。

PEC 転送はハードウェアバイトカウンタによって管理されるので、スレーブモードで SMBus とインタフェースするときには SBC ビットをセットする必要があります。PEC は、PECBYTE ビットがセットされ、RELOAD ビットがクリアされたとき、NBYTES - 1 データの転送後に転送されます。RELOAD がセットされた場合、PECBYTE は効果がありません。

注意： I2C が有効なときには、PECEN 設定の変更はできません。

表 508. SMBus の PEC 設定

モード	SBC ビット	RELOAD ビット	AUTOEND ビット	PECBYTE ビット
マスタ Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
マスタ Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
スレーブ Tx/Rx と PEC	1	0	x	1

タイムアウト検出

タイムアウト検出は、I2C_TIMEOUTR レジスタの TIMOUTEN および TEXTEN ビットをセットすることによって有効になります。SMBus 仕様で指定された最大時間の前にタイムアウトを検出するようにタイマをプログラムする必要があります。

- t_{TIMEOUT} チェック

t_{TIMEOUT} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] ビットを、 t_{TIMEOUT} パラメータをチェックするようにタイマ再ロード値でプログラムする必要があります。SCL ローレベルタイムアウトを検出するためには、TIDLE ビットを 0 に設定する必要があります。

その場合、タイマは、I2C_TIMEOUTR レジスタの TIMOUTEN をセットすることによって有効になります。

SCL が $(\text{TIMEOUTA} + 1) \times 2048 \times t_{\text{I2CCLK}}$ より長い時間、ローに設定された場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。

表 509 を参照してください。

注意： TIMOUTEN ビットがセットされているときには、TIMEOUTA[11:0] ビットおよび TIDLE ビットの設定変更はできません。

- $t_{\text{LOW:SEXT}}$ および $t_{\text{LOW:MEXT}}$ チェック

ペリフェラルがマスタとして設定されているか、スレーブとして設定されているかに応じて、12 ビットの TIMEOUTB タイマは、スレーブの場合は $t_{\text{LOW:SEXT}}$ をチェックするように、マスタの場合は $t_{\text{LOW:MEXT}}$ をチェックするように、設定する必要があります。標準では最大値のみが規定されているので、両方について同じ値を選ぶことができます。タイマを有効にするには、I2C_TIMEOUTR レジスタの TEXTEN ビットをセットします。

SMBus ペリフェラルが、 $(\text{TIMEOUTB} + 1) \times 2048 \times t_{\text{I2CCLK}}$ より長い時間および [バスアイドル検出](#) セクションで述べられているタイムアウト間隔で、累積 SCL ストレッチを実行した場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。

表 510 を参照してください。

注意： TEXTEN ビットがセットされているときには、TIMEOUTB 設定変更はできません。

バスアイドル検出

t_{IDLE} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] フィールドを、 t_{IDLE} パラメータが得られるように、タイマ再ロード値でプログラムする必要があります。SCL および SDA ハイレベルタイムアウトを検出するためには、TIDLE ビットを 1 に設定する必要があります。タイマを有効にするには、I2C_TIMEOUTR レジスタの TIMEOUTEN ビットをセットします。

SCL および SDA の両方のラインが $(TIMEOUTA + 1) \times 4 \times t_{I2CCLK}$ より長い間ハイのままであった場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。

表 511 を参照してください。

注意 : TIMEOUTEN がセットされているときに、TIMEOUTA および TIDLE 設定を変更することはできません。

48.4.13 SMBus : I2C_TIMEOUTR レジスタの設定例

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 48.3 を参照してください。

- $t_{TIMEOUT}$ の最大時間を 25 ms に設定 :

表 509. TIMEOUTA の設定例 (最大 $t_{TIMEOUT} = 25$ ms)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{TIMEOUT}$
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$

- $t_{LOW:SEXT}$ および $t_{LOW:MEXT}$ の最大時間を 8 ms に設定 :

表 510. TIMEOUTB の設定例

f_{I2CCLK}	TIMEOUTB[11:0] ビット	TEXTEN ビット	$t_{LOW:EXT}$
8 MHz	0x1F	1	$32 \times 2048 \times 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 \times 2048 \times 62.5 \text{ ns} = 8 \text{ ms}$

- t_{IDLE} の最大時間を 50 μ s に設定

表 511. TIMEOUTA の設定例 (最大 $t_{IDLE} = 50$ μ s)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	t_{IDLE}
8 MHz	0x63	1	1	$100 \times 4 \times 125 \text{ ns} = 50 \mu\text{s}$
16 MHz	0xC7	1	1	$200 \times 4 \times 62.5 \text{ ns} = 50 \mu\text{s}$

48.4.14 SMBus スレーブモード

このセクションは、SMBus 機能がサポートされるときにのみ適用されます（[セクション 48.3](#) を参照してください）。

I²C スレーブ転送管理（[セクション 48.4.8](#) を参照）に加えて、SMBus をサポートするために、追加のソフトウェアフローが用意されています。

SMBus スレーブトランスミッタ

IP が SMBus で使用されるときには、SBC は、プログラムされたデータバイト数の終わりの PEC 送信を有効にするため、1 にプログラムする必要があります。PECBYTE ビットがセットされているときには、NBYTES[7:0] でプログラムされたバイト数には PEC 送信が含まれます。その場合、TXIS 割込みの合計数は NBYTES - 1 であり、NBYTES - 1 データ転送後にマスタが追加のバイトをリクエストした場合、I2C_PECR レジスタの内容が自動的に送信されます。

注意： PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 645. N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフロー

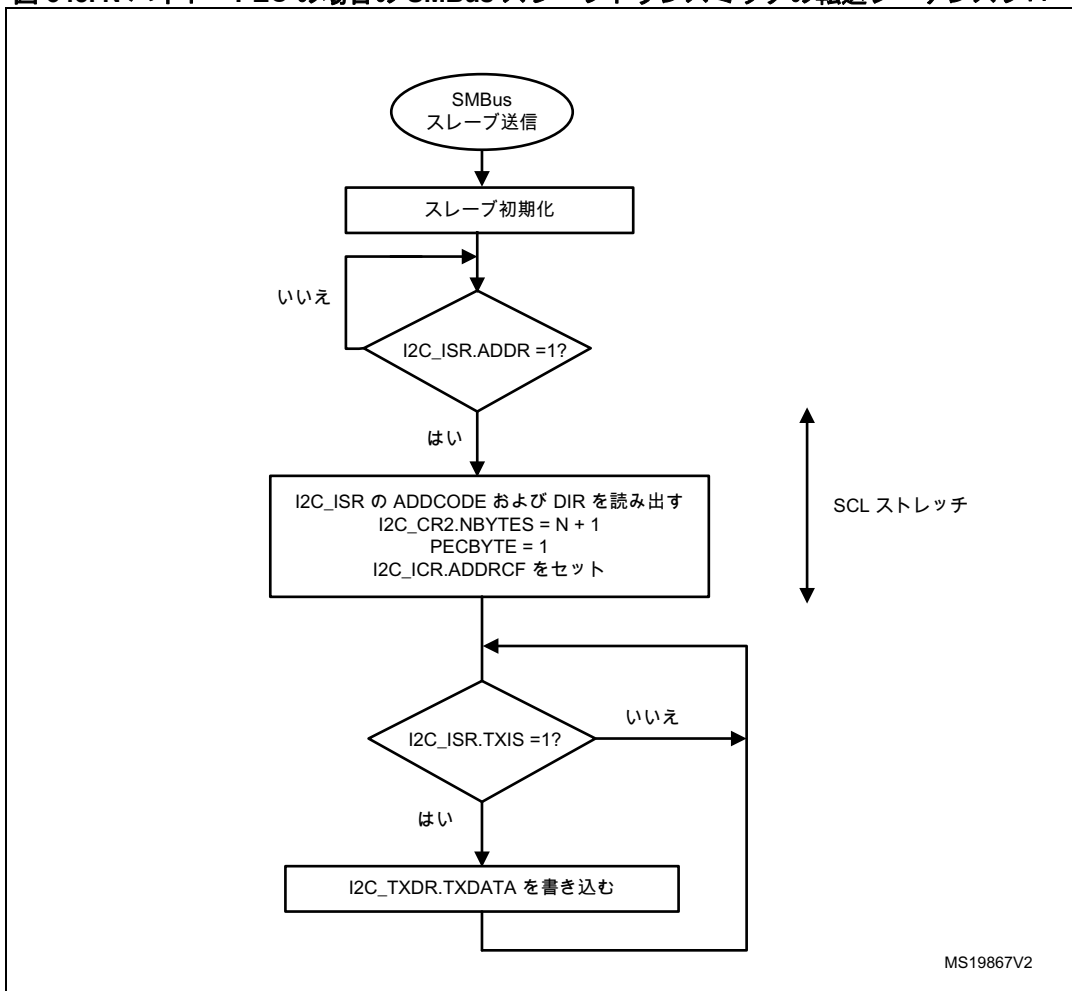
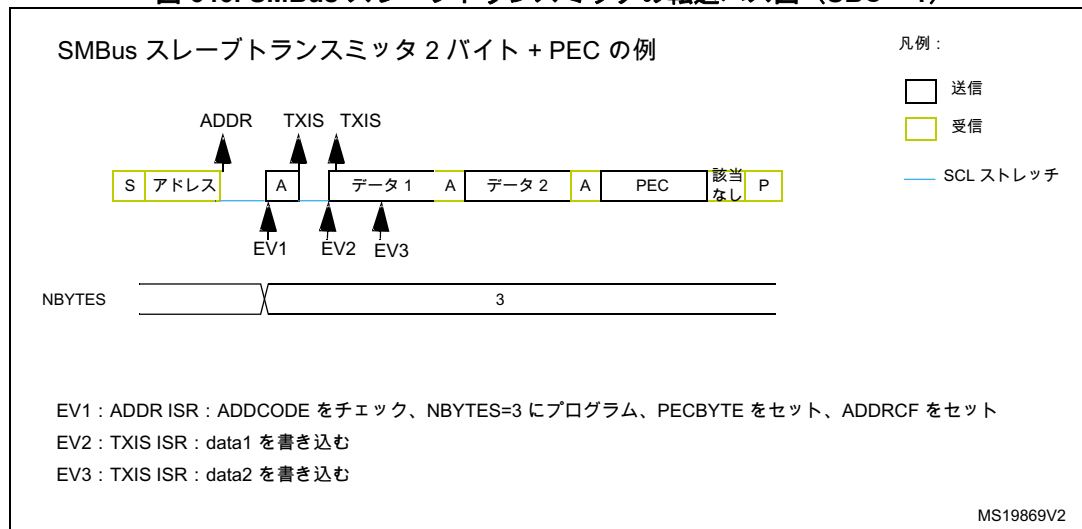


図 646. SMBus スレーブトランスミッタの転送バス図 (SBC = 1)



SMBus スレーブレシーバ

I2C が SMBus モードで使用されるときには、SBC は、プログラムされたデータバイト数の終わりの PEC チェックを可能にするため、1 にプログラムする必要があります。各バイトの ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD = 1)。詳細については、[スレーブバイト制御モード](#)を参照してください。

PEC バイトをチェックするためには、RELOAD ビットをクリアして、PECBYTE ビットをセットする必要があります。この場合、NBYTES - 1 データが受信された後、次の受信バイトが内部 I2C_PECR レジスタの内容と比較されます。ACK ビットの値にかかわらず、比較が一致しなかった場合は NACK が自動的に生成され、比較が一致した場合は ACK が自動的に生成されます。PEC バイトが受信されると、他のデータと同様に I2C_RXDR レジスタにコピーされ、RXNE フラグがセットされます。

PEC が不一致の場合、PECERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

ACK ソフトウェア制御が不要な場合は、PECBYTE = 1 をプログラムし、同じ書き込み操作で NBYTES を連続フローで受信するバイト数にプログラムします。NBYTES - 1 が受信された後、次の受信バイトが PEC であるかどうかチェックされます。

注意： PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 647. N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフロー

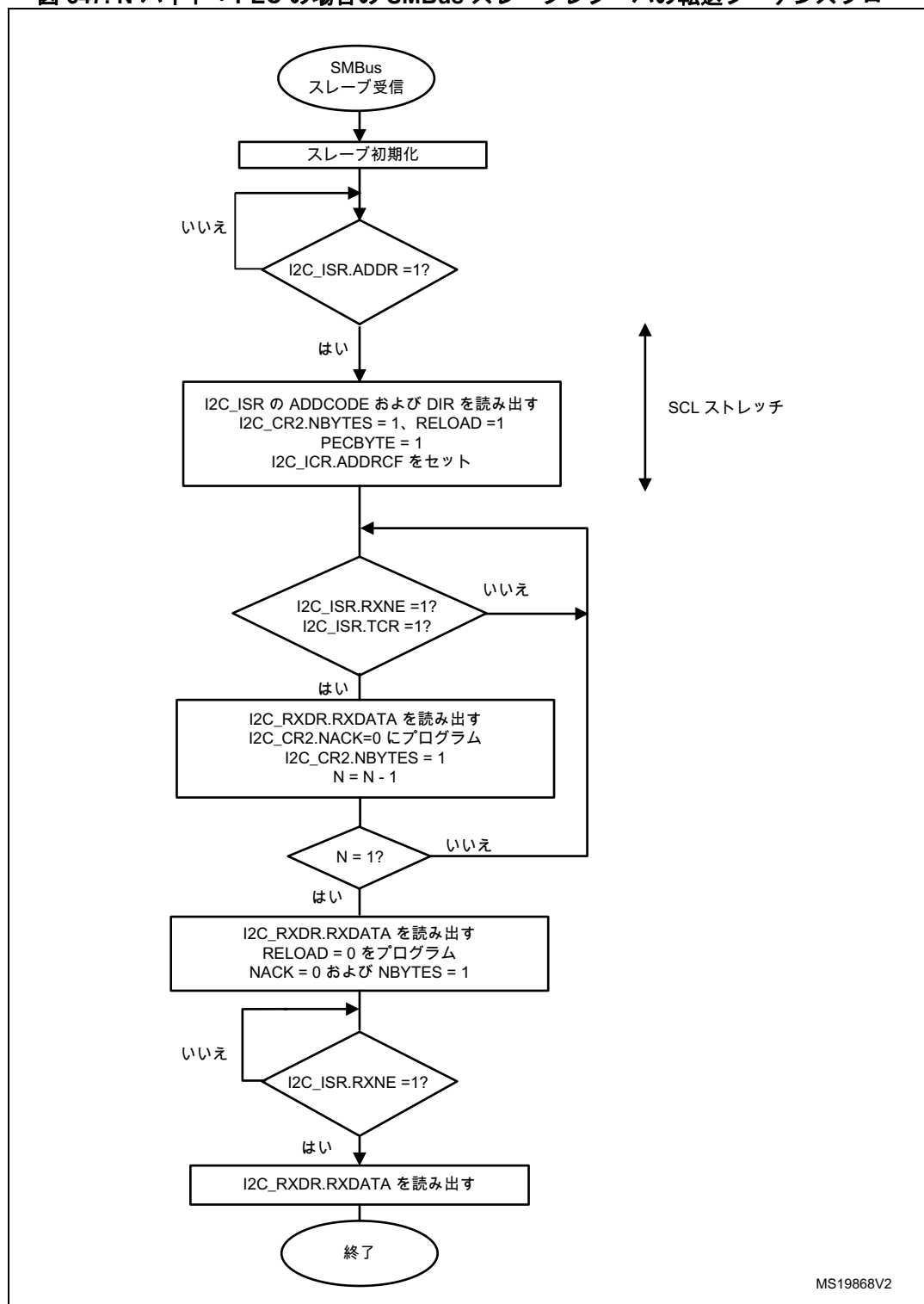
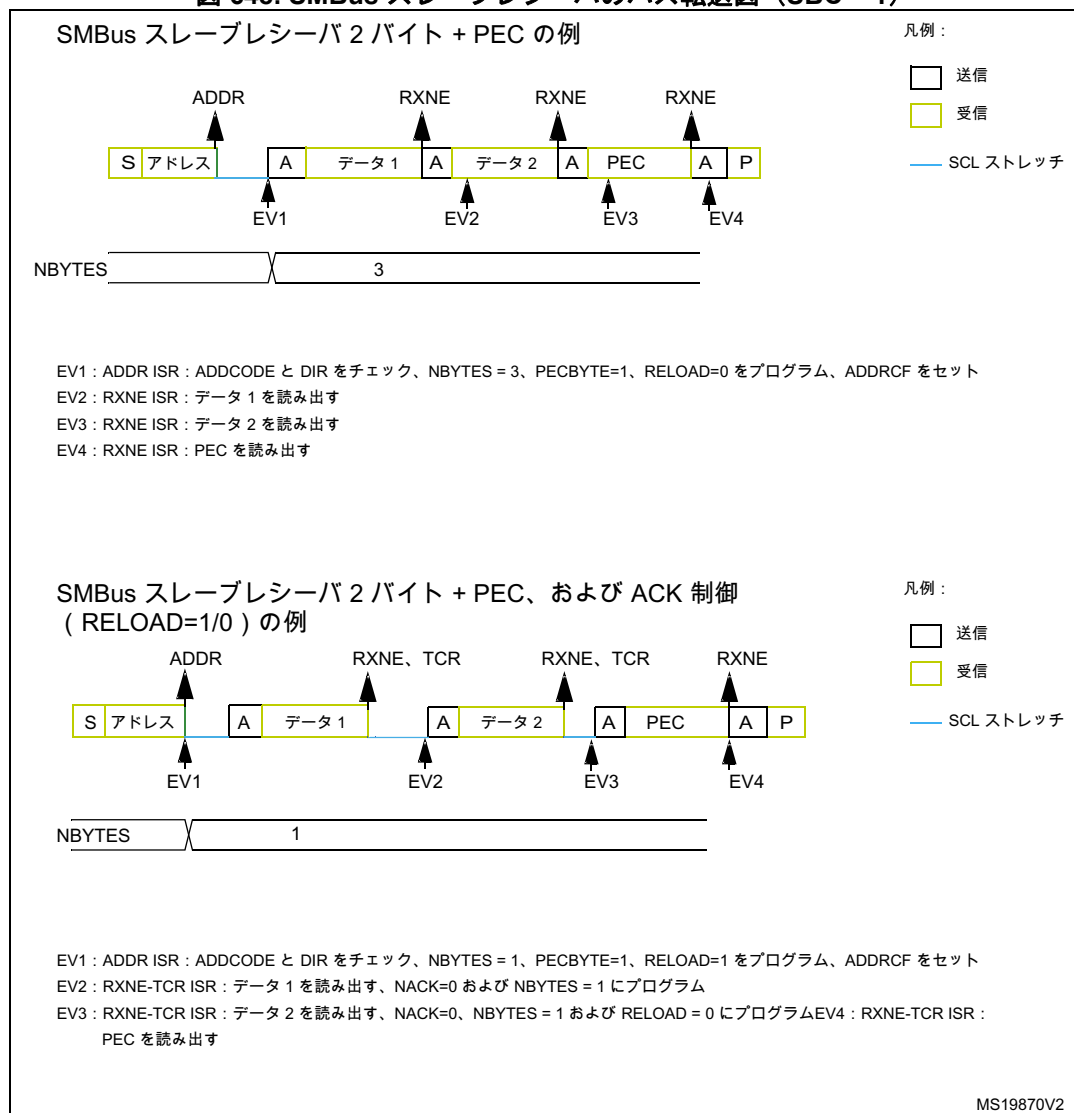


図 648. SMBus スレーブレシーバのバス転送図 (SBC = 1)



このセクションは、SMBus 機能がサポートされるときにのみ適用されます (セクション 48.3 を参照してください)。

I²C マスタ転送管理 (セクション 48.4.9 を参照) に加えて、SMBus をサポートするために、追加のソフトウェアフローが用意されています。

SMBus マスタトランスミッタ

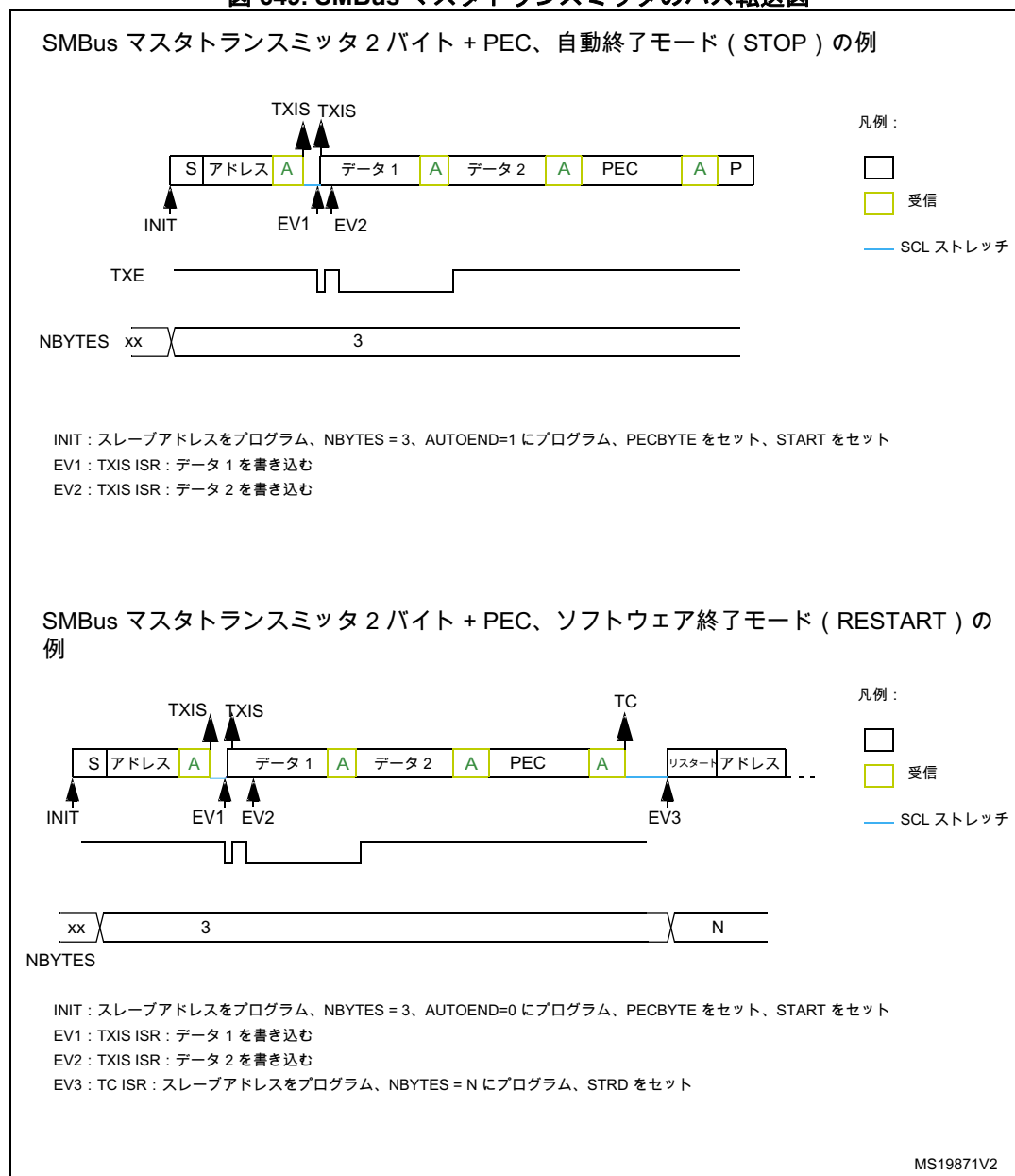
SMBus マスタが PEC を送信したいときには、START ビットをセットする前に、PECBYTE ビットをセットする必要があります。バイト数を NBYTES[7:0] フィールドでプログラムする必要があります。この場合、TXIS 割込みの合計数は NBYTES - 1 になります。したがって、NBYTES = 0x1 のときに PECBYTE ビットがセットされた場合、I2C_PECR レジスタの内容が自動的に送信されます。

SMBus マスタが PEC 後に STOP コンディションを送信したい場合は、自動終了モードを選択してください (AUTOEND = 1)。この場合、PEC 送信に続いて、STOP コンディションが自動的に送信されます。

SMBus マスタが PEC 後に RESTART コンディションを送信したい場合は、ソフトウェアモードを選択してください (AUTOEND = 0)。この場合、NBYTES - 1 が送信されると、PEC 送信後に I2C_PECR レジスタの内容が送信され、TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションを TC 割込みサブルーチンでプログラムする必要があります。

注意 : PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 649. SMBus マスタトランスミッタのバス転送図



SMBus マスタレシーバ

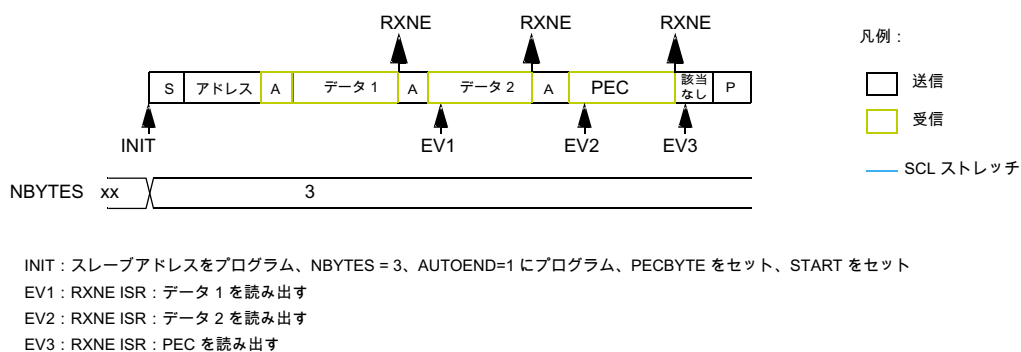
SMBus マスタが転送終了時に PEC を受信してから STOP を受信したいときには、自動終了モードを選択できます (AUTOEND = 1)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります、スレーブアドレスをプログラムする必要があります。この場合、NBYTES - 1 データが受信された後、次の受信バイトが I2C_PECR レジスタの内容と自動的に照合されます。PEC バイトに対して NACK 応答が与えられた後、STOP コンディションが送信されます。

SMBus マスタが転送終了時に PEC バイトを受信してから RESTART を受信したいときには、ソフトウェアモードを選択する必要があります (AUTOEND = 0)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります、スレーブアドレスをプログラムする必要があります。この場合、NBYTES - 1 データが受信された後、次の受信バイトが I2C_PECR レジスタの内容と自動的に照合されます。PEC バイト受信後に TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションは、TC 割込みサブルーチンでプログラムできます。

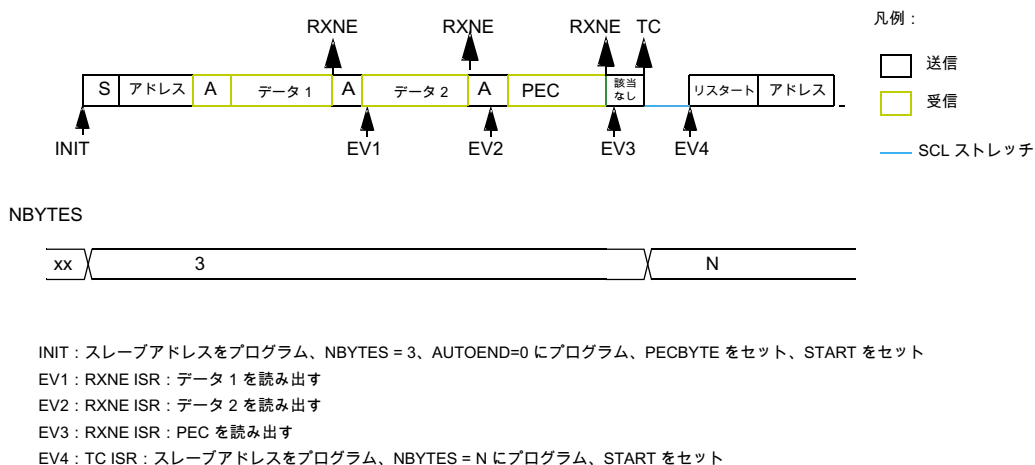
注意： PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 650. SMBus マスタレシーバのバス転送図

SMBus マスタレシーバ 2 バイト + PEC、自動終了モード (STOP) の例



SMBus マスタレシーバ 2 バイト + PEC、ソフトウェア終了モード (RESTART) の例



MS19872V2

48.4.15 アドレス一致時に STOP モードからウェイクアップ

このセクションは、STOP モードからのウェイクアップ機能がサポートされるときにのみ適用されません（セクション 48.3 を参照）。

I²C は、アドレス指定されたとき、MCU を STOP モードからウェイクアップできます（APB クロックはオフ）。すべてのアドレッシングモードがサポートされます。

STOP モードからのウェイクアップを有効にするには、I2C_CR1 レジスタの WUPEN ビットをセットします。STOP モードからのウェイクアップを可能にするには、HSI および CSI のみ オシレータを I2CCLK のクロックソースとして選択する必要があります。

STOP モード中、HSI および CSI のみ はオフです。START が検出されると、I²C インタフェースは HSI および CSI のみ をオンに切り替えて、HSI および CSI のみ がウェイクアップするまで SCL ローをストレッチします。

HSI および CSI のみ は、アドレス受信に使用されます。

アドレス一致の場合、I²C は、MCU のウェイクアップ時間の間、SCL ローをストレッチします。ストレッチは、ADDR フラグがソフトウェアによってクリアされたときにリリースされ、転送は通常通りに続行されます。

アドレスが一致しなかった場合、HSI および CSI のみ は再びオフになり、MCU はウェイクアップしません。

注： I²C クロックがシステムクロックの場合、または WUPEN = 0 の場合、START 受信後も HSI および CSI のみはオンになりません。

ADDR 割込みによってのみ、MCU をウェイクアップします。したがって、I²C がマスタとして、または ADDR フラグのセット後にアドレス指定されたスレーブとして転送を行っているときには、STOP モードに入らないでください。これを管理するには、ADDR 割込みルーチンで SLEEPDEEP ビットをクリアして、STOPF フラグのセット後にのみ再びオンにセットします。

注意： デジタルフィルタは、STOP モードからのウェイクアップ機能と互換性がありません。DNF ビットが 0 でない場合、WUPEN ビットをセットしても効果はありません。

注意： この機能は、I2C クロックソースが HSI および CSI のみ オシレータのときのみ使用できます。

注意： STOP モードからのウェイクアップ機能の正しい動作を保証するには、クロックストレッチを有効にする必要があります（NOSTRETCH = 0）。

注意： STOP モードからのウェイクアップが無効な場合（WUPEN = 0）、STOP モードに入る前に、I²C ペリフェラルが無効にする必要があります（PE = 0）。

48.4.16 エラー条件

次のエラーは、通信エラーを引き起こす可能性のある条件です。

バスエラー（BERR）

バスエラーは、START または STOP コンディションが検出され、それが 9 SCL クロックパルスの倍数の後に位置していなかったときに検出されます。START または STOP コンディションは、SCL がハイとのときに SDA エッジが発生した場合に検出されます。

バスエラーフラグは、I²C がマスタまたはアドレス指定されたスレーブとして転送に関与する場合にのみ（すなわち、スレーブモードのアドレスフェーズでないとき）、セットされます。

スレーブモードで START または RESTART の誤配置が検出された場合、I²C は、正しい START コンディションの場合と同様に、アドレス認識状態に入ります。

バスエラーが検出されると、I2C_ISR レジスタの BERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

アービトレーション喪失 (ARLO)

アービトレーション喪失は、SDA ラインでハイレベルが送信されたが、SCL 立上がりエッジでローレベルがサンプリングされたときに検出されます。

- マスタモードでは、アービトレーション喪失は、アドレスフェーズ、データフェーズ、およびデータ確認応答フェーズで検出されます。この場合、SDA および SCL ラインはリリースされ、START 制御ビットがハードウェアによってクリアされ、マスタは自動的にスレーブモードに切り替わります。
- スレーブモードでは、アービトレーション喪失は、データフェーズとデータ確認応答フェーズで検出されます。この場合、転送は中止され、SCL および SDA ラインがリリースされます。

アービトレーション喪失が検出されると、I2C_ISR レジスタの ARLO フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

オーバーラン/アンダーランエラー (OVR)

オーバーランまたはアンダーランエラーは、スレーブモードで NOSTRETCH = 1 のとき、および次のときに検出されます：

- 受信時、新しいバイトが受信され、RXDR レジスタがまだ読み出されていないとき。新しい受信バイトは失われ、新しいバイトへの応答として NACK が自動的に送信されます。
- 送信時：
 - STOPF = 1 で、最初のデータバイトが送信されなければならないとき。TXE = 0 の場合、I2C_TXDR レジスタの内容が送信され、そうでない場合は 0xFF が送信されます。
 - 新しいバイトが送信されるべきときに、I2C_TXDR レジスタがまだ書き込まれていなかった場合、0xFF が送信されます。

オーバーランまたはアンダーランエラーが検出されると、I2C_ISR レジスタの OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

パケットエラーチェックエラー (PECERR)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます ([セクション 48.3](#) を参照してください)。

PEC エラーは、受信した PEC バイトが I2C_PECR レジスタの内容と一致しなかったときに検出されます。正しくない PEC の受信後、NACK が自動的に送信されます。

PEC エラーが検出されると、I2C_ISR レジスタの PECERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

タイムアウトエラー (TIMEOUT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます ([セクション 48.3](#) を参照してください)。

タイムアウトエラーは、次のような条件で発生します：

- TIDLE = 0 であり、SCL が TIMEOUTA[11:0] ビットで定義された時間だけローのままであった場合：これは SMBus タイムアウトの検出に使用されます。
- TIDLE = 1 であり、SDA および SCL が TIMEOUTA[11:0] ビットで定義された時間だけハイのままであった場合：これはバスアイドル状態の検出に使用されます。
- マスタ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間 (SMBus $t_{\text{LOW:MEXT}}$ パラメータ) に達した場合。
- スレーブ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間 (SMBus $t_{\text{LOW:SEXT}}$ パラメータ) に達した場合。

マスタモードでタイムアウト違反が検出されると、STOP コンディションが自動的に送信されます。

スレーブモードでタイムアウト違反が検出されると、SDA および SCL ラインが自動的にリリースされます。

タイムアウトエラーが検出されると、I2C_ISR レジスタの TIMEOUT フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割込みが生成されます。

アラート (ALERT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます ([セクション 48.3](#) を参照してください)。

ALERT フラグは、I2C インタフェースがホストとして設定され (SMBHEN = 1)、アラートピン検出が有効であり (ALERTEN = 1)、SMBA ピンで立下がりエッジが検出されたときにセットされます。I2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割込みが生成されます。

48.4.17 DMA リクエスト

DMA を使用した送信

送信について DMA (Direct Memory Access) を有効にするには、I2C_CR1 レジスタの TXDMAEN ビットをセットします。TXIS ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 16 : 617 ページの汎用ダイレクト・メモリ・アクセス・コントローラ \(GPDMA\)](#) を参照) を使用して設定された SRAM 領域から I2C_TXDR レジスタにロードされます。

データのみが DMA で転送されます。

- マスタモード：初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます (送信されたスレーブアドレスを DMA で転送することはできません)。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。[マスタトランスミッタ](#)を参照してください。
- スレーブモードでは：
 - NOSTRETCH = 0 では、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
 - NOSTRETCH = 1 では、アドレス一致イベントの前に DMA を初期化する必要があります。
- SMBus をサポートする場合：PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブトランスミッタ](#) および [SMBus マスタトランスミッタ](#)を参照してください。

注： DMA が送信に使用される場合、TXIE ビットが有効である必要はありません。

DMA を使用した受信

受信について DMA (Direct Memory Access) を有効にするには、I2C_CR1 レジスタの RXDMAEN ビットをセットします。RXNE ビットがセットされているときには、データは、I2C_RXDR レジスタから DMA ペリフェラル([セクション 16:617 ページの汎用ダイレクト・メモリ・アクセス・コントローラ \(GPDMA\)](#)) を参照) を使用して設定された SRAM 領域にロードされます。データのみ (PEC を含む) が DMA で転送されます。

- マスタモード、初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。
- NOSTRETCH = 0 のスレーブモードでは、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
- SMBus がサポートされている場合 ([セクション 48.3](#)を参照)、PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブレシーバ](#)および[SMBus マスタレシーバ](#)を参照してください。

注： DMA が受信に使用される場合、RXIE ビットが有効である必要はありません。

48.4.18 デバッグモード

マイクロコントローラがデバッグモードに入ると (コア停止)、DBG モジュールの DBG_I2Cx_ 設定ビットに応じて、SMBus タイムアウトは、通常の動作を続行するか、あるいは停止します。

48.5 I²C 低電力モード

表 512. 低電力モードが I²C に与える影響

モード	説明
SLEEP	影響はありません。I ² C 割込みによって、デバイスは SLEEP モードから復帰します。
STOP ⁽¹⁾	I ² C レジスタの内容は保たれます。 – WUPEN = 1 かつ I2C が内部オシレータ (HSI および CSI のみ) によってクロック供給されている：アドレス認識が機能します。I2C アドレス一致条件によって、デバイスは STOP モードから復帰します。 – WUPEN = 0：STOP モードに入る前に I2C を無効にする必要があります。
STANDBY	I2C ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. 各インスタンスでサポートされる STOP モードについては、[セクション 48.3](#) を参照してください。特定の STOP モードからのウェイクアップがサポートされていない場合、この STOP モードに入る前に、インスタンスを無効にする必要があります。

48.6 I²C 割込み

次の表に、I²C 割込みリクエストの一覧を示します。

表 513. I²C 割込みリクエスト

項目 (割込みの略称)		割込みイベント	イベント フラグ	有効制御 ビット	割込みのクリア方法	SLEEP モードの 終了	STOP モードの 終了	STANDBY モードの 終了	
I ² C	I2C_EV	受信バッファノット エンプティ	RXNE	RXIE	I2C_RXDR レジスタを 読み出す	可	不可	不可	
		送信バッファ割込み ステータス	TXIS	TXIE	I2C_TXDR レジスタに 書き込む				
		STOP 検出割込み フラグ	STOPF	STOPIE	STOPCF = 1 を書き込む。				
		転送完了再ロード	TCR	TCIE	I2C_CR2 の NBYTES[7:0] に 0 以外 を書き込む		可 ⁽¹⁾		
		転送完了	TC		START = 1 または STOP = 1 を書き込む				
		アドレス一致	ADDR	ADDRIE	ADDRCF = 1 を 書き込む。				可 ⁽¹⁾
		NACK 受信	NACKF	NACKIE	NACKCF = 1 を書き込む。		不可		
	I2C_ER	バスエラー	BERR	ERRIE	BERRCF = 1 を書き込む。	可	不可	不可	
		アービトレーション 喪失	ARLO		ARLOCF = 1 を書き込む。				
		オーバーラン/ アンダーラン	OVR		OVRCF = 1 を書き込む。				
		PEC エラー	PECERR		PECERRCF = 1 を 書き込む				
		タイムアウト/ t _{LOW} エラー	TIMEOUT		TIMEOUTCF = 1 を書込む				
		SMBus アラート	ALERT		ALERTCF = 1 を書き込む				

1. ADDR 一致イベントは、I²C インスタンスが STOP モードからのウェイクアップ機能をサポートしている場合にのみデバイスを STOP モードからウェイクアップすることができます。セクション 48.3 を参照してください。

48.7 I²C レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタは、ワード（32 ビット）単位でアクセスされます。

48.7.1 I²C 制御レジスタ 1 (I2C_CR1)

アドレスオフセット：0x00

リセット値：0x0000 0000

アクセス：書込みアクセスが進行中のときに書込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書込みアクセスが完了するまで、2 番目の書込みアクセスにウェイト状態が挿入されます。2番目の書込みアクセスの遅延は、最大 $2 \times i2c_pclk + 6 \times i2c_ker_ck$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STOPFA CLR	ADDRAC LR	Res.	Res.	Res.	Res.	Res.	FMP	PECEN	ALERTE N	SMBDE N	SMBHE N	GCEN	WUPEN	NOSTR ETCH	SBC
rw	rw						rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMAE N	TXDMAE N	Res.	ANF OFF	DNF[3:0]				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **STOPFA CLR** : STOP 検出フラグ (STOPF) 自動クリア

0 : STOPF フラグは、ハードウェアによってセットされ、ソフトウェアによって STOPCF ビットをセットすることでクリアされます。

1 : STOPF フラグは、ハードウェアによってクリア状態に保たれます。このモードは NOSTRETCH スレーブモードで使用でき、次のデータ送信の前に STOPF フラグがクリアされない場合のオーバーランエラーを回避できます。これにより、ペリフェラルからの割込みなしで、DMA のみによるスレーブデータ管理が可能になります。

ビット 30 **ADDRACL R** : アドレス一致フラグ (ADDR) 自動クリア

0 : ADDR フラグは、ハードウェアによってセットされ、ソフトウェアによって ADDR CF ビットをセットすることでクリアされます。

1 : ADDR フラグは、ハードウェアによってクリア状態に保たれます。このモードはスレーブモードで使用でき、I²C が 1 つのスレーブアドレスのみを有効にする場合に、ADDR クロックストレッチを回避できます。これにより、ペリフェラルからの割込みなしで、DMA のみによるスレーブデータ管理が可能になります。

ビット 29:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **FMP** : 高速モードプラス 20 mA 駆動イネーブル

0 : 20 mA I/O 駆動は無効です。

1 : 20 mA I/O 駆動は有効です。

ビット 23 **PECEN** : PEC 有効化

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

注： **SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。** [セクション 48.3](#) を参照してください。

ビット 22 **ALERTEN** : SMBus アラート有効

0 : SMBus アラートピン (SMBA) はホストモード (SMBHEN = 1) ではサポートされません。デバイスモード (SMBHEN = 0) では、SMBA ピンが解放され、アラート応答アドレスヘッダが無効になります (0001100x の後には NACK が続きます)。

1 : SMBus アラートピンはホストモード (SMBHEN = 1) でサポートされます。デバイスモード (SMBHEN = 0) では、SMBA ピンがローに駆動され、アラート応答アドレスヘッダが有効になります (0001100x の後には ACK が続きます)。

注 : **ALERTEN = 0** のときには、**SMBA** ピンを標準 GPIO として使用できます。

SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

ビット 21 **SMBDEN** : SMBus デバイスデフォルトアドレス有効

0 : デバイスデフォルトアドレス無効。アドレス 0b1100001x は NACK されます。

1 : デバイスデフォルトアドレス有効。アドレス 0b1100001x は ACK されます。

注 : **SMBus** 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

ビット 20 **SMBHEN** : SMBus ホストアドレス有効

0 : ホストアドレス無効。アドレス 0b0001000x は NACK されます。

1 : ホストアドレス有効。アドレス 0b0001000x は ACK されます。

注 : **SMBus** 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

ビット 19 **GCEN** : 同報イネーブル

0 : 同報は無効です。アドレス 0b00000000 は NACK されます。

1 : 同報は有効です。アドレス 0b00000000 は ACK されます。

ビット 18 **WUPEN** : STOP モードからのウェイクアップ有効

0 : STOP モードからのウェイクアップは無効です。

1 : STOP モードからのウェイクアップは有効です。

注 : **STOP** モードからのウェイクアップ機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

注 : **WUPEN** は、**DNF = 0000** のときのみセットできます。

ビット 17 **NOSTRETCH** : クロックストレッチ無効

このビットは、スレーブモードでのクロックストレッチを無効にするために使用されます。マスターモードではクリアされたままでなければなりません。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

注 : このビットにプログラムできるのは、**I²C** が無効 (**PE = 0**) なときだけです。

ビット 16 **SBC** : スレーブバイト制御

このビットは、スレーブモードでのハードウェアバイト制御を有効にするために使用されます。

0 : スレーブバイト制御無効

1 : スレーブバイト制御有効

ビット 15 **RxDMAEN** : DMA 受信リクエスト有効

0 : DMA モードは受信に無効

1 : DMA モードは受信に有効

ビット 14 **TxDMAEN** : DMA 送信リクエスト有効

0 : DMA モードは送信に無効

1 : DMA モードは送信に有効

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **ANFOFF** : アナログノイズフィルタ OFF

0 : アナログノイズフィルタ有効

1 : アナログノイズフィルタ無効

注: このビットにプログラムできるのは、I²C が無効 (PE = 0) なときだけです。

ビット 11:8 **DNF[3:0]** : デジタルノイズフィルタ

これらのビットは、SDA および SCL 入力のデジタルノイズフィルタを設定するために使用されます。デジタルフィルタは、最大 DNF[3:0] * t_{12CCLK} の長さのスパイクを除去します。

0000 : デジタルフィルタ無効

0001 : デジタルフィルタは有効であり、最大 1 t_{12CCLK} の除去能力を持ちます。

.....

1111 : デジタルフィルタは有効であり、最大 15 t_{12CCLK} の除去能力を持ちます。

注: アナログフィルタが有効な場合、デジタルフィルタがそれに追加されます。

このフィルタがプログラムできるのは、I²C が無効 (PE = 0) なときだけです。

ビット 7 **ERRIE** : エラー割込み有効

0 : エラー検出割込み無効

1 : エラー検出割込み有効

注: 次のようなエラーが発生すると、割込みが生成されます:

アービトレーション喪失 (ARLO)

バスエラー検出 (BERR)

オーバーラン/アンダーラン (OVR)

タイムアウト検出 (TIMEOUT)

PEC エラー検出 (PECERR)

アラートピンイベント検出 (ALERT)

ビット 6 **TCIE** : 転送完了割込み有効化

0 : 転送完了割込みは無効です。

1 : 転送完了割込みは有効です。

注: 次のようなイベントが発生すると、割込みが生成されます:

転送完了 (TC)

転送完了再ロード (TCR)

ビット 5 **STOPIE** : STOP 検出割込み有効

0 : STOP 検出 (STOPF) 割込み無効

1 : STOP 検出 (STOPF) 割込み有効

ビット 4 **NACKIE** : 非確認応答受信割込み有効

0 : 非確認応答 (NACKF) 受信割込み無効

1 : 非確認応答 (NACKF) 受信割込み有効

ビット 3 **ADDRIE** : アドレス一致割込み有効 (スレーブのみ)

0 : アドレス一致 (ADDR) 割込み無効

1 : アドレス一致 (ADDR) 割込み有効

ビット 2 **RXIE** : RX 割込み有効

0 : 受信 (RXNE) 割込み無効

1 : 受信 (RXNE) 割込み有効

ビット 1 **TXIE** : TX 割込み有効

0 : 送信 (TXIS) 割込み無効

1 : 送信 (TXIS) 割込み有効

ビット 0 PE : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注 : PE = 0 のとき、I²C SCL および SDA ラインは解放されます。内部ステートマシンおよびステータスビットはリセット値に戻されます。クリアされたときには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たれる必要があります。

48.7.2 I²C 制御レジスタ 2 (I2C_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : 書込みアクセスが進行中のときに書込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書込みアクセスが完了するまで、2 番目の書込みアクセスにウェイト状態が挿入されます。2 番目の書込みアクセスの遅延は、最大 $2 \times i2c_pclk + 6 \times i2c_ker_ck$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD10 R	ADD10	RD_ WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 PECBYTE : パケットエラーチェックバイト

このビットはソフトウェアによってセットされ、PEC が転送されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、また、PE = 0 のとき、ハードウェアによってクリアされます。
0 : PEC 転送なし。

1 : PEC 送信／受信がリクエストされます。

注 : このビットに 0 を書き込んでも、効果はありません。

このビットは、RELOAD がセットされているときには効果がありません。

このビットは、SBC = 0 のとき、スレーブモードでは効果がありません。

SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

ビット 25 AUTOEND : 自動終了モード (マスタモード)

このビットは、ソフトウェアによってセット／クリアされます。

0 : ソフトウェア終了モード : NBYTES データが転送されると TC フラグがセットされ、SCL ローをストレッチします。

1 : 自動終了モード : NBYTES データが転送されると、STOP コンディションが自動的に送信されます。

注 : このビットは、スレーブモードまたは RELOAD ビットがセットされているときには効果がありません。

ビット 24 RELOAD : NBYTES 再ロードモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : 転送は、NBYTES データの転送後、完了します (STOP または RESTART が続きます)。

1 : 転送は、NBYTES データの転送後に完了しません (NBYTES が再ロードされます)。NBYTES データが転送されると TCR フラグがセットされ、SCL ローをストレッチします。

ビット 23:16 NBYTES[7:0] : バイト数

送受信されるバイト数は、ここでプログラムされます。このフィールドは、SBC = 0 のスレーブモードでは効果がありません。

注： START ビットがセットされているときに、これらのビットを変更することはできません。

ビット 15 NACK : NACK 生成 (スレーブモード)

このビットはソフトウェアによってセットされ、NACK が送信されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、または PE = 0 のとき、ハードウェアによってクリアされます。

0 : 現在の受信バイト後に ACK が送信されます。

1 : 現在の受信バイト後に NACK が送信されます。

注： このビットに 0 を書き込んでも、効果はありません。

このビットは、スレーブモードでのみ使用されます：マスタレシーバモードでは、NACK ビットの値にかかわらず、STOP または RESTART コンディション前の最後のバイト後に NACK が自動的に生成されます。

スレーブレシーバ NOSTRETCH モードでオーバーランが発生すると、NACK ビットの値にかかわらず、NACK が自動的に生成されます。

ハードウェア PEC チェックが有効なとき (PECBYTE = 1)、PEC 確認応答値は NACK 値に依存しません。

ビット 14 STOP : STOP 生成 (マスタモード)

このビットはソフトウェアによってセットされ、STOP コンディションが検出されたとき、または PE = 0 のときにハードウェアによってクリアされます。

マスタモード：

0 : STOP 生成なし。

1 : 現在のバイト転送後の STOP 生成。

注： このビットに 0 を書き込んでも、効果はありません。

ビット 13 START : START 生成

このビットはソフトウェアによってセットされ、START とアドレスシーケンスが送信された後、アービトラージ喪失によって、スレーブモードでのアドレス一致によって、タイムアウトエラー検出によって、または PE = 0 のときに、ハードウェアによってクリアされます。

0 : START 生成なし。

1 : RESTART/START 生成：

I²C がすでにマスタモードであり、AUTOEND = 0 の場合、このビットをセットすると、RELOAD = 0 のとき、NBYTES 転送の終了後に REPEATED START コンディションが生成されます。

そうでない場合、バスがフリーになると、このビットをセットすることで START コンディションが生成されます。

注： このビットに 0 を書き込んでも、効果はありません。

バスが BUSY の場合、または I²C がスレーブモードの場合でも、START ビットをセットできます。

このビットは、RELOAD がセットされているときには効果がありません。

ビット 12 HEAD10R : 読出し方向のみの 10 ビットアドレスヘッダ (マスタレシーバモード)

0 : マスタは完全な 10 ビットスレーブアドレス読出しシーケンスを送信します：START + 2 バイトの書き込み方向の 10 ビットアドレス + RESTART + 読出し方向の 10 ビットアドレスの最初の 7 ビット。

1 : マスタは 10 ビットアドレスの最初の 7 ビットのみを送信し、その後に読出し方向を送信します。

注： START ビットがセットされているときに、このビットを変更することはできません。

ビット 11 ADD10 : 10 ビットアドレッシングモード (マスタモード)

0 : マスタは 7 ビットアドレッシングモードで動作します。

1 : マスタは 10 ビットアドレッシングモードで動作します。

注： START ビットがセットされているときに、このビットを変更することはできません。

ビット 10 **RD_WRN** : 転送方向 (マスタモード)

0 : マスタは書込み転送をリクエストします。

1 : マスタは読出し転送をリクエストします。

注 : **START** ビットがセットされているときに、このビットを変更することはできません。

ビット 9:0 **SADD[9:0]** : スレーブアドレス (マスタモード)

7 ビットアドレッシングモード (ADD10=0) :

SADD[7:1] には、送信される 7 ビットのスレーブアドレスを書き込む必要があります。SADD[9]、SADD[8]、および SADD[0] ビットは無視されます。

10 ビットアドレッシングモード (ADD10=1) :

SADD[9:0] には、送信される 10 ビットのスレーブアドレスを書き込む必要があります。

注 : **START** ビットがセットされているときに、これらのビットを変更することはできません。

48.7.3 I²C Own Address 1 レジスタ (I2C_OAR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : 書込みアクセスが進行中のときに書込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書込みアクセスが完了するまで、2 番目の書込みアクセスにウェイト状態が挿入されます。2 番目の書込みアクセスの遅延は、最大 2 x i2c_pclk + 6 x i2c_ker_ck です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1 [9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA1EN** : Own Address 1 有効

0 : Own Address 1 は無効です。受信されたスレーブアドレス OA1 は NACK されます。

1 : Own Address 1 は有効です。受信されたスレーブアドレス OA1 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OA1MODE** : Own Address 1 10 ビットモード

0 : Own Address 1 は 7 ビットアドレスです。

1 : Own Address 1 は 10 ビットアドレスです。

注 : このビットは、**OA1EN = 0** のときのみ書き込むことができます。

ビット 9:0 **OA1[9:0]** : インタフェース専用スレーブアドレス

7 ビットアドレスモード : OA1[7:1] には 7 ビットの専用スレーブアドレスが含まれます。OA1[9]、OA1[8]、および OA1[0] ビットは無視されます。

10 ビットアドレスモード : OA1[9:0] には 10 ビットの専用スレーブアドレスが含まれます。

注 : これらのビットは、**OA1EN = 0** のときのみ書き込むことができます。

48.7.4 I²C Own Address 2 レジスタ (I2C_OAR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : 書込みアクセスが進行中のときに書込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書込みアクセスが完了するまで、2 番目の書込みアクセスにウェイト状態が挿入されます。2 番目の書込みアクセスの遅延は、最大 $2 \times i2c_pclk + 6 \times i2c_ker_ck$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]				OA2 [7:1]						Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA2EN** : Own Address 2 有効

0 : Own Address 2 は無効です。受信されたスレーブアドレス OA2 は NACK されます。

1 : Own Address 2 は有効です。受信されたスレーブアドレス OA2 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **OA2MSK[2:0]** : Own Address 2 マスク

000 : マスクなし。

001 : OA2[1] はマスクされ、無視されます。OA2[7:2] のみ比較されます。

010 : OA2[2:1] はマスクされ、無視されます。OA2[7:3] のみ比較されます。

011 : OA2[3:1] はマスクされ、無視されます。OA2[7:4] のみ比較されます。

100 : OA2[4:1] はマスクされ、無視されます。OA2[7:5] のみ比較されます。

101 : OA2[5:1] はマスクされ、無視されます。OA2[7:6] のみ比較されます。

110 : OA2[6:1] はマスクされ、無視されます。OA2[7] のみ比較されます。

111 : OA2[7:1] はマスクされ、無視されます。比較は行われず、すべての（予約済みを除く）7 ビット受信アドレスが確認応答されます。

注 : これらのビットは、OA2EN = 0 のときのみ書き込むことができます。

OA2MSK が 0 でなくなると、予約済み I2C アドレス (0b0000xxx および 0b1111xxx) は、比較が一致した場合でも確認応答されません。

ビット 7:1 **OA2[7:1]** : インタフェースアドレス

7 ビットアドレスモード : 7 ビットアドレス

注 : これらのビットは、OA2EN = 0 のときのみ書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

48.7.5 I²C タイミングレジスタ (I2C_TIMINGR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 **PRESC[3:0]** : タイミングプリスケアラ

このフィールドは、データのセットアップおよびホールドカウンタ (I2C のタイミングを参照) と SCL ハイおよびローレベルカウンタ (I2C マスタ初期化 を参照) に使用されるクロック周期 t_{PRESC} を生成するために、i2c_ker_ck をプリスケールするのに使用されます。

$$t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{I2CLK}}$$

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **SCLDEL[3:0]** : データセットアップ時間

このフィールドは、SDA エッジと SCL 立上がりエッジの間に遅延 t_{SCLDEL} を生成するために使用されます。NOSTRETCH = 0 でのマスタモードおよびスレーブモードでは、SCL ラインは、 t_{SCLDEL} の間ローにストレッチされます。

$$t_{\text{SCLDEL}} = (\text{SCLDEL} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLDEL} は、 $t_{\text{SU:DAT}}$ タイミングを生成するために使用されます。

ビット 19:16 **SDADEL[3:0]** : データホールド時間

このフィールドは、SCL 立下がりエッジと SDA エッジの間に遅延 t_{SDADEL} を生成するために使用されます。NOSTRETCH = 0 でのマスタモードおよびスレーブモードでは、SCL ラインは、 t_{SDADEL} の間ローにストレッチされます。

$$t_{\text{SDADEL}} = \text{SDADEL} \times t_{\text{PRESC}}$$

注 : t_{SDADEL} は、 $t_{\text{HD:DAT}}$ タイミングを生成するために使用されます。

ビット 15:8 **SCLH[7:0]** : SCL ハイ周期 (マスタモード)

このフィールドは、マスタモードで SCL ハイ周期を生成するために使用されます。

$$t_{\text{SCLH}} = (\text{SCLH} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLH} は、 $t_{\text{SU:STO}}$ および $t_{\text{HD:STA}}$ タイミングを生成するためにも使用されます。

ビット 7:0 **SCLL[7:0]** : SCL ロー周期 (マスタモード)

このフィールドは、マスタモードで SCL ロー周期を生成するために使用されます。

$$t_{\text{SCLL}} = (\text{SCLL} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLL} は、 t_{BUF} および $t_{\text{SU:STA}}$ タイミングを生成するためにも使用されます。

注 : このレジスタは、I2C が無効 (PE=0) のときに設定する必要があります。

注 : STM32CubeMX ツールは、I²C 設定ウィンドウの I2C_TIMINGR コンテンツを計算し、提供します。

48.7.6 I²C タイムアウトレジスタ (I2C_TIMEOUTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

アクセス : 書込みアクセスが進行中のときに書込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書込みアクセスが完了するまで、2 番目の書込みアクセスにウェイト状態が挿入されます。2 番目の書込みアクセスの遅延は、最大 $2 \times i2c_pclk + 6 \times i2c_ker_ck$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **TEXTEN** : 拡張クロックタイムアウト有効

0 : 拡張クロックタイムアウト検出は無効です。

1 : 拡張クロックタイムアウト検出は有効です。 $t_{LOW:EXT}$ を超える累積 SCL ストレッチが I2C インタフェースによって行われると、タイムアウトエラーが検出されます (TIMEOUT = 1)。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **TIMEOUTB[11:0]** : バスタイムアウト B

このフィールドは、累積クロック拡張タイムアウトを設定するために使用されます :

マスタモードでは、マスタ累積クロックロー拡張時間 ($t_{LOW:MEXT}$) が検出されます。

スレーブモードでは、スレーブ累積クロックロー拡張時間 ($t_{LOW:SEXT}$) が検出されます。

$$t_{LOW:EXT} = (TIMEOUTB + TIDLE = 01) \times 2048 \times t_{2CCLK}$$

注 : これらのビットは、**TEXTEN = 0** のときのみ書き込むことができます。

ビット 15 **TIMOUTEN** : クロックタイムアウト有効

0 : SCL タイムアウト検出は無効です。

1 : SCL タイムアウト検出は有効です。SCL が $t_{TIMEOUT}$ (TIDLE = 0) を超えてローであるか、 t_{IDLE} (TIDLE = 1) を超えてハイであった場合、タイムアウトエラーが検出されます (TIMEOUT = 1)。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TIDLE** : アイドルクロックタイムアウト検出

0 : TIMEOUTA は、SCL ロータイムアウトの検出に使用されます。

1 : TIMEOUTA は、SCL と SDA の両方のハイタイムアウト (バスアイドル条件) の検出に使用されます。

注 : このビットは、**TIMOUTEN = 0** のときのみ書き込むことができます。

ビット 11:0 **TIMEOUTA[11:0]** : バスタイムアウト A

このフィールドは、以下を設定するために使用されます :

TIDLE = 0 のときの SCL ロータイムアウト条件 $t_{TIMEOUT}$

$$t_{TIMEOUT} = (TIMEOUTA + 1) \times 2048 \times t_{2CCLK}$$

TIDLE = 1 のときのバスアイドル条件 (SCL と SDA の両方のハイ)

$$t_{IDLE} = (TIMEOUTA + 1) \times 4 \times t_{2CCLK}$$

注 : これらのビットは、**TIMOUTEN = 0** のときのみ書き込むことができます。

注 : SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。[セクション 48.3](#) を参照してください。

48.7.7 I²C 割込みおよびステータスレジスタ (I2C_ISR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0001

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 **ADDCODE[6:0]** : アドレス一致コード (スレーブモード)

これらのビットは、アドレス一致イベントが発生したときに (ADDR = 1)、受信したアドレスで更新されます。

10 ビットアドレスの場合、ADDCODE は 10 ビットのヘッダとその後のアドレスの 2 つの MSB を示します。

ビット 16 **DIR** : 転送方向 (スレーブモード)

このフラグは、アドレス一致イベントが発生したときに (ADDR = 1)、更新されます。

0 : 書き込み転送、スレーブはレシーバモードになります。

1 : 読出し転送、スレーブはトランスミッタモードになります。

ビット 15 **BUSY** : バスビジー

このフラグは、バスで通信が進行中であることを示します。START コンディションが検出されたときに、ハードウェアによってセットされます。STOP コンディションが検出されたとき、または PE = 0 のときにハードウェアによってクリアされます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERT** : SMBus アラート

このフラグは、SMBHEN = 1 (SMBus ホスト設定)、ALERTEN = 1、および SMBALERT イベント (立下がりエッジ) が SMBA ピンで検出されたときに、ハードウェアによってセットされます。ALERTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注 : このビットは、PE = 0 のとき、ハードウェアによってクリアされます。**SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 48.3 を参照してください。**ビット 12 **TIMEOUT** : タイムアウトまたは t_{LOW} 検出フラグ

このフラグは、タイムアウトまたは拡張クロックタイムアウトが発生したときに、ハードウェアによってセットされます。TIMEOUTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注 : このビットは、PE = 0 のとき、ハードウェアによってクリアされます。**SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 48.3 を参照してください。**

ビット 11 PECERR : 受信時の PEC エラー

このフラグは、受信した PEC が PEC レジスタの内容に一致しないときに、ハードウェアによってセットされます。正しくない PEC の受信後、NACK が自動的に送信されます。PECCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 48.3](#) を参照してください。

ビット 10 OVR : オーバーラン／アンダーラン（スレーブモード）

このフラグは、NOSTRETCH = 1 のスレーブモードで、オーバーラン／アンダーランエラーが発生したときに、ハードウェアによってセットされます。OVRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 9 ARLO : アービトレーション喪失

このフラグは、アービトレーション喪失の場合に、ハードウェアによってセットされます。ARLOCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 8 BERR : バスエラー

このフラグは、ペリフェラルが転送に関与しているので、START または STOP コンディションの誤配置が検出されたときに、ハードウェアによってセットされます。このフラグは、スレーブモードのアドレスフェーズではセットされません。BERRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 7 TCR : 転送完了再ロード

このフラグは、RELOAD = 1 で、NBYTES データが転送されたときに、ハードウェアによってセットされます。NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

このフラグは、マスタモード、または SBC ビットがセットされているときのスレーブモードでのみ使用されます。

ビット 6 TC : 転送完了（マスタモード）

このフラグは、RELOAD = 0、AUTOEND = 0、および NBYTES データが転送されたときに、ハードウェアによってセットされます。START ビットまたは STOP ビットがセットされたときに、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 5 STOPF : STOP 検出フラグ

このフラグは、バス上で STOP コンディションが検出され、ペリフェラルがこの転送に関与しているときに、ハードウェアによってセットされます：

- － マスタとして。ただし、STOP コンディションがペリフェラルによって生成される場合。
- － または、スレーブとして。ただし、ペリフェラルがこの転送中にアドレス指定されていた場合。

STOPCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 4 NACKF : 非確認応答受信フラグ

このフラグは、バイト送信後に NACK を受信したときに、ハードウェアによってセットされます。NACKCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 3 ADDR : アドレス一致（スレーブモード）

このビットは、受信したスレーブアドレスが有効なスレーブアドレスの 1 つに一致したときに、ハードウェアによってセットされます。ADDRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 2 **RXNE** : 受信データレジスタノットエンプティ (レシーバ)

このビットは、受信データが I2C_RXDR レジスタにコピーされ、読み出す準備ができたときに、ハードウェアによってセットされます。I2C_RXDR が読み出されたときにクリアされます。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 1 **TXIS** : 送信割込みステータス (トランスミッタ)

このビットは、I2C_TXDR レジスタが空であり、送信データを I2C_TXDR レジスタに書き込む必要があるときに、ハードウェアによってセットされます。次の送信データが I2C_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、NOSTRETCH = 1 のときのみ、ソフトウェアによって 1 を書き込んで、TXIS イベントを生成することができます (TXIE = 1 の場合に割込み、または TXDMAEN = 1 の場合に DMA リクエストを生成)。

注： このビットは、PE = 0 のとき、ハードウェアによってクリアされます。

ビット 0 **TXE** : 送信データレジスタエンプティ (トランスミッタ)

このビットは、I2C_TXDR レジスタが空のときに、ハードウェアによってセットされます。次の送信データが I2C_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、ソフトウェアによって 1 を書き込んで、送信データレジスタ I2C_TXDR を一掃できます。

注： このビットは、PE = 0 のとき、ハードウェアによってセットされます。

48.7.8 I²C 割込みクリアレジスタ (I2C_ICR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERTC F	TIMOUT CF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF F	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERTCF** : アラートフラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの ALERT フラグがクリアされます。

注： SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 48.3 を参照してください。

ビット 12 **TIMOUTCF** : タイムアウト検出フラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの TIMEOUT フラグがクリアされます。

注： SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 48.3 を参照してください。

ビット 11 **PECCF** : PEC エラーフラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの PECERR フラグがクリアされます。

注： SMBus 機能がサポートされていない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 48.3 を参照してください。

ビット 10 **OVRCF** : オーバーラン/アンダーランフラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの OVR フラグがクリアされます。

ビット 9 **ARLOCF** : アービトレーション喪失フラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの ARLO フラグがクリアされます。

ビット 8 **BERRCF** : バスエラーフラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの BERRF フラグがクリアされます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **STOPCF** : STOP 検出フラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの STOPF フラグがクリアされます。

ビット 4 **NACKCF** : 非確認応答フラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの NACKF フラグがクリアされます。

ビット 3 **ADDRCF** : アドレス一致フラグクリア

このビットに 1 を書き込むと、I2C_ISR レジスタの ADDR フラグがクリアされます。このビットに 1 を書き込むと、I2C_CR2 レジスタの START ビットもクリアされます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

48.7.9 I²C PEC レジスタ (I2C_PECR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PEC[7:0]** : パケットエラーチェックレジスタ

PECEN=1 のとき、このフィールドは内部 PEC を含みます。

PEC は、PE = 0 のとき、ハードウェアによってクリアされます。

注 : SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。[セクション 48.3](#) を参照してください。

48.7.10 I²C 受信データレジスタ (I2C_RXDR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXDATA[7:0]** : 8 ビットの受信データ

I²C バスから受信したデータバイト。

48.7.11 I²C 送信データレジスタ (I2C_TXDR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXDATA[7:0]** : 8 ビットの送信データ

I²C バスに送信されるデータバイト。

注 : これらのビットは、TXE = 1 のときのみ書き込むことができます。

48.7.12 I²C レジスタマップ

次の表に、I²C のレジスタマップとリセット値を示します。

表 514. I²C レジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x0	I2C_CR1	STOPFACLR	ADDRACLR	Res.	Res.	Res.	Res.	Res.	FMP	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]			ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE			
	リセット値	0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x4	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]												
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x8	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1 [9:0]											
	リセット値																	0					0	0	0	0	0	0	0	0	0	0			
0xC	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2 [7:1]					Res.						
	リセット値																	0					0	0	0	0	0	0	0	0	0				
0x10	I2C_TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL [3:0]			SDADEL [3:0]			SCLH[7:0]					SCLL[7:0]														
	リセット値	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	I2C_TIMEOCTR	TEXTEN		Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]													
	リセット値	0				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0		
0x18	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE			
	リセット値								0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1		
0x1C	I2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECFCF	OVRFCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.		
	リセット値																			0	0	0	0	0	0			0	0	0					
0x20	I2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]									
	リセット値																									0	0	0	0	0	0	0	0		
0x24	I2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]									
	リセット値																									0	0	0	0	0	0	0	0		
0x28	I2C_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]									
	リセット値																									0	0	0	0	0	0	0	0		

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

49 I3C (Improved inter-integrated circuit)

49.1 概要

I3C インタフェースは、I3C バス上で接続された、このデバイスと他のセンサやホスト・プロセッサなどとの間の通信を処理します。

I3C バスは、従来の I²C バスの改良を目的とした、2 線でシリアル・シングルエンドのマルチドロップ・バスです。

I3C SDR 専用ペリフェラルは、MIPI® I3C 仕様 v1.1 に規定されたすべての機能を実装しています。I3C バス固有のシーケンス、プロトコル、アービトレーション、およびタイミングを制御でき、コントローラ（旧マスタ）やターゲット（旧スレーブ）として機能します。

コントローラとして機能する場合、I3C ペリフェラルは一部後方互換性を保持したまま I²C インタフェースを改良します。これにより、I²C ターゲットは、従来の I²C 高速モード（Fm）または従来の I²C 高速モードプラス（Fm+）で I3C バスを操作できます（ただし、後者がクロックストレッチを行わない場合に限る）。

I3C ペリフェラルは、CPU をオフロードするために DMA と組み合わせて使用できます。

49.2 I3C の主な機能

I3C ペリフェラルは、以下をサポートしています。

- MIPI® I3C 仕様 v1.1（詳細については、表 518 を参照）、例：
 - I3C SDR 専用 1 次コントローラ
 - I3C SDR 専用 2 次コントローラ
 - I3C SDR 専用ターゲット
- 最大 12.5MHz までの I3C SCL バスクロック周波数
- APB スレーブポート経由のホストアプリケーションからのレジスタ設定
- キューに登録されたデータの転送：
 - I3C バスで送信するデータバイト/ワードの送信 FIFO（TX-FIFO）
 - I3C バスで受信するデータバイト/ワードの受信 FIFO（RX-FIFO）
 - FIFO ごとの、専用 DMA チャンネルのあるオプションの DMA モード
- キューに登録された制御/ステータス転送（コントローラが以下の場合）：
 - I3C バスで送信する制御ワードの制御 FIFO（C-FIFO）
 - I3C バスで受信するステータスワードのオプションのステータス FIFO（S-FIFO）
 - FIFO ごとの、専用 DMA チャンネルのあるオプションの DMA モード
- メッセージ：
 - Fm/Fm+ の従来の I²C ターゲットへの従来の I²C 読出し/書込みメッセージ
 - I3C SDR 読出し/書込みプライベート・メッセージ
 - I3C SDR ブロードキャスト CCC メッセージ（詳細については、表 524 を参照）
 - I3C SDR 読出し/書込み直接 CCC メッセージ（詳細については、表 524 を参照）

- フレームレベルの管理（コントローラが以下の場合）：
 - オプションの C-FIFO および TX-FIFO のプリロード
 - 複数のメッセージのカプセル化
 - オプションの I3C バスでのアービトレーション可能なヘッダ生成
 - エラー・リカバリのための I3C バスでの HDR 終了パターン生成
- プログラム可能なバスタイミング（コントローラが以下の場合）：
 - SCL ハイ/ロー周期
 - SDA ホールド時間
 - バスフリー時間（最小）
 - バス使用可能/アイドル条件時間
 - クロックのストール時間
- ターゲットによるリクエスト管理：
 - コントローラ使用時、最大4ターゲット同時サポート
 - 帯域内割込み、プログラム可能な IBI ペイロード（最大 4 バイト）およびペンディング中の読出し通知サポートあり
 - バス制御リクエスト、リカバリ・フロー・サポートおよびハンドオフ遅延あり
 - ホットジョイン・メカニズム
- HDR 終了パターンの検出（ターゲットの場合）：
- バスエラー管理：
 - $x = 0, 1, 2, 3$ の CEx（コントローラの場合）
 - $x = 0, 1 \dots 6$ の TEx（ターゲットの場合）
 - バス制御スイッチエラーおよびリカバリ
 - ターゲットのリセット
- 個別のプログラム可能なイベントベースの管理：
 - フラグ報告やクリア制御によるイベントごとの識別
 - フラグポーリング、および/またはイベントごとのプログラム可能なイネーブルの割込みによる、ホストアプリケーション通知
 - エラータイプ識別
- コントローラとして STOP モードからウェイクアップ（[セクション 49.3.2](#) を参照）：
 - ペイロードなしの帯域内割込みによる
 - ホットジョイン・リクエストによる
 - コントローラ - 機能リクエストによる
- ターゲットとして STOP モードからウェイクアップ（[セクション 49.3.2](#) を参照）：
 - リセット・パターンによる
 - 開始の見逃しによる
- マルチクロック・ドメイン管理：
 - SCL クロックに加え、RCC によって個別にプログラムされたクロックソースから駆動された、個別の APB クロックおよびカーネルクロック
 - カーネルクロックと APB クロックの最小動作周波数に対する、アプリケーション駆動の SCL クロック（クロックの制約については、[セクション 49.6.2](#) を参照）

49.3 I3C の実装

49.3.1 I3C のインスタンス化

デバイスには、単一の I3C インスタンスがあります。

49.3.2 低電力モードからの I3C のウェイクアップ

ペリフェラルは、デバイスを低電力モードからウェイクアップできます。詳細については、[表 515](#) を参照してください。ウェイクアップ機能の詳細については、[セクション 49.13](#) を参照してください。

表 515. I3C のウェイクアップ

ウェイクアップ
SVOS3 から STOP モード

49.3.3 I3C の FIFO

FIFO は[表 516](#) の定義に従って実装されます。

表 516. I3C FIFO の実装

FIFO	内容	単位	サイズ (単位)	コントローラ/ターゲットとして使用 (理由)
C-FIFO	(32 ビット) 制御ワード	ワード	2	コントローラ (フレームが複数の制御ワードに基づいている場合があり、ターゲットに該当しない)
S-FIFO	(32 ビット) ステータスワード			コントローラ (ターゲット: ステータスがレジスタ・モードのみ)
TX-FIFO	送信データ	バイト	8	コントローラおよびターゲット
RX-FIFO	受信データ			

49.3.4 I3C トリガ

この製品ではこの機能を使用できません。I3C ペリフェラルへの入力として接続されているハードウェアトリガ信号がありません。

49.3.5 I3C 割込み

割込みマッピングは、[表 517](#) に詳しく記載されているとおりに実装されています。

表 517. I3C 割込み

信号名	信号タイプ	説明
i3c_err_it	O	エラーの割込みライン
i3c_evt_it	O	イベントの割込みライン

49.3.6 I3C MIPI® サポート

I3C ペリフェラルは、表 518 の定義に従って、MIPI 仕様 v1.1 をサポートします。

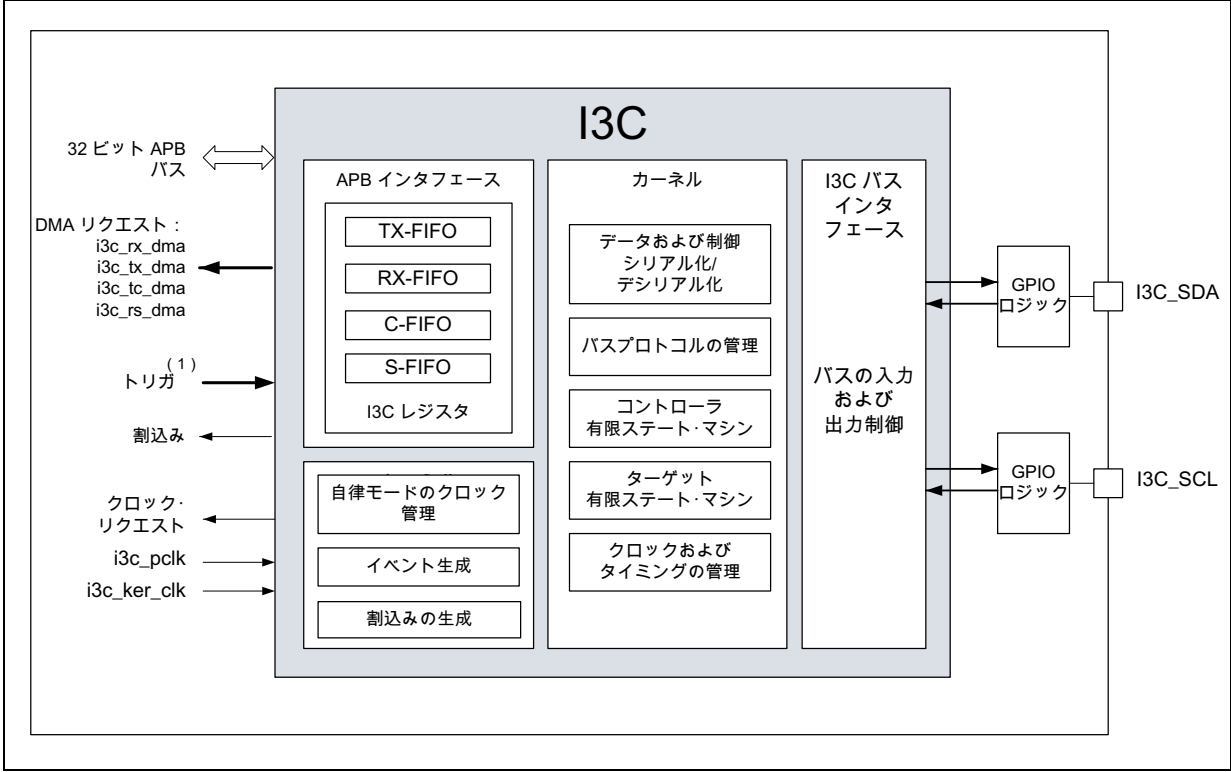
表 518. I3C ペリフェラルのコントローラ/ターゲット機能と MIPI v1.1

機能	MIPI I3C v1.1	I3C ペリフェラル		コメント
		コントローラの場合	ターゲットの場合	
I3C SDR メッセージ	X	X	X	-
従来の I ² C メッセージ (Fm/Fm+)	X	X	-	コントローラならびに I3C バスが、従来の I ² C ターゲットと混在している場合に必須。 ターゲットの場合、MIPI v1.1 ではオプション。
HDR DDR メッセージ	X	-	-	MIPI v1.1 ではオプション
HDR-TSL/TSP、HDR-BT	X	-	-	MIPI v1.1 ではオプション
動的なアドレス割り当て	X	X	X	-
静的アドレス	X	X	-	I ² C バスでペリフェラルをターゲットとした（意図的な）サポートなし
グループアドレス	X	X	-	MIPI v1.1 ではオプション
CCC	X	X	X	必須、ならびにオプションの一部の CCC をサポート（コントローラ/ターゲットの場合は表 524 を参照）。
エラー検出およびリカバリ	X	X	X	-
帯域内割込み（MDB あり）	X	X	X	-
2 次コントローラ	X	X	X	-
ホットジョイン・メカニズム	X	X	X	-
ターゲットのリセット	X	X	X	-
同期タイミング制御	X	X	-	MIPI v1.1 ではオプション
非同期タイミング制御 0	X	X	-	コントローラの場合、MIPI v1.1 で必須。ターゲットの場合、MIPI v1.1 ではオプション。
非同期タイミング制御 1、2、3	X	-	-	MIPI v1.1 ではオプション
デバイス間トンネリング	X	X	-	MIPI v1.1 ではオプション
マルチレーン・データ転送	X	-	-	MIPI v1.1 ではオプション
監視デバイスの早期終了	X	-	-	MIPI v1.1 ではオプション

49.4 I3C ブロック図

図 651に I3C ブロック図を示します。

図 651. I3C ブロック図



1. この機能は実装によって使用できない場合があります。セクション 49.3.4 : I3C トリガを参照してください。

49.5 I3C ピンおよび内部信号

表 519. I3C の入出力ピン

ピン名	信号タイプ	説明
I3C_SDA	双方向	I3C バス・シリアル・データ・ライン
I3C_SCL	双方向	I3C バス・シリアル・クロック・ライン

表 520. I3C 内部入力／出力信号

信号名	信号タイプ	説明
i3c_pclk	I	APB クロック
i3c_ker_clk	I	カーネルクロック (I3CCLK とする)
i3c_pclk_req	O	APB クロックリクエスト
i3c_ker_clk_req	O	カーネルクロックリクエスト
i3c_it ⁽¹⁾	O	グローバル割り込みライン
i3c_err_it ⁽¹⁾	O	エラーの割り込みライン

表 520. I3C 内部入力／出力信号 (続き)

信号名	信号タイプ	説明
i3c_evt_it ⁽¹⁾	O	イベントの割込みライン
i3c_rx_dma	O	RX-FIFO から受信したバイト/ワードを読み出すための DMA リクエスト。
i3c_tx_dma	O	TX-FIFO に送信するバイト/ワードを書き込むための DMA リクエスト。
i3c_tc_dma	O	ペリフェラルがコントローラとして機能する場合の、C-FIFO に送信する制御ワードを書き込むための DMA リクエスト。
i3c_rs_dma	O	ペリフェラルがコントローラとして機能する場合の、S-FIFO からステータスワードを読み出すための DMA リクエスト。

1. この信号は実装によって異なります。セクション 49.3.5 を参照してください。

49.6 I3C のリセットおよびクロック

49.6.1 I3C リセット

システムリセット時に I3C ペリフェラルがリセットされます。

一方で、このソフトウェアでは、リセットおよびクロックコントローラ (RCC) の、対応するリセット制御ビット (I3CxRST) を書き込むことで、特にペリフェラルをリセットできます。詳細については、本書の RCC のセクションを参照してください。

また、ターゲットとして機能する場合、有効化したペリフェラル (I3C_CFGR レジスタで EN = 1) は、I3C バスの帯域内リセット・パターンをコントローラから受信できます。ソフトウェアはその後、ブロードキャストまたは直接 RSTACT CCC の以前の受信時に I3C_DEVR0 レジスタの RSTACT[1:0] に登録されたとおりに、リクエストした操作を実行するよう通知されます (I3C_EVR レジスタで RSTF = 1 の場合、および/または対応する割込みが有効化された場合)。詳細については、表 524 および セクション 49.16.16 を参照してください。

このリセット割込み通知は、I3C ペリフェラル (通常は V_{CORE} ドメインにある) はアクティブになっている低電力モードからウェイクアップするために使用できます。

対応する低電力モードの詳細については、PWR セクションのパワー・マネージメントを参照してください。

49.6.2 I3C のクロックおよび要件

図 651 に示すように、I3C ペリフェラルにはいくつかのクロックドメインが実装されています。

- SCL バスクロック : I3C バスインタフェース用
 - コントローラの場合 : ユーザは、[コントローラの初期化および転送用の設定更新 \(コントローラの場合\)](#) に要約されるように、[I3C タイミングレジスタ 0 \(I3C_TIMINGR0\)](#)、[I3C タイミングレジスタ 1 \(I3C_TIMINGR1\)](#)、および [I3C タイミングレジスタ 2 \(I3C_TIMINGR2\)](#) をプログラミングすることで、SCL/SDA タイミングをセットする必要があり、また調整することができます。
 - ターゲットの場合 : ユーザは、[ターゲットの初期化](#) に要約されるように、[I3C タイミングレジスタ 1 \(I3C_TIMINGR1\)](#) をプログラミングすることで、バス使用可能条件 (帯域内割込みまたはコントローラ機能リクエストの場合は t_{AVAIL}) ならびにバスアイドル条件 (ホットジョイン・リクエストの場合は t_{IDLE}) をセットし、適合する必要があります。

- I3CCLK カーネルクロック : I3C プロトコルの管理、データおよび制御のシリアル化/デシリアル化、コントローラとターゲットの有限ステート・マシン、バスクロック、およびタイミングの管理のため
- APB クロック : APB インタフェース、DMA インタフェース、イベント、および割込み生成のため

APB クロックとカーネルクロックは、RCC によって個別にプログラムされたクロックソースから駆動されます（[セクション 11 : リセットおよびクロック制御 \(RCC\)](#) を参照）。

I3C カーネルクロックの要件（コントローラの場合）

バスの SCL クロックの目的値に応じて、アプリケーションでは I3CCLK カーネルクロックの周波数が、SCL クロックの周波数の 2 倍以上であることを保証する必要があります。

注 : $F_{SCL\ max} = 12.9\text{MHz}$ を維持することは、I3CCLK カーネルクロックの周波数が 25.8MHz を超えていることを意味します。

I3C カーネルクロックの要件（ターゲットの場合）

バスの SCL クロックの目的値に応じて、アプリケーションでは次の制約を満たす I3CCLK カーネルクロックの最小動作周波数を保証する必要があります。

1. I3CCLK カーネルクロックの周期 $< t_{HIGH}$ (SCL クロックのハイ周期)
 - $t_{HIGH\ min} = 24\text{ ns}$ 。41.7MHz を超える周波数ではこの制約が保証されます。この制約は、I3C バス/コントローラによって緩和できます。
2. I3CCLK カーネルクロックの周期 $< t_{CASr}$ (反復 START コンディション後のクロック)
 - $t_{CASr\ min} = t_{CAS\ min} / 2 = 19.2\text{ ns}$ 。52MHz を超える周波数ではこの制約が保証されます。この制約は、I3C バス/コントローラによって緩和できます。
3. I3CCLK カーネルクロックの 2 周期 $< t_{LOW_OD}$ (オープンドレインの SCL クロックのロー周期)
 - $t_{LOW_OD\ min} = 200\text{ ns}$ 。10MHz を超える周波数ではこの制約が保証されます。この制約は、I3C バス/コントローラによって緩和できます。
4. I3CCLK カーネルクロックの周波数 $>$ SCL クロックの周波数の 2.5 倍
 - $F_{SCL\ max} = 12.9\text{ MHz}$ 。32.3MHz を超える周波数ではこの制約が保証されます。この制約は、I3C コントローラによって緩和できます。

APB クロックの要件

バスの SCL クロックの目的値に応じて、アプリケーションでは APB クロックの最小動作周波数を保証する必要があります。

$$APB \text{ クロック周期} < 3x (\text{SCL クロック周期}) - I3CCLK \text{ カーネルクロック周期}$$

$$\text{つまり、} F_{APB} > [F_{SCL} \times F_{I3CCLK} / (3x F_{I3CCLK} - F_{SCL})]$$

注 : この式は、APB の周波数が 5MHz の最小値になるまで簡略化できます。

49.7 I3C ペリフェラルの状態とプログラミング

49.7.1 I3C ペリフェラルの状態

I3C ペリフェラルは I3C バスコントローラまたは I3C ターゲットの役割を果たします。どの場合でも (図 652 および図 653 を参照)、ペリフェラルは次のいずれかの状態です。

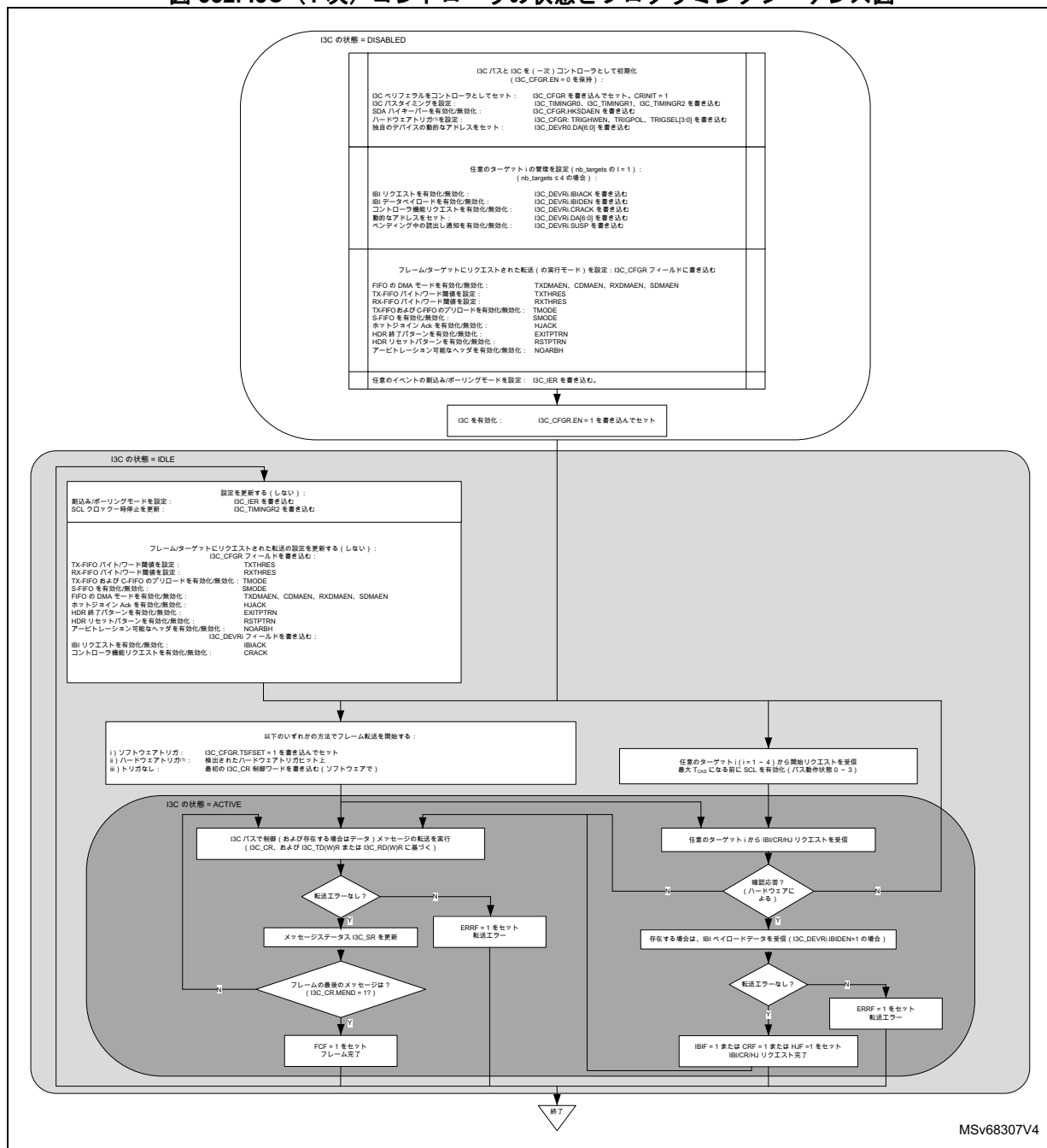
- 無効状態 :
 - I3C リセット後 (システムリセットまたは RCC からの I3C リセット)、ペリフェラルは無効状態になります。
 - ソフトウェアを I3C_CFGR レジスタで 1 ビット EN にセットしている場合、ペリフェラルは別の設定レジスタの値を考慮して、(有効および) アイドル状態に切り替わります。
- アイドル状態 :
 - 有効 (EN = 1) にすると、ペリフェラルは I3CCLK および SCL クロックドメインをアクティブにし、I3C バスでの通信が可能になります。
 - このソフトウェアでは I3C ペリフェラル設定を部分的に更新できます。詳細については、[転送用の設定更新 \(コントローラの場合\)](#) および [I3C ペリフェラルの設定の更新 \(ターゲットの場合\)](#) を参照してください。
 - ペリフェラルは以下のいずれかの場合に (有効および) アクティブ状態になります。
 - a) ソフトウェアが転送を開始したとき (コントローラの場合 : ソフトウェアがフレーム転送を開始したとき。ターゲットの場合 : ソフトウェアが IBI/CR/HJ リクエストを開始したとき)。
 - b) ハードウェアがバスの別の I3C デバイスからリクエストを受信したとき (コントローラの場合 : ターゲットからの開始リクエスト後、最大 T_{CAS} 時間。ターゲットの場合 : ブロードキャスト/直接 CCC またはプライベート読出し/書き込みを受信したとき)。
- アクティブ状態 :
 - ペリフェラルがバスで転送を実行します。
 - リクエストされた転送が完了すると、ソフトウェアは [I3C イベントレジスタ \(I3C_EVR\)](#) からイベントによって通知され、対応する割込みが [I3C 割込み有効レジスタ \(I3C_IER\)](#) によって有効になります。ペリフェラルはアイドル状態に切り替わりますが、引き続きバス上での通信が可能で、また (部分的に) 再設定できます。
 - a) コントローラの場合 : イベント/フラグが発生している場合は、フレーム完了 (FCF)、IBI/コントローラ機能/ホットジョイン・リクエスト完了 (IBIF/CRF/HJF)、または転送エラー (ERRF) などを意味する
 - b) ターゲットの場合 : イベント/フラグが発生している場合は、動的なアドレス割り当て完了 (DAUPDF)、IBI 完了 (IBIENDF)、コントローラ機能取得完了 (CRUPDF)、ブロードキャスト/直接 CCC 完了 (xxUPDF/RSTF/GETF/STAF)、プライベート読出し/書き込み完了 (FCF)、または転送エラー (ERRF) などを意味する

注 : ソフトウェアは、ペリフェラルを部分的にリセットして無効化できます (EN = 0 と書き込む) (SCL クロックドメイン内および I3CCLK カーネルクロックドメイン内のサブパート)。イベント、割込み、およびクロックリクエストの生成にも影響します。以前に書き込まれた APB レジスタの設定は保持されますが、変更されません。

49.7.2 I3C コントローラの状態とプログラミングシーケンス

図 652 に、(1 次) コントローラとして機能する I3C ペリフェラルのプログラミングシーケンス全体を示します。これには、このセクションで説明する状態遷移、主なサブタスク、および条件が含まれます。

図 652. I3C (1 次) コントローラの状態とプログラミングシーケンス図



1. この機能は実装によって使用できない場合があります。セクション 49.3.4 : I3C トリガを参照してください。

コントローラの初期化

コントローラが無効状態の場合は (I3C_CFGR レジスタで EN = 0)、ソフトウェアを以下のように初期化する必要があります。

- 次のフィールドを持つ **I3C 設定レジスタ (I3C_CFGR)** を設定します。
 - CRINIT = 1 : I3C バスコントローラ
 - HKSDAEN : SDA 有効/無効時のハイキーパー
- I3C バスタイミングを設定します。
 - a) **I3C タイミングレジスタ 0 (I3C_TIMINGR0)** :
 - 従来の I²C および I3C オープンドレイン/プッシュプルでの SCL クロックのハイ時間周期 (t_{DIG_H} 、 $t_{DIG_H_MIXED}$)
 - 従来の I²C および I3C オープンドレイン/プッシュプルでの SCL クロックのロー時間周期 (t_{DIG_L} 、 $t_{DIG_OD_L}$)
 - b) **I3C タイミングレジスタ 1 (I3C_TIMINGR1)** :
 - プッシュプルでの SDA ホールド時間 (t_{HD_PP})
 - バスフリー条件時間 (I3C t_{CAS} 、従来の I²C t_{BUF})
 - I3C 反復 START タイミング (t_{CASr} 、 t_{CBSr})
 - I3C STOP タイミング (t_{CBP})
 - SCL クロックローの最大ストール。ENTDAA CCC ($t_{STALLDAA}$)、または従来の I²C の ACK/NACK または I3C 転送のアドレスフェーズまたは書込みデータ転送のパリティビット、または従来の I²C 書込みの ACK/NACK データフェーズ、または I3C 読出し転送の遷移ビット、または従来の I²C 書込み (t_{STALL}) の ACK/NACK フェーズをストールする。コントローラとして使用される場合は、ペリフェラル自体が必要とするときに、SCL クロックローのストールを調整するために行われる
 - コントローラ機能のハンドオフ手順の $t_{NEWCRLock}$ (GETACCCR CCC 後)
 - c) **I3C タイミングレジスタ 2 (I3C_TIMINGR2)** :
 - SCL クロックローのストール時間。フェーズごとに個別に有効化または無効化でき、アドレス指定された I3C ターゲットまたは従来の I²C ターゲットでの SDA ハンドオフが必要な場合は、クロックローのストールを調整するために行われる
- 独自の動的なアドレスを設定します (**I3C 独自デバイス特性レジスタ (I3C_DEVR0)** の DA [6:0] フィールド)。
- 任意のデバイスのターゲット x の管理を設定します (**I3C デバイス x 特性レジスタ (I3C_DEVRx)** で $x = 1 \sim x \leq 4$)。
- フレーム転送またはターゲットがリクエストした転送の実行モードを設定します (次のフィールドを持つ **I3C 設定レジスタ (I3C_CFGR)**)。
 - TXDMAEN、CDMAEN、RXDMAEN、SDMAEN : TX-FIFO、C-FIFO、RX-FIFO、S-FIFO について、それぞれ DMA モードを有効化/無効化
 - TXTHRES、RXTHRES : それぞれ TX-FIFO および RX-FIFO バイト/ワード閾値
 - TMODE : 送信モード (TX-FIFO および C-FIFO の両方のプリロードの有効化/無効化)
 - SMODE : S-FIFO の有効化/無効化
 - EXITPTRN、RSTPTRN : 終了、リセット・パターンの有効化/無効化
 - HJACK : ホットジョイン確認応答の有効化/無効化
 - NOARBH : アービトレーション可能なヘッダの有効化/無効化
- 任意のイベントから割込み生成またはポーリングモードを設定します (**I3C 割込み有効レジスタ (I3C_IER)**)。

これにより、ソフトウェアは I3C ペリフェラルを有効化できます (EN = 1 にセット)。

注： ソフトウェアは I3C_CFGR を有効化する際にすべてのフィールドに一度に書き込むことができます。

コントローラによるフレーム転送の開始

コントローラが有効状態の場合 (I3C_CFGR レジスタで EN = 1)、ソフトウェアは次のいずれかの設定方法によってフレーム転送を開始できます。

1. ソフトウェアによるトリガ：書き込み時に I3C_CFGR レジスタで TSFSET = 1 をセット
 - これにより、ハードウェアは I3C_EVR レジスタで CFNFF = 1 のフラグを立て、最初の制御ワード I3C_CR の書き込みをリクエストします。
2. トリガなし：ソフトウェアによる最初の制御ワード I3C_CR の書き込み時

その後、フレームの開始方法にかかわらず、I3C ペリフェラルはアクティブ状態に切り替わります。制御ワードは I3C フレームの最後のメッセージではないため (I3C_CR レジスタで MEND = 0 の間)、また転送エラーがないため (I3C_EVR レジスタで ERRF = 1 の間)、ハードウェアは次の制御ワードのリクエストと、フレームの転送を続けます。

- a) C-FIFO が DMA モードで設定されていない場合 (I3C_CFGR レジスタで CDMAEN = 0)、ソフトウェアは I3C_EVR レジスタのフラグ CFNFF = 1 に続けて、または対応する割り込みが有効な場合 (I3C_IER レジスタで CFNFIE = 1 の場合) はそれに続けて、次の制御ワードを書き込みます。
- b) C-FIFO が DMA モードで設定されている場合 (CDMAEN = 1)、次の制御ワードは、アサートされた I3C DMA リクエスト (i3c_tc_dma) に従い、割り当てられた DMA チャンネルによって自動的にプッシュされ、書き込まれます。

ターゲットによる転送の開始または受信

コントローラが有効状態の場合 (I3C_CFGR レジスタで EN = 1) で、同時にコントローラが転送を開始する可能性がある場合、ターゲットは開始リクエストを発行して転送を開始できます (SDA をローに駆動)。これには、コントローラによる I3C_DEVR0 レジスタ経由のホットジョイン・リクエスト、IBI リクエスト、またはコントローラ機能リクエストの許可が必要です。

この場合、コントローラ・ソフトウェアにフレームの転送を開始する意図がなくても、ハードウェアはターゲットからホットジョイン/帯域内割り込み/コントローラ機能のリクエストを受信するために、アクティブ状態に切り替わります (SCL クロックを、バス動作状態 0、1、2、または 3 に応じてそれぞれ 1 μ s、100 μ s、2 ms、または 50 ms に定義された最大 t_{CAS} 時間より前に有効化します)。

ターゲットによる I3C バス転送の実行と、コントローラとしての関連プログラミングの詳細については、[セクション 49.9](#) の該当する図を参照してください。

- [図 664：IBI 転送 \(コントローラ/ターゲットの場合\)](#)
- [図 665：ホットジョイン・リクエスト転送 \(コントローラ/ターゲットの場合\)](#)
- [図 666：コントローラ機能リクエスト転送 \(コントローラ/ターゲットの場合\)](#)

(コントローラによる) フレーム転送の実行

有効な場合、コントローラはバス上でフレーム転送を実行します。これは、最後のメッセージ (I3C_EVR レジスタで FCF = 1) が完了するか、転送エラー (I3C_EVR レジスタで ERRF = 1) が発生し、対応する割り込みが発生するまで実行されます。これは、ソフトウェアによって明示的に書き込まれた、または割り当てられた DMA チャンネルによってプッシュされた、I3C_CR および I3C_TD(W)R レジスタに基づきます。また、ソフトウェアまたは割り当てられた DMA チャンネルによって明示的に読み出された I3C_RD(W)R に基づきます。その後、I3C コントローラは元のアイドル状態に切り替わります。

コントローラによる I3C バス転送の実行と、コントローラとしての関連プログラミングの詳細については、[セクション 49.9](#) の図を参照してください。

- [図 654](#) : I3C CCC メッセージ (コントローラの場合)
- [図 655](#) : I3C ブロードキャスト ENTDAACCCC (コントローラの場合)
- [図 656](#) : I3C ブロードキャスト、直接読出しおよび直接書き込み RSTACTCCC (コントローラの場合)
- [図 661](#) : I3C プライベート読出し/書き込みメッセージ (コントローラの場合)
- [図 663](#) : 従来の I2C 読出し/書き込みメッセージ (コントローラの場合)

[図 652](#) には、FIFO (TX-FIFO、RX-FIFO、C-FIFO、および S-FIFO) の管理は含まれません。詳細については、[セクション 49.10](#) を参照してください。

各完了メッセージに転送エラーがない場合、ハードウェアは [I3C ステータスレジスタ \(I3C_SR\)](#) を更新することで I3C バスでの交換転送を報告します。これを読み出せるかどうかは、S-FIFO が無効 (I3C_CFGR レジスタで SMODE = 0) な場合、ソフトウェアによって異なります。

- 直接 CCC 読出しまたはプライベート読出しの転送時に、最後のメッセージの完了 (I3C_EVR レジスタで FCF = 1) または転送エラー (I3C_EVR レジスタで ERRF = 1) と対応する割込み (有効な場合) に加え、ステータスレジスタ I3C_SR で S-FIFO が無効であれば (I3C_CFGR レジスタで SMODE = 0)、ターゲットによって読出し転送が途中で終了した場合に、I3C_EVR レジスタで RXTGTENDF = 1 にすることで、また対応する割込みが有効な場合はそれによって、ソフトウェアに通知されます。これにより、ソフトウェアは I3C_SR を読み出して、実行した転送の詳細を得ることができます。

あるいは、S-FIFO が有効 (I3C_CFGR レジスタで SMODE = 1) な場合は、各実行メッセージについてステータスレジスタ I3C_SR を読み出す必要があります。これは、ソフトウェアが直接読み出すか、(I3C_EVR レジスタの SFNEF = 1、および有効な場合は対応する割込みで通知)、DMA 経由 (I3C_CFGR レジスタで SDMAEN = 1 の場合) で、読出しがターゲットによって途中で終了されているかどうかにかかわらず、行われます。フレームの完了 (I3C_EVR レジスタで FCF = 1) は、最後のメッセージのステータスを読み出した後にのみ行われます (S-FIFO は空)。詳細については、[セクション 49.10.4](#) を参照してください。

転送用の設定更新 (コントローラの場合)

アイドル状態では、ソフトウェアは次の転送の前に I3C ペリフェラルの設定を更新できました。

- SCL クロックのストールを [I3C タイミングレジスタ 2 \(I3C_TIMINGR2\)](#) によって変更してください。
- 割込み/ポーリング・モード・ポリシーを [I3C 割込み有効レジスタ \(I3C_IER\)](#) によって変更してください。
- [I3C 設定レジスタ \(I3C_CFGR\)](#) の以下のフィールドを変更してください。
 - TXTHRES、RXTHRES
 - TMODE、SMODE
 - TXDMAEN、CDMAEN、RXDMAEN、SDMAEN
 - EXITPTRN、RSTPTRN
 - NOARBH

- ソフトウェアおよび/または DMA によって次に実行されるフレーム転送の制御ワード、ステータスワード、読出し/書込みデータを変更/準備してください。
 - I3C メッセージ制御レジスタ (I3C_CR)、I3C メッセージ制御レジスタ [オルタネート] (I3C_CR) を更新。
 - I3C ステータスレジスタ (I3C_SR)
 - I3C 送信データバイトレジスタ (I3C_TDR)、I3C 送信データワードレジスタ (I3C_TDWR) を更新。
 - I3C 受信データバイトレジスタ (I3C_RDR)、I3C 受信データワードレジスタ (I3C_RDWR) を更新。
- 通常、ブロードキャスト/直接 DISEC/ENEC CCC の発行および完了後：
 - ホットジョイン確認応答ポリシーを I3C_CFGR レジスタのビット HJACK によって変更してください。
 - 任意のターゲット x への IBI/CR 確認応答ポリシーを I3C デバイス x 特性レジスタ (I3C_DEVRx) によって変更してください。

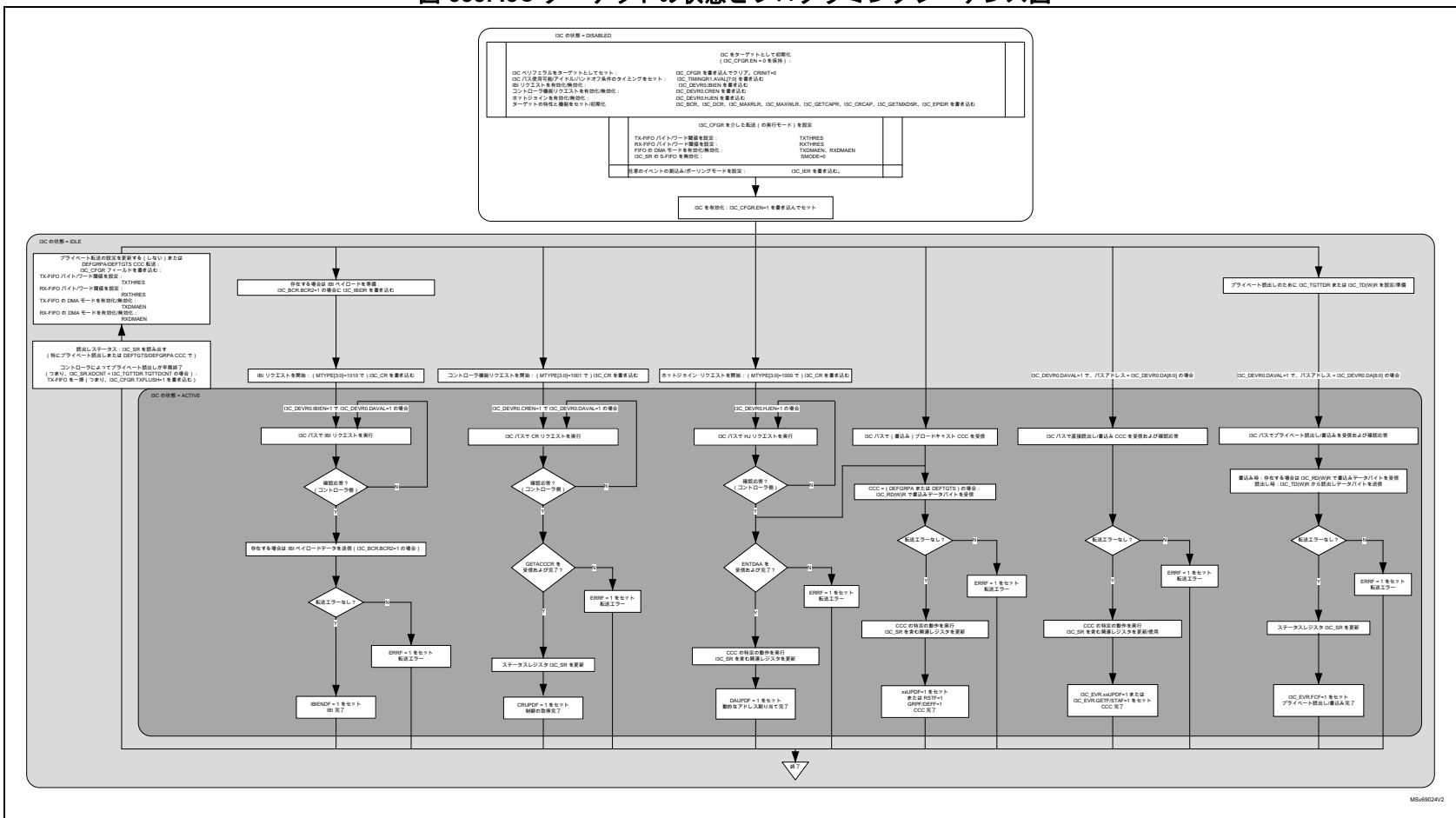
レジスタの使用方法和 I3C ペリフェラルのコントローラとしての役割については、[セクション 49.8.1](#) に示しています。

コントローラとして機能する場合のスタティック/ダイナミック・レジスタ・フィールドの使用方法については、[表 522](#) に示しています。

49.7.3 I3C ターゲットの状態とプログラミングシーケンス

[図 653](#) に、ターゲットとして機能するペリフェラルのプログラミングシーケンス全体を示します。これには、このセクションで説明する状態遷移、主なサブタスク、および条件が含まれます。

図 653. I3C ターゲットの状態とプログラミングシーケンス



ターゲットの初期化

ターゲットが無効状態の場合は (I3C_CFGR レジスタで EN = 0)、ソフトウェアを以下のように初期化する必要があります。

- I3C ペリフェラルをターゲットとしてセットします。I3C_CFGR レジスタで CRINIT = 0 を書き込み、クリアしてください。
- I3C バスタイミングを **I3C タイミングレジスタ 1 (I3C_TIMINGR1)** によってセットします。以下について、AVAL[7:0] を書き込んでください。
 - IBI またはコントローラ機能のリクエストのバス使用可能条件時間 (t_{AVAL})
 - ホットジョイン・リクエストのバスアイドル条件時間 (t_{IDLE})
 - コントローラ機能のハンドオフ手順の $t_{NEWCRLOCK}$ (GETACCCR CCC 後)
- ターゲットによるリクエストを設定します。以下で **I3C 独自デバイス特性レジスタ (I3C_DEVR0)** を書き込んでください。
 - IBIEN : 帯域内割込み (IBI と呼ばれる) リクエストの有効化/無効化
 - CREN : コントローラ機能リクエストの有効化/無効化
 - HJEN : ホットジョイン・リクエストの有効化/無効化
- ターゲットの特性と機能を初期化します。以下を書き込んでください。
 - **I3C バス特性レジスタ (I3C_BCR)**
 - **I3C デバイス特性レジスタ (I3C_DCR)**
 - **I3C 最大読出し長レジスタ (I3C_MAXRLR)**
 - **I3C 最大書込み長レジスタ (I3C_MAXWLR)**
 - **I3C 取得のケーパシタリティ・レジスタ (I3C_GETCAPR)**
 - **I3C コントローラ機能のケーパシタリティ・レジスタ (I3C_CRCAPR)**
 - **I3C 最大データ速度の取得レジスタ (I3C_GETMXDSR)**
 - **I3C 拡張された専用 ID レジスタ (I3C_EPIDR)**
- 転送の実行モードを設定します。(次のフィールドを持つ **I3C 設定レジスタ (I3C_CFGR)**)。
 - TXDMAEN, RXDMAEN : TX-FIFO および RX-FIFO について、それぞれ DMA モードを有効化/無効化
 - TXTHRES, RXTHRES : それぞれ TX-FIFO および RX-FIFO バイト/ワード閾値
 - S-FIFO の無効化 : SMODE = 0 (デフォルト/リセット値)
- 任意のイベントから割込み生成またはポーリングモードを設定します (**I3C 割込み有効レジスタ (I3C_IER)**)。

これにより、ソフトウェアは I3C ペリフェラルを有効化できます (書き込んで EN = 1 にセット)。

注 : ソフトウェアは I3C_CFGR レジスタを有効化する際にすべてのフィールドに一度に書き込むことができます。

コントローラからの (ブロードキャスト CCC、直接読出し/書込み CCC、またはプライベート読出し/書込み) メッセージの受信

ターゲットがアイドル状態の場合 (I3C_CFGR レジスタで EN = 1)、ターゲットはコントローラから I3C バスで通信メッセージを受信し、アクティブ状態に切り替える準備ができています。

通常、可能な場合はオプションでブロードキャスト ENEC/DISEC CCC を受信した後に、まずアクティブなターゲットがブロードキャスト ENTDAACCCC を受信しており、その後動的なアドレスに割り当てられます。I3C_EVR レジスタのイベント DAUPF が 1 まで引き上げられ、関連する割込みが生成され (有効な場合)、ターゲットはアイドル状態に戻ります。

その後、アイドル状態のターゲットは、その他のブロードキャスト CCC メッセージ、または直接読出し/書込み CCC、またはプライベート読出し/書込みメッセージをコントローラから受信できるようになります。

コントローラによる I3C バス転送の実行と、更新された I3C レジスタおよびフィールドを含むターゲットとしての関連プログラミングの詳細については、[セクション 49.9](#) の図を参照してください。

- [図 657](#) : I3C CCC メッセージ (ターゲットの場合)
- [図 658](#) : I3C ブロードキャスト ENTDAACCC (ターゲットの場合)
- [図 659](#) : I3C ブロードキャスト DEFTGTS CCC (ターゲットの場合)
- [図 660](#) : I3C ブロードキャスト DEFGRPA CCC (ターゲットの場合)
- [図 662](#) : I3C プライベート読出し/書込みメッセージ (ターゲットの場合)

[図 653](#) には、FIFO 管理 (TX-FIFO、RX-FIFO) は含まれません。詳細については、[セクション 49.10](#) を参照してください。

メッセージステータスレジスタの読出し

受信し、完了した各メッセージに転送エラーがない場合、ハードウェアは [I3C ステータスレジスタ \(I3C_SR\)](#) を更新することで I3C バスでの交換転送を報告します。これは、[I3C イベントレジスタ \(I3C_EVR\)](#) の対応するフラグ、または [I3C 割込み有効レジスタ \(I3C_IER\)](#) で有効化されている対応する割込みによって通知された後に、ソフトウェアが読み出すことができます。

[I3C ステータスレジスタ \(I3C_SR\)](#) は、以下のメッセージの後にソフトウェアで読み出す必要があります。

- プライベート読出し : 交換したデータバイト数を取得します。これは、コントローラが予想よりも早くターゲットによる転送を終了している場合があるためです。(I3C_SR レジスタの XDCNT[15:0] が I3C_TGTTDR レジスタの TGTTDCNT[15:0] より低い場合)。その場合、ソフトウェアは TX-FIFO を一掃する必要があります (I3C_CFGR レジスタで TXFLUSH = 1 を書き込みます)。
- DEFTGTS CCC または DEFGRPA CCC : RX-FIFO の受信データバイト数を取得します。

(ターゲットによる) 転送の開始

ターゲットが最初に無効状態からアイドル状態に移行した場合 (ソフトウェアが I3C_CFGR レジスタに EN = 1 を書き込んだ場合)、同時にコントローラからブロードキャスト CCC を受信できるようにするために、ソフトウェアはホットジョイン・リクエストを開始できます (ソフトウェアが I3C_CR レジスタに MTYPE[3:0] = 1000 を書き込んだ場合)。これにより、可能であれば (I3C_DEVR0 レジスタで HJEN = 1)、次の ENTDAACCC に参加できるようになります。

より一般的に、場合によってはコントローラによるフレーム転送の送信と同時に動的なアドレスが割り当てられると (I3C_EVR レジスタで DAUPF = 1)、ソフトウェアはコントローラへの IBI (帯域内割込みリクエスト)、または I3C_CR レジスタへの関連する制御ワードを書き込むことで、コントローラ機能リクエストを開始できます。

ターゲットによる I3C バス転送の実行と、ターゲットとしての関連プログラミングの詳細については、[セクション 49.9](#) の図を参照してください。

- [図 664](#) : IBI 転送 (コントローラ/ターゲットの場合)
- [図 665](#) : ホットジョイン・リクエスト転送 (コントローラ/ターゲットの場合)
- [図 666](#) : コントローラ機能リクエスト転送 (コントローラ/ターゲットの場合)

I3C ペリフェラルの設定の更新（ターゲットの場合）

アイドル状態では、ソフトウェアは次の転送の前に I3C ターゲットの設定を更新できました。

- 割込み/ポーリング・モード・ポリシーを [I3C 割込み有効レジスタ \(I3C_IER\)](#) によって変更してください。
- [I3C 設定レジスタ \(I3C_CFGR\)](#) の以下のフィールドを変更してください。
 - TXTHRES、RXTHRES
 - TXDMAEN、RXDMAEN
- IBI 転送を開始する前に (MTYPE[3:0] = 1010 で [I3C メッセージ制御レジスタ \[オルタネート\] \(I3C_CR\)](#) を書き込む)、ペイロード (I3C_BCR レジスタで BCR2 = 1 の場合) がある場合は、[I3C IBI ペイロード・データ・レジスタ \(I3C_IBIDR\)](#) を変更/準備してください。
- プライベート読出し、またはコントローラからの直接 CCC 読出しを (GETSTATUS CCC から) 受信する前に、定義された送信するデータバイト数とともにプリロードする TX-FIFO を無効または有効にするには、[I3C ターゲット送信設定レジスタ \(I3C_TGTTDR\)](#) を変更/準備してください。

レジスタの使用法と I3C ペリフェラルのターゲットとしての役割については、[セクション 49.8.1](#) に示しています。

ターゲットとして機能する場合のスタティック/ダイナミック・レジスタ・フィールドの使用法については、[表 523](#) に示しています。

49.8 I3C レジスタおよびプログラミング

49.8.1 I3C レジスタのセット（コントローラ/ターゲットの場合）

[表 521](#) にレジスタおよびその使用方法、ならびに I3C ペリフェラルの役割を示します。

表 521. I3C レジスタの使用法

レジスタ	コントローラとして使用	ターゲットとして使用
I3C_CR	X	X
I3C_CFGR	X	X
I3C_RDR	X	X
I3C_RDWR	X	X
I3C_TDR	X	X
I3C_TDWR	X	X
I3C_IBIDR	X	X
I3C_TGTTDR	-	X
I3C_SR	X	X
I3C_SER	X	X
I3C_RMR	X	X
I3C_EVR	X	X
I3C_IER	X	X
I3C_CEV	X	X
I3C_DEVR0	X	X

表 521. I3C レジスタの使用方法（続き）

レジスタ	コントローラとして使用	ターゲットとして使用
I3C_DEVRx x = 1 4	X	-
I3C_MAXRLR	-	X
I3C_MAXWLR	-	X
I3C_TIMINGR0	X	-
I3C_TIMINGR1	X	X
I3C_TIMINGR2	X	-
I3C_BCR	-	X
I3C_DCR	-	X
I3C_GETCAPR	-	X
I3C_CRCAPR	-	X
I3C_GETMXDSR	-	X
I3C_EPIDR	-	X

49.8.2 コントローラとしてのI3C レジスタとフィールドの使用方法とペリフェラルの状態

I3C ペリフェラルがコントローラとして機能する場合、[表 522](#) にレジスタとその使用方法、ならびにコントローラの状態を示します（無効、アイドル、およびアクティブ）。

表 522. I3C レジスタ/フィールドの使用方法とコントローラの状態

レジスタ	コントローラとして使用	無効状態でのみ書込み可能	アイドル状態では通常書込み/読出し	アイドル状態またはアクティブ状態では通常書込み/読出し
I3C_CR	X	-	-	X
I3C_CFGR	X	CRINIT HKSDAEN EN ⁽¹⁾	書込み：{CRINIT, HKSDAEN} 以外の 以下の使用済みフィールド CDMAEN SDMAEN TXDMAEN RXDMAEN TMODE SMODE TXTHRES RXTHRES HJACK EXITPTRN RSTPTRN NOARBH ⁽²⁾ CFLUSH SFLUSH TXFLUSH RXFLUSH TSFSET	-

表 522. I3C レジスタ/フィールドの使用方法和コントローラの状態 (続き)

レジスタ	コントローラ として使用	無効状態でのみ 書き込み可能	アイドル状態では 通常書き込み/読出し	アイドル状態またはアクティブ 状態では通常書き込み/読出し
I3C_RDR	X	-	-	読出し
I3C_RDWR	X	-	-	読出し
I3C_TDR	X	-	-	書き込み
I3C_TDWR	X	-	-	書き込み
I3C_IBIDR	X	-	-	読出し
I3C_TGTTDR			-	
I3C_SR	X	-	-	読出し
I3C_SER	X	-	読出し	-
I3C_RMR	X	-	-	RADD[6:0] IBIRD CNT[2:0] の 読出し
I3C_EVR	X	-	-	読出し (コントローラ機能の フィールド) : HJF CRF IBIF FCF ERRF RXTGTENDF RXFNEF TXFNFF SFNEF CFNFF RXLASTF TXLASTF TXFEF CFEF
I3C_IER	X	xIE の書き込み (x = HJ、CR、IBI、FC、ERR、RXTGTEND、RXFNE、TXFNF、SFNE、 CFNF ⁽²⁾)		
I3C_CEV	X	-	-	書き込み (コントローラ機能の フィールド、I3C_EVR を参照)
I3C_DEVR0	X	DA[6:0] の書き込み ⁽²⁾		-
I3C_DEVRx x = 1 4	X	SUSP、IBIDEN、IBIACK、CRACK、DA[6:0] の書き込み ⁽²⁾		-
I3C_MAXRLR			-	
I3C_MAXWLR			-	
I3C_TIMINGR0	X	X	-	-
I3C_TIMINGR1	X	X	-	-
I3C_TIMINGR2	X	書き込み ⁽²⁾		
I3C_BCR			-	
I3C_DCR			-	
I3C_GETCAPR			-	

表 522. I3C レジスタ/フィールドの使用方法和コントローラの状態（続き）

レジスタ	コントローラ として使用	無効状態でのみ 書き込み可能	アイドル状態では 通常書き込み/読出し	アイドル状態またはアクティブ 状態では通常書き込み/読出し
I3C_CRCAPR			-	
I3C_GETMXDSR			-	
I3C_EPIDR			-	

1. I3C_CFGR レジスタのビット EN が書き込まれ、無効状態にセットされます（同じビットが 0 の場合）。このフィールドは、アイドル状態でも書き込みおよびデアサート可能です。
2. これらのフィールドは通常、バス設定中に無効状態で書き込まれ、初期化されます。EN = 0 の場合は書き込み保護されていません。他の状態の場合は書き込みも更新もできます。

49.8.3 I3C レジスタとフィールドの使用方法和ペリフェラルの状態（ターゲットの場合）

ペリフェラルがターゲットとして機能する場合、表 523 にレジスタとその使用方法、ならびに I3C ターゲットの状態を示します（無効、アイドル、およびアクティブ）。

表 523. I3C レジスタ/フィールドの使用方法和ターゲットの状態

レジスタ	ターゲット として使用	無効状態でのみ 書き込み可能	アイドル状態では 通常書き込み/読出し	アイドル状態またはアクティブ 状態では通常書き込み/読出し
I3C_CR	X	-	MTYPE[3:0] DCNT[2:0]	-
I3C_CFGR	X	CRINIT EN ⁽¹⁾	書き込み：CRINIT 以外の以下の 使用済みフィールド： {TXDMAEN RXDMAEN TXTHRES RXTHRES} ⁽²⁾ TXFLUSH RXFLUSH	-
I3C_RDR	X	-	-	読出し
I3C_RDWR	X	-	-	読出し
I3C_TDR	X	-	-	書き込み
I3C_TDWR	X	-	-	書き込み
I3C_IBIDR	X	-	書き込み	-
I3C_TGTTDR	X	-	書き込み	-
I3C_SR	X	-	-	読出し DIR、XDCNT[15:0]
I3C_SER	X	-	DOVR、STALL、PERR、 CODERR[3:0] の読出し	-
I3C_RMR	X	-	-	RCODE[7:0] の読出し

表 523. I3C レジスタ/フィールドの使用方法とターゲットの状態 (続き)

レジスタ	ターゲットとして使用	無効状態でのみ書き込み可能	アイドル状態では通常書き込み/読出し	アイドル状態またはアクティブ状態では通常書き込み/読出し
I3C_EVR	X	-	-	読出し (ターゲット機能のフィールド) : GRPF DEFF INTUPDF ASUPDF RSTF MRLUPDF MWLUPDF DAUPDF STAF GETF WKPF CRUPDF IBIENDF ERRF FCF RXFNEF TXFNFF TXLASTF TXFEF
I3C_IER	X	xIE の書き込み (x = GRP、DEF、INTUPD、ASUPD、RST、MRLUPD、MWLUPD、DAUPD、STA、GET、WKP、CRUPD、IBIEND、ERR、FC、RXFNE、TXFNFF) ⁽²⁾		
I3C_CEV	X	-	-	書き込み (ターゲット機能のフィールド、I3C_EVR を参照)
I3C_DEVR0	X	HJEN CREN IBIEN	RSTVAL、RSTACT[1:0]、および AS[1:0] の読出し	-
I3C_DEVRx x = 1 4		-		
I3C_MAXRLR	X	X	-	-
I3C_MAXWLR	X	X	-	-
I3C_TIMINGR0		-		
I3C_TIMINGR1	X	AVAL[7:0]	-	-
I3C_TIMINGR2		-		
I3C_BCR	X	X	-	-
I3C_DCR	X	X	-	-
I3C_GETCAPR	X	X	-	-
I3C_CRCAPR	X	X	-	-

表 523. I3C レジスタ/フィールドの使用方法とターゲットの状態（続き）

レジスタ	ターゲットとして使用	無効状態でのみ書き込み可能	アイドル状態では通常書き込み/読出し	アイドル状態またはアクティブ状態では通常書き込み/読出し
I3C_GETMXDSR	X	X	-	-
I3C_EPIDR	X	X	-	-

1. I3C_CFGR レジスタのビット EN が書き込まれ、無効状態にセットされます（同じビットが 0 の場合）。このフィールドは、アイドル状態でも書き込みおよびデアサート可能です。
2. これらのフィールドは通常、I3C バス設定中に無効状態で書き込まれ、初期化されます。EN = 0 の場合は、書き込み保護されていません。

49.9 I3C バス転送およびプログラミング

49.9.1 I3C コマンド・セット（CCC）（コントローラ/ターゲットの場合）

サポートされている I3C コマンド・セットのリスト（例：CCC（コモン・コマンド・コード）のリスト）と、コントローラまたはターゲットとして機能するペリフェラルによる処理方法の概要を表 524 に示します。

表 524. サポートされている I3C CCC のリスト (コントローラ/ターゲットの場合)

CCC 名	CCC 値	読出し／書込み	バイト定義のあり/なし サブコマンドバイトのあり/なし	オプション・データ・バイトのあり/なし	コントローラ/ターゲット	ターゲットとして使用、発生している I3C_EVR イベント	ターゲットの場合：具体的な操作	
ブロードキャスト CCC								
ENEC	0x00	書込み	定義/ サブコマンド バイトなし	1 データバイトあり (ターゲット・イベント・バイトの有効化)	X	X、INTUPDF	I3C_DEVR0 の更新および有効化：HJEN、CREN、IBIEN（存在する場合）	
DISEC	0x01			1 データバイトあり (ターゲット・イベント・バイトの無効化)	X	X、INTUPDF	I3C_DEVR0 の更新および無効化：HJEN、CREN、IBIEN（存在する場合）	
ENTASx x = 0 3	0x02 0x05			データバイトなし	X	X、ASUPDF	I3C_DEVR0.AS[1:0] の更新	
RSTDAA	0x06			-	X	X、DAUPDF	I3C_DEVR0.DAVAL = 0 のクリア	
ENTDAA	0x07			-	X	X、DAUPDF	I3C_DEVR0: DA[6:0] の更新と DAVAL = 1 のセット	
DEFTGTS	0x08			[1+ 4x (1+ number_of_targets)] x データバイトあり	X	X、DEFF	I3C_RDR/ I3C_RDWR の更新。 図 659を参照してください。	
SETMWL	0x09			2 データバイトあり	X	X、MWLUPDF	I3C_MAXWLR の更新	
SETMRL	0x0A			2 または 3 データ バイトあり	X	X、MRLUPDF	I3C_MAXRLR の更新	
ENTTM	0x0B			1 データバイトあり	X	-		
SETXTIME	0x28		サブコマンド バイトあり	1 データバイト以上 あり/なし	X			
SETAASA	0x29		定義/ サブコマンド バイトなし	データバイトなし	X			
RSTACT	0x2A		定義バイトあり (0x00、0x01、 または 0x02)	複数データバイト あり	X	X、リセット・ パターン 検出後 RSTF	I3C_DEVR0: RSTACT[1:0] の更新と RSTVAL = 1 のセット	
DEFGRPA	0x2B		定義/ サブコマンド バイトなし		X	X、GRPF	I3C_RDR/ RDWR の更新。 図 660を参照してください。	
RSTGRPA	0x2C		データバイトなし		X	-	-	

表 524. サポートされている I3C CCC のリスト (コントローラ/ターゲットの場合) (続き)

CCC 名	CCC 値	読出し/ 書き込み	バイト定義の あり/なし サブコマンド バイトの あり/なし	オプション・データ ・バイトのあり/なし	コントローラ/ターゲット	ターゲット として使用、 発生している I3C_EVR イベント	ターゲットの場合： 具体的な操作
直接 CCC							ACK の場合の操作 (I3C ターゲットアドレス = I3C_DEVR0.DA[6:0] および I3C_DEVR0.DAVAL = 1 の場合) (それ以外の場合は NACK)
ENEC	0x80	書き込み	定義/ サブコマンド バイトなし	1 データバイトあり (ターゲット・イベン ト・バイトの有効化)	X	X、INTUPDF	I3C_DEVR0 の更新および有効 化 : HJEN、CREN、IBIEN (存在 する場合)
DISEC	0x81			1 データバイトあり (ターゲット・イベン ト・バイトの無効化)	X	X、INTUPDF	I3C_DEVR0 の更新および無効 化 : HJEN、CREN、IBIEN (存在 する場合)
ENTASx x = 0 3	0x82.. .0x85			データバイトなし	X	X、ASUPDF	I3C_DEVR0.AS[1:0] の更新
SETDASA	0x87			データバイトなし	X	-	-
SETNEWDA	0x88			1 データバイトあり	X	X、DAUPDF	更新 I3C_DEVR0 : DA[6:0] (および DAVAL = 1 のセット)
SETMWL	0x89			2 データバイトあり	X	X、 MWLUPDF	I3C_MAXWLR の更新
SETMRL	0x8A			2 または 3 データ バイトあり	X	X、MRLUPDF	I3C_MAXRLR の更新
GETMWL	0x8B	読出し	定義/ サブコマンド バイトなし	2 データバイトあり	X	X、GETF	I3C_MAXWLR[15:0] からデータバ イトを返す。 セクション 49.16.19 を参照してください。
GETMRL	0x8C			2 または 3 データ バイトあり	X	X、GETF	I3C_MAXRLR[15:0] からデータバ イトを返す。I3C_BCR.BCR2 = 1 の場合は I3C_MAXRLR.IBIP[2:0] から 3 番目のバイトを返す。 セク ション 49.16.18 を参照してくだ さい。

表 524. サポートされている I3C CCC のリスト（コントローラ/ターゲットの場合）（続き）

CCC 名	CCC 値	読出し／ 書き込み	バイト定義の あり/なし サブコマンド バイトの あり/なし	オプション・データ ・バイトのあり/なし	コントローラ ハブ	ターゲット として使用、 発生している I3C_EVR イベント	ターゲットの場合： 具体的な操作
GETPID	0x8D	読出し	定義/ サブコマンド バイトなし	6 データバイトあり	X	X、GETF	I3C_EPIDR からデータバイトを返す。 セクション 49.16.28 を参照してください。
GETBCR	0x8E			1 データバイトあり	X	X、GETF	I3C_BCR[7:0] からデータバイトを返す。 セクション 49.16.23 を参照してください。
GETDCR	0x8F				X	X、GETF	I3C_DCR[7:0] を返す。 セクション 49.16.24 を参照してください。
GETSTATUS	0x90		定義バイトあり、またはなし (TGTSTAT、PRECR)	2 データバイトあり (PRECR のフォーマット 1 またはフォーマット 2)	X	X、フォーマット 1 の場合 STAF X、フォーマット 2 の場合 GETF	セクション 49.9.9 に詳しく記載されているとおりに 2 データバイトを返す。
GETACCCR	0x91		定義/ サブコマンド バイトなし	1 データバイトあり	X	X、CRUPDF	I3C_DEVR0.DA[6:0] からパリティビット付きでデータバイトを返す。
GETMXDS	0x94		定義バイトあり、またはなし (WRRDTURN、CRHDLY)	2 データバイト (フォーマット 1) または 5 データ バイト (WRRDTURN の フォーマット 2 またはフォーマット 3) または 1 データ バイト (CRHDLY の フォーマット 3) あり	X	X、GETF	I3C_GETMXDSR からデータバイトを返す。 セクション 49.16.27 を参照してください。
GETCAPS	0x95	書き込み	定義バイトあり、またはなし (TGTSTAT、CRCAPS)	3 データバイト (TGTSTAT の フォーマット 1 またはフォーマット 2) または 2 データ バイト (CRCAPS のフォーマット 2) あり	X	X、GETF	I3C_GETCAPR から 3 GETCAPx データバイトを返す (セクション 49.16.25 を参照)、または I3C_CRCAPR から 2 CRCAPx データバイトを返す (セクション 49.16.26 を参照)。
D2DXFER	0x97		定義バイトあり	定義バイトあり	X	-	-
SETXTIME	0x98		サブコマンド バイトあり	サブコマンドバイト あり	X		
GETXTIME	0x99	読出し	定義/ サブコマンド バイトなし	定義/サブコマンド バイトなし	X		

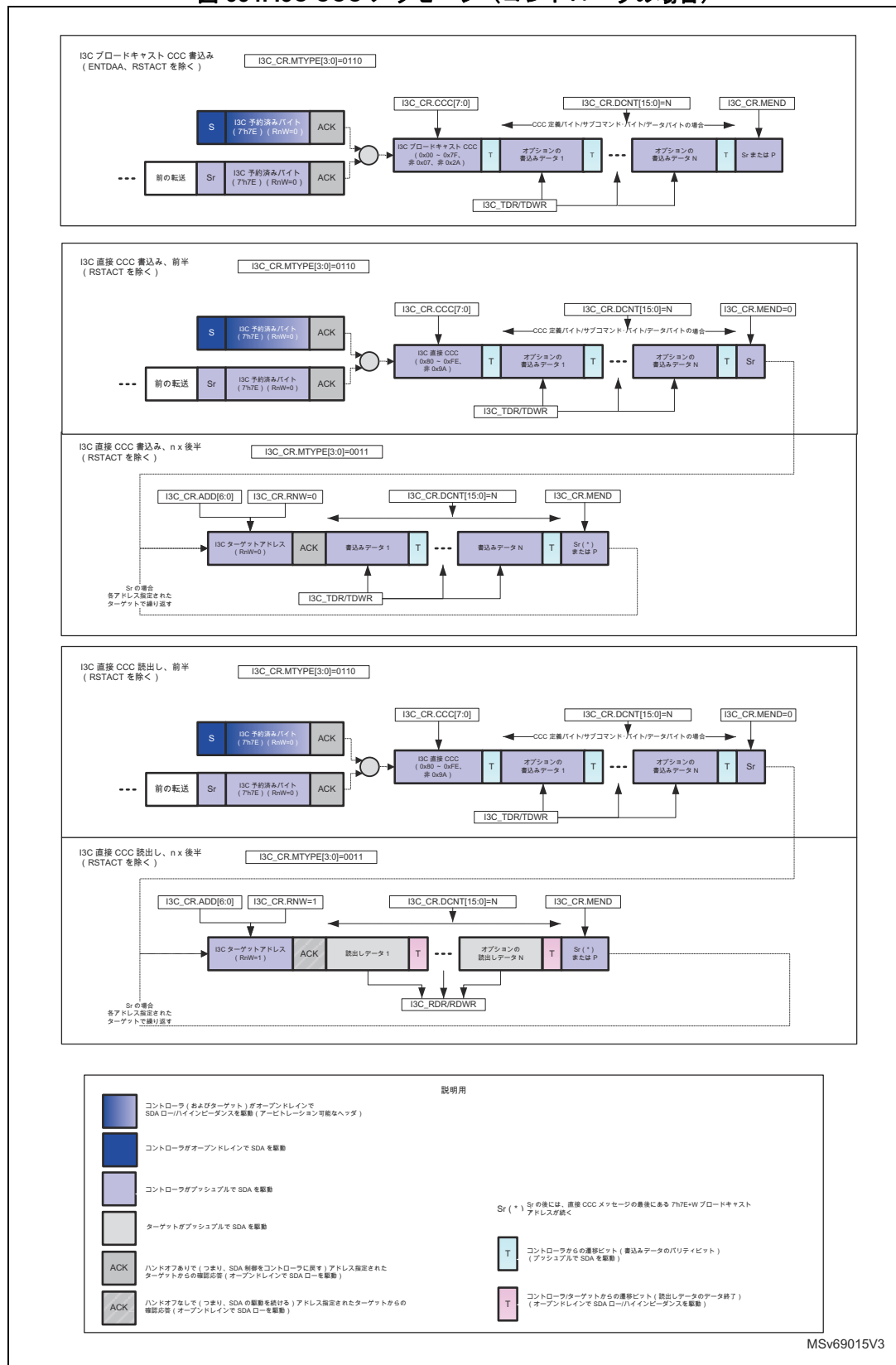
表 524. サポートされている I3C CCC のリスト（コントローラ/ターゲットの場合）（続き）

CCC 名	CCC 値	読み出し／書き込み	バイト定義のあり/なし サブコマンド バイトのあり/なし	オプション・データ ・バイトのあり/なし	コントローラ/ターゲット	ターゲットとして使用、発生している I3C_EVR イベント	ターゲットの場合：具体的な操作
RSTACT	0x9A	読み出し／書き込み	定義バイトあり (0x00、0x01、または 0x02)	定義バイトあり (0x00、0x01、または 0x02)	X	X、リセット・パターン検出の場合 RSTF	読み出し：I3C_DEVR0 レジスタの RSTACT[1:0] からデータバイトを返す。 書き込み：I3C_DEVR0: RSTACT[1:0] の更新と RSTVAL = 1 のセット
SETGRPA	0x9B	書き込み	定義/サブコマンドバイトなし	定義/サブコマンドバイトなし	X	-	-
RSTGRPA	0x9C				X		

49.9.2 I3C ブロードキャスト/直接 CCC 転送（ENTDAA、RSTACT を除く）（コントローラの場合）

図 654 に、I3C バスで通信され、またコントローラとして機能している場合にプログラムされたとおりの、I3C ブロードキャスト CCC 書き込み転送（ENTDAA、RSTACT を除く）、ならびに直接 CCC 読み出し/書き込み転送を示します。

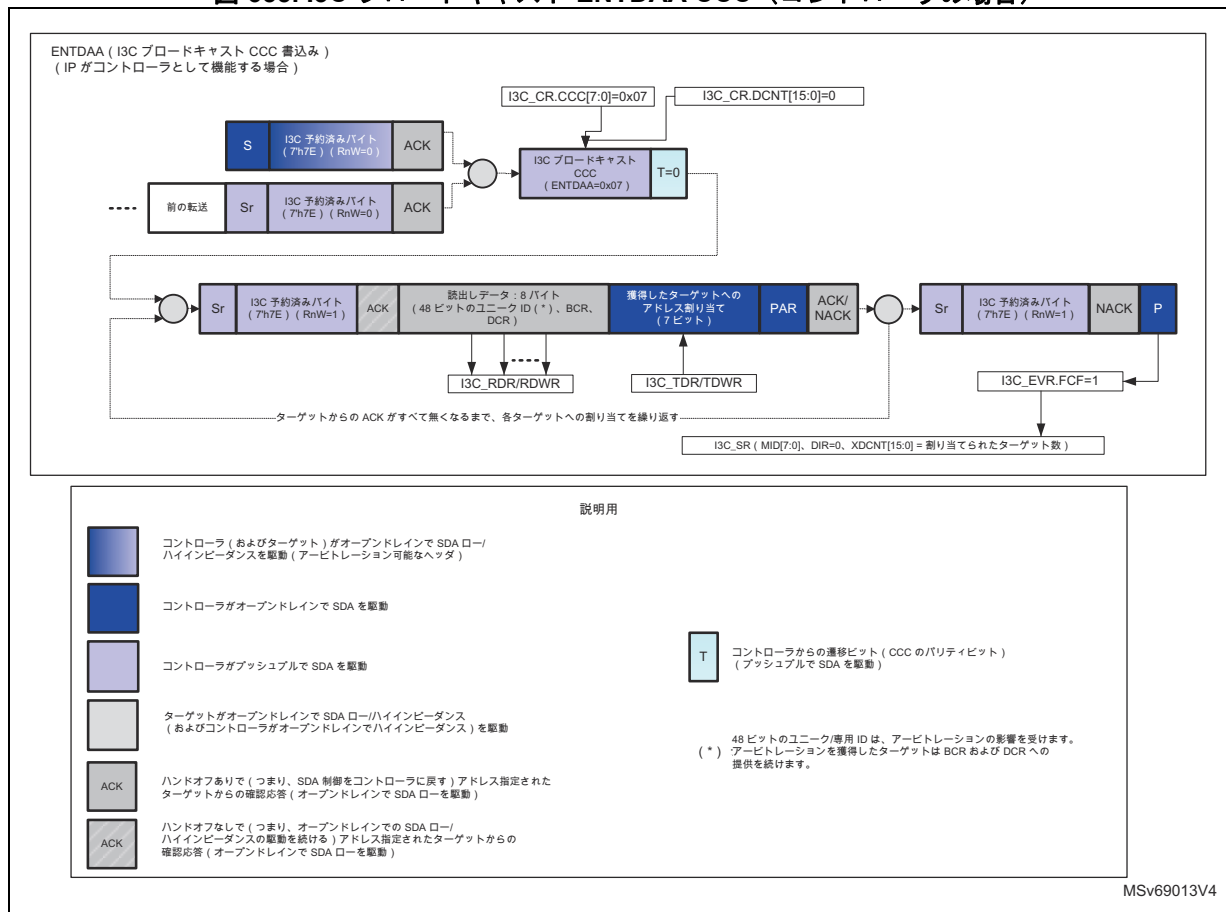
図 654. I3C CCC メッセージ (コントローラの場合)



49.9.3 I3C ブロードキャスト ENTDAACCC 転送（コントローラの場合）

図 655 に、I3C バスで通信され、またコントローラとして機能している場合にプログラムされたとおりの、I3C ブロードキャスト ENTDAACCC を示します。

図 655. I3C ブロードキャスト ENTDAACCC (コントローラの場合)



49.9.4 I3C ブロードキャスト/直接 RSTACT CCC 転送（コントローラの場合）

図 656 に、I3C バスで通信され、またコントローラとして機能している場合にプログラムされたとおりの、I3C ブロードキャスト（書込み）、直接書込みおよび読出し RSTACT CCC を示します。

RSTACT
(I3CブロードキャストCCC書込み)

RSTACT
(I3C直接CCC書込み)、前半

RSTACT
(I3C直接CCC書込み) : n x 後半

RSTACT
(I3C直接CCC読出し)、前半

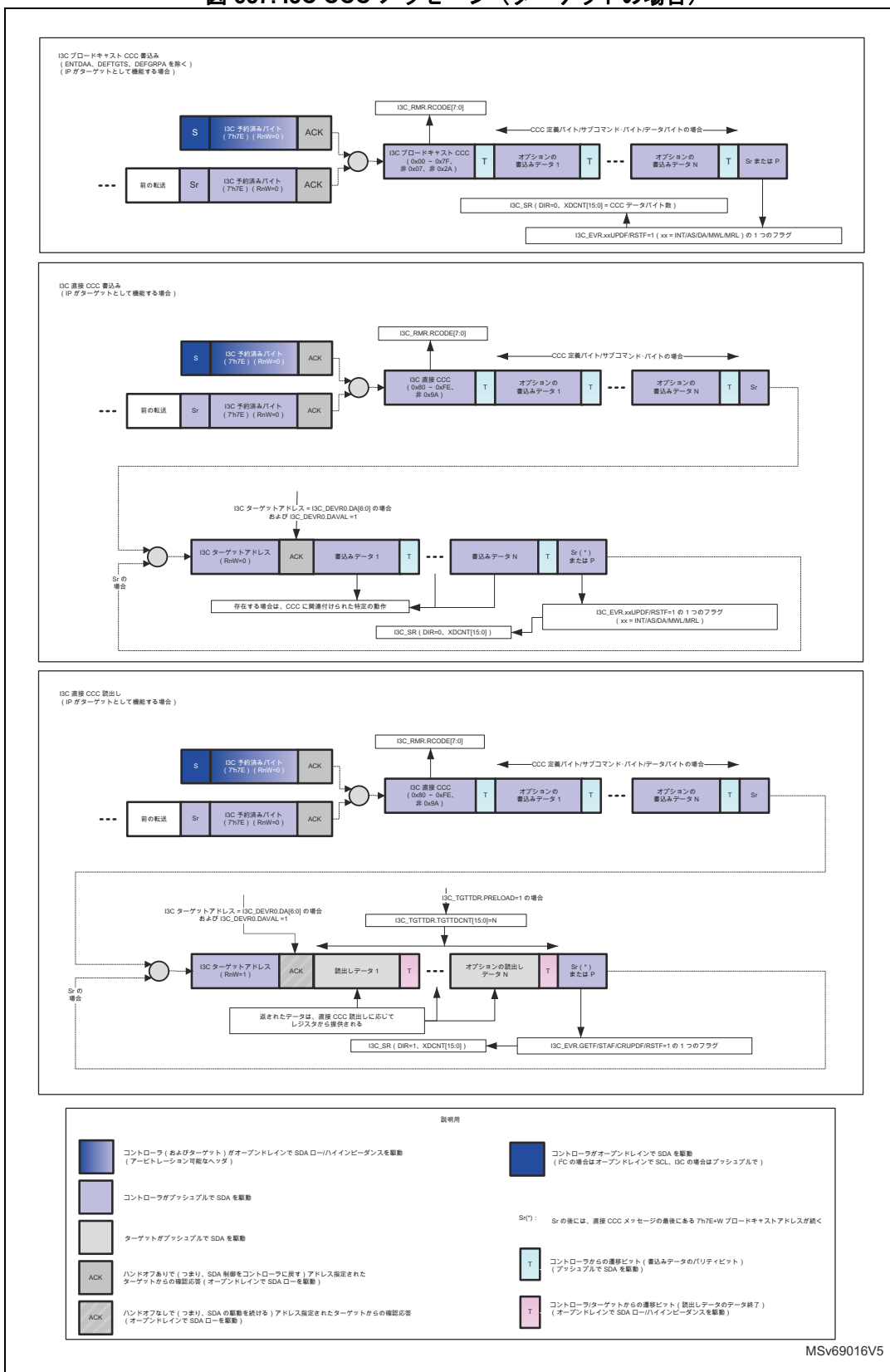
RSTACT
(I3C直接CCC読出し) : n x 後半

説明用	
[Blue Box]	コントローラ（およびターゲット）がオープンドレインで SDA ロー/ハイインピーダンスを駆動（アービトレーション可能なヘッダ）
[Light Blue Box]	コントローラがプルアップで SDA を駆動
[Grey Box]	ターゲットがプルアップで SDA を駆動
[Dark Grey Box]	ACK
[Light Grey Box]	ACK
[T Box]	コントローラからの遷移ビット（書き込みデータのバリディット） （プッシュプルで SDA を駆動）
[P Box]	コントローラ・ターゲットからの遷移ビット（読み出しデータのデータ終了） （オープンドレインで SDA ロー/ハイインピーダンスを駆動）
Sr(*)	Sr の後には、直接 CCC メッセージの最後にある 7hFE+W ブロードキャスト アドレスが続く

49.9.5 I3C ブロードキャスト/直接 CCC 転送 (ENTDAA、DEFTGTS、DEFGRPA を除く) (ターゲットの場合)

[図 657](#) に、I3C バスで通信され、またターゲットとして機能している場合にプログラムされたおりの、I3C ブロードキャスト CCC 書き込み転送 (ENTDAA、DEFTGTS、DEFGRPA を除く)、ならびに直接 CCC 読み出し/書き込み転送を示します。

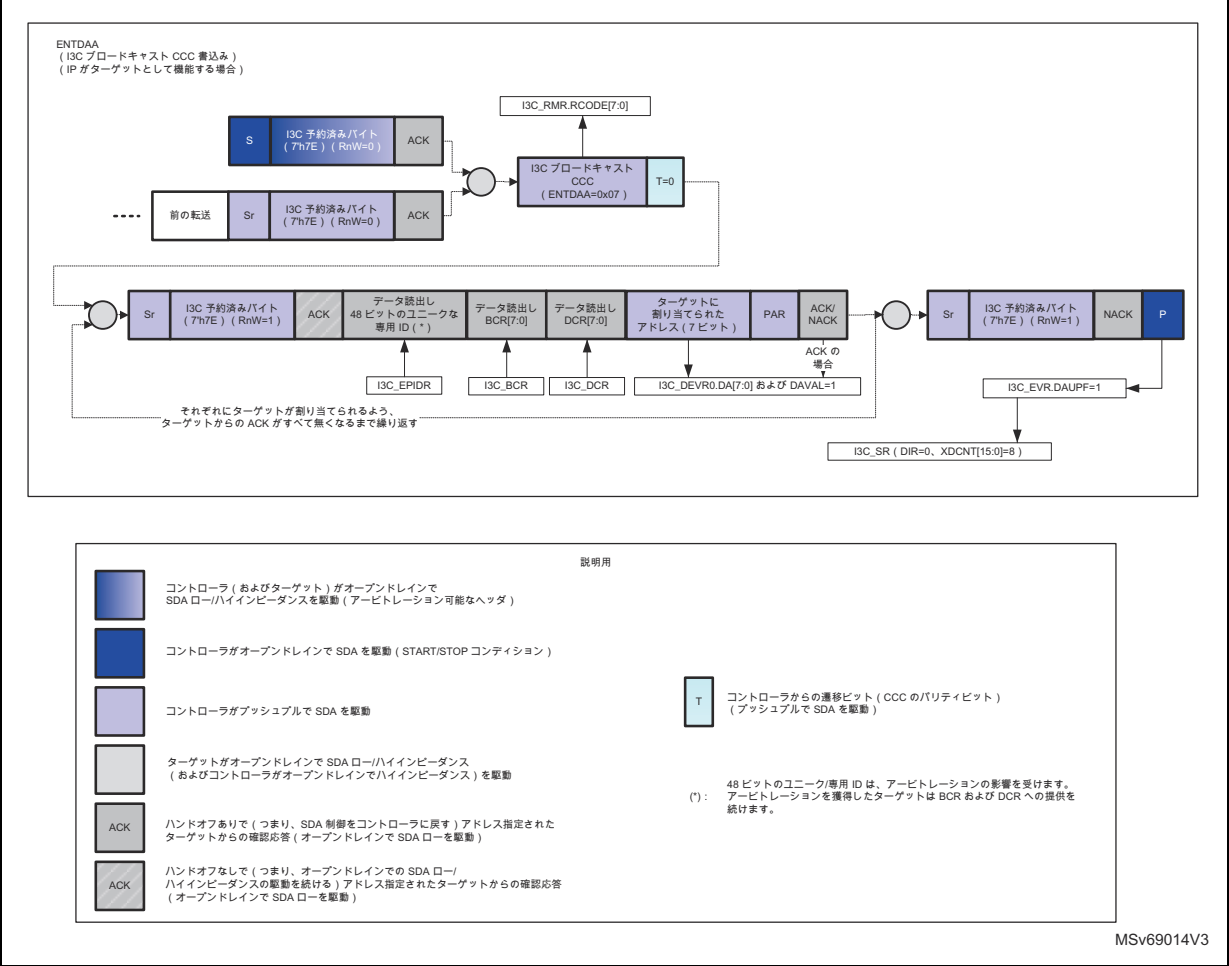
図 657. I3C CCC メッセージ (ターゲットの場合)



49.9.6 I3C ブロードキャスト ENTDAACCC 転送（ターゲットの場合）

図 658 に、I3C バスで通信され、またターゲットとして機能している場合にプログラムされた通りの、I3C ブロードキャスト ENTDAACCC を示します。

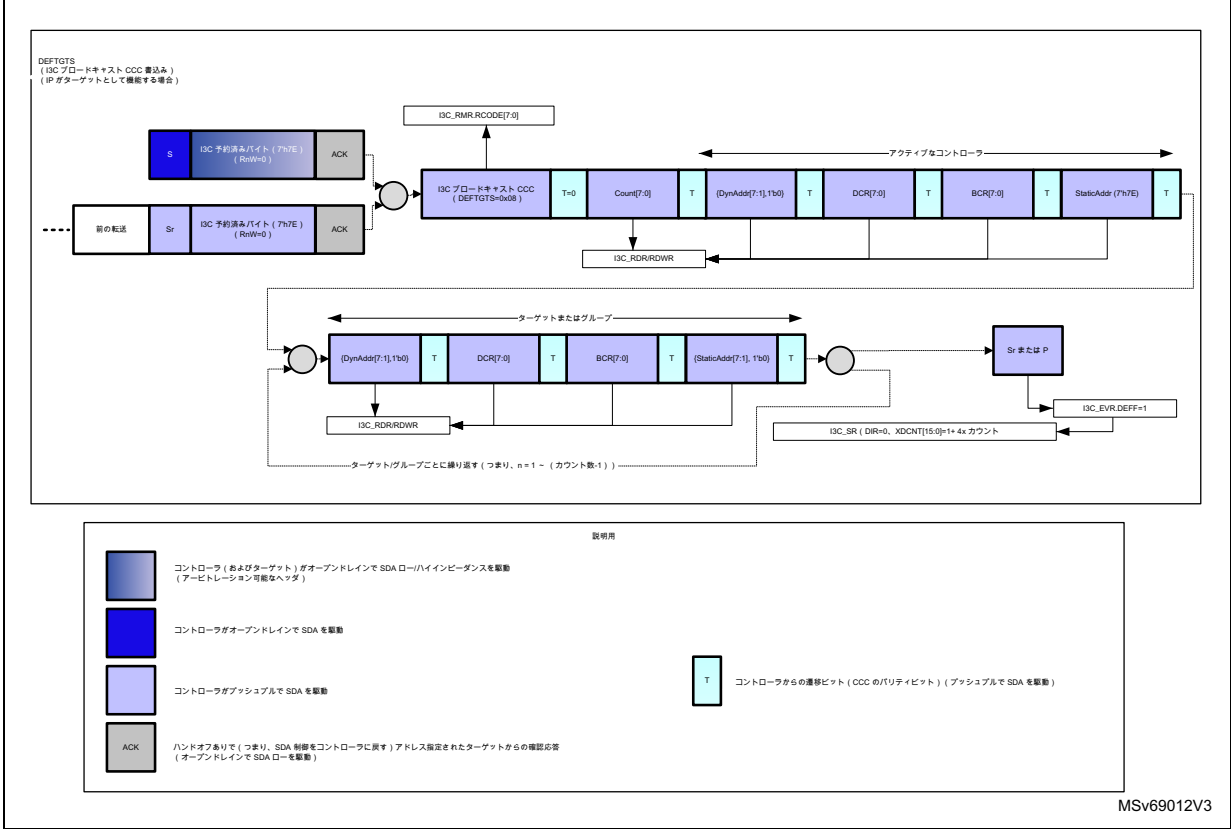
図 658. I3C ブロードキャスト ENTDAACCC（ターゲットの場合）



49.9.7 I3C ブロードキャスト DEFTGTS CCC 転送 (ターゲットの場合)

図 659 に、I3C バスで通信され、またターゲットとして機能している場合にプログラムされた通りの、I3C ブロードキャスト DEFTGTS CCC を示します。

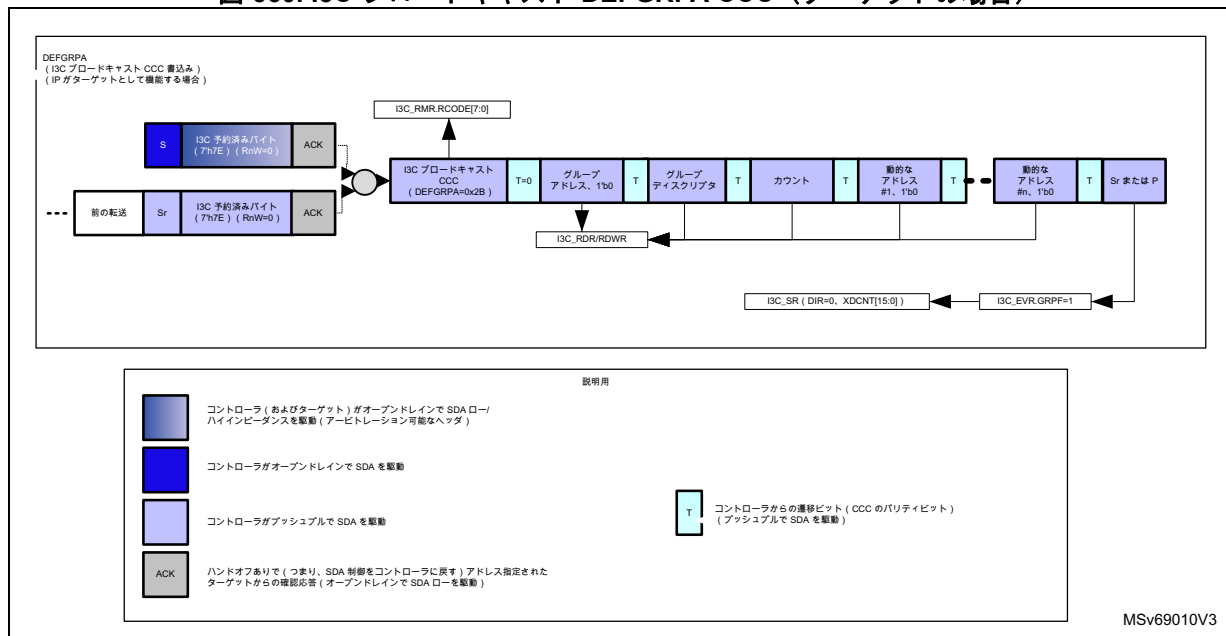
図 659. I3C ブロードキャスト DEFTGTS CCC (ターゲットの場合)



49.9.8 I3C ブロードキャスト DEFGRPA CCC 転送（ターゲットの場合）

図 660 に、I3C バスで通信され、またターゲットとして機能している場合にプログラムされた通りの、I3C ブロードキャスト DEFGRPA CCC を示します。

図 660. I3C ブロードキャスト DEFGRPA CCC（ターゲットの場合）



49.9.9 I3C 直接 GETSTATUS CCC 応答（ターゲットの場合）

I3C がターゲットとして機能する場合、ハードウェアは GETSTATUS CCC の受信時に、フォーマット 1（定義バイトなし、または定義バイト TGTSTAT = 0x00 あり）、またはフォーマット 2（定義バイト PRECR = 0x91 あり）の 2 データバイトを返します。

I3C バスの返されたフォーマット 1 の 2 バイト STATUS[15:0] は、次のとおりです。

- STATUS[15:14] = 00（未使用）
- STATUS[13] = 1：以前の GETSTATUS CCC 以降に開始の見逃しが検出された場合。それ以外の場合は 0
- STATUS[12] = 1：以前の GETSTATUS CCC 以降にオーバーラン/アンダーランエラーが検出された場合。それ以外の場合は 0
- STATUS[11] = 1：以前の GETSTATUS CCC 以降、SDR 読出し中に、SCL が 125 μ s より長い時間安定していることが検出された場合。それ以外の場合は 0
- STATUS[10:8] = 000 ~ 110：以前の GETSTATUS CCC 以降にプロトコルエラーが検出された場合（STATUS[5] = 1 の場合）、ターゲットエラー TEx に対応するエンコードされた値 $x = 0 \sim 6$ 。それ以外の場合は 000
- STATUS[7:6] = 00（ハンドオフ手順の準備完了）
- STATUS[5] = 1：以前の GETSTATUS CCC 以降にプロトコルエラーが検出された場合。それ以外の場合は 0
- STATUS[4] = 0（予約済み）
- STATUS[3:1] = 000（未使用）

- ペンディング中の割込みがある場合は STATUS[0] = 1 (I3C_CR レジスタに IBI が設定されている場合、I3C_DEVR0. レジスタで IBIEN = 1 および DAVAL = 1 の場合、IBI がまだコントローラによって確認応答されておらず、DISEC によって無効化されていない場合)、それ以外の場合は 0

I3C バスの返されたフォーマット 2 の 2 バイト STATUS[15:0] は、次のとおりです。

- STATUS[15:8] = 0000 0000 (未使用)
- STATUS[7:2] = 00000 (未使用)
- STATUS[1] = 1。受信した DEFTGTS または受信した DEFGRPA CCC がまだソフトウェア処理中の場合、また関連イベントがまだソフトウェアによってクリアされていない場合 (I3C_EVR レジスタで DEFF = 1 または GRPF = 1)。コントローラは GETACCR CCC の発行を待たなければならない (それ以外の場合は確認応答されない)
- STATUS[0] = 1。DEFTGTS または DEFGRPA CCC を見逃している場合がある。このビットは、開始の見逃しが検出された場合 (I3C_EVR レジスタで WKPF = 1) にアサートされ、I3C_EVR レジスタで DEFF = 1 または GRPF = 1 の場合にデアサートされる。

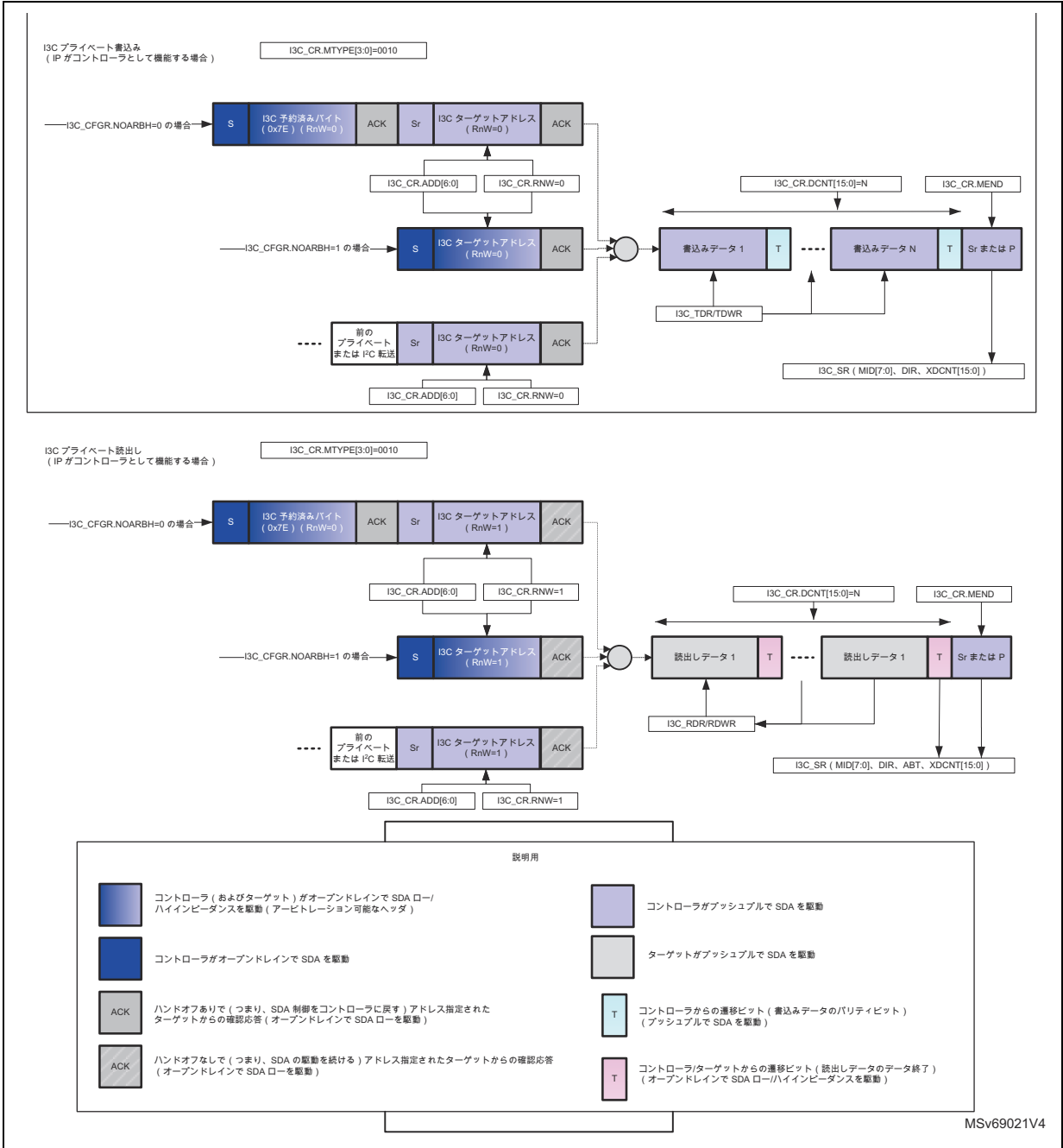
フォーマット 1 の GETSTATUS CCC の完了報告は、I3C_EVR レジスタで STAF = 1 にすることで、また対応する割込みが有効な場合 (I3C_IER レジスタで STAIE = 1 の場合) はそれによって行われます。

フォーマット 2 の GETSTATUS CCC の完了報告は、I3C_EVR レジスタで GETF = 1 にすることで、また対応する割込みが有効な場合 (I3C_IER レジスタで GETIE = 1 の場合) はそれによって行われます。

49.9.10 I3C プライベート読み出し/書き込み転送（コントローラの場合）

図 661 に、I3C バスで通信され、またコントローラとして機能している場合にプログラムされたとおりの、プライベート読み出し/書き込み転送を示します。

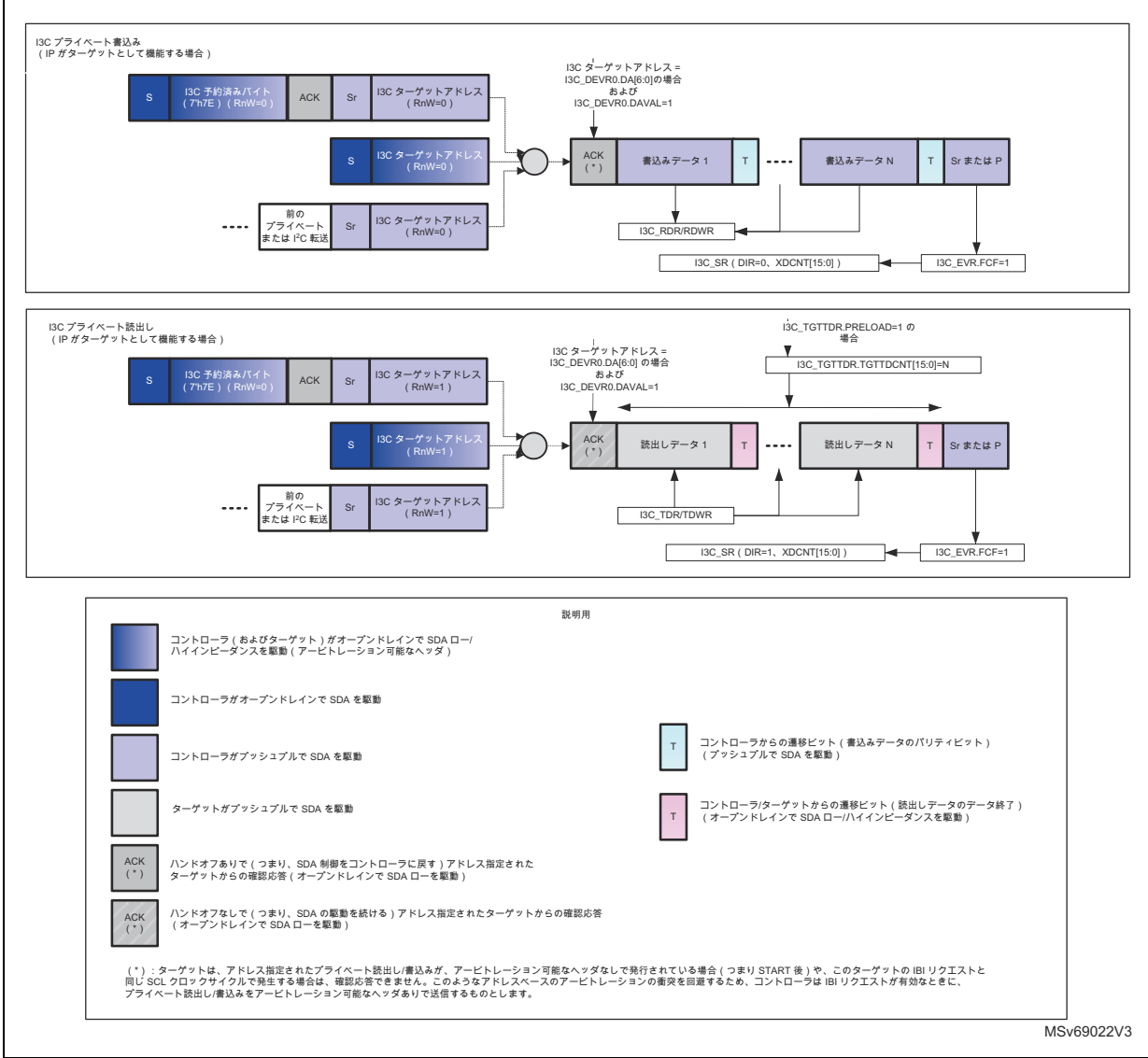
図 661. I3C プライベート読み出し/書き込みメッセージ（コントローラの場合）



49.9.11 I3C プライベート読み出し/書き込み転送（ターゲットの場合）

図 662 に、I3C バスで通信され、またターゲットとして機能している場合にプログラムされた通りの、I3C プライベート読み出し/書き込み転送を示します。

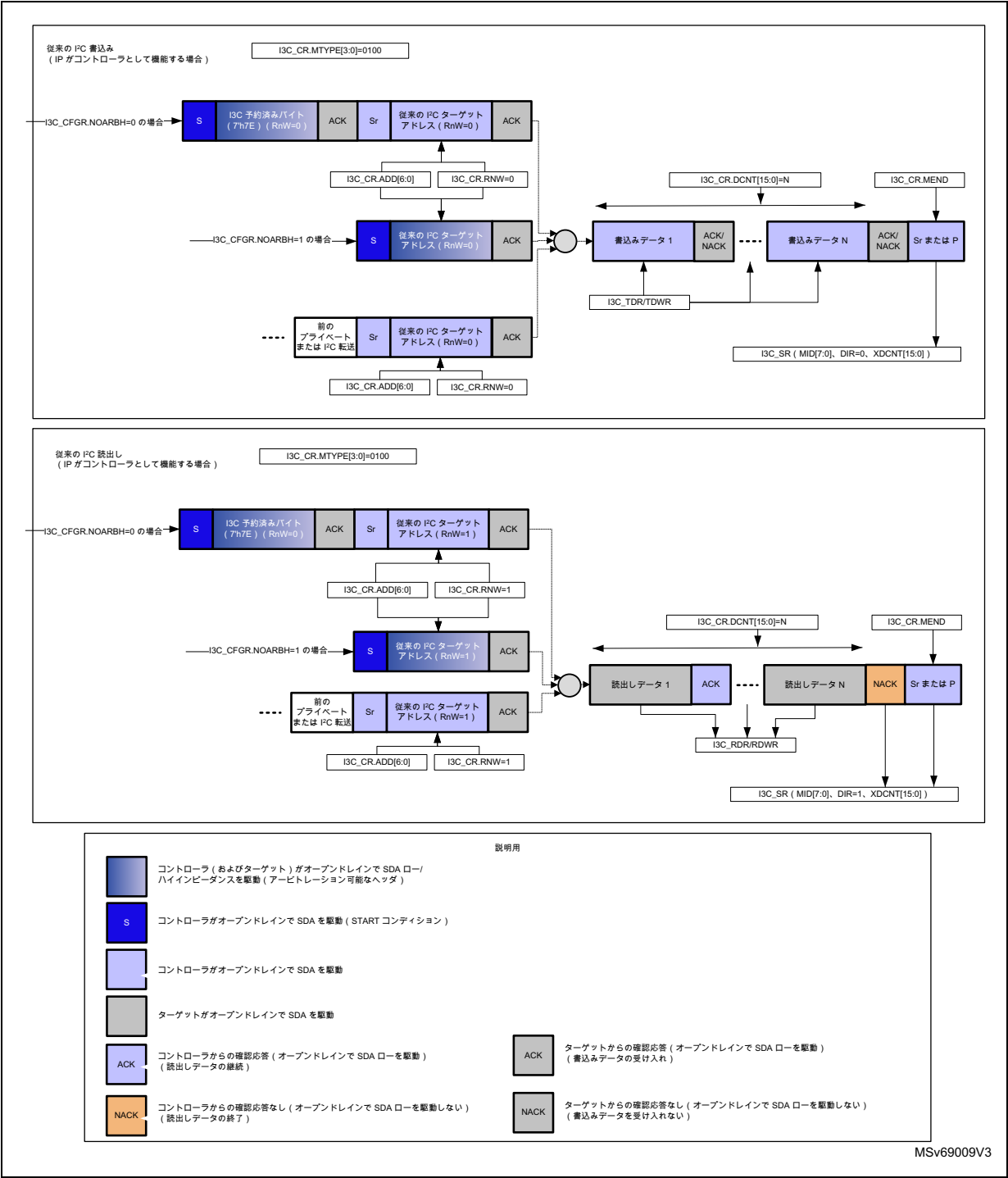
図 662. I3C プライベート読み出し/書き込みメッセージ（ターゲットの場合）



49.9.12 従来の I²C 読出し/書込み転送（コントローラの場合）

図 663 に、I3C バスで通信され、またコントローラとして機能している場合にプログラムされたとおりの、従来の I²C 読出し/書込み転送を示します。

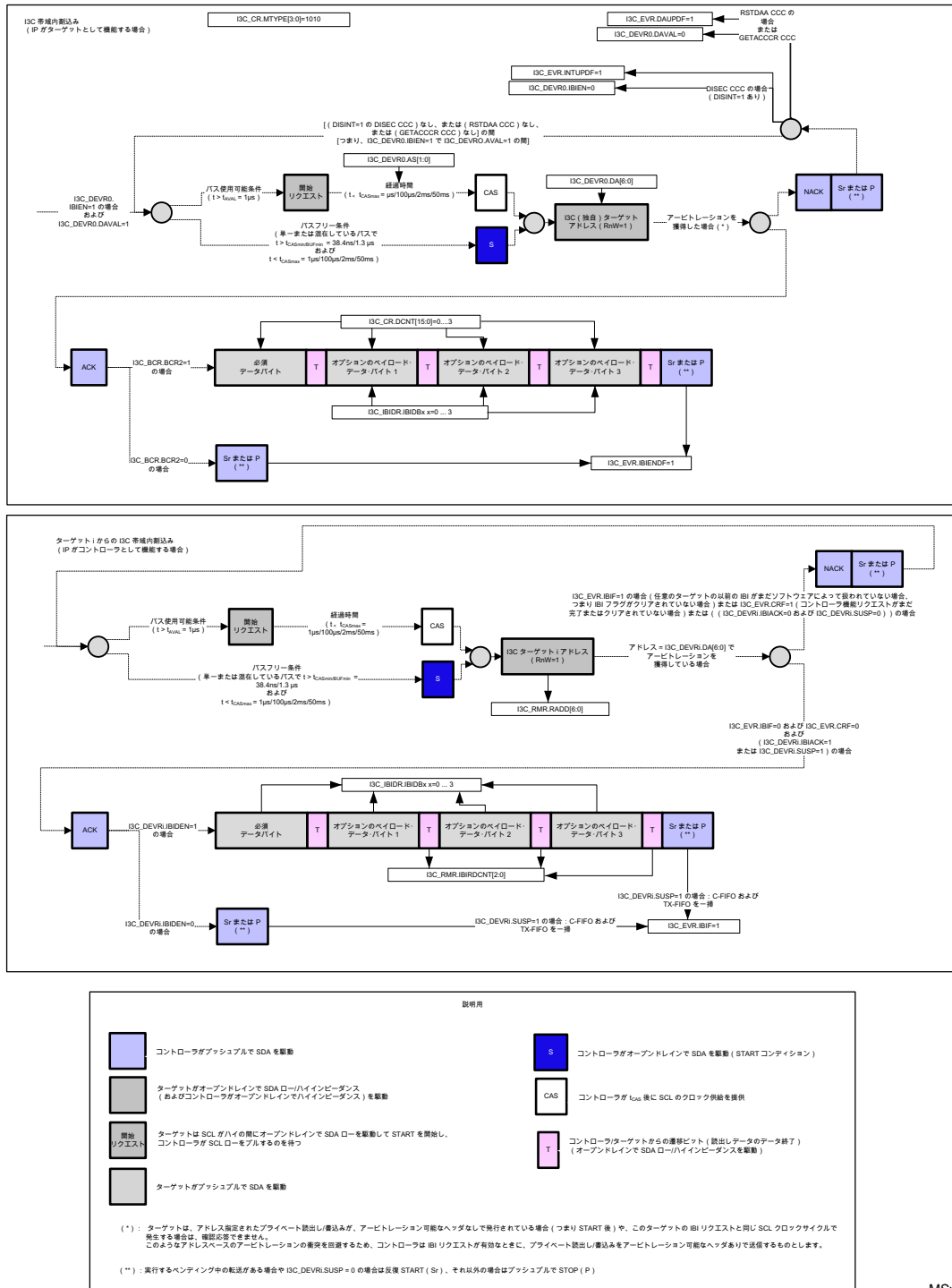
図 663. 従来の I²C 読出し/書込みメッセージ（コントローラの場合）



49.9.13 I3C IBI 転送（コントローラ/ターゲットの場合）

図 664 に、I3C バスで通信され、またターゲットとして機能しているかコントローラとして受信されている場合にプログラムされたとおりの、IBI（帯域内割込み）転送を示します。

図 664. IBI 転送 (コントローラ/ターゲットの場合)

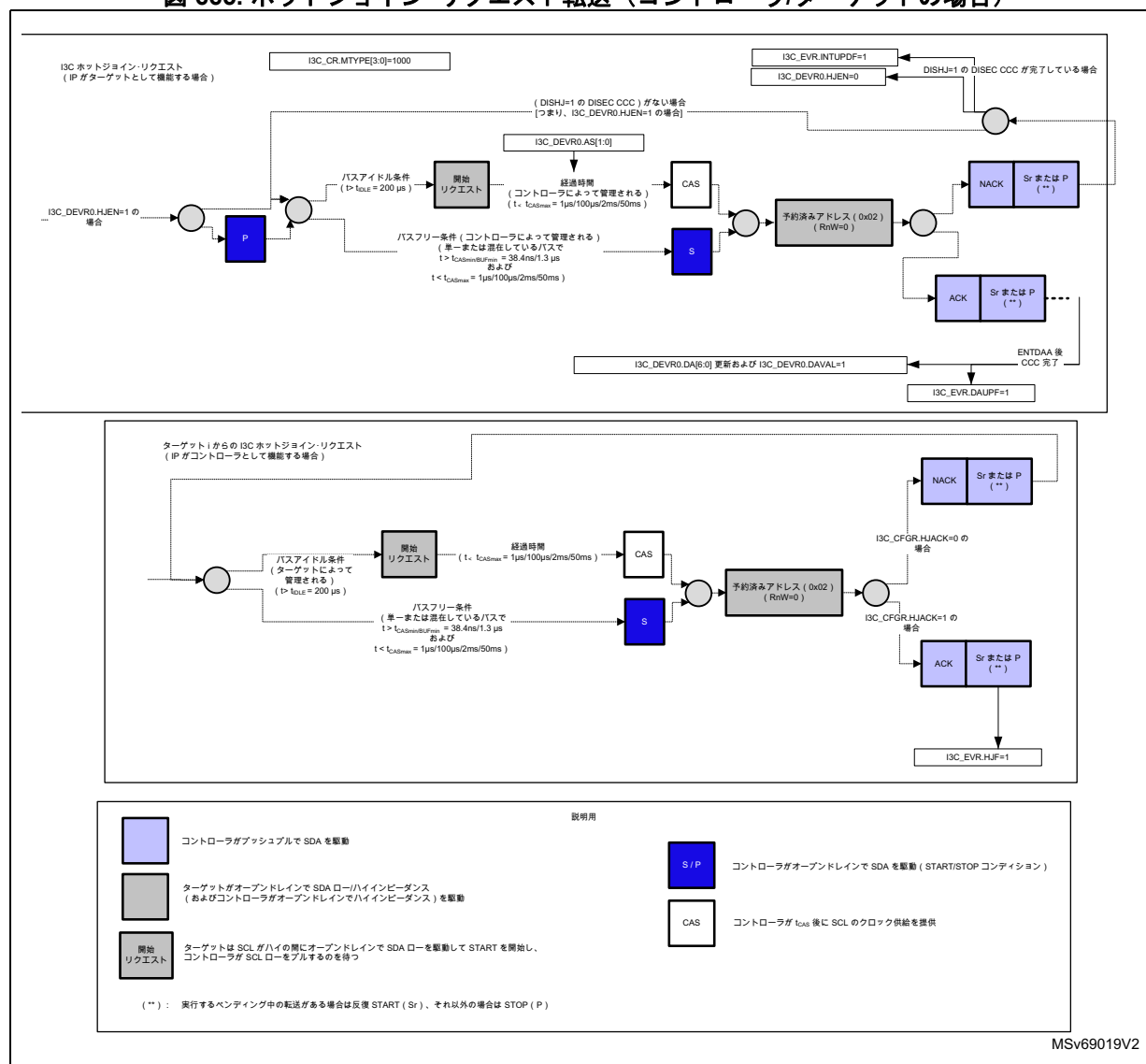


ペリフェラルがコントローラとして機能する場合、I3C_IBIDR レジスタは IBI データペイロードを受信するために使用されます。結果として、ターゲットからの IBI リクエストは 4 バイトのデータペイロードを超えてはなりません。この帯域内割込みのコンテキストではより多くの交換すべき情報があり、コントローラソフトウェアはプライベート読出しを発行する必要があります。

49.9.14 I3C ホットジョイン・リクエスト転送（コントローラ/ターゲットの場合）

図 665 に、I3C バスで通信され、またターゲットとして機能しているかコントローラとして受信されている場合にプログラムされたとおりの、ホットジョイン・リクエスト転送を示します。

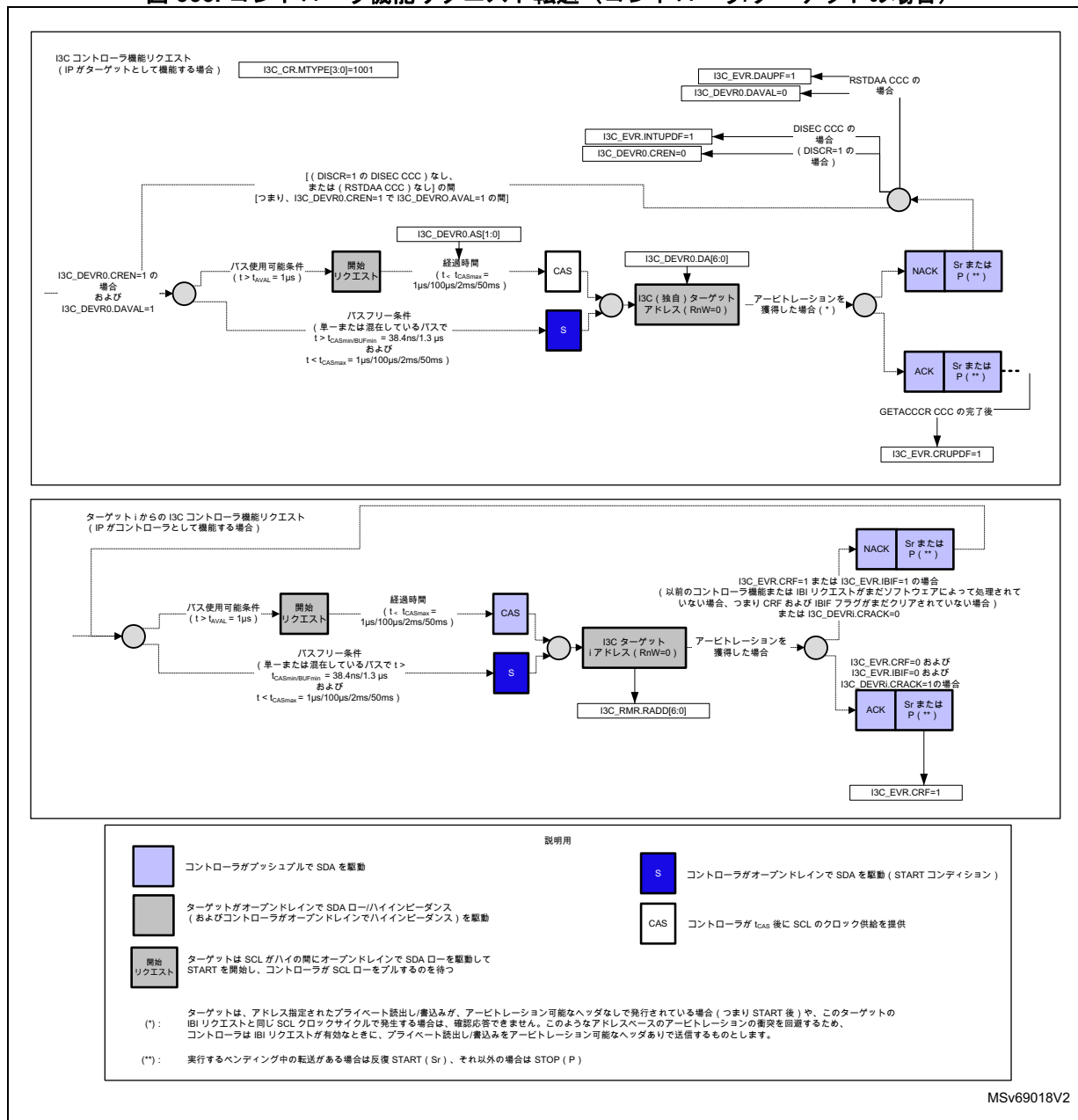
図 665. ホットジョイン・リクエスト転送（コントローラ/ターゲットの場合）



49.9.15 I3C コントローラ機能リクエスト転送（コントローラ/ターゲットの場合）

図 666 に、I3C バスで通信され、またターゲットとして機能しているかコントローラとして受信されている場合にプログラムされたとおりの、コントローラ機能リクエスト転送を示します。

図 666. コントローラ機能リクエスト転送（コントローラ/ターゲットの場合）



49.10 I3C FIFO 管理（コントローラの場合）

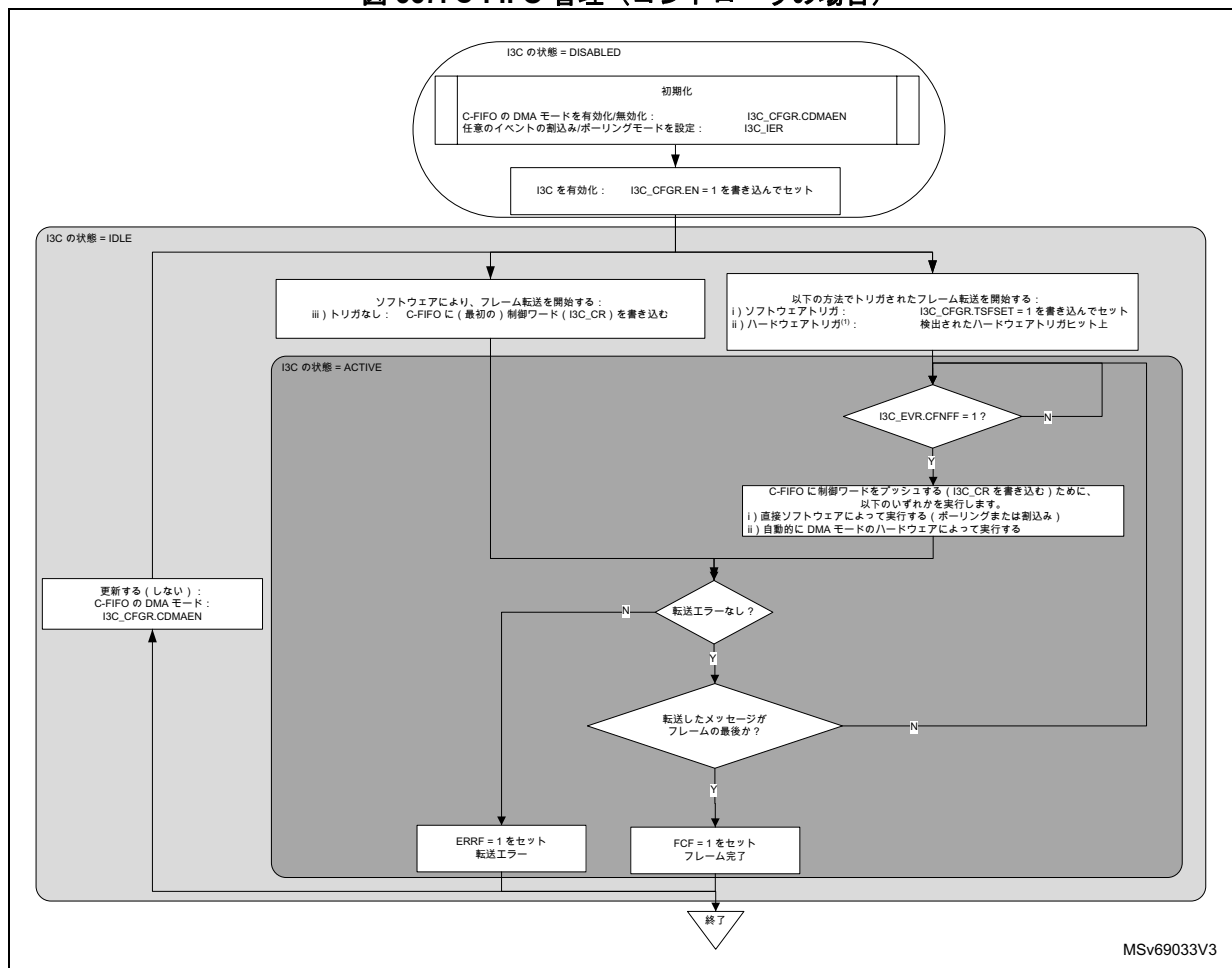
49.10.1 C-FIFO 管理（コントローラの場合）

コントローラの場合、[セクション 49.9](#) の図に示すように、C-FIFO は以下のいずれかの転送中に使用できます。

- ブロードキャスト CCC ([図 654](#)、[図 655](#)、[図 656](#))
- 直接読み出し/書き込み CCC ([図 656](#))
 - コマンドパート、最初のメッセージ
 - データパート、次のメッセージ
- プライベート読み出し/書き込み ([図 661](#))
- 従来の I2C 読み出し/書き込み ([図 663](#))
- ソフトウェアによるエラー・リカバリ（次のヘッダメッセージまで SCL 強制停止、続いて HDR 終了パターン）

[図 667](#) に、I3C ペリフェラルがコントローラとして機能する場合に、I3C バスで制御ワードをキューに入れるための C-FIFO の管理について示します。

図 667. C-FIFO 管理（コントローラの場合）



1. この機能は実装によって使用できない場合があります。[セクション 49.3.4](#) を参照してください。

最初に、ソフトウェアは I3C_CFGR レジスタの CDMAEN を使用して、以下のいずれかのように書き込むために C-FIFO 管理を初期化する必要があります。

- 直接ソフトウェアによって (CDMAEN = 0 の場合)、制御ワードレベルで：
 - ポーリングモードを使用 (I3C_IER レジスタで CFNFIE = 0) : I3C_CR レジスタに明示的に書き込む前に、ハードウェアによって次の制御ワードを待つようリクエストされます (I3C_CR レジスタで CFNFF = 1)。
 - 有効化された割り込み通知を使用 (CFNFIE = 1 の場合)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_tc_dma) から対応する DMA リクエストへ (CDMAEN = 1 の場合) :
 - ブロックレベルで設定したように、DMA はメモリ転送元バッファから I3C_CR レジスタに自動的に制御ワードをプッシュ/書き込んでいます。これは、転送エラーが発生しない限り、フレームが完了するまで行われます (フレームの最後のメッセージ後に I3C バスで STOP が送信されます)。

どの場合でも、C-FIFO が空で、新しい制御ワードで再開を送信する必要がある場合、C-FIFO アンダーランが報告されます (I3C_EVR レジスタで ERRF = 1、また I3C_SER レジスタで COVR = 1)。I3C_IER レジスタで ERRIE = 1 にして有効にした場合は、割り込みが生成されます。

C-FIFO 管理の DMA モードは I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

コントローラの場合、転送エラーが発生した場合 (I3C_EVR レジスタで ERRF = 1)、C-FIFO はハードウェアによって自動的に一掃されます。

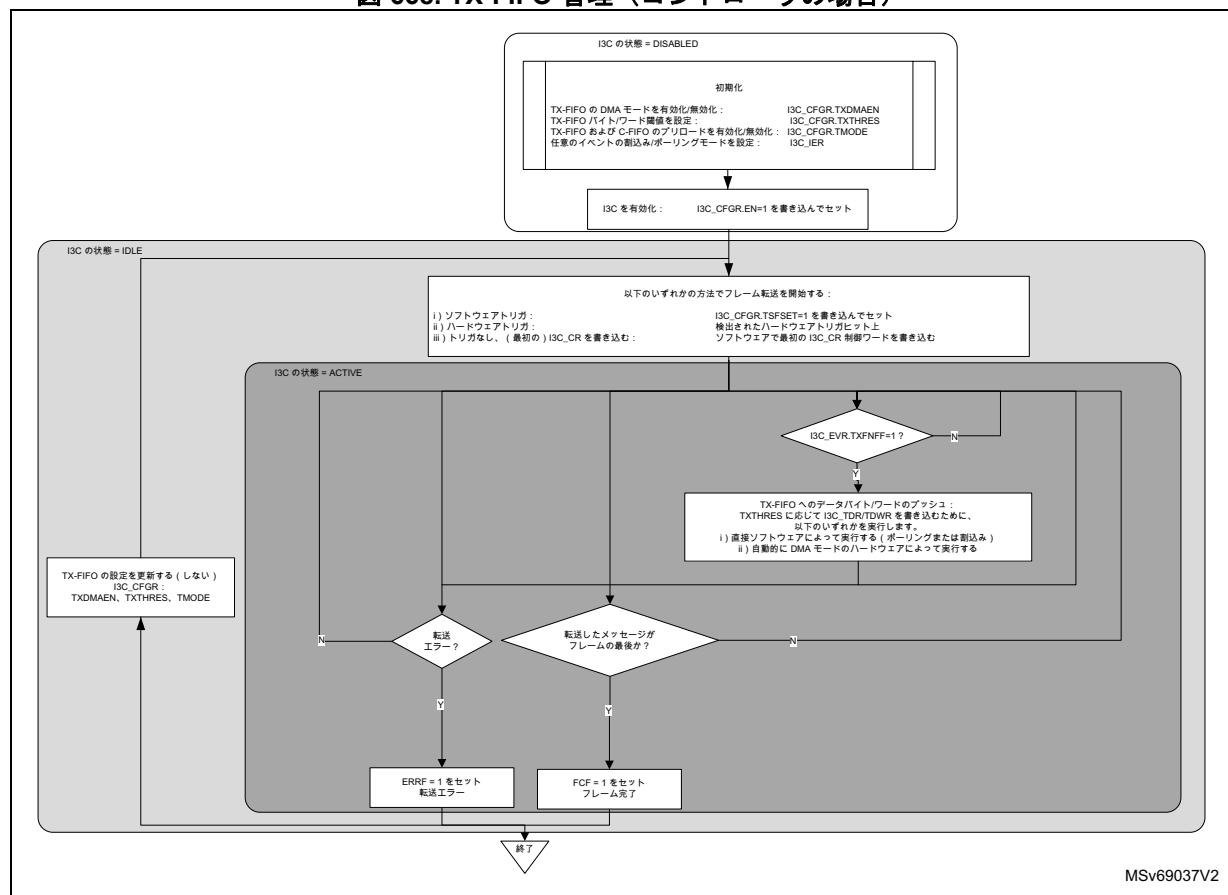
49.10.2 TX-FIFO 管理 (コントローラの場合)

コントローラの場合、[セクション 49.9](#) の図に示すように、TX-FIFO は以下のいずれかの転送中に使用できます。

- ブロードキャストまたは直接 CCC (ENTDAA および RSTACT を含む) : 定義/サブコマンドバイトまたはデータバイトがある場合 ([図 654](#)、[図 655](#)、[図 656](#))
- プライベート書き込み ([図 661](#))
- 従来の I2C 書き込み ([図 663](#))

[図 668](#) に、I3C ペリフェラルがコントローラとして機能する場合に、I3C バスで送信されるデータバイトまたはワードをキューに入れるための、TX-FIFO の管理について示します。

図 668. TX-FIFO 管理 (コントローラの場合)



MSv69037V2

最初に、ソフトウェアは I3C_CFGR レジスタの以下のフィールドを使用して、TX-FIFO 管理を初期化する必要があります。

- TXDMAEN : TX-FIFO の DMA モードの有効化/無効化
- TXTHRES : TX-FIFO へのデータバイトまたはワードのプッシュ
- TMODE : TX-FIFO および C-FIFO プリロードの有効化/無効化

次に、I3C_CFGR レジスタのビット TXDMAEN により、TX-FIFO は以下のいずれかのように書き込まれます。

- 直接ソフトウェアによって (TXDMAEN = 0 の場合)、バイト/ワードレベルで :
 - ポーリングモードを使用 (I3C_IER レジスタで TXFNFIE = 0) : I3C_CFGR レジスタのビット TXTHRES に応じて、I3C_TDR または I3C_TDWR レジスタに明示的に書き込む前に、ハードウェアによって次のデータバイト/ワードを待つようリクエストされます (I3C_CR レジスタで CFNFF = 1)。
 - 有効化された割り込み通知を使用 (TXFNFIE = 1 の場合)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_tx_dma) から対応する DMA リクエストへ (TXDMAEN = 1 の場合) :
 - DMA ブロックレベルで設定したように、DMA はメモリ転送元バッファから (I3C_CFGR. レジスタのビット TXTHRES に応じて) I3C_TDR または I3C_TDWR レジスタに自動的にデータバイト/ワードをプッシュ/書き込んでいます。これは、転送エラーが発生しない限り、フレームが完了するまで行われます (フレームの最後のメッセージ後に I3C バスで STOP が送信されます)。

I3C メッセージは START または反復 START から開始し、STOP または反復 STOP で終了します。メッセージ・レベルでは、I3C_EVR レジスタで TXLASTF = 1 にすると、送信する最後のデータバイト/ワードにフラグが設定されます。I3C フレームが（反復 START で分けられた）複数のメッセージで説明されている場合、ソフトウェアはこのイベントを使用して、次のメッセージのバイト/ワードが格納されているバッファのポインタを更新できます。

フレームの完了の報告時（I3C_EVR レジスタで FCF = 1、また対応する割込みが有効）、TX-FIFO は空です。

TX-FIFO が空で、データバイトを I3C バスで送信する必要がある場合、TX-FIFO アンダーランが報告されます（I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1）。I3C_IER レジスタで ERRIE = 1 にして有効にした場合は、割込みが生成されます。

TX-FIFO 管理の設定は I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

コントローラの場合、転送エラーが発生した場合（ERRF = 1）、TX-FIFO はハードウェアによって自動的に一掃されます。

C-FIFO/TX-FIFO のプリロードなし

表 516 で定義されているように、C-FIFO のサイズは 2 ワードで、TX-FIFO のサイズは 8 バイトです。

C-FIFO/TX-FIFO プリロードが設定されていない場合（I3C_CFGR レジスタで TMODE = 0）、最初の制御ワードが C-FIFO に書き込まれるとすぐに、I3C ペリフェラルは I3C バスで START を送信します。その後、I3C_CR レジスタをデコードし、このメッセージ内で必要な場合は、次のデータバイト/ワードの書き込みを求めます。I3C バスに送信するためにハードウェアによって必要とされる次の制御ワードを検出すると（I3C バスに反復 START を送信する必要がある場合、または C-FIFO が使用可能なスペースを得られる場合）、すぐにこの制御ワードを最後のメッセージまで C-FIFO に書き込むことが求められます（I3C_CR レジスタで MEND = 1）。

同様に、I3C バスに送信するためにハードウェアによって必要とされる別のデータバイト/ワードを検出すると（I3C バスに反復 START を送信する必要がある場合、および TX-FIFO がフルではない場合、およびこの I3C メッセージ中にデータバイト/ワードを送信する必要がある場合）、このデータバイト/ワードを TX-FIFO に書き込む必要があります。

C-FIFO および TX-FIFO のプリロード

バスで START を送信する前に C-FIFO/TX-FIFO プリロードが設定されたら（I3C_CFGR レジスタで TMODE = 1）、I3C ペリフェラルは以下に示すように可能な限り C-FIFO と TX-FIFO の両方がロードされるのを待ちます。

- 最初の制御ワードが C-FIFO に書き込まれるのを待ちます。
- データバイト/ワードがある場合は、TX-FIFO サイズまで、最初の制御ワード（I3C_CR レジスタで RNW = 0 および DCNT[15:0] = 0 の場合）によって定義されたとおりに TX-FIFO に書き込まれるのを待ちます。
- TX-FIFO がフルではなく、また最初の制御ワードがフレームの最後ではない場合（I3C_CR レジスタの MEND = 0）：
 - 2 番目の制御ワードが C-FIFO に書き込まれるのを待ちます。これで C-FIFO がフルになります。
 - TX-FIFO がフルではない場合、データバイト/ワードがある場合は、TX-FIFO サイズまで、2 番目の制御ワード（I3C_CR レジスタで RNW = 0 および DCNT[15:0] = 0）によって定義されたとおりに TX-FIFO に書き込まれるのを待ちます。

次に、I3C バスに送信するためにハードウェアによって必要とされる次の制御ワードを検出すると（I3C バスに反復 START を送信する必要がある場合）、すぐにこの制御ワードを最後のメッセージまで C-FIFO に書き込むことが求められます（I3C_CR レジスタで MEND = 1）。

同様に、I3C バスに送信するためにハードウェアによって必要とされる次のデータバイト/ワードを検出すると (I3C バスに反復 START を送信する必要がある場合、および TX-FIFO がフルではない場合、およびこの I3C メッセージ中にデータバイト/ワードを送信する必要がある場合)、このデータバイト/ワードを TX-FIFO に書き込む必要があります。

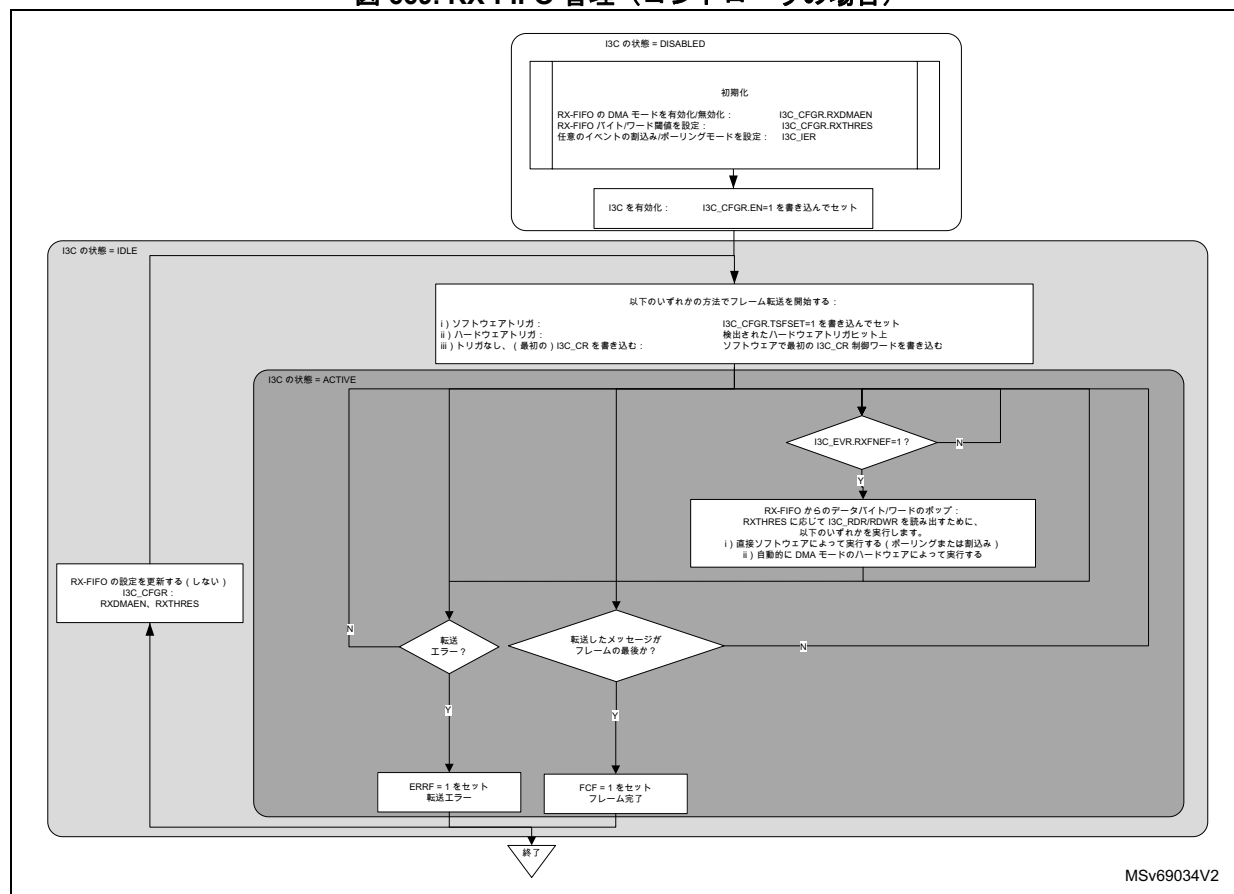
49.10.3 RX-FIFO 管理 (コントローラの場合)

コントローラの場合、[セクション 49.9](#) の図に示すように、RX-FIFO は以下のいずれかの転送中に使用されます。

- ブロードキャスト ENTDAACCC ([図 655](#))
- 直接 CCC 読出し ([図 654](#))、直接 RSTACT CCC 読出し ([図 656](#)) を含む
- プライベート読出し ([図 661](#))
- 従来の I2C 読出し ([図 663](#))

[図 669](#) に、I3C ペリフェラルがコントローラとして機能する場合に、I3C バスから受信したデータバイトまたはワードをキューに入れ、ポップするための、RX-FIFO の管理について示します。

図 669. RX-FIFO 管理 (コントローラの場合)



最初に、ソフトウェアは I3C_CFGR レジスタの以下のフィールドを使用して、RX-FIFO 管理を初期化する必要があります。

- RXDMAEN : RX-FIFO の DMA モードの有効化/無効化
- RXTHRES : RX-FIFO からのデータバイトまたはワードのポップ

次に、RX-FIFO は RXDMAEN に応じて、以下のいずれかのように読み出されます。

- 直接ソフトウェアによって (RXDMAEN = 0)、バイト/ワードレベルで：
 - ポーリングモードを使用 (I3C_IER レジスタで RXFNEIE = 0) : I3C_CFGFR レジスタのビット RXTHRES に応じて、I3C_RDR または I3C_RDWR レジスタに明示的に読み出す前に、ハードウェアによって次のデータバイト/ワードを待つようリクエストされます (I3C_CR レジスタで RXFNEF = 1)。
 - 有効化された割込み通知を使用 (I3C_IER レジスタで RXFNEIE = 1 の場合)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_rx_dma) から対応する DMA リクエストへ (RXDMAEN = 1 の場合)：
 - DMA ブロックレベルで設定したように、DMA は (ビット RXTHRES に応じて) I3C_RDR または I3C_RDWR レジスタから自動的にデータバイト/ワードをポップ/読み出して、これらをメモリ転送先バッファに書き込んでいます。これは、転送エラーが発生しない限り、フレームが完了するまで行われます (フレームの最後のメッセージ後に I3C バスに STOP が送信されます)。

I3C メッセージは START または反復 START から開始し、STOP または反復 STOP で終了します。メッセージ・レベルでは、I3C_EVR レジスタで IRLASTF = 1 にすると、I3C バスから最後の受信データバイト/ワードにフラグが設定されます。I3C フレームが (反復 START で分けられた) 複数のメッセージで説明されている場合、ソフトウェアはこのイベントを使用して、次のメッセージのデータバイト/ワードが格納されているバッファのポインタを更新できます。

RX-FIFO がフルで、データバイトを I3C バスで受信した場合、RX-FIFO オーバーランが報告されます (I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1)。I3C_IER レジスタで ERRIE = 1 にして有効にした場合は、割込みが生成されます。

RX-FIFO 管理の設定は I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

ターゲットからの読出しの早期終了

プライベート読出しメッセージはアドレス指定されたターゲットによって早期完了 (途中終了とも言える) することができます。

- RXDMAEN = 1 の場合：
 - ソフトウェアは、DMA リクエスト i3c_rx_dma に、ペリフェラルの早期終了を可能にする DMA チャンネル x を割り当てる必要があります (DMA の実装セクションを参照)。
 - ソフトウェアは、DMA_CxTR2 レジスタで PFREQ = 1 にすることで DMA ペリフェラルのフロー制御モードを有効にし、(早期かどうかを問わず) DMA ブロックを完了できるようにするために、DMA チャンネル x を設定する必要があります。
 - その後、ブロック完了時、ソフトウェアは DMA_Cx および/または DMA_CxSAR レジスタの BR1.BNDT[15:0] を読み出して、有効な DMA 転送バイト数を取得できます。
- RXDMAEN = 0 の場合：
 - メッセージ・レベルでは、I3C_EVR レジスタで RXFNEF = 1、RXLASTF = 1 にすると、I3C バスから最後の受信データバイト/ワードにフラグが設定されます。

どの場合でも、S-FIFO が無効な場合 (I3C_CFGFR レジスタで SMODE = 0 の場合)、I3C_EVR レジスタで RXTGTENDF = 1 にすることで、また対応する割込みが有効な場合はそれによって、読出しが早期終了することがソフトウェアに通知されます。その後、ソフトウェアは最後のメッセージに関する情報を確認するためにステータスレジスタ I3C_SR を読み出すことができ、途中終了した読出し転送 (I3C_SR レジスタで XDCNT[15:0]) の受信データバイト数を取得します。

どの場合でも、S-FIFO が有効な場合 (I3C_CFGFR レジスタで SMODE = 1)、(I3C_CFGFR レジスタで SDMAEN に応じて) ソフトウェアまたは DMA は、各メッセージについてステータスレジスタ I3C_SR を読み出す必要があります。途中終了した読出しメッセージの有効な受信データバイト数は、I3C_SR レジスタの XDCNT[15:0] によって (次に ABT = 1 で) 報告されます。

詳細については、[I3C ステータスレジスタ \(I3C_SR\)](#) および[セクション 49.10.4](#) を参照してください。

49.10.4 S-FIFO 管理（コントローラの場合）

コントローラの場合、各転送メッセージの [I3C ステータスレジスタ \(I3C_SR\)](#) を読み出すために、ソフトウェアは S-FIFO を使用できます。

S-FIFO が無効なステータスレジスタの読出し

I3C_CFGR レジスタで SMODE = 0 の場合、S-FIFO は無効で、ステータスレジスタは通常のレジスタとして読み出せます。

- レジスタの内容は、新しいメッセージを転送する際にハードウェアによって上書きされます。
- I3C_SR には、最後の転送メッセージのステータスが含まれています。
- SCL クロックは、ステータスレジスタが読み出されていない場合はストールしません。

プライベート読出しがターゲットによって途中終了した特定のケースで、SMODE = 0 の場合：

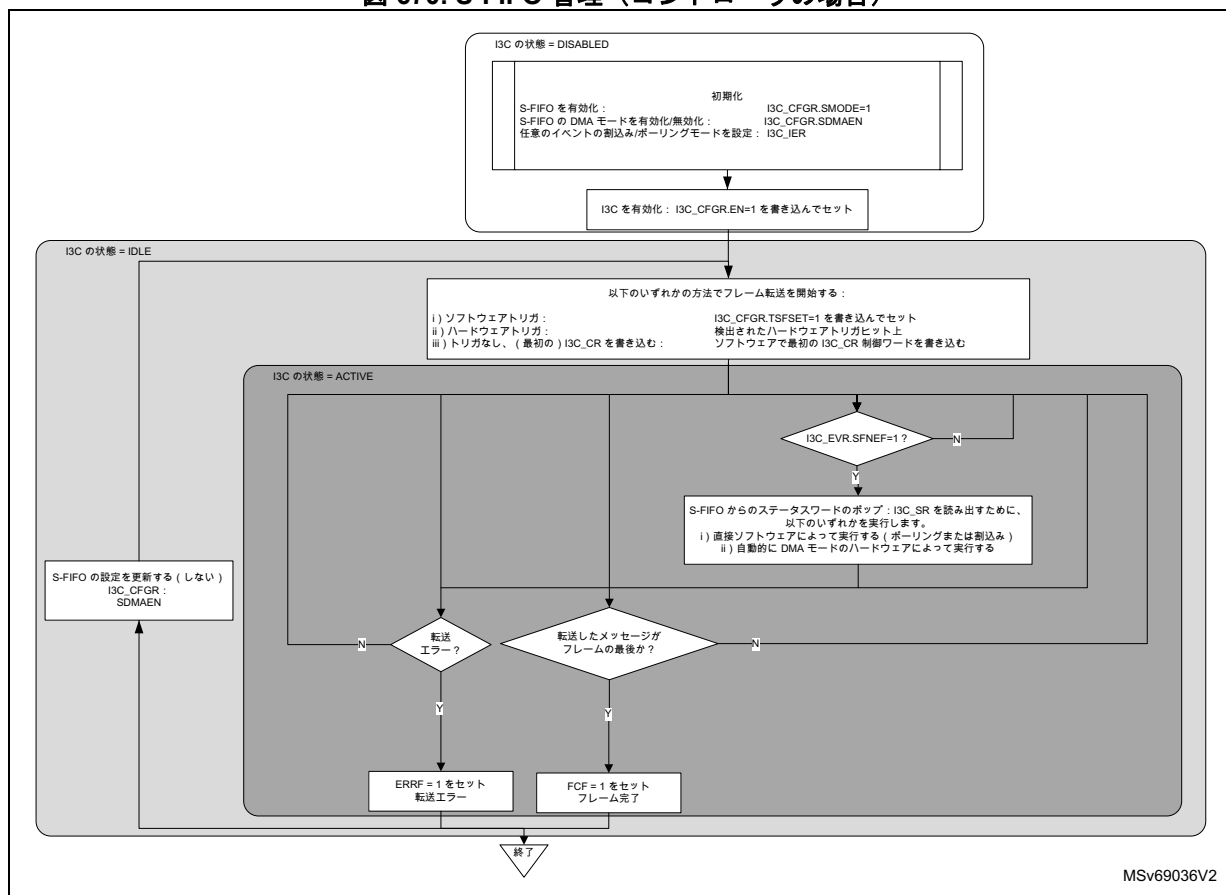
- ソフトウェアへの通知は、I3C_EVR レジスタのフラグ RXTGTENDF = 1 によって、また対応する割込みが有効な場合はそれによって行われます。
- 次に、ソフトウェアがイベントフラグをクリアするまでは、I3C_CEVN レジスタで CRXTGTENDF = 1 と書き込み、セットしてください。
 - これ以上のデータバイトを I3C バスで受信したり、ハードウェアによって I3C_RDR/I3C_RDWR レジスタに書き込んだりできない
 - I3C_SR を更新できない
 - SCL クロックが必要に応じてストールする

通常、I3C_EVR レジスタで FCF = 1、または ERRF = 1、または RXTGTENDF = 1 の場合、I3C_SR を読み出せます。プライベート読出しの有効な受信データバイト数を取得するには、ターゲットからの早期終了後に I3C_SR レジスタの XDCNT[15:0] を読み出すことができます([セクション 49.10.3](#) を参照)。

S-FIFO が有効なステータスレジスタの読出し

I3C_CFGR レジスタで SMODE = 1 の場合、S-FIFO は有効です。[図 670](#) に、I3C ペリフェラルがコントローラとして機能する場合に、I3C バス上の各実行メッセージについてステータスワードをキューに入れ、ポップするための、S-FIFO の管理について示します。

図 670. S-FIFO 管理 (コントローラの場合)



MSv69036V2

最初に、ソフトウェアは I3C_CFGR レジスタの SDMAEN フィールド (S-FIFO の DMA モードの有効化/無効化) を使用して、S-FIFO 管理を初期化する必要があります。次に、S-FIFO は SDMAEN に応じて、以下のいずれかのように読み出されます。

- 直接ソフトウェアによって (SDMAEN = 0 の場合) :
 - ポーリングモードを使用 (I3C_IER レジスタで SFNEIE = 0) : I3C_SR レジスタに明示的に読み出す前に、ハードウェアによって次のステータスワードを待つようリクエストされます (I3C_CR レジスタで SFNEF = 1)。
 - 有効化された割込み通知を使用 (SFNEIE = 1)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_rs_dma) から対応する DMA リクエストへ (SDMAEN = 1 の場合) :
 - DMA ブロックレベルで設定したように、DMA は I3C_SR レジスタから自動的にステータスワードをポップ/読み出して、これらをメモリ転送先バッファに書き込んでいます。これは、転送エラーが発生しない限り、フレームが完了するまで行われます (フレームの最後のメッセージ後に I3C バスに STOP が送信されます)。

各メッセージステータスを読み出す必要がありますが、読み出せない場合は、S-FIFO がフルの場合、オーバーランエラーがハードウェアによってアサートされ (I3C_EVR レジスタで ERRF = 1、I3C_SER レジスタで COVR = 1)、次のメッセージのステータスを書き込む必要があります。I3C_IER レジスタの ERRIE=1 の場合、対応する割込みが発生します。

フレームの完了 (I3C_EVR レジスタで FCF = 1) は、S-FIFO が空の場合にのみ報告されます。

S-FIFO 管理の設定は I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

49.11 I3C FIFO 管理（ターゲットの場合）

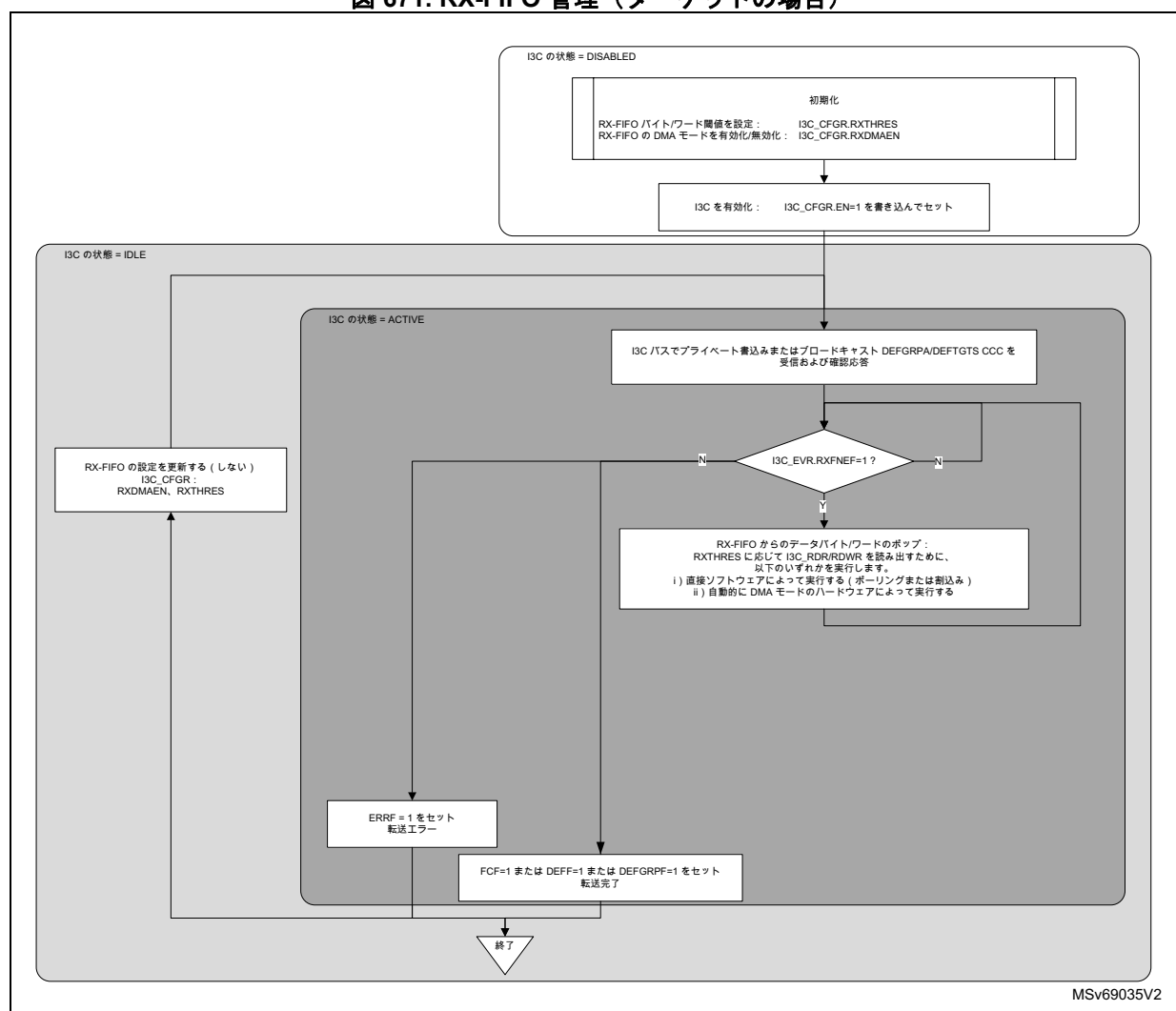
49.11.1 RX-FIFO 管理（ターゲットの場合）

ターゲットの場合、[セクション 49.9](#) の図に示すように、RX-FIFO は以下のいずれかの受信され、確認応答された転送中に使用されます。

- ブロードキャスト DEFTGTS CCC ([図 659](#))
- ブロードキャスト DEFGRPA CCC ([図 660](#))
- プライベート書き込み ([図 662](#))

[図 671](#) に、I3C ペリフェラルがターゲットとして機能する場合に、I3C バスから受信したデータバイトまたはワードをキューに入れ、ポップするための、RX-FIFO の管理について示します。

図 671. RX-FIFO 管理（ターゲットの場合）



最初に、ソフトウェアは I3C_CFGR レジスタの以下のフィールドを使用して、RX-FIFO 管理を初期化する必要があります。

- RXDMAEN : RX-FIFO の DMA モードの有効化/無効化
- RXTHRES : RX-FIFO からのデータバイトまたはワードのポップ

次に、RX-FIFO は RXDMAEN に応じて、以下のいずれかのように読み出されます。

- 直接ソフトウェアによって (RXDMAEN = 0 の場合)、バイト/ワードレベルで：
 - ポーリングモードを使用 (I3C_IER レジスタで RXFNEIE = 0) : I3C_CGFR レジスタの RXTHRES に応じて、I3C_RDR または I3C_RDWR レジスタに明示的に読み出す前に、ハードウェアによって次のデータバイト/ワードを待つようリクエストされます (I3C_CR レジスタで RXFNEF = 1)。
 - 有効化された割り込み通知を使用 (RXFNEIE = 1 の場合)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_rx_dma) から対応する DMA リクエストへ (RXDMAEN = 1 の場合)：
 - DMA ブロックレベルで設定したように、DMA は (RXTHRES に応じて) I3C_RDR または I3C_RDWR レジスタから自動的にデータバイト/ワードをポップ/読み出して、これらをメモリ転送先バッファに書き込んでいます。これは、転送エラーが発生しない限り、転送が完了するまで行われます (I3C_EVR レジスタでは、プライベート書き込みの場合は FCF = 1、DEFTGTS CCC の場合は GRPF = 1、または DEFTGTS CCC の場合は DEFF = 1)。

RX-FIFO がフルで、新しいデータバイトを I3C バスで受信した場合、RX-FIFO オーバーランが報告されます (I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1)。有効な場合は対応する割り込みも報告されます。

転送完了時 (I3C_EVR レジスタで FCF = 1、または GRPF = 1、または DEFF = 1)：

- RX-FIFO は空です。
- 完了したプライベート書き込み/DEFTGTS/DEFTGTS 転送に対応する RX データバッファがソフトウェアで未処理の間は (つまり、ソフトウェアが対応するフラグの書き込みをクリアせず、I3C_CEVR レジスタの CFCF / CDEFF / CGRPF を 1 にセットするまでは)、次のプライベート書き込み/DEFTGTS CCC/DEFTGTS CCC が受信され、データバイトをハードウェアによって RX-FIFO に書き込む必要がある場合、RX-FIFO オーバーラン (I3C_EVR レジスタで ERRF = 1、I3C_SER レジスタで DOVR = 1) によって、また対応する割り込みが有効な場合はそれによって、転送エラーが報告されます。

RX-FIFO 管理の設定は I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

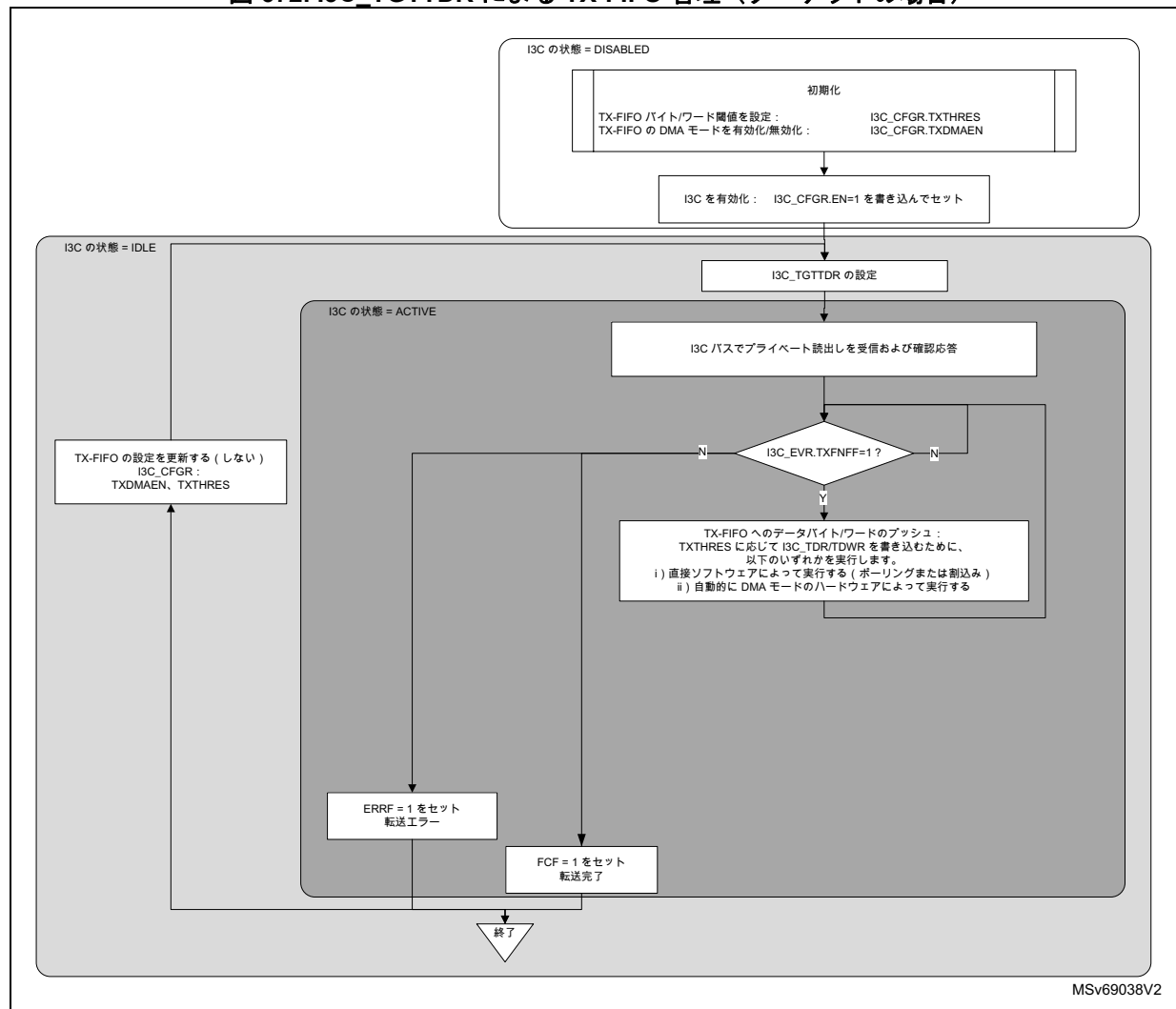
49.11.2 TX-FIFO 管理 (ターゲットの場合)

主な仕組み

ターゲットの場合、[セクション 49.9](#) の図に示すように、TX-FIFO はプライベート読み出し中にのみ使用されます ([図 662](#))。

[図 672](#) に、I3C ペリフェラルがターゲットとして機能する場合に、I3C バスで送信されるデータバイトまたはワードをキューに入れ、プッシュするための、TX-FIFO の管理について示します。

図 672. I3C_TGTTDR による TX-FIFO 管理 (ターゲットの場合)



最初に、ソフトウェアは TX-FIFO 管理を初期化し、I3C_CFGR に以下を書き込む必要があります。

- TXDMAEN : TX-FIFO の DMA モードの有効化/無効化
- TXTHRES : TX-FIFO へのデータバイトまたはワードのプッシュ

次に、I3C バスでプライベート読出しを受信する前に、ソフトウェアは TX-FIFO をプリロードするためにデータバイト数 (シングルアクセスで TGTTDCNT[15:0] と PRELOAD = 1 を書き込む) で **I3C ターゲット送信設定レジスタ (I3C_TGTTDR)** を設定する必要があります。これにより、ターゲットからのデータバイトを I3C バスで送信できるようになります。

- PRELOAD = 1 で TGTTDCNT[15:0] > TX-FIFO サイズの場合、TX-FIFO は最初に FIFO サイズまでプリロードされます。
- PRELOAD = 1 で TGTTDCNT[15:0] ≤ TX-FIFO サイズの場合、TX-FIFO は TGTTDCNT[15:0] までプリロードされます。

注 : TX-FIFO サイズは 8 バイトです (表 516 を参照)。

TXDMAEN によって、TX-FIFO は以下のいずれかの方法でプリロードされます。

- 直接ソフトウェアによって (TXDMAEN = 0)、バイト/ワードレベルで：
 - ポーリングモードを使用 (I3C_IER レジスタで TXFNFIEN = 0) : I3C_CFGFR レジスタの TXTHRES に応じて、I3C_TDR または I3C_TDWR レジスタに明示的に書き込む前に、ハードウェアによって次のデータバイト/ワードを待つようリクエストされます (I3C_CR レジスタで CFNFF = 1)。
 - 有効化された割り込み通知を使用 (TXFNFIEN = 1 の場合)
- 割り当てられた DMA チャンネルによって、I3C ペリフェラル (i3c_tx_dma) から対応する DMA リクエストへ (TXDMAEN = 1) :
 - DMA ブロックレベルで設定したように、DMA はメモリ転送元バッファから (TXTHRES に応じて) I3C_TDR または I3C_TDWR レジスタに自動的にデータバイト/ワードをプッシュ/書き込んでいます。これは、転送エラーが発生しない限り、転送が完了するまで行われます (IEC_EVR レジスタで FCF = 1)。

その後、プライベート読出しを続けるために TX-FIFO にロードする I3C_TGTTDR レジスタに TGTDCNT[15:0] が残っている場合 (PRELOAD = 1 にセットし、TGTDCNT[15:0] > TX-FIFO サイズにする)、またコントローラによるプライベート読出しがまだ完了していない場合は、直接ソフトウェアによって、または割り当てられた DMA チャンネルによって、同様に動作します。

- TXTHRES = 0 : バイトが I3C バスで送信された場合、次のバイトは TX-FIFO にプリロードされます。
- TXTHRES = 1:4 バイトが I3C バスで送信された場合、次のワードは TX-FIFO にプリロードされます。

ターゲットまたはコントローラのどちらかがデータバイト転送を最初に終了したときに、プライベート読出しの転送が完了します (I3C_EVR レジスタで FCF = 1)。

転送完了時に、ソフトウェアは以下のいずれかを行うことができます。

- I3C_SR レジスタの XDCNT[15:0] の読出し : 有効な送信データバイト数
- I3C_TGTTDR レジスタの TGTDCNT[15:0] の読出し : I3C バスでロードされ、送信される残りのバイト数
- TX-FIFO の一掃 : I3C_CFGFR レジスタで TXFLUSH = 1 を書き込んでセットする (かどうか)、別のプライベート読出しを続ける (かどうか) はユーザアプリケーションに応じて異なる

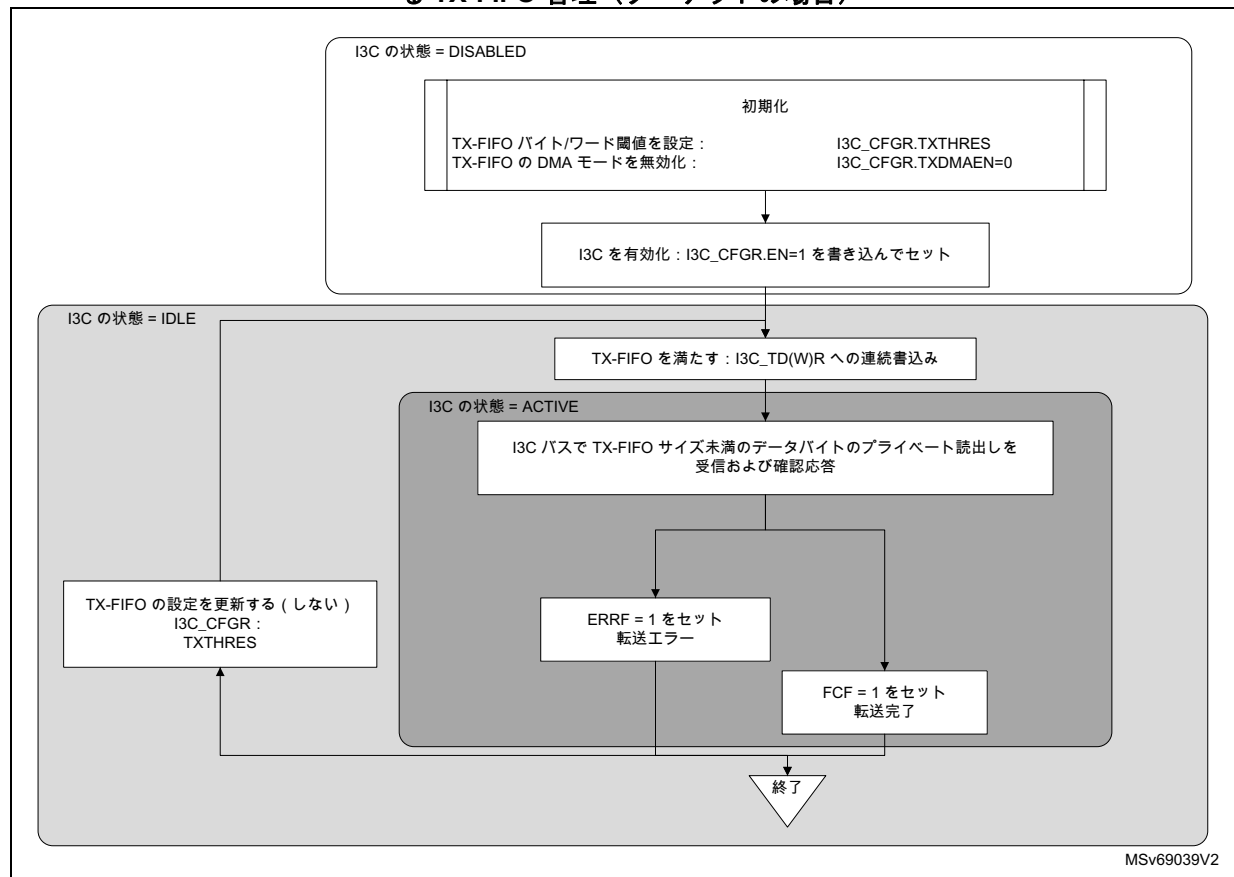
TX-FIFO が空で、データバイトを I3C バスで送信する必要がある場合、TX-FIFO アンダーランが報告されます (I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1)。I3C_EVR レジスタの ERRIE = 1 で有効にした場合は、割り込みが生成されます。

TX-FIFO 管理の設定は I3C ペリフェラルがアクティブ状態ではない場合に変更できます。

TX-FIFO サイズよりバイト数が少ない場合の I3C_TGTTDR を使用しない方法

図 673 に示すように、I3C_TGTTDR レジスタを使用しない方法として、DMA を使用しない場合は (TDMAEN = 0)、I3C バスで読み出されるデータバイト数が TX-FIFO サイズより少なければ、XTHRES に応じて、I3C_TDR または I3C_I3C_TDWR への連続した書込みにより、ソフトウェアは直接準備して、TX-FIFO にソフトウェアによって直接読み出されるバイト数を書き込むことができます。

図 673. TX-FIFO サイズよりバイト数が少ない読出しの I3C_TGTTDR を使用しないソフトウェアによる TX-FIFO 管理 (ターゲットの場合)



49.12 I3C エラー管理

49.12.1 コントローラのエラー管理

表 525 では、I3C バスのエラー条件と、I3C ペリフェラルがコントローラとして機能する場合の、それぞれに対応する検出、操作、および報告をエニユメレーションします。

表 525. I3C コントローラのエラー管理

エラータイプ	説明	エラー検出	コントローラの操作	報告されたエラー ⁽¹⁾
CE0	不正にフォーマットされた CCC (例: 返されたデータバイトが少ない)	直接 CCC 読出しで、ターゲットによって途中終了した読出しデータ ⁽²⁾	ハードウェアは STOP を送信します。	ERRF = 1 および PERR = 1 および CODERR[3:0] = 0000
CE1	監視エラー	従来の I ² C 読出し終了時に正しくない ACK が検出された	ハードウェアは SCL 実行状態を 9 クロックサイクルの間維持し、別のバイトと交換できるようにします。その後、再度 NACK、STOP の順で送信します。	ERRF = 1 および PERR = 1 および CODERR[3:0] = 0001
		I3C SDR 読出し終了時に RESTART または STOP を生成できない	SCL は実行状態を維持します。ソフトウェアは、I3C_CR レジスタ (ヘッダメッセージ) に制御ワード MTYPER[3:0] = 0000 を書き込むことで SCL を停止でき、その後次のメッセージを送信するまでに通常は少なくとも 150 μs 待つ必要があります。	
		CE1 エラー後に START を生成できない		
CE2	ブロードキャストアドレスに対してレスポンスなし (0b111_1110)	昇格フォールトまたはリセット・パターンのメッセージ以外のメッセージ中に、NACK されたヘッダ (0b111_1110 + RnW = 0) が検出された	ハードウェアは、HDR 終了パターン、STOP の順で送信します。	ERRF = 1 および PERR = 1 および CODERR[3:0] = 0010
CE3	コントローラ・ハンドオフの失敗	新しいコントローラが、SDA がローになった後 (テスト・ヘッダによる START またはターゲットからの開始リクエスト)、およびその動作状態によって定義された遅延時間経過後に、SCL をローに駆動していない	ハードウェアは、START + 0b111_1110 + RnW = 0 を送信し、続けてターゲットから ACK/NACK を送信し、停止します。	ERRF = 1 および PERR = 1 および CODERR[3:0] = 0011
-	GETACCCR CCC に返された、パリティビットを含む 7 ビットのターゲットアドレスが正しくない	正しくない動的なアドレスおよび/またはパリティビットが検出された	ハードウェアは、RESTART + 0b111_1110 + RnW = 0 を送信し、続けてターゲットから ACK/NACK を送信し、停止することで、GETACCCR CCC をキャンセルします。	ERRF = 1 および DERR = 1
-	アドレス指定されたターゲットが直接 CCC 読出しで NACK された (1 回目)	NACK が検出された	ハードウェアは、RESTART + 7 ビットと同じターゲット・アドレス + RnW = 1 を送信することで、1 回再試行します。	-

表 525. I3C コントローラのエラー管理 (続き)

エラー タイプ	説明	エラー検出	コントローラの操作	報告されたエラー ⁽¹⁾
-	アドレス指定されたターゲットが、直接 CCC 書き込み、I3C プライベート読み出し/書き込み、従来の I ² C、または直接 CCC 読み出しのいずれかで NACK された (2 回目)	NACK が検出された	ハードウェアは STOP を送信します。	ERRF = 1 および ANACK = 1
-	パリティビットを含む割り当てられた/送信された動的なアドレスが ENTDAACCC で NACK された (1 回目)	NACK が検出された	ハードウェアは、RESTART + 0b111_1110 + RnW = 1、ACK および (最優先) ターゲットからの 8 バイトの読み出しデータ、割り当てられたアドレス + パリティビットの順で送信することで、割り当てループを 1 回再試行します。	-
-	パリティビットを含む割り当てられた/送信された動的なアドレスが ENTDAACCC で NACK された (2 回目)	NACK が検出された	ハードウェアは STOP を送信します。	ERRF = 1 および DNACK = 1
-	従来の I2C 書き込みで書き込みデータが NACK された			
-	時間内の書き込み/読み出しが行われていない制御ワード、ステータスワード、送信データ、または読み出しデータと I3C バスタイミング	SCL のストールのタイムアウト	ハードウェアは STOP を送信します。	ERRF = 1 および (COVR = 1 または DOVR = 1)

1. I3C_EVR レジスタの ERRF、I3C_SER レジスタの PERR、CODERR[3:0]、DERR、ANACK、DNACK、COVR、および DOVR。
2. MIPI v1.1 : GETCAPS CCC で、受信データバイト数は 2、3、または 4 です。ただし、MIPI v1.0 と一致しているターゲットは、以前に名付けた GETHDRCAP CCC に従って最初のバイトのみを返せます。結果的に、数字が 4 未満の場合、CE0 は生成されません。GETMXDS CCC で、受信データバイト数は 2 または 5 です。数字が 5 未満の場合、CE0 は生成されません。

49.12.2 ターゲットのエラー管理

表 526 では、I3C バスのエラー条件と、I3C ペリフェラルがターゲットとして機能する場合の、対応する検出、操作、および報告をエニユメレーションします。

表 526. I3C ターゲットのエラー管理

エラータイプ	説明	エラー検出	次の操作 (コントローラによって 送信されている場合)	報告されたエラー ⁽¹⁾
TE0	無効なブロードキャストアドレス (0b111_1110+RnW = 0) または 無効な 7 ビットの動的なアドレス + RnW = 1 (DAA 割り当て後)	START または反復 START 後に禁止されたアドレスが検出された	ハードウェアは HDR 終了パターンを待つ	ERRF = 1 および PERR = 1 および ODERR[3:0] = 1000
TE1	CCC コード	パリティエラーのある CCC コードが検出された		ERRF = 1 および PERR = 1 および ODERR[3:0] = 1001
TE2	書き込みデータ	I3C プライベート書き込みメッセージで、パリティエラーのある書き込みデータバイトが検出された	ハードウェアは STOP または反復 START を待つ	ERRF = 1 および PERR = 1 および ODERR[3:0] = 1010
TE3	動的なアドレスのアービトレーション中に割り当てられたアドレス	ENTDAA CCC で、パリティエラーのある割り当てられた動的なアドレスが検出された		ERRF = 1 および PERR = 1 および ODERR[3:0] = 1011
TE4	動的なアドレスのアービトレーションの Sr 後に 0b111_1110 + RnW = 1 が失われた	動的なアドレスのアービトレーションの Sr 後に {0b111_1110 + RnW = 1} が失われたことが検出された		ERRF = 1 および PERR = 1 および ODERR[3:0] = 1100
TE5	CCC 検出後のトランザクション	CCC 直接読み出し/書き込みで、アドレスフェーズで示された方向に対して無効な CCC の方向が検出された		ERRF = 1 および PERR = 1 および ODERR[3:0] = 1101
TE6	監視エラー	SDR データ読み出し (CCC 直接読み出しまたはプライベート読み出しまたは IBI) で、SDA が予期しない値で駆動していることが検出された	ハードウェアは SDA を解放し、STOP または反復 START を待つ	ERRF = 1 および PERR = 1 および ODERR[3:0] = 1110
-	SDR データ読み出しでの SCL の監視	SDR データ読み出し (CCC 直接読み出しまたはプライベート読み出しまたは IBI) で、SCL が 125 μ s より長い時間安定していることが検出された		ERRF = 1 および STALL = 1
-	書き込まれていない/間に合わなかった送信データと I3C バスタイミング	プライベート読み出しで、送信データがソフトウェアまたは DMA によって時間内に書き込まれていない	ハードウェアは STOP または反復 START を検出する	ERRF = 1 および DOVR = 1
-	登録が間に合わなかった受信データと I3C バスタイミング	プライベート書き込みまたは DEFTGTS または DEFGRPA CCC で、受信データがソフトウェアまたは DMA によって時間内に読み出されていない		

1. I3C_EVR レジスタの ERRF、I3C_SER レジスタの PERR、ODERR[3:0]、STALL、および DOVR。

49.13 低電力モードからの I3C のウェイクアップ

49.13.1 STOP からのウェイクアップ

ユーザは、I3C ペリフェラルまでのクロックデータパスをセットするために、最初のリセットおよびクロックコントローラ (RCC) を設定する必要があります。この場合、I3C カーネルクロックにはソースオシレータ、I3C APB クロックにはソースオシレータを選択し、クロック周波数をセットします。RCC で初期化する際、ユーザは所定の I3C ペリフェラルが RUN/SLEEP モードおよび STOP モードで個別に機能するように、I3C クロックを有効にする必要があります。RCC プログラミングの詳細については、[セクション 11：リセットおよびクロック制御 \(RCC\)](#) を参照してください。

I3C ハードウェアは、デバイスが RUN、SLEEP、または STOP モードの場合は、いつでも専用のクロックゲートを自動的に管理し、そのカーネルクロックと APB クロックに RCC への個別のクロックリクエストの出力信号を生成します。

STOP モードに移行する際、 V_{CORE} ドメインには電源が供給され、デフォルトで任意のクロックオシレータが無効化され、またどのような場合でも、ドメイン内でシステムクロックまたはペリフェラルクロックが実行されることはありません。

コントローラの場合：MDB、ホットジョイン・リクエスト、またはコントローラ機能リクエストなしで IBI でウェイクアップ

ペリフェラルがコントローラとして機能する場合、製品が低電力モードに移行する前に、ソフトウェアは ENTASx CCC を発行する必要があります。通常 $x = 0, 1, 2$ 、または 3 で、それぞれ $1 \mu s$ 、 $100 \mu s$ 、 $2 ms$ 、または $50 ms$ のインターバルが経過する前に、I3C コントローラがアイドル状態を終了したり、I3C バスで通信したりすることが予期されていないことを、ターゲットに通知します。この遅延は、コントローラの T_{CAS} 遅延を定義します。これにより、START コンディションの後に、SCL バスクロックがローで実行するようセットされます。STOP モードの場合、より具体的には、CCC を ENTASx (値 $x = 1, 2$ 、または 3) に制限する必要があります。

STOP からの I3C ウェイクアップの仕組みは通常、次のとおりです。

- 最初に I3C ペリフェラルがシステムにカーネルクロックをリクエストします。
 - I3C バスで検出された START コンディションによる (SCL がハイである間、SDA ラインがローに駆動されていることが検出される)。
 - ホットジョイン/帯域内割込み/コントローラ機能リクエストについて、外部 I3C ターゲットデバイスで起動されたコントローラとして。カーネルクロックが供給され、実行されると、I3C ハードウェアは内部タイマを使用して対応する T_{CAS} 時間の経過を待ち、次にロー SCL を駆動して SCL をトグルし続けます。これにより、ターゲットは I3C バスでホットジョイン/帯域内割込み/コントローラ機能リクエストを実行できます。
- システムは I3C カーネルクロックのソースオシレータを有効にすると、クロックが準備に入ります (HSI から数ミリ秒後)。ユーザは、起動の遅延時間を短縮するため、消費電力と引き換えに STOP モードでソースオシレータのオン状態を維持します。
 - I3C ペリフェラルは、I3C バスで STOP コンディションが生成されるまで、カーネルクロックリクエストを維持します。
- その後カーネルクロックが実行され、I3C ペリフェラルはコントローラとしてシステムに APB クロックをリクエストします。
 - 受信した IBI リクエストの ACK されたアドレスで ([図 664](#))、APB クロックリクエストは当該 IBIF がクリアされるまで維持されます。
IBI に MDB がない場合：APB クロックがまだ供給されていなくても、STOP は通常 I3C バスで生成されます。
 - 受信したコントローラ機能リクエストの ACK されたアドレスで ([図 665](#))、APB クロックリクエストは当該 CRF がクリアされるまで維持されます。

- ホットジョイン・リクエストの ACK されたアドレスで (図 666)、APB クロックリクエストは当該 HJF がクリアされるまで維持されます。
- 4. システムに I3C APB クロックリクエストが通知され、PWR モジュールのパワー・マネージメント・ユニットが起こされます。
- 5. レギュレータには、RUN モードの電圧を上げる必要がある場合に、追加の遅延が必要になる可能性があります。
- 6. システムが、APB クロックを駆動するシステムクロックを有効にします。システムと APB クロックに選択したオシレータソースが I3C カーネルクロックを駆動しているものと異なる場合、追加の遅延が発生します。その場合、システムは遅延後に準備できる、システムクロックのソースオシレータを有効にします。
- 7. APB クロックを実行した状態で、ペリフェラルはそのステータスレジスタとデータレジスタでの I3C 転送を記録できます。バス転送完了後、ペリフェラルは対応するフラグ (IBIF/CRF/HJF) を生成し、有効化された割込みは CPU をウェイクアップできます。

ターゲットの場合：リセット・パターンでウェイクアップ

ターゲットのリセット・パターンは、コントローラをウェイクアップし、ターゲットの低電力モードからのリセットを可能にする固有の仕組みです。RSTACT CCC 内および帯域内リセット・パターン生成内の両方で構成されています。

ターゲットの場合、リセット・パターンでの STOP からの I3C ウェイクアップのシーケンスは次のとおりです。

1. ペリフェラルがバス上でリセット・パターンを検出すると (SCL がローに保持されている間の 14 SDA 遷移)、I3C_EVR レジスタの RSTF フラグをセットするために、システムに APB クロックをリクエストします。
2. システムに I3C APB クロックリクエストが通知され、PWR モジュールのパワー・マネージメント・ユニットが起こされます (HSI から数ミリ秒後)。
3. レギュレータには、RUN モードの電圧を上げる必要がある場合に、追加の遅延が必要になる可能性があります。
4. システムは数ミリ秒後に準備できる、システムクロックのソースオシレータを有効にします。
5. APB クロックを実行した状態で、I3C ペリフェラルは I3C_EVR レジスタの RSTF フラグを立てることができ、また有効化された割込みによって CPU をウェイクアップできます。

ターゲットの場合：開始の見逃しによるウェイクアップ

1. ペリフェラルがシステムにカーネルクロックをリクエストします。
 - I3C バスで検出された START コンディションによる (SCL がハイである間、SDA ラインがローに駆動されていることが検出される)。
 - コントローラであるかターゲットであるかにかかわらず、外部 I3C デバイスで起動されたターゲットとして。
 - SCL が外部コントローラによってローに駆動される前にカーネルクロックが供給されていない場合、I3C ターゲットは I3C バスの開始を見逃したことを検出し、カーネルクロックが供給されるのを待ちます。送信されるはずだった I3C コントローラメッセージを見逃している (NACK している) 場合があります。または、コントローラから別のアドレス指定されたターゲット、または別のターゲットから IBI/CR/HJF 開始リクエストへの、I3C CCC またはプライベート・メッセージを見逃している場合があります。
2. システムは I3C カーネルクロックのソースオシレータを有効にすると、クロックが準備に入ります (HSI の場合、数ミリ秒後)。ユーザは、起動の遅延時間を短縮するため、消費電力と引き換えに STOP モードでソースオシレータのオン状態を維持します。特に、コントローラが $x = 0$ ($1 \mu s T_{CAS}$ 遅延あり) の動作状態である場合。
3. その後カーネルクロックが実行され、非ユーザ・テキストによって調整された WKPF フラグを立てるために、I3C ペリフェラルはターゲットとしてシステムに APB クロックをリクエストします。

4. システムに I3C APB クロックリクエストが通知され、PWR モジュールのパワー・マネージメント・ユニットが起こされます (HSI から数ミリ秒後)。
5. レギュレータには、RUN モードの電圧を上げる必要がある場合に、追加の遅延が必要になる可能性があります。
6. システムが、APB クロックを駆動するシステムクロックを有効にします。システムと APB クロックに選択したオシレータソースが I3C カーネルクロックを駆動しているものと異なる場合、追加の遅延が発生します。その場合、システムは数ミリ秒後に準備できる、システムクロックのソースオシレータを有効にします。
7. APB クロックを実行した状態で：
 - 非ユーザ・テキストによって調整された開始の見逃しフラグ (I3C_EVR レジスタの WKPF) が立てられます。発生した場合は、有効化された割込みにより、CPU をウェイクアップできます。これにより、I3C バストランザクションの見逃しがわかりますが、ターゲットがアドレス指定されているかどうかはわかりません。ソフトウェアが、このインターバルの間にコントローラによって再度アドレス指定されない場合は、STOP モードに復帰するためにタイムアウトを使用してください。

49.14 低電力モードの I3C

表 527. 低電力モードの影響

モード	説明
SLEEP	影響はありません。I3C 割込みによって、デバイスは SLEEP モードを終了します。
STOP ⁽¹⁾	STOP モードに移行するときに、I3C レジスタの内容は保持されます。 I3C ハードウェアは、専用のクロックゲートを自動的に管理し、そのカーネルクロックと APB クロックに RCC へのクロックリクエストの出力信号を生成します。I3C 割込みによって、デバイスはウェイクアップして STOP モードを終了できます。自律 DMA が STOP モードで動作可能な場合、I3C 転送が発生したり、メモリ間のデータ転送を自律 DMA で支援したりすることができます。 詳細については、 セクション 49.13 を参照してください。
STANDBY	I3C ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. STOP モードがサポートされているかどうか、またどれがサポートされているかについては、[表 515](#) を参照してください。

49.15 I3C 割込み

表 528. I3C 割込みリクエスト

項目 (割込みの略称)		割込みイベント	使用方法		割込み イネーブル： I3C_IER の フィールド	イベント フラグ： I3C_EVR の フィールド	イベントのクリア 方法： I3C_CEVCR の フィールドへの 1 の書き込み
			ハードウェア	ソフトウェア			
I3C	EVT	制御ワードをリクエスト	X	-	CFNFIE	CFNFF	-
		ステータスワードを使用可能	X	-	SFNEIE	SFNEF	-
		送信データをリクエスト	X	X	TXFNIE	TXFNFF	-
		受信データを使用可能	X	X	RXFNEIE	RXFNEF	-
		コントローラ：フレーム転送が完了 ターゲット：プライベート転送が完了	X	X	FCIE	FCF	CFCF
		ターゲットによってプライベート読み出し転送を途中終了 (I3C_CFGR レジスタで SMODE = 0)	X	-	RXTGTENDIE	RXTGTENDF	CRXTGTENDF
		IBI リクエストを受信	X	-	IBIIE	IBIF	CIBIF
		コントローラ機能リクエストを受信	X	-	CRIE	CRF	CCRF
		ホットジョイン・リクエストを受信	X	-	HJIE	HJF	CHJF
		IBI リクエストが完了	-	X	IBIENDIE	IBIENDF	CIBIENDF
		I3C バスの開始を見逃した	-	X	WKPIE	WKPF	CWKPF
		直接 GETACCR CCC を受信	-	X	CRUPDIE	CRUPDF	CCRUPDF
		直接 GETSTATUS CCC を受信	-	X	STAIE	STAF	CSTAF
		任意の直接 GETxxx CCC (GETSTATUS を除く) を受信	-	X	GETIE	GETF	CGETF
		動的なアドレスを更新 (ブロードキャスト ENTDA または RSTDA、または直接 SETNEWDA を受信)	-	X	DAUPDIE	DAUPDF	CDAUPDF
		直接 SETMWL CCC を受信	-	X	MWLUPDIE	MWLUPDF	CMWLUPDF
		直接 SETMRL CCC を受信	-	X	MRLUPDIE	MRLUPDF	CMRLUPDF
		リセット・パターンを検出	-	X	RSTIE	RSTF	CRSTF
		バス動作状態を更新 (直接/ブロードキャスト ENTASx CCC を受信)	-	X	ASUPDIE	ASUPDF	CASUPDF
		ブロードキャスト/直接 ENEC/DISEC CCC を受信	-	X	INTUPDIE	INTUPDF	CINTUPDF
	ブロードキャスト DEFTGTS CCC を受信	-	X	DEFIE	DEFF	CDEFF	
	ブロードキャスト DEFGRPA CCC を受信	-	X	GRPIE	GRPF	CGRPF	
	ERR	エラーが発生	X	X	ERRIE	ERRF	CERRF

49.16 I3C レジスタ

I3C レジスタには 32 ビットワード整列アドレスでアクセスする必要があります。

注： I3C_RDR および I3C_TDR レジスタには、それぞれ RX-FIFO の読出しと TX-FIFO の書込みに対して 1 つの最下位ビット・データ・バイトでアクセスする必要があります。

49.16.1 I3C メッセージ制御レジスタ (I3C_CR)

アドレスオフセット：0x000

リセット値：0x0000 0000

このレジスタは、I3C バスに送信するメッセージを制御するために使用する必要があります。

- I3C がコントローラとして機能する場合（ビット [30] = MTYPE[3] = 0）：送信する CCC コードがない場合、ビット [29:27] = MTYPE[2:0] は 110 とは異なります。それ以外の場合は、オルタネート・レジスタの説明 [セクション 49.16.2](#) を考慮する必要があります。
- I3C がターゲットとして機能する場合（ビット [30] = MTYPE[3] = 1）。

I3C がコントローラとして機能する場合：

- 制御 FIFO (C-FIFO) がフルではない場合 (I3C_EVR レジスタで CFNFF = 1)、このレジスタへの書込みは C-FIFO に新しい制御ワードをプッシュすることを意味します。これは、I3C_CFGR レジスタの CDMAEN で定義されているように、ソフトウェアによって、または自動的に DMA によって行われます。
- C-FIFO が空で、新しい制御ワードで再開を送信する必要がある場合、I3C ハードウェアは制御 FIFO エラーのアンダーランフラグ (I3C_SER レジスタで COVR = 1) をアサートします。I3C_IER レジスタの ERRIE = 1 で有効にした場合は、割込みが生成されます。
- フレームの最後のメッセージを完了後 (I3C_CR レジスタで MEND = 1 のメッセージ)、I3C ハードウェアはフレーム完了フラグ (I3C_EVR レジスタで FCF = 1) と、有効な場合は対応する割込みをアサートします。

I3C がターゲットとして機能する場合、このレジスタはレジスタ・モードで使用されます。

- ソフトウェアは、I3C バスでコマンド (IBI、コントローラ機能、またはホットジョイン・リクエスト) を開始するために、このレジスタに書き込みます。
- C-FIFO は無効で、また制御ワードの DMA モード也没有せん。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEND	MTYPE[3:0]					Res.	Res.	Res.	ADD[6:0]						RNW
w	w	w	w	w				w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCNT[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **MEND**：メッセージ終了タイプ/フレームの最後のメッセージ (I3C がコントローラとして機能する場合)
 0：コントローラからのこのメッセージの後に、別のメッセージの送信が必要になる前に、反復 START (Sr) が続く
 1：コントローラからのこのメッセージは、フレームの最後のメッセージであるため、STOP (P) で終了する

ビット 30:27 **MTYPE[3:0]** : メッセージタイプ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

条件 : I3C が I3C コントローラとして機能する場合

0000 : 次の制御ワードが実行されるまで、**SCL クロックを強制停止**

ビット [26:0] は無視されます。CE1 エラー検出時 (I3C_EVR レジスタで ERRF = 1、および I3C_SER レジスタで CODERR[3:0] = 0001)、開始/再開/終了は生成できないようになっています。ソフトウェアは、リカバリから「動けない」SCL に、このメッセージタイプを使用する必要があります。表 525 を参照してください。

0001 : **ヘッダメッセージ**

ビット [26:0] は無視されます。アドレス指定されたターゲットが、失敗した暫定的な GETSTATUS 後の昇格段階として、プライベート/直接メッセージに ACK で応答していない場合、ソフトウェアはこれを、I3C_CFGR レジスタで EXITPTRN = 1 によりプログラムする必要があります。これにより、ヘッダが ACK されているか NACK されているかにかかわらず (I3C バスが HDR モードであるとターゲットがみなさないようにするため)、HDR 終了パターンがバスに送信されます。昇格の処理については、表 525 と MIPI 仕様を参照してください。

0010 : **プライベート・メッセージ** (図 661 を参照)

ビット [23:17] (ADD[6:0]) は、送信された 7 ビットの動的なアドレスです。

ビット [16] (RNW) は、送信された RnW ビットです。

ビット [15:0] (DCNT[15:0]) は、プログラムされたデータバイト数です。

プライベート転送メッセージは :

- $\{S / S + 0b111_1110 + RnW = 0 + Sr / Sr^*\} + 7$ ビットの DynAddr + RnW + (8 ビットのデータ + T)* + Sr/P。
- S (START) 後、I3C_CFGR レジスタのビット NOARBH に応じて、アービトレーション可能なヘッダ (0b111_1110 + RnW = 0) が挿入されたりされなかったりします。
- $Sr^* : Sr$ (反復 START) 後、ハードウェアが P (STOP) で終わらずに前回のメッセージに続く場合は、(0b111_1110 + RnW = 0) が必要に応じて自動的に挿入されます。

0011 : **直接メッセージ (I3C SDR 直接 CCC コマンドの後半)** (図 654 を参照)

ビット [23:17] (ADD[6:0]) は、送信された 7 ビットの動的なアドレスです。

ビット [16] (RNW) は、送信された RnW ビットです。

ビット [15:0] (DCNT[15:0]) は、プログラムされたデータバイト数です。

直接転送メッセージは : $Sr + 7$ ビット DynAddr + RnW + (8 ビットのデータ + T)* + Sr/P

0100 : **従来の I²C メッセージ** (図 663 を参照)

ビット [23:17] (ADD[6:0]) は、送信された 7 ビットの静的なアドレスです。

ビット [16] (RNW) は、送信された RnW ビットです。

ビット [15:0] (DCNT[15:0]) は、プログラムされたデータバイト数です。

従来の I²C 転送メッセージは :

- $\{S / S + 0b111_1110 + RnW = 0 + Sr / Sr^*\} + 7$ ビットの StaAddr + RnW + (8 ビットのデータ + T)* + Sr/P。
- S 後、NOARBH に応じて、アービトレーション可能なヘッダ (0b111_1110 + RnW = 0) が挿入されたりされなかったりします。
- $Sr^* : Sr$ (反復 START) 後、(0b111_1110 + RnW = 0) が必要に応じて自動的に挿入されます (ハードウェアが P (STOP) で終わらずに前回のメッセージに続く場合)。

その他 : 予約済み

条件 : I3C が I3C ターゲットとして機能する場合

1000 : **ホットジョイン・リクエスト (W)** (図 665 を参照)

転送されたホットジョイン・リクエストは {S +} 0b000_0010 addr + RnW = 0 です。

ターゲットが許可されている場合 (I3C_DEVR0 レジスタで HJEN = 1)、制御ワードの書込みによってホットジョイン・リクエストが開始されます。開始リクエスト (SDA ロー) を発行し、コントローラによる SCL クロックの有効化を待つハードウェアによって、バスアイドル条件後はアクティブに、またはコントローラが同時メッセージを開始する場合はパッシブに行われます。

1001 : **コントローラ機能リクエスト (W)** (図 666 を参照)

転送されたコントローラ機能リクエストは {S +} DA[6:0] + RnW = 0 (I3C_DEVR0 レジスタの DA) です。ターゲットが許可されている場合 (I3C_DEVR0 レジスタで CREN = 1 および DAVAL = 1)、制御ワードの書込みによってコントローラ機能リクエストが開始されます。開始リクエスト (SDA ロー) を発行し、コントローラによる SCL クロックの有効化を待つハードウェアによって、バスアイドル条件後はアクティブに、またはコントローラが同時メッセージを開始する場合はパッシブに行われます。

1010 : **IBI (帯域内割込み) リクエスト (R)** (図 664 を参照)

ビット [15:0] (DCNT[15:0]) は、存在する場合は IBI データペイロード数です (最初の MDB を含む)。

転送された IBI リクエストは {S +} DA[6:0] + RnW = 1 + オプションの IBI データペイロードです。ターゲットが許可されている場合 (I3C_DEVR0 レジスタで IBIEN = 1 および DAVAL = 1)、制御ワードの書込みによって IBI リクエストが開始されます。開始リクエスト (SDA ロー) を発行し、コントローラによる SCL クロックの有効化を待つハードウェアによって、バスアイドル条件後はアクティブに、またはコントローラが同時メッセージを開始する場合はパッシブに行われます。

コントローラから確認応答した場合、送信された IBI ペイロード・データ (I3C_BCR レジスタの BCR2 に応じてオプション) は I3C_CR レジスタおよび I3C_IBIDR の DCNT[15:0] によって定義され、I3C_IBIDR レジスタの IBIP[2:0] で定義された IBI ペイロード・データ・サイズに合わせてプログラムする必要があります。

その他 : 予約済み

ビット 26:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 **ADD[6:0]** : 7 ビットの I3C の動的なターゲットアドレス/I²C の静的なターゲットアドレス (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフィールドは MTYPE[3:0] = 0010 (プライベート・メッセージ)、または MTYPE[3:0] = 0011 (直接メッセージ)、または MTYPE[3:0] = 0100 (従来の I²C メッセージ) の場合に使用されます。

ビット 16 **RNW** : 読出し/非書込みメッセージ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフィールドは MTYPE[3:0] = 0010 (プライベート・メッセージ)、または MTYPE[3:0] = 0011 (直接メッセージ)、または MTYPE[3:0] = 0100 (従来の I²C メッセージ) の場合に、I3C バスで RnW ビットを送信するために使用されます。

0 : 書込みメッセージ

1 : 読出しメッセージ

ビット 15:0 **DCNT[15:0]** : 読出し/書込みメッセージ中に転送されるバイト単位のデータ数 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

I3C がコントローラとして機能する場合、このフィールドは MTYPE[3:0] = 0010 (プライベート・メッセージ)、または MTYPE[3:0] = 0011 (直接メッセージ)、または MTYPE[3:0] = 0100 (従来の I²C メッセージ) の場合にバスに交換したデータバイト数をセットするために使用されます。プライベート・メッセージ、または従来の I²C 読出し/書込みメッセージの場合、このフィールドは非 null である必要があります。

I3C がターゲットとして機能する場合、このフィールドは MTYPE[3:0] = 1010 (IBI リクエスト) で、IBI データ・ペイロード (I3C_BCR レジスタで BCR2 = 1 の場合に送信されるデータ) の場合に、IBI データ・ペイロードのバイト数 (1、2、3、または 4) をセットするために使用されます。

最大 64 KB - 1 のリニア・エンコード

0x0000 : 転送データなし

0x0001 : 1 バイト

0x0002 : 2 バイト

.....

0xFFFF : 64 KB - 1 バイト

49.16.2 I3C メッセージ制御レジスタ [オルタネート] (I3C_CR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

コントローラが CCC (CCC ブロードキャスト、CCC 直接、または CCC 入力 HDR など、CCC のタイプにかかわらず) を送信しなければならない場合、メッセージを制御するには、この書込みレジスタの説明を使用する必要があります。

これは、レジスタ I3C_CR が MTYPE[3:0] = 0110 の場合の代替説明です。それ以外の場合は、[セクション 49.16.1](#) を参照してください。

制御 FIFO (C-FIFO と呼ばれる) がフルではない場合 (I3C_EVR レジスタで CFNFF = 1)、このレジスタへの書込みは C-FIFO に新しい制御ワードをプッシュすることを意味します。これは、I3C_CFGR レジスタの CDMAEN ビットで定義されているように、ソフトウェアによって、または自動的に DMA によって行われます。

フレームの最後のメッセージを完了後 (I3C_CR レジスタで MEND = 1 のメッセージ)、I3C ハードウェアはフレーム完了フラグ (I3C_EVR レジスタで FCF = 1) と、有効な場合は対応する割込みをアサートします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEND	MTYPE[3:0]					Res.	Res.	Res.	CCC[7:0]						
w	w	w	w	w				w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCNT[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **MEND** : メッセージ終了タイプ/フレームの最後のメッセージ (I3C がコントローラとして機能する場合)
 0 : コントローラからのこのメッセージの後に、別のメッセージの送信が必要になる前に、反復 START (Sr) が続く
 1 : コントローラからのメッセージは、フレームの最後のメッセージであるため、STOP (P) で終了する

ビット 30:27 **MTYPE[3:0]** : メッセージタイプ (I3C がコントローラとして機能する場合)

条件 : I3C が I3C コントローラとして機能する場合

0110 : ブロードキャスト/直接 CCC コマンド (表 524、図 654、図 655、図 656 を参照)

ビット [23:16] (CCC[7:0]) は、送信された 8 ビットの CCC コードです。

ビット [15:0] (DCNT[15:0]) は、CCC 定義バイト数、または CCC サブコマンドバイト数、または CCC データバイト数です。

ビット [23] = CCC[7] = 1 の場合 : I3C SDR 直接 CCC コマンドの前半。

転送された直接 CCC コマンド (前半) のメッセージは :

- {S / S + 0b111_1110 + RnW = 0 / Sr+*} + (直接 CCC + T) + (8 ビットのデータ + T)* + Sr
- S (START) 後、I3C_CFGR レジスタの NOARBH に応じて、アービトレーション可能なヘッダ (0b111_1110 + RnW = 0) が挿入されたりされなかったりします。
- Sr+* : Sr (反復 START) 後、ハードウェアは自動的に (0b111_1110 + R/W) を挿入します。

ビット [23] = CCC[7] = 0 の場合 : I3C SDR ブロードキャスト CCC コマンド (特定の ENTDAAs を含む、図 655 を参照)

転送されたブロードキャスト CCC コマンドのメッセージは :

- {S / S + 0b111_1110 + RnW = 0 / Sr+*} + (ブロードキャスト CCC + T) + (8 ビットのデータ + T)* + Sr/P
- S (START) 後、NOARBH に応じて、アービトレーション可能なヘッダ (0b111_1110 + RnW = 0) が挿入されたりされなかったりします。
- Sr+* : Sr (反復 START) 後、ハードウェアは自動的に (0b111_1110 + R/W) を挿入します。

その他 : 予約済み

ビット 26:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **CCC[7:0]** : 8 ビット CCC コード (I3C がコントローラとして機能する場合)

ビット [23] = CCC[7] = 1 の場合 : I3C SDR 直接 CCC コマンドの前半。

ビット [23] = CCC[7] = 0 の場合 : I3C SDR ブロードキャスト CCC コマンド (ENTDAAs を含む)。

ビット 15:0 **DCNT[15:0]** : CCC 定義バイト、または CCC サブコマンドバイト、または CCC データバイトとして転送する、バイト単位の CCC コマンドの関連データ数です。

最大 64 KB - 1 のリニア・エンコード

0x0000 : 転送データなし

注 : ENTDAAs ブロードキャスト CCC 送信時に必須の値 (図 655 を参照)。

0x0001 : 1 バイト

注 : RSTACT 直接/ブロードキャスト CCC 送信時に必須の値 (図 656 を参照)。

0x0002 : 2 バイト

.....

0xFFFF : 64 KB - 1 バイト

49.16.3 I3C 設定レジスタ (I3C_CFGR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

このレジスタは、以下を設定するために使用されます。

- I3C がコントローラまたはターゲットとして機能する場合に適用される機能 : RX-FIFO および TX-FIFO の管理 (RXDMAEN、RXTHRES、RXFLUSH、TXDMAEN、TXTHRES、TXFLUSH)、I3C ペリフェラルの役割 (CRINIT)
- I3C がコントローラとして機能する場合の専用機能 : フレームベースの制御ワードのトリガ (TSFSET)、FIFO 管理 (TMODE、SMODE、SFLUSH、SDMAEN、CDMAEN)、およびその他 (HJACK、HKSDAEN、EXITPTRN、RSTPTRN、NOARBH)

設定フィールド CRINIT、HKSDAEN は、EN = 0 の場合にのみ変更できます。この条件は、EN が 1 にセットされているときに、同時に変更された場合は順守されます (別の書き込み操作時に、後で EN をセットする必要はありません)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TSFSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFLUSH	CDMAEN	TMODE	SMODE	SFLUSH	SDMAEN
	w									w	rw	rw	rw	w	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXTHRES	TXFLUSH	TXDMAEN	Res.	RXTHRES	RXFLUSH	RXDMAEN	HJACK	Res.	HKSDAEN	EXITPTRN	RSTPTRN	いいえ ARBH	CRINIT	EN
	rw	w	rw		rw	w	rw	rw		rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **TSFSET** : フレーム転送のセット (ソフトウェアトリガ) (I3C がコントローラとして機能する場合)

このビットは、書き込みのみ可能です。I3C が I3C コントローラとして機能する場合 :

0 : クリアなし

1 : このビットをセットすると、ハードウェアによって I3C_EVR レジスタのフラグ CFNFF がアサートされ、フレーム転送が開始される (C-FIFO はフルではなく、制御ワードが必要)

注 : このビットがセットされていない場合、ソフトウェアはフレーム転送を開始するために、代替方法として C-FIFO が空 (I3C_EVR レジスタで CFEF = 1) のうちに、最初の制御ワードレジスタ (I3C_CR) を直接書き込むことができます。その後、最初に書き込まれた制御ワードにメッセージ終了 (I3C_CR レジスタで MEND = 0) のタグが付けられていない場合は、ハードウェアによって CFNFF がアサートされます。

ビット 29:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **CFLUSH** : C-FIFO の一掃 (I3C がコントローラとして機能する場合)

このビットは、書き込みのみ可能です。

0 : クリアなし

1 : C-FIFO を一掃する

ビット 20 **CDMAEN** : C-FIFO DMA リクエストイネーブル (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合 :

0 : C-FIFO の DMA モードを無効化

- ソフトウェアは、必要に応じて特定のフレームで制御ワードを C-FIFO に書き込んでプッシュします (I3C_CR レジスタを書き込む)。
- 次の制御ワード転送は、I3C_EVR レジスタのフラグ CFNFF = 1 でのポーリング、または割り込み通知 (I3C_IER レジスタで CFNFIE = 1 にすることで有効化) により、ソフトウェアが書き込むことができます。

1 : C-FIFO の DMA モードを有効化

- DMA は、必要に応じて特定のフレームで制御ワードを C-FIFO に書き込んでプッシュします (I3C_CR レジスタを書き込む)。
- 次の制御ワード転送は、プログラムされたハードウェアによって自動的に書き込まれます (I3C およびプログラムされた DMA チャンネルからアサートされた C-FIFO DMA リクエストによって)。

ビット 19 **TMODE** : 送信モード (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、C-FIFO と TX-FIFO の管理と I3C バスに送信されたフレームに使用されます。

0 : C-FIFO と TX-FIFO は、フレーム転送の送信を開始する前にプリロードされません。

フレーム転送は、C-FIFO に最初の制御ワードが出現するとすぐに開始されます。

1 : C-FIFO と TX-FIFO は、フレーム転送の送信を開始する前に、最初にプリロードされます。詳細については、[セクション 49.10.2](#) を参照してください。

ビット 18 **SMODE** : S-FIFO の有効化/ステータス受信モード (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは I3C バスで交換されたメッセージのステータス FIFO (S-FIFO) を有効にするために使用されます。

I3C がターゲットとして機能する場合、このビットはクリアする必要があります。

0 : S-FIFO は無効です。

- ステータスレジスタ (I3C_SR) は FIFO メカニズムなしで使用されます。
- 新しいステータスレジスタの内容が読み出されていない場合、SCL はストールされません。
- ステータスレジスタは、ハードウェアによって上書きされる前に読み出す必要があります。
- I3C_CFGR レジスタで SDMAEN = 0 である必要があります。

1 : S-FIFO は有効です。

- 各メッセージステータスを読み出す必要があります。
- S-FIFO がフルで次のメッセージステータスを読み出す必要がある場合、SCL はストールされます。
- 最大 SCL クロックのストール時間の後に、S-FIFO オーバーランエラーが報告されます。

ビット 17 **SFLUSH** : S-FIFO の一掃 (I3C がコントローラとして機能する場合)

このビットは、I3C がコントローラとして機能する場合にのみ書き込みおよび使用可能です。

0 : クリアなし

1 : S-FIFO を一掃する

ビット 16 **SDMAEN** : S-FIFO DMA リクエストイネーブル (I3C がコントローラとして機能する場合)

このビットは、I3C_CFGR レジスタで SMODE = 0 (S-FIFO は無効) の場合にクリアする必要があります。つまり、S-FIFO が無効な場合、DMA モードは使用できません。これにより、ステータスレジスタ I3C_SR は、読み出しできたり、できなかったりします。

このビットは、SMODE = 1 (S-FIFO は有効) の場合にセットまたはクリアできます。つまり、ソフトウェアまたは割り当てられた DMA チャンネルによって、各メッセージについてステータスレジスタ I3C_SR を読み出す必要があります。

0 : 読出しステータスレジスタ I3C_SR の DMA モードは無効です。

- SMODE = 0 : ソフトウェアは、フレーム完了後 (I3C_EVR レジスタで FCF = 1) またはエラー発生後 (I3C_EVR レジスタで ERRF = 1) に I3C_SR レジスタを読み出すことができます。これらのレジスタフラグでのポーリング、または割り込み通知を使用 (I3C_IER レジスタで FCIE = 1 および ERRIE = 1)。

- SMODE = 1 : ソフトウェアは、アサートされた各フラグ SFNEF = 1 後に、S-FIFO (I3C_SR レジスタの読出し) からステータスワードを読み出すかポップする必要があります。このレジスタフラグでのポーリング、または割り込み通知を使用 (I3C_IER レジスタで SFNEIE = 1)。

1 : 読出しステータスレジスタ I3C_SR の DMA モードは有効です。

- I3C_CFGR レジスタで SMODE = 1 である必要があります (S-FIFO は有効)。

- DMA は S-FIFO からステータスワードを読み出してポップします (I3C_SR レジスタを読み出します)。

- ステータスワードは、プログラムされたハードウェアによって自動的に読み出されます (I3C およびプログラムされた DMA チャンネルからアサートされた S-FIFO DMA リクエストによって)。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TXTHRES** : TX-FIFO 閾値 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

この閾値は、I3C_EVR レジスタに TXFNFF フラグがセットされている場合に、TX-FIFO レベルと比較して定義します (結果的に、DMA TX リクエストがアサートされていて TXDMAEN = 1 である場合)。

0 : 1 バイト閾値

TXFNFF は、(I3C_TDR の) TX-FIFO に 1 バイトを書き込む必要がある場合にセットされます。

1 : 1 ワード/4 バイト閾値

TXFNFF は、(I3C_TDWR レジスタの) TX-FIFO に 1 ワード/4 バイトを書き込む必要がある場合にセットされます。最後の送信データ数が 4 バイトの倍数でない場合 (I3C_SR レジスタで XDCNT[1:0] = 00)、ハードウェアは最後のワードの関連する 1、2、または 3 の有効な最下位ビット・バイトのみが考慮され、I3C バスで送信されます。

ビット 13 **TXFLUSH** : TX-FIFO の一掃 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このビットは、書き込みのみ可能です。

I3C がターゲットとして機能する場合、このビットはコントローラがデータ読出しをアボートした場合 (T ビットでローに駆動)、また残りのデータが TX-FIFO にある場合 (ABT = 1、および I3C_SR レジスタで XDCNT[15:0] < I3C_TGTTDR レジスタで TGTTDCNT[15:0]) に、プライベート読出しで TX-FIFO を一掃するために使用できます。

0 : クリアなし

1 : TX-FIFO を一掃する

ビット 12 **TXDMAEN** : TX-FIFO DMA リクエストイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : TX-FIFO の DMA モードを無効化

- ソフトウェアは、TX-FIFO に I3C バス経由で送信されるデータバイト/ワードを書き込んでプッシュします (I3C_TDR または I3C_TDWR レジスタに書き込みます)。

- 次のデータバイト/ワードは、フラグ TXFNFF = 1 フラグでのポーリング、または割り込み通知 (TXFNIE = 1 で有効化) を使用して、ソフトウェアで書き込む必要があります。

1 : TX-FIFO の DMA モードを有効化

- DMA は、TX-FIFO にデータバイト/ワードを書き込んでプッシュします (I3C_TDR または I3C_TDWR レジスタに書き込みます)。

- 次のデータバイト/ワード転送は、プログラムされたハードウェアによって自動的にプッシュされます (I3C およびプログラムされた DMA チャンネルからアサートされた TX-FIFO DMA リクエストによって)。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **RXTHRES** : RX-FIFO 閾値 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

この閾値は、I3C_EVR レジスタに RXFNEF フラグがセットされている場合に、RX-FIFO レベルと比較して定義します (結果的に、DMA RX リクエストがアサートされていて RXDMAEN = 1 である場合)。

0 : 1 バイト閾値

RXFNEF は、(I3C_RDR レジスタの) RX-FIFO に 1 バイトを書き込む必要がある場合にセットされます。

1 : 1 ワード/4 バイト閾値

RXFNEF は、(I3C_RDWR) RX-FIFO で 1 ワード/4 バイトが読み出される場合にセットされます。最後の受信データが 4 バイトの倍数でない場合、ソフトウェアは最後のワードの関連する 1、2、または 3 の有効な最下位ビット・バイトのみが考慮されます。有効な受信データバイト数は I3C_SR レジスタの XDCNT[15:0] によって報告されます。

ビット 9 **RXFLUSH** : RX-FIFO の一掃 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このビットは、書込みのみ可能です。

0 : クリアなし

1 : RX-FIFO を一掃する

ビット 8 **RXDMAEN** : RX-FIFO DMA リクエストイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : RX-FIFO の DMA モードを無効化

- ソフトウェアは RX-FIFO からのデータバイト/ワードを読み出してポップします (I3C_RDR または I3C_RDWR レジスタで読み出します)。

- 次のデータバイト/ワードは、I3C_EVR レジスタのフラグ RXFNEF = 1 でのポーリング、または割り込み通知 (I3C_IER レジスタで RXFNEIE = 1 にすることで有効化) により、ソフトウェアで読み出す必要があります。

1 : RX-FIFO の DMA モードを有効化

- DMA は RX-FIFO からのデータバイト/ワードを読み出してポップします (I3C_RDR または I3C_RDWR レジスタで読み出します)。

- 次のデータバイト/ワードは、プログラムされたハードウェアによって自動的に読み出されます (I3C およびプログラムされた DMA チャンネルからアサートされた RX-FIFO DMA リクエストによって)。

ビット 7 **HJACK** : ホットジョイン・リクエスト確認応答 (I3C がコントローラとして機能する場合)

0 : ホットジョイン・リクエストは確認応答されない

NACK された後で、コントローラは最初にプログラムされたとおりに続行します (ホットジョインを行うターゲットは NACK を認識しており、後で別のホットジョイン・リクエストを送信する必要があります)。

1 : ホットジョイン・リクエストが確認応答される

ACK された後で、コントローラは最初にプログラムされたとおりに続行します。ソフトウェアは、HJ 割り込みによって通知され (I3C_EVR レジスタでフラグ HJF がセットされる)、後で ENTDAА シーケンスを開始する必要があります。これにより、ターゲット・イベントの無効化コマンド (DISHJ = 1 の DISEC) を含む他のホットジョイン・リクエストを防ぐことができます。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **HKSDAEN** : ハイキーパーを SDA ライン上で有効化 (I3C がコントローラとして機能する場合)

0 : ハイキーパーは無効です。

1 : ハイキーパーは有効です。T ビットではオープン・ドレイン・クラス・プルアップの代わりにウィークプルアップが有効になります。

注 : このビットは、I3C_CFGR レジスタで EN = 0 の場合にのみ変更できます。

ビット 4 EXITPTRN : HDR 終了パターンの有効化 (I3C がコントローラとして機能する場合)

このビットは、処理中のフレームがない場合にのみ変更できます。

0 : HDR 修正パターンは、発行されたメッセージヘッダの後には送信されません (I3C_CR レジスタで MTYP[3:0] = 0001)。これは、コントローラ機能のハンドオフ後に問題が疑われる場合に、バスの所有権を試すために、ヘッダを送信するために使用されます (新しいコントローラでは、動作状態で定義された遅延未満で以前のコントローラ機能にアクセスしているため、コントローラ機能はアサートされません)。

1 : HDR 修正パターンは、発行されたメッセージヘッダの後には送信されます (MTYP[3:0] = 0001)。これは、プライベート・メッセージや直接読み出し CCC にターゲットが応答しない場合にコントローラのエラー検出と昇格の処理に使用されています。

HDR 終了パターンは、ACK または NACK された任意のメッセージヘッダ {S/Sr + 0x7E addr + W} に送信されます。

ビット 3 RSTPTRN : HDR リセットパターンの有効化 (I3C がコントローラとして機能する場合)

このビットは、処理中のフレームがない場合にのみ変更できます。

0 : フレーム終了時に標準 STOP が送信されます。

1 : HDR リセットパターンは、RSTACT CCC コマンドを含む送信されたフレームを停止する前に挿入されます。

ビット 2 NOARBH : 開始後のアービトレーション可能なヘッダなし (I3C がコントローラとして機能する場合)

このビットは、処理中のフレームがない場合にのみ変更できます。

0 : アービトレーション可能なヘッダ (0b111_1110 + RnW = 0) は、開始後、従来の I²C メッセージまたは I3C SDR プライベート読み出し/書き込みメッセージ (デフォルト) の前に送信されます。

1 : アービトレーション可能なヘッダなし。

- 従来の I²C メッセージまたは I3C SDR プライベート読み出し/書き込みメッセージの場合、ターゲットアドレスは開始直後に送信されます。

- これはより有益なオプションです (0x7E のアービトレーション可能なヘッダでの送信が役に立たない場合)。ただし、アドレス指定されたターゲットデバイスが IBI またはコントローラ機能リクエストを同時に送信できないことを、コントローラが確認した場合にのみ使用されます (IBI または MR について、コントローラがオープンドレインモードで送信したアドレスと、ターゲットデバイスが開始後に送信できる同じアドレスの間で、解釈に誤りがなく、競合の可能性がないかを確認するため)。

ビット 1 CRINIT : コントローラ/ターゲットの最初の機能

このビットは、I3C_CFGR レジスタで EN = 0 の場合にのみ変更できます。

0 : ターゲット機能

EN = 1 をセットして有効にすると、ペリフェラルは最初にターゲットとして機能します。I3C は、最終的にコントローラ機能を取得するまでは、SCL ラインを駆動せず、SDA プルアップを有効にしません。

1 : コントローラ機能

EN = 1 をセットして有効にすると、ペリフェラルは最初にコントローラとして機能します。I3C コントローラ機能により、最終的に I3C の二次コントローラにコントローラ機能を提供するまで、SCL ラインを駆動して SDA プルアップを有効にします。

ビット 0 EN : I3C は有効です (I3C がコントローラ/ターゲットのどちらで機能する場合でも)。

0 : I3C は無効です。

- レジスタを除き、ペリフェラルはリセットされます (部分リセット)。

- EN をクリアする前に、I3C がコントローラとして機能する場合、DISEC CCC を使用してすべてのターゲットリクエストを無効にする必要があります。

- I3C がターゲットとして機能する場合、部分リセットが必要でない限り、ソフトウェアは I3C を無効にしてはなりません。

1 : I3C は有効です。

この状態では、一部のレジスタフィールドは変更できません (I3C_CFGR の CRINIT、HKSDAEN など)

49.16.4 I3C 受信データバイトレジスタ (I3C_RDR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

このレジスタは、RX-FIFO がバイトベースの読出しアクセス (I3C_CFGR レジスタで RXTHRES = 0) で設定されている場合に、I3C バスから受信データを読み出すために使用されます。その後、

- 読出し時、I3C_RDR は、最下位ビット位置にある受信データバイトと 32 ビットのデータワード、および 0 として読み出されたその他の予約済みビットを返します。
- I3C_CFGR レジスタで RXDMAEN = 1 の場合：プログラムされた I3C および DMA は、関連する連続した読出しをハードウェアによって自動的に管理します。
- RXDMAEN = 0 の場合：読み出す前に、ソフトウェアは、次の受信データバイトを RX-FIFO の非エンプティフラグ (I3C_EVR レジスタで RXFNEF = 1) または対応する割込み (有効な場合) で読み出す必要があることを通知されるまで、待つ必要があります。読み出すべき特定のメッセージの最後の受信バイトがコントローラとして機能する場合は、I3C_EVR レジスタで RXLASTF = 1 としてもマークされます。
- RX-FIFO がフルで、新しいバイトを受信し、待たずに RX-FIFO にプッシュできなくなった場合、ソフトウェアには、I3C_EVR レジスタのエラーフラグ ERRF = 1 および I3C_SER レジスタのデータオーバーランフラグ DOVRF = 1 (また有効な場合は対応する割込み) によって通知されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDB0 [7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 RDB0[7:0] : I3C バスでの 8 ビット受信データ。

49.16.5 I3C 受信データワードレジスタ (I3C_RDWR)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

このレジスタは、RX-FIFO が 32 ビットワードベースの読出しアクセス (I3C_CFGR レジスタで RXTHRES = 1) で設定されている場合に、I3C バスから受信データを読み出すために使用されます。その後、

- I3C_CFGR レジスタで RXDMAEN = 1 の場合：プログラムされた I3C および DMA は、関連する連続した読出しをハードウェアによって自動的に管理します。
- RXDMAEN = 0 の場合：読み出す前に、ソフトウェアは最初に、次の受信データワードを RX-FIFO の非エンプティフラグ (I3C_EVR レジスタで RXFNEF = 1) または対応する割込み (有効な場合) で読み出す必要があることを通知されるまで、待つ必要があります。読み出すべき特定のメッセージの最後の受信ワード/バイトがコントローラとして機能する場合は、I3C_EVR レジスタで RXLASTF = 1 としてもマークされます。
- RX-FIFO が保持しているバイト数が 4 バイト未満の場合、I3C_RDWR を読み出すことで、null バイトでパディングされたデータワードが返されます。最下位ビット位置で使用可能なバイト数は、最上位ビット位置でゼロバイトでパディングされます。

- RX-FIFO がフルで、新しいバイトを受信し、待たずに RX-FIFO にプッシュできなくなった場合、ソフトウェアには、I3C_EVR レジスタのエラーフラグ ERRF = 1 および I3C_SER レジスタのデータオーバーラン DOVR = 1 (また有効な場合は対応する割込み)によって通知されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB3 [7:0]								RDB2 [7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB1 [7:0]								RDB0 [7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **RDB3[7:0]** : 8 ビットの受信データ (I3C バスの最新バイト)。

ビット 23:16 **RDB2[7:0]** : 8 ビットの受信データ (I3C バスの RDB1 の次のバイト)。

ビット 15:8 **RDB1[7:0]** : 8 ビットの受信データ (I3C バスの RDB0 の次のバイト)。

ビット 7:0 **RDB0[7:0]** : 8 ビットの受信データ (I3C バスの一番早いバイト)。

49.16.6 I3C 送信データバイトレジスタ (I3C_TDR)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタは、I3C バス経由で送信されるデータバイトを書き込むために使用されます。

このレジスタは、送信 FIFO (TX-FIFO) へのバイトベースの書き込みアクセスを実装します。これは、I3C_CFGR レジスタで TXTHRES = 0 の場合に使用されます。

I3C がコントローラとして機能する場合 :

- I3C_CFGR レジスタで TXDMAEN = 1 および TXTHRES = 0 の場合 : プログラムされた I3C および DMA は、関連する連続した書き込みをハードウェアによって自動的に管理します。
- TXDMAEN = 0 および TXTHRES = 0 の場合 : 書き込む前に、ソフトウェアは、次のデータバイトを TX-FIFO の非フルフラグ (I3C_EVR レジスタで TXFNFF = 1) または対応する割込み (有効な場合) で書き込む必要があることを通知されるまで、待つ必要があります。書き込むべき特定のメッセージの最後の送信バイトが I3C_EVR レジスタで TXLASTF = 1 としてもマークされます。

I3C がターゲットとして機能する場合 :

- I3C_TGTTDR レジスタで TXDMAEN = 1 のときに、TXTHRES = 0 で PRELOAD = 1 の場合 : プログラムされた I3C および DMA は、I3C_TDR レジスタへの関連する連続した書き込み数を、I3C_TGTTDR レジスタの TGTTDCNT[15:0] に従って、ハードウェアによって自動的に管理します。
- TXDMAEN = 0 で TXTHRES = 0 の場合 :
 - I3C_TGTTDR レジスタで PRELOAD = 1 の場合 : 書き込む前に、ソフトウェアは、次のデータバイトを TX-FIFO の非フルフラグ (TXFNFF = 1) または対応する割込み (有効な場合) で書き込む必要があることを通知されるまで、待つ必要があります。最初にプログラムされた最後の送信バイト TGTTDCNT[15:0] は I3C_EVR レジスタで TXLASTF = 1 としてもマークされます。
 - PRELOAD = 0 の場合 : 書き込む前に、ソフトウェアは、最大 8 個の次のデータバイト (TX-FIFO サイズ) を TX-FIFO のエンプティフラグ (I3C_EVR レジスタで TXFEF = 1) または対応する割込み (有効な場合) で書き込めることが通知されるまで、待つ必要があります。ソフトウェアが別のデータバイトの書き込みを必要とする場合は、TX-FIFO が空 (TXFEF = 1) になるまで待つ必要があります。送信されるこのデータバイトは、データの

アンダーラン・エラー・フラグを回避するため、SCL クロック周期が 9 回経過するまでに書き込みます (I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1)。

TX-FIFO が空で、コントローラが送信されるデータバイトをこれ以上待てない場合、ソフトウェアには I3C_EVR レジスタのエラーフラグ ERRF = 1 によって、また I3C_SER レジスタのデータのアンダーランフラグ DOF = 1 (および有効な場合は対応する割込み) によって通知されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDB0 [7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 TDB0[7:0] : I3C バスで送信する 8 ビットデータ。

49.16.7 I3C 送信データワードレジスタ (I3C_TDWR)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

このレジスタは、I3C バス経由で送信される 32 ビットデータワードを書き込むために使用されます。

このレジスタは、送信 FIFO (TX-FIFO) へのワードベースの書き込みアクセスを実装します。これは、I3C_CFGR レジスタで TXTHRES = 1 の場合に使用されます。

I3C がコントローラとして機能する場合 :

- I3C_CFGR レジスタで TXDMAEN = 1 および TXTHRES = 1 の場合、プログラムされた I3C および DMA は、関連する連続した書き込みをハードウェアによって自動的に管理します。
- TXDMAEN = 0 および TXTHRES = 1 の場合 : 書き込む前に、ソフトウェアは、次のデータワード/バイトを TX-FIFO の非フルフラグ (I3C_EVR レジスタで TXFNFF = 1) または対応する割込み (有効な場合) で書き込む必要があることを通知されるまで、待つ必要があります。TX-FIFO に書き込むべき特定のメッセージの最後の送信ワード/バイトが I3C_EVR レジスタで TXLASTF = 1 としてもマークされます。

I3C がターゲットとして機能する場合 :

- I3C_TGTTDR レジスタで TXDMAEN = 1 のときに、TXTHRES = 1 で PRELOAD = 1 の場合 : プログラムされた I3C および DMA は、I3C_TDWR レジスタへの関連する連続した書き込み数を、I3C_TGTTDR レジスタの TGTTDCNT[15:0] に従って、ハードウェアによって自動的に管理します。
- TXDMAEN = 0 で TXTHRES = 1 の場合 :
 - PRELOAD = 1 の場合 : 書き込む前に、ソフトウェアは、次のデータワードを TX-FIFO の非フルフラグ (I3C_EVR レジスタで TXFNFF = 1) または対応する割込み (有効な場合) で書き込む必要があることを通知されるまで、待つ必要があります。I3C_TGTTDR レジスタで最初にプログラムされた最後の送信ワード TGTTDCNT[15:0] は I3C_EVR レジスタで TXLASTF = 1 としてもマークされます。
 - PRELOAD = 0 の場合 : 書き込む前に、ソフトウェアは、最大 2 個の次のデータワード (TX-FIFO サイズ) を TX-FIFO のエンプティフラグ (I3C_EVR レジスタで TXFEF = 1) または対応する割込み (有効な場合) で書き込めることが通知されるまで、待つ必要があります。ソフトウェアが別のデータワードの書き込みを必要とする場合は、TX-FIFO が空 (TXFEF = 1) になるまで待つ必要があります。送信される次のデータワードは、データの

アンダーラン・エラー・フラグを回避するため、SCL クロック周期が 9 回経過するまでに書き込みます (I3C_EVR レジスタで ERRF = 1 および I3C_SER レジスタで DOVR = 1)。

TX-FIFO が空で、コントローラ/ターゲットが送信されるデータバイトをこれ以上待てない場合、ソフトウェアには I3C_EVR レジスタのエラーフラグ ERRF = 1 によって、また I3C_SER レジスタのデータのアンダーランフラグ DOF = 1 (および有効な場合は対応する割込み) によって通知されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDB3 [7:0]								TDB2 [7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDB1 [7:0]								TDB0 [7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

- ビット 31:24 **TDB3[7:0]** : 8 ビットの送信データ (I3C バスの最新バイト)。
- ビット 23:16 **TDB2[7:0]** : 8 ビットの送信データ (I3C バスの TDB1[7:0] の次のバイト)。
- ビット 15:8 **TDB1[7:0]** : 8 ビットの送信データ (I3C バスの TDB0[7:0] の次のバイト)。
- ビット 7:0 **TDB0[7:0]** : 8 ビットの送信データ (I3C バスの一番早いバイト)。

49.16.8 I3C IBI ペイロード・データ・レジスタ (I3C_IBIDR)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

このレジスタは、IBI ペイロード・データに使用されます。

I3C がターゲットとして機能する場合 :

- I3C_BCR レジスタで BCR2 = 0 の場合、このレジスタは不要です。
- BCR2 = 1 の場合、このレジスタはソフトウェアによって書き込まれなければなりません。
 - IBI ペイロードとして I3C バスに送信するため。その後、IBI リクエスト (I3C_CR レジスタで MTTYPE[3:0] = 1010) はコントローラによって確認応答されます。IBI データのペイロード・サイズは、I3C_CR レジスタの DCNT[15:0] により定義されます。
 - 最大の (静的) ペイロード・データ・サイズは I3C_MAXRLR レジスタの IBIP[2:0] で与えられます。1、2、3、または 4 バイトのいずれかです。
 - DCNT[15:0] は 1 (必須のデータバイト MDB[7:0] の場合) と最大 IBIP[2:0] の間にセットする必要があります。

I3C がコントローラとして機能する場合 : I3C_DEVRx レジスタ で IBIAACK = 1 の場合は、I3C バスでターゲット x からの IBI リクエストを確認応答したとき :

- I3C_DEVRx レジスタで IBIDEN = 0 (ターゲット x から BCR[2] = 0 を受信) の場合、このレジスタは不要です。
- IBIDEN = 1 の場合 (ターゲット x から受信した BCR[2] = 1) :
 - このレジスタは、I3C バスで受信した IBI ペイロード・データから、ハードウェアによって内部的に書き込まれます (これには、最初の必須のデータバイト MDB[7:0] も含まれます)。
 - ペイロードの最後のバイトを I3C_IBIDR で受信した場合 (ターゲットが T ビット = 0 で駆動)、I3C_EVR レジスタの IBIF フラグ (および有効な場合は対応する割込み) がセットされます。
 - その後、ソフトウェアは、受信され、記録された I3C_RMR レジスタの RADD[6:0] の 7 ビットアドレスにより、ターゲット x を識別できます。
 - ソフトウェアは、I3C_RMR レジスタの IBIRDCNT[2:0] で効果的に受信され、記録されたバイト数に従って、このレジスタと割込みバイトを読み出すことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBIDB3 [7:0]								IBIDB2 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIDB1 [7:0]								IBIDB0 [7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **IBIDB3[7:0]** : 8 ビットの IBI ペイロード・データ (I3C バスの最新バイト)。

ビット 23:16 **IBIDB2[7:0]** : 8 ビットの IBI ペイロード・データ (I3C バスの IBIDB1[7:0] 後の最新バイト)。

ビット 15:8 **IBIDB1[7:0]** : 8 ビットの IBI ペイロード・データ (I3C バスの IBIDB0[7:0] 後の最新バイト)。

ビット 7:0 **IBIDB0[7:0]** : 8 ビットの IBI ペイロード・データ (I3C バスで一番早いバイト、MDB[7:0] の必須のデータバイト)。

49.16.9 I3C ターゲット送信設定レジスタ (I3C_TGTTDR)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは TX-FIFO のデータバイト数をプリロードするために使用する必要があります。これは、使われている DMA モードにかかわらず (I3C_CFGR レジスタの TXDMAEN とは別)、コントローラによって許可され、確認応答された場合に、I3C バスで送信できるようにするためです。

I3C がターゲットとして機能する場合、別の方法として、送信するデータバイト数が TX-FIFO サイズ (8 バイト) 以下の場合、ソフトウェアは、TX-FIFO エンプティフラグ (I3C_EVR レジスタで TXFEF = 1) をポーリングすることで、または対応する有効化された割込みによって、直接 I3C_TDR レジスタ (または I3C_CFGR レジスタの TXTHRES によっては I3C_TDWR レジスタ) を使用して、バイトを書き込むことができます。

どの場合でも、I3C ターゲットでは SCL ラインをストレッチ/ストールできないため、ソフトウェアは I3C_EVR レジスタで DOR = 1 にすることで TX-FIFO アンダーランを識別できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRE LOAD
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGTTDCNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **PRELOAD** : TX-FIFO のプリロード (I3C がターゲットとして設定されている場合)

このビットは、TX-FIFO にプリロードし、送信するバイト数を書き込まれ、定義されたときと同じアクセスで、ソフトウェアによって書き込まれ、アサートされなければなりません。

このビットは、送信するすべてのデータバイトが TX-FIFO にロードされると、ハードウェアによってクリアされます。

0 : TX-FIFO プリロードなし

1 : TX-FIFO プリロード

ビット 15:0 **TGTTDCNT[15:0]** : バイト単位の送信データカウンタ (I3C がターゲットとして設定されている場合)

このビットフィールドは、プリロードし、送信するバイト数を定義するために、PRELOAD がアサートされたときと同じアクセスで、ソフトウェアによって書き込まれなければなりません。

このビットフィールドは、ハードウェアによって更新され、読み出し時に TX-FIFO にロードする残りのバイト数を報告します。

49.16.10 I3C ステータスレジスタ (I3C_SR)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

このレジスタは、I3C バスで交換されたメッセージのステータスを読み出すために使用されます。

- FIFO モード : I3C がコントローラとして機能する場合で S-FIFO が I3C_CFGR レジスタで SMODE = 1 にすることで有効化されている場合 :
 - ソフトウェアは、DMA モード (SDMAEN = 0) ではない場合に、読み出すステータスレジスタ (または有効な場合は対応する割込み) があれば、I3C_EVR レジスタの SFNEF = 1 によって通知されます。
 - DMA モード (SDMAEN = 1) で、プログラムされた I3C と DMA は、関連する連続した読出しをハードウェアによって自動的に管理します。
 - ソフトウェアは、S-FIFO のオーバーフローについて、I3C_SER レジスタの COVR = 1 (および I3C_EVR レジスタの ERRF = 1、および有効な場合は対応する割込み) によって通知されます。
- レジスタモード : I3C_CFGR レジスタで SMODE = 0 の場合
 - ソフトウェアは、このレジスタを読み出すために、I3C_SER レジスタのフラグ FCF と I3C_EVR レジスタの ERRF = 1 (および有効な場合は対応する割込み) を使用できます。
 - このレジスタは、新しいメッセージの完了時に、ハードウェアによって通知なしで上書き可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MID[7:0]								Res.	Res.	Res.	Res.	Res.	DIR	ABT	Res.
r	r	r	r	r	r	r	r						r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XDCNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **MID[7:0]** : 特定のフレームのメッセージ識別子/カウンタ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合 : このビットフィールドは、I3C_SR ステータスレジスタが指す相手宛ての制御ワードメッセージ (I3C_CR) を識別します。

フレームの最初のメッセージは MID[7:0] = 0 として識別されます。

このビットフィールドは、I3C 経由の新しいメッセージの制御ワード (I3C_CR) 完了時に (ハードウェアによって) インクリメントされます。このフィールドは、新しいフレームを開始するたびにリセットされます。

ビット 23:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **DIR** : メッセージの方向

I3C がコントローラ/ターゲットのどちらで機能する場合でも、このビットは I3C バス上の関連メッセージの方向を示します。

0 : 書込み

1 : 読出し

注 : ENTDAACCC は書込みコマンドとみなされます。

ビット 17 **ABT** : プライベート読み出しメッセージのターゲットによる途中終了 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットはターゲットによって送信されたプライベート読み出しデータが途中で終了するかどうかを示します (ターゲットが早めに T ビットをローで駆動、または I3C_CR レジスタのプログラムされた読み出しデータバイト数 DCNT[15:0] に関するコントローラの期待)。

0 : ターゲットからの早期完了なし

1 : ターゲットからの早期完了あり

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **XDCNT[15:0]** : データカウンタ

条件 : 動的なアドレス割り当て処理中 (ENTDAA CCC)

- I3C がコントローラとして機能する場合 : バス上で検出されたターゲット数

- I3C がターゲットとして機能する場合 : 送信されたバイト数

条件 : 他の転送の場合、メッセージ中

- I3C がコントローラ/ターゲットのどちらで機能する場合でも : メッセージ中に I3C バスから読み出された、または送信されたデータバイト数

49.16.11 I3C ステータス・エラー・レジスタ (I3C_SER)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

この読み出しレジスタは、ハードウェアによってエラーが発生し、I3C_EVR レジスタのエラーフラグ ERRF = 1 (および有効な場合は対応する割込み) によりソフトウェアに通知された場合に、エラーに関する詳細な情報を得るために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DERR	DNACK	ANACK	COVR	DOVR	STALL	PERR	CODERR[3:0]			
					r	r	r	r	r	r	r	r	r	r	r

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DERR** : データエラー (I3C がコントローラとして機能する場合)

0 : 検出されたエラーなし

1 : 受信したターゲットアドレスまたは/およびパリティビットが一致しない場合に、コントローラがコントローラ機能のハンドオフ手順 (GETACCCR CCC、旧 GETACCMST) 中にデータエラーを検出しました。アクティブなコントローラはコントローラ機能を保持します。

ビット 9 **DNACK** : 確認応答されていないデータ (I3C がコントローラとして機能する場合)

0 : 検出されたエラーなし

1 : コントローラによって、データバイトがターゲットに確認応答されていないことが検出された :

i) 従来の I2C 書き込み転送時

ii) ENTDAA 手順中に動的なアドレスを送信する 2 回目の試行時

ビット 8 **ANACK** : 確認応答されていないアドレス (I3C がコントローラとして設定されている場合)

0 : 検出されたエラーなし

1 : コントローラによって、静的/動的なアドレスがターゲットに確認応答されていないことが検出された :

- i) 従来の I2C 読出し/書き込み転送時
- ii) 直接 CCC 書き込み転送時
- iii) 直接 CCC 読出し転送の 2 回目の試行時
- iv) プライベート読出し/書き込み転送時

ビット 7 **COVR** : C-FIFO アンダーランまたは S-FIFO オーバーラン (I3C がコントローラとして機能する場合)

0 : 検出されたエラーなし

1 : コントローラが次のいずれかを検出 :

- i) C-FIFO アンダーラン : 制御 FIFO は空で再開を送信する必要があること
- ii) S-FIFO オーバーラン : S-FIFO がフルで、新しいメッセージが終了したこと

ビット 6 **DOVR** : RX-FIFO オーバーランまたは TX-FIFO アンダーラン

0 : 検出されたエラーなし

1 : コントローラであるかターゲットであるかにかかわらず、ハードウェアが次のいずれかを検出 :

- i) TX-FIFO アンダーラン : TX-FIFO が空で、書き込みデータバイトを送信する必要がある
- ii) RX-FIFO オーバーラン : RX-FIFO がフルで新しいデータバイトを受信

ビット 5 **STALL** : SCL ストールエラー (I3C がターゲットとして機能する場合)

0 : 検出されたエラーなし

1 : I3C SDR データ読出し中 (直接 CCC 読出し、プライベート読出し、または IB 中) に、SCL が 125 μ s より長い時間安定していることがターゲットによって検出された

ビット 4 **PERR** : プロトコルエラー

0 : 検出されたエラーなし

1 : コントローラであるかターゲットであるかにかかわらず、ハードウェアによってプロトコルエラーが検出されました。詳細については、CODERR[3:0] を参照してください。

ビット 3:0 **CODERR[3:0]** : プロトコルエラーのコード/タイプ

0000 : CE0 エラー (CCC 送信後のトランザクション) :

コントローラが不正にフォーマットされた CCC を検出

0001 : CE1 エラー (監視エラー) :

コントローラがバス上の送信データが想定と異なることを検出

0010 : CE2 エラー (ブロードキャストアドレスに対して応答なし) :

コントローラが確認応答されていないブロードキャストアドレス (0b111_1110) を検出

0011 : CE3 エラー (コントローラ機能のハンドオフの失敗) :

コントローラがコントローラ機能のハンドオフ後に新しいコントローラがバスを駆動しなかったことを検出

1000 : TE0 エラー (無効なブロードキャストアドレス 0b111_1110 + W) :

ターゲットが無効なブロードキャストアドレス 0b111_1110 + W を検出

1001 : TE1 エラー (CCC コード) :

ターゲットがパリティチェック (対 T ビット) により CCC コード上でパリティエラーを検出

1010 : TE2 エラー (書き込みデータ) :

ターゲットがパリティチェック (対 T ビット) により書き込みデータ上でパリティエラーを検出

1011 : TE3 エラー (動的なアドレスのアービトレーション中に割り当てられたアドレス) :

ターゲットがパリティチェック (対 PAR ビット) により動的なアドレスのアービトレーション中に割り当てられたアドレス上でパリティエラーを検出

1100 : TE4 エラー (動的なアドレスのアービトレーション中に Sr の後の 0b111_1110 + R が失われた) :

ターゲットが動的なアドレスのアービトレーション中に 0b111_1110 + R が失われていることを検出

1101 : TE5 エラー (CCC 検出後のトランザクション) :

ターゲットが不正にフォーマットされた CCC を検出

1110 : TE6 エラー (監視エラー) :

ターゲットがバス上の送信データが想定と異なることを検出

その他 : 予約済み

49.16.12 I3C 受信メッセージレジスタ (I3C_RMR)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

I3C がコントローラとして機能する場合、この読み出しレジスタは、受信したターゲットアドレスと、IBI の受信したデータペイロードのサイズを記録するために使用されます。

I3C がターゲットとして機能する場合、この読み出しレジスタは、受信した CCC コードを記録するために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RADD[6:0]						Res.	
								r	r	r	r	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCODE[7:0]								Res.	Res.	Res.	Res.	Res.	IBIRDCNT[2:0]		
r	r	r	r	r	r	r	r						r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 **RADD[6:0]** : 受信したターゲットアドレス (I3C がコントローラとして設定されている場合)

I3C がコントローラとして設定されている場合、このフィールドは、確認応答された IBI またはコントローラ機能リクエスト中に、ターゲットから受信した動的なアドレスを記録します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **RCODE[7:0]** : 受信した CCC コード (I3C がターゲットとして設定されている場合)

I3C がターゲットとして設定されている場合、このフィールドは受信した CCC コードを記録します。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **IBIRD CNT[2:0]** : IBI の受信したペイロード・データ・カウンタ (I3C がコントローラとして設定されている場合)

I3C がコントローラとして設定されている場合、このフィールドは I3C_IBIDR レジスタで効果的に受信したデータバイト数を記録します。

49.16.13 I3C イベントレジスタ (I3C_EVR)

アドレスオフセット : 0x050

リセット値 : 0x0000 0003

これは、イベントフラグの報告に使用される読み出しレジスタです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GRPF	DEFF	INT UPDF	AS UPDF	RSTF	MRL UPDF	MWL UPDF	DA UPDF	STAF	GETF	WKPF	Res.	HJF	CR UPDF	CRF	IBI ENDF
r	r	r	r	r	r	r	r	r	r	r		r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIF	Res.	Res.	Res.	ERRF	RXTGTE NDF	FCF	Res.	RX LASTF	TX LASTF	RX FNEF	TX FNFF	SFNEF	CFNFF	TXFEF	CFEF
r				r	r	r		r	r	r	r	r	r	r	r

ビット 31 **GRPF** : グループ・アドレス・フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合 (通常はコントローラ対応)、このフラグは、ブロードキャスト DEFTGTS CCC (グループアドレスの一覧を定義) を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは、コントローラ機能を得たときのために、受信データを格納できます。

このフラグは、I3C_CR レジスタの対応する CGRPF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 30 **DEFF** : DEFTGTS フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合 (通常はコントローラ対応)、このフラグは、ブロードキャスト DEFTGTS CCC (ターゲットの一覧を定義) を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは、コントローラ機能を得たときのために、受信データを格納できます。

このフラグは、I3C_CEVR レジスタの対応する CDEFF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 29 **INTUPDF** : 割込み/コントローラ機能/ホットジョイン更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは直接またはブロードキャスト ENEC/DISEC CCC (ターゲット・イベントの有効化/無効化) を受信したことを示すために、ハードウェアによってアサートされます。ここで、ターゲット・イベントは割込み/IBI リクエスト、コントローラ機能リクエスト/またはホットジョインリクエストのいずれかです。

その後、ソフトウェアは I3C_DEVR0 レジスタの IBIEN、CREN、または HJEN をそれぞれ読み出す必要があります。

このフラグは、I3C_CEVR レジスタの対応する CINTUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 28 ASUPDF : 動作状態更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは直接またはブロードキャスト ENTASx CCC (x = 0 3) を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは I3C_DEVR0 レジスタの AS[1:0] を読み出す必要があります。

このフラグは、I3C_CEVR レジスタの対応する CASUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 27 RSTF : リセット・パターン・フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグはリセットパターンを検出したことを示すために、ハードウェアによってアサートされます (SCL がローの間の 14 SDA 遷移、続けて反復 START、その後 STOP)。

その後、STOP モードでないときに、ソフトウェアは要求されるリセットレベルを知るために I3C_DEVR0 レジスタの RSTACT[1:0] と RSTVAL を読み出す必要があります。

- RSTVAL = 1 の場合 : RSTF がアサートされたら (および/または有効な場合は対応する割込み)、I3C_DEVR0 レジスタの RSTACT[1:0] ではソフトウェアによって実行されるリセット操作を要求します。
- RSTVAL = 0 の場合 : RSTF がアサートされたら (および/または有効な場合は対応する割込み)、ソフトウェアは最初に検出されたリセットパターンの後で I3C リセットを発行し、2 番目のときにはシステムリセットを発行する必要があります。

STOP モードでは、対応する割込みをデバイスのウェイクアップに使用できます。

このフラグは、I3C_CEVR レジスタの対応する CRSTF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 26 MRLUPDF : 最大読出し長更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは直接 SETMRL CCC (最大読出し長) を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは最大読出し長の値を取得するために I3C_MAXRLR レジスタの MRL[15:0] を読み出す必要があります。

このフラグは、I3C_CEVR レジスタの対応する CMRLUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 25 MWLUPDF : 最大書込み長更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは直接 SETMWL CCC (最大書込み長) を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは最大書込み長の値を取得するために I3C_MAXRLR レジスタの MWL[15:0] を読み出す必要があります。

このフラグは、I3C_CEVR レジスタの対応する CMWLUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 24 DAUPDF : 動的なアドレス更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグはブロードキャスト ENTDA, RSTDA, および直接 SETNEWDA CCC のいずれかで動的なアドレスの更新を受信したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは動的なアドレス更新を取得するために I3C_DEVR0 レジスタの DA[6:0] および DAVAL を読み出す必要があります。

このフラグは、I3C_CEVR レジスタの対応する CDAUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 23 STAF : 状況取得フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグはフォーマット 1 の直接 GETSTATUS CCC (定義バイトなし、または定義バイト TGTSTAT あり) を受信したことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVR レジスタの対応する CSTAF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 22 **GETF** : 取得フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグはフォーマット 1 の GETSTATUS を除いた (ただしフォーマット 2 の GETSTATUS は含む) 取得タイプの任意の直接 CCC (GET*** CCC) を受信したことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CGETF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 21 **WKPF** : ウェイクアップ/開始の見逃しフラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは開始が検出されたこと (SDA 立下りエッジとそれに続く SCL 立ち上がりエッジ) を示すために、ハードウェアによってアサートされます。ただし、次の SCL 立下りエッジでは、I3C カーネルクロックは (まだ) ゲートされています。このため、I3C バスのトランザクションは、ターゲットによって失われた可能性があります。

対応する割込みを使用して、デバイスを低電力 (SLEEP または STOP) モードからウェイクアップすることができます。

このフラグは、I3C_CEVF レジスタの対応する CWKPF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **HJF** : ホットジョイン・フラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフラグはホットジョイン・リクエストを受信したことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CHJF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 18 **CRUPDF** : コントローラ機能更新フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグはコントローラ機能のハンドオフ手順の完了後にコントローラ機能を取得できたことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CCRUPDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 17 **CRF** : コントローラ機能リクエストフラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフラグはコントローラ機能リクエストが (ハードウェアによって) 確認応答され、完了したことを示すために、ハードウェアによってアサートされます。その後、ソフトウェアは、コントローラ機能のハンドオフ手順に GETACCCR CCC (コントローラ機能の取得と許可) を発行する必要があります。

このフラグは、I3C_CEVF レジスタの対応する CCRF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 16 **IBIENDF** : IBI 終了フラグ (I3C がターゲットとして機能する場合)

I3C がターゲットとして機能する場合、このフラグは IBI 転送を受信し、完了したこと (存在する場合は、コントローラによって IBI が確認応答され、IBI データバイトが読み出されている) を示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CIBIENDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 15 **IBIF** : IBI フラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフラグは IBI リクエストを受信したことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CIBIF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **ERRF** : フラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このフラグは発生したエラーを示すために、ハードウェアによってアサートされます。その後、ソフトウェアはエラータイプを取得するために I3C_SER を読み出す必要があります。

このフラグは、I3C_CEVF レジスタの対応する CERRF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 10 **RXTGTENDF** : ターゲットによる読出し終了フラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、また S-FIFO が無効である場合にのみ (I3C_CFGR レジスタで SMODE = 0)、ターゲットによって読出し転送が途中終了したことを示すために、このフラグがハードウェアによってアサートされます。その後、ソフトウェアは最後のメッセージに関する情報を確認するためにステータスレジスタ I3C_SR を読み出す必要があります、途中終了した読出し転送 (I3C_SR レジスタで XDCNT) の受信データバイト数を取得します。

このフラグは、I3C_CEVF レジスタの対応する CRXTGTENDF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 9 **FCF** : フレーム完了フラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

I3C がコントローラとして機能する場合、このフラグは、たとえば STOP が発行されたときに、フレームが I3C バス上で (正常に) 完了したことを示すために、ハードウェアによってアサートされます。

I3C がターゲットとして機能する場合、このフラグは、たとえばコントローラによって次の STOP や 反復 START が発行されたときに、このターゲットとの間でアドレス指定されたメッセージが I3C バス上で (正常に) 完了したことを示すために、ハードウェアによってアサートされます。

このフラグは、I3C_CEVF レジスタの対応する CFCF ビットに、ソフトウェアが 1 を書き込むことでクリアされます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **RXLASTF** : 最後に読み出されたデータバイト/ワードのフラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフラグは、メッセージの最後のデータバイト/ワード (I3C_CFGR レジスタの RXTHRES による) を RX-FIFO から読み出す必要があることを示すために、ハードウェアによってアサートされます。このフラグは、メッセージの最後のデータバイト/ワードが読み出されたときに、ハードウェアによってアサートされます。

ビット 6 **TXLASTF** : 最後に書き込まれたデータバイト/ワードのフラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このフラグは、メッセージの最後のデータバイト/ワード (I3C_CFGR レジスタの TXTHRES による) を TX-FIFO に書き込む必要があることを示すために、ハードウェアによってアサートされます。このフラグは、メッセージの最後のデータバイト/ワードが書き込まれたときに、ハードウェアによってアサートされます。

ビット 5 **RXFNEF** : RX-FIFO ノットエンプティフラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このフラグは、データバイトを S-FIFO から読み出す必要があるかないかを示すために、ハードウェアによってアサート/デアサートされます。

注 : ソフトウェアは、RX-FIFO から読み出す (RXTHRES に応じて I3C_RDR または I3C_RDWR から読み出す) 前に、(ポーリングまたは有効化された割込みによって) RXFNEF = 1 を待つ必要があります。

ビット 4 **TXFNFF** : TX-FIFO ノットフルフラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このフラグは、データバイト/ワードを TX-FIFO に書き込む必要があるかないかを示すために、ハードウェアによってアサート/デアサートされます。

注 : ソフトウェアは、TX-FIFO に書き込む (TXTHRES に応じて I3C_TDR または I3C_TDWR に書き込む) 前に、(ポーリングまたは有効化された割込みによって) TXFNFF = 1 を待つ必要があります。

注 : I3C がターゲットとして機能する場合、ソフトウェアが I3C_TDR/I3C_TDWR への書き込みには TXFNFF フラグを使用する場合は、TX-FIFO プリロードを設定し、セットしておく必要があります (I3C_TGTTDR レジスタの PRELOAD)。

ビット 3 **SFNEF** : S-FIFO ノットエンプティフラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、また S-FIFO が有効である場合にのみ (I3C_CFGR レジスタで SMODE = 1)、ステータスワードを S-FIFO から読み出す必要があることを示すために、このフラグがハードウェアによってアサートされます。このフラグは、ステータスワードを S-FIFO から読み出しはならないことを示すために、ハードウェアによってデアサートされます。

ビット 2 **CFNFF** : C-FIFO ノットフルフラグ (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフラグは制御ワードを C-FIFO に書き込む必要があることを示すために、ハードウェアによってアサートされます。このフラグは、制御ワードを C-FIFO に書き込んでいないことを示すために、ハードウェアによってデアサートされます。

注 : ソフトウェアは、C-FIFO に書き込む (I3C_CR に書き込む) 前に、(ポーリングまたは有効化された割込みによって) CFNFF = 1 を待つ必要があります。

ビット 1 **TXFEF** : TX-FIFO エンプティフラグ (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

このフラグは、TX-FIFO が空であることを示すために、ハードウェアによってアサートされます。

このフラグは、TX-FIFO が空でないことを示すために、ハードウェアによってデアサートされます。

ビット 0 **CFEF** : C-FIFO エンプティフラグ (I3C がコントローラとして機能する場合)

このフラグは、コントローラの場合は C-FIFO が空であること、またターゲットの場合は I3C_CR レジスタに制御ワード (IBI/CR/HJ リクエストなし) が含まれないことを示すために、ハードウェアによってアサートされます。

このフラグは、コントローラの場合は C-FIFO が空でないこと、またターゲットの場合は I3C_CR レジスタに 1 つの制御ワード (ペンディング中の IBI/CR/HJ リクエスト) が含まれていることを示すために、ハードウェアによってデアサートされます。

注 : I3C がコントローラとして機能する場合、C-FIFO と TX-FIFO プリロードが設定されているときは (I3C_CFGR レジスタで TMODE = 1)、ソフトウェアは新しいフレーム転送を開始する前に TXFEF = 1 と CFEF = 1 を待つ必要があります。

49.16.14 I3C 割込み有効レジスタ (I3C_IER)

アドレスオフセット : 0x054

リセット値 : 0x0000 0000

このレジスタは、以下の各イベント/フラグの割込みを、ビットレベルで有効化/無効化するために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GRPIE	DEFIE	INTUPDIE	ASUPDIE	RSTIE	MRLUPDIE	MWLUPDIE	DAUPDIE	STAE	GETIE	WKPIE	Res.	HJIE	CRUPDIE	CRIE	IBENDIE
r	r	r	r	r	r	r	r	r	r	r		r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIE	Res.	Res.	Res.	ERRIE	RXTGTENDIE	FCIE	Res.	Res.	Res.	RXFNEIE	TXFNIE	SFNEIE	CFNIE	Res.	Res.
r				r	r	r				r	r	r	r		

ビット 31 **GRPIE** : DEFGRPA CCC 割込みイネーブル (I3C がターゲットとして機能する場合)

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 30 **DEFIE** : DEFTGTS CCC 割込みイネーブル (I3C がターゲットとして機能する場合)

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 29 **INTUPDIE** : ENEC/DISEC CCC 割込みイネーブル (I3C がターゲットとして機能する場合)

0 : 割込みは無効です。

1 : 割込みは有効です。

- ビット 28 **ASUPDIE** : ENTASx CCC 割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 27 **RSTIE** : リセットパターン割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 26 **MRLUPDIE** : SETMRL CCC 割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 25 **MWLUPDIE** : SETMWL CCC 割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 24 **DAUPDIE** : ENTDAARSTDAA/SETNEWDA CCC 割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 23 **STAIE** : フォーマット 1 の GETSTATUS CCC 割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 22 **GETIE** : GETxxx CCC 割込みイネーブル (フォーマット 1 の GETSTATUS を除く) (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 21 **WKPIE** : ウェイクアップ割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 20 予約済みであり、リセット値に保持する必要があります。
- ビット 19 **HJIE** : ホットジョイン割込みイネーブル (I3C がコントローラとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 18 **CRUPDIE** : コントローラ機能更新割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 17 **CRIE** : コントローラ機能リクエスト割込みイネーブル (I3C がコントローラとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 16 **IBIENDIE** : IBI 終了割込みイネーブル (I3C がターゲットとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 15 **IBIIE** : IBI リクエスト割込みイネーブル (I3C がコントローラとして機能する場合)
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 14:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **ERRIE** : エラー割込みイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)
0 : 割込みは無効です。
1 : 割込みは有効です。

ビット 10 **RXTGTENDIE** : ターゲットによる読み出し終了割り込みイネーブル (I3C がコントローラとして機能する場合)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 9 **FCIE** : フレーム完了割り込みイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 8:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RXFNEIE** : RX-FIFO ノットエンプティ割り込みイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 4 **TXFNIE** : TX-FIFO ノットフル割り込みイネーブル (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 3 **SFNEIE** : S-FIFO ノットエンプティ割り込みイネーブル (I3C がコントローラとして機能する場合)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 2 **CFNIE** : C-FIFO ノットフル割り込みイネーブル (I3C がコントローラとして機能する場合)

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

49.16.15 I3C クリアイベントレジスタ (I3C_CEVR)

アドレスオフセット : 0x058

リセット値 : 0x0000 0000

この書き込みレジスタは、I3C_EVR レジスタの対応するイベントフラグを、ビットレベルで個別にクリアするために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGRP	CDEFF	CINTUPDF	CASUPDF	CRSTF	CMRLUPDF	CMWLUPDF	CDAUPDF	CSTAF	CGETF	CWKPF	Res.	CHJF	CCRUPDF	CCRF	CBIENDF
w	w	w	w	w	w	w	w	w	w	w		w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIBIF	Res.	Res.	Res.	CERRF	CRXTGTENDF	CFCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w				w	w	w									

ビット 31 **CGRP** : DEFGRPA CCC フラグをクリア (I3C がターゲットとして機能する場合)

0 : 影響なし。

1 : GRPF をクリアします。

ビット 30 **CDEFF** : DEFTGTS CCC フラグをクリア (I3C がターゲットとして機能する場合)

0 : 影響なし。

1 : DEFF をクリアします。

- ビット 29 **CINTUPDF** : ENEC/DISEC CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : CINTUPDF をクリアします。
- ビット 28 **CASUPDF** : ENTASx CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : ASUPDF をクリアします。
- ビット 27 **CRSTF** : リセット・パターン・フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : RSTF をクリアします。
- ビット 26 **CMRLUPDF** : SETMRL CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : MRLUPDF をクリアします。
- ビット 25 **CMWLUPDF** : SETMWL CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : MWLUPDF をクリアします。
- ビット 24 **CDAUPDF** : ENTDA/RTDAA/SETNEWDA CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : DAUPDF をクリアします。
- ビット 23 **CSTAF** : フォーマット 1 の GETSTATUS CCC フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : STAF をクリアします。
- ビット 22 **CGETF** : GETxxx CCC フラグをクリア (フォーマット 1 の GETSTATUS を除く) (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : GETF をクリアします。
- ビット 21 **CWKPF** : ウェイクアップ・フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : WKPF をクリアします。
- ビット 20 予約済みであり、リセット値に保持する必要があります。
- ビット 19 **CHJF** : ホットジョイン・フラグをクリア (I3C がコントローラとして機能する場合)
0 : 影響なし。
1 : HJF をクリアします。
- ビット 18 **CCRUPDF** : コントローラ機能更新フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : CRUPDF をクリアします。
- ビット 17 **CCRF** : コントローラ機能リクエストフラグをクリア (I3C がコントローラとして機能する場合)
0 : 影響なし。
1 : CRF をクリアします。
- ビット 16 **CIBIENDF** : IBI 終了フラグをクリア (I3C がターゲットとして機能する場合)
0 : 影響なし。
1 : IBIENDF をクリアします。
- ビット 15 **CIBIF** : IBI リクエストフラグをクリア (I3C がコントローラとして機能する場合)
0 : 影響なし。
1 : IBIF をクリアします。
- ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **CERRF** : エラーフラグをクリア (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : 影響なし。

1 : ERRF をクリアします。

ビット 10 **CRXTGTENDF** : ターゲットによる読み出し終了フラグをクリア (I3C がコントローラとして機能する場合)

0 : 影響なし。

1 : RXTGTENDF をクリアします。

ビット 9 **CFCF** : フレーム完了フラグをクリア (I3C がコントローラ/ターゲットのどちらで機能する場合でも)

0 : 影響なし。

1 : FCF をクリアします。

ビット 8:0 予約済みであり、リセット値に保持する必要があります。

49.16.16 I3C 独自デバイス特性レジスタ (I3C_DEVR0)

アドレスオフセット : 0x060

リセット値 : 0x0000 0000

I3C ペリフェラルがターゲットとして機能する場合、このレジスタは独自のデバイス特性の書き込みまたは読み出しに使用されます。

I3C ペリフェラルがコントロールとして機能する場合、フィールド DA[6:0] は独自の動的なアドレスの書き込みおよび格納に使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTVAL	RSTACT[1:0]		AS[1:0]		HJEN	Res.	CREN	IBIEN
							r	r	r	r	r	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DA[6:0]							DAVAL
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **RSTVAL** : リセット操作が有効 (I3C がターゲットとして機能する場合)

このビットは、ブロードキャストまたは直接書き込み RSTACT CCC (ターゲットのリセット操作) の受信時に RSTACT[1:0] フィールドが更新され、またそれが有効であることを示すために、ハードウェアによってアサートされます。

このビットは、ターゲットがフレーム開始を受信したときにハードウェアによってクリアされます。

デバイスが STOP モードでない場合 :

- RSTVAL = 1 の場合 : I3C_EVR レジスタの RSTF がアサートされたら (および/または有効な場合は対応する割込み)、I3C_DEVR0 レジスタの RSTACT[1:0] ではソフトウェアによって実行されるリセット操作を要求します。
- RSTVAL = 0 の場合 : RSTF がアサートされたら (および/または有効な場合は対応する割込み)、ソフトウェアは最初に検出されたリセットパターンその後 I3C リセットを発行し、2 番目のときにはシステムリセットを発行する必要があります。

STOP モードでは、対応する割込みをデバイスのウェイクアップに使用できます。

ビット 23:22 **RSTACT[1:0]**: 受信したリセット・パターンでのリセット操作/レベル (I3C がターゲットとして機能する場合)

この読み出しフィールドは、直接読み出し RSTACT CCC の受信時に、I3C バスの対応するデータバイトを返すために、ハードウェアによって使用されます。

この読み出しフィールドは、ブロードキャストまたは直接書き込み RSTACT CCC (ターゲットのリセット操作) を受信したときに、ハードウェアによって更新されます。

定義バイト 0x00、0x01、および 0x02 のみがマップされ、RSTACT[1:0] = 定義バイト [1:0] になります。

00: リセット操作なし。

01: リセットの第 1 レベル: アプリケーション・ソフトウェアで以下を行う必要があります。

a) I3C 設定レジスタのイネーブルビットの書き込みおよびクリア (EN = 0 を書き込む) による、ペリフェラルの部分リセット。これにより、I3C APB レジスタの内容を変更することなく (EN ビットは除く)、I3C バスインタフェースと I3C カーネルサブパートがリセットされます。

b) RCC (リセットおよびクロックコントローラ) レジスタの I3C リセット制御ビットの書き込みおよびセットによる、ペリフェラルの完全リセット (そのすべてのレジスタを含む)。

10: リセットの第 2 レベル: アプリケーション・ソフトウェアで、システムリセットとも呼ばれるウォーム・リセットを発行する必要があります。これには ([セクション 11: リセットおよびクロック制御 \(RCC\)](#) を参照)、ピンリセット (NRST = 0) と同じ効果があります。

– ソフトウェアは、デバイスが Cortex[®]-M によって制御されている場合に、AITR レジスタの SYSRESETREQ 制御ビットを書き込み、セットします。

– ソフトウェアは、デバイスが Cortex[®]-A によって制御されている場合に、RCC_GRSTCSETR レジスタで SYSRST = 1 を書き込み、セットします。

11: リセット操作なし。

ビット 21:20 **AS[1:0]**: 動作状態 (I3C がターゲットとして機能する場合)

この読み出しフィールドは、ENTASx CCC (動作状態を入力、x = 0 -3) を受信したときに、ハードウェアによって更新されます。

00: 動作状態 0

01: 動作状態 1

10: 動作状態 2

11: 動作状態 3

ビット 19 **HJEN**: ホットジョイン・リクエスト・イネーブル (I3C がターゲットとして機能する場合)

このビットは、EN = 0 のときにソフトウェアによって最初に関き込まれ、DISHJ = 1 (クリア) の DISEC CCC を受信したときと、ENHJ = 1 (セット) の ENEC CCC を受信したときに、ハードウェアによって更新されます。このビットは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

0: ホットジョイン・リクエストは無効です。

1: ホットジョイン・リクエストは有効です。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CREN**: コントローラ機能リクエスト・イネーブル (I3C がターゲットとして機能する場合)

このビットは、EN = 0 のときにソフトウェアによって最初に関き込まれ、DISCR = 1 (クリア) の DISEC CCC を受信したときと、ENCN = 1 (セット) の ENEC CCC を受信したときに、ハードウェアによって更新されます。このビットは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

0: コントローラ機能リクエストは無効です。

1: コントローラ機能リクエストは有効です。

ビット 16 **IBIEN**: IBI リクエスト・イネーブル (I3C がターゲットとして機能する場合)

このビットは、EN = 0 のときにソフトウェアによって最初に関き込まれ、DISINT = 1 (クリア) の DISEC CCC を受信したときと、ENINT = 1 (セット) の ENEC CCC を受信したときに、ハードウェアによって更新されます。このビットは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

0: IBI リクエストは無効です。

1: IBI リクエストは有効です。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:1 **DA[6:0]** : 7 ビットの動的なアドレス

I3C がコントローラとして機能する場合、このフィールドは独自の動的なアドレスを定義するためにソフトウェアによって書き込むことができます。

I3C がターゲットとして機能する場合、このフィールドはブロードキャスト ENTDA CCC または直接 SETNEWDAA CCC のどちらかを受信したときに、ハードウェアによって更新されます。

ビット 0 **DAVAL** : 動的なアドレスが有効 (I3C がターゲットとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、たとえばコントローラ機能のハンドオフの前に、独自の動的なアドレスを検証するためにソフトウェアによって書き込むことができます。

I3C がターゲットとして機能する場合、このビットはブロードキャスト ENTDA CCC または直接 SETNEWDAA CCC の確認応答時にハードウェアによってアサートされ、またこのフィールドはブロードキャスト RSTDAA CCC の確認応答時にハードウェアによってクリアされます。

49.16.17 I3C デバイス x 特性レジスタ (I3C_DEVRx)

アドレスオフセット : $0x060 + 0x4 * x$, ($x = 1$ から 4)

リセット値 : $0x0000\ 0000$

I3C ペリフェラルがコントローラとして機能する場合、このレジスタは、I3C 経由で任意のターゲット x に応じて通信するために、デバイスターゲット x の一部の特性を、コントローラからの関連する管理とともに定義し、格納するために使用されます。その後、ハードウェアは、ターゲット x からの許可された IBI または/およびコントローラ機能リクエストを自律的に特定し、確認応答すること、存在する場合は予想された IBI ペイロード・データを受信すること、および I3C_EVR レジスタの対応するフラグ IBIF/CRF (および有効な場合は対応する割込み) 経由でソフトウェアに通知することができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	IBIDEN	CRACK	IBIACK
r												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DA[6:0]							Res.
								rw	rw	rw	rw	rw	rw	rw	

ビット 31 **DIS** : DA[6:0] 書き込み無効 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、一度ソフトウェアを IBIACK= 1 または CRACK= 1 にセットすると、設定された DA[6:0] および IBIDEN の値をロックするために、この読み出しビットはハードウェアによってセットされます (DIS = 1)。

次に、DA[6:0]、IBIDEN、または SUSP を変更できるようにするには、ソフトウェアはターゲット x に割り当てられたこれらの 3 つの値を変更する前に、DIS がハードウェアによってデアサートされるのを待つ必要があります (DIS = 0 でポーリング)。実際、コントローラが DA[6:0]、IBIDEN、または SUSP を変更する場合、ターゲットは IBI またはコントローラ機能をリクエストできます。

0 : I3C_DEVRx レジスタの DA[7:0] および IBIDEN への書き込みが許可されます。

1 : DA[7:0] および IBIDEN への書き込みが無効化/ロックされます。

ビット 30:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **SUSP** : 受信した IBI で I3C 転送をサスペンド/停止 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、ペンディング中の読出し通知機能 (MDB[7:5] = 3'b101 を受信) を持つターゲット x から IBI を受信するために使用できます。

このビットがセットされた場合、IBI を受信し、完了すると (I3C_EVR レジスタで IBIF = 1)、I3C バスで STOP が送信され、C-FIFO と TX-FIFO の両方がハードウェアによって自動的に一掃されます。次の制御メッセージがプログラムされていても、コントローラの実行フローが停止されます。IBI が完了すると、コントローラのソフトウェアは、IBI リクエストを開始したターゲットデバイスに、プライベート読出しなどの新しい制御ワードを発行できます

0 : C-FIFO および TX-FIFO は、ターゲット x からの IBI リクエストが確認応答され、完了した後に一掃されません。また、次の制御ワードの有無に応じて、反復 START または STOP が送信されます。

1 : ターゲット x から IBI リクエストを受信した後で、I3C 転送が停止され、C-FIFO と TX-FIFO の両方が一掃されます。

ビット 18 **IBIDEN** : IBI データインエーブル (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、受信した I3C_RDR でブロードキャスト ENTDAА または直接 GETBCR CCC 中にターゲット x から受信した BCR[2] ビットを格納するために、ソフトウェアによって書き込まれなければなりません。

このフィールドに書き込んでも、I3C_DEVRx レジスタで DIS = 1 の場合、影響はありません。

0 : ターゲット x からの確認応答された IBI に続くデータバイトなし

1 : ターゲット x からの確認応答された IBI に続く必須のデータバイト MDB[7:0]

ビット 17 **CRACK** : コントローラ機能リクエスト確認応答 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、ターゲット x からのコントローラ機能リクエストの受信時に、I3C バスに適用される確認応答ポリシーを定義するためにソフトウェアによって書き込まれます。

0 : ターゲット x からのコントローラ機能リクエストを NACK する必要があります。

NACK された後で、メッセージは最初にプログラムされたとおりに続行します (ターゲットは NACK を認識しており、後で別のコントローラ機能リクエストを送信できます)。

1 : ターゲット x からのコントローラ機能リクエスト (7 ビットの動的なアドレス DA[6:0] を持つ) を ACK する必要があります。

- フィールド DIS は、DA[6:0] がソフトウェアによって変更されないように保護するために、ハードウェアによってアサートされます。その間、ハードウェアは現在の DA[6:0] をカーネルクロックドメインに内部的に格納できます。

ACK された後で、メッセージは最初にプログラムされたとおりに続行します。ソフトウェアはコントローラ機能リクエスト・フラグ (I3C_EVR レジスタで CRF = 1) および/または有効な場合は対応する割込みによって通知されます。リクエスト側の 2 次コントローラにコントローラ機能を効果的に付与する場合、ソフトウェアは GETACCCR (旧 GETACCMST) に続けて STOP を発行する必要があります。

- このデバイスまたは他のデバイスの CRACK の設定とは関係なく、追加のコントローラ機能リクエストは、I3C_EVR レジスタのコントローラ機能リクエスト・フラグ (CRF) と IBI フラグ (IBIF) が両方ともクリアされるまで NACK されます。

ビット 16 **IBIACK** : IBI リクエスト確認応答 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このビットは、ターゲット x からの IBI リクエストの受信時に、I3C バスに適用される確認応答ポリシーを定義するためにソフトウェアによって書き込まれます。

0 : ターゲット x からの IBI リクエストを NACK する必要があります。

- NACK された後で、メッセージは最初にプログラムされたとおりに続行します (ターゲットは NACK を認識しており、後で別の IBI リクエストを送信できます)。

1 : ターゲット x からの IBI リクエスト (7 ビットの動的なアドレス DA[6:0] を持つ) を ACK する必要があります。

- フィールド DIS は、DA[6:0] がソフトウェアによって変更されないように保護するために、ハードウェアによってアサートされます。その間、ハードウェアは現在の DA[6:0] をカーネルクロックドメインに内部的に格納できます。

- ACK された後で、コントローラは I3C_DEVRx.IBIDEN に応じて IBI ペイロード・データ (存在する場合) を記録します。

- ソフトウェアは、IBI フラグ (IBIF = 1) および/または有効な場合は対応する割込みによって通知されます。

- このデバイスまたは他のデバイスの IBIACK の設定とは関係なく、追加の IBI リクエストは、IBI リクエスト・フラグ (IBIF) とコントローラ機能リクエスト・フラグ (CRF) が両方ともクリアされるまで NACK されます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:1 **DA[6:0]** : ターゲット x への割り当てられた I3C の動的なアドレス (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合、このフィールドは、コントローラがブロードキャスト ENTDA またはターゲット x によって確認応答された直接 SETNEWDA CCC によって送信する 7 ビットの動的なアドレスを格納するために、ソフトウェアによって書き込まなければなりません。

このフィールドに書き込んでも、I3C_DEVRx レジスタで読出しフィールド DIS = 1 の場合、影響はありません。

ビット 0 予約済みであり、リセット値に保持する必要があります。

49.16.18 I3C 最大読出し長レジスタ (I3C_MAXRLR)

アドレスオフセット : 0x090

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは GETMRL または SETMRL CCC の間にコントローラと交換された最大読出し長の値を、それぞれセットまたは取得するために使用されます。このレジスタは、IBI データ・ペイロード・サイズをセットするためにも使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによって書き込むことができます。

プライベート読出しメッセージの受信時に、送信データ数が I3C_MAXRLR レジスタの MRL[15:0] に達すると、ターゲットは (T ビット = 0 を駆動して) データ送信を終了します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IBIP[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **IBIP[2:0]** : バイト単位での IBI ペイロード・データの最大サイズ (I3C がターゲットとして機能する場合)
 このフィールドは、IBI リクエストの確認応答後にコントローラに送信されるデータバイトの最大数をセットするために、EN = 0 のときにソフトウェアによって最初書き込まれます。このフィールドは、SETMRL コマンドを受信したときに、ハードウェアによって更新されます (これにより、場合によっては IBIP[2:0] も更新されます)。

ソフトウェアは、MRLUPF による MRL 更新について、I3C_EVR レジスタの MRLUPF によって、また対応する割込みが有効な場合はそれによって、通知されます。

000 : null ペイロードデータサイズ (IC3_BCR レジスタで BCR2 = 0 の場合にのみ可能)

001 : 1 バイト (必須のデータバイト MDB[7:0])

010 : 2 バイト (最初の MDB[7:0] を含む)

011 : 3 バイト (最初の MDB[7:0] を含む)

100 : 4 バイト (最初の MDB[7:0] を含む)

その他 : 100 と同じ

ビット 15:0 **MRL[15:0]** : 最大データ読出し長 (I3C がターゲットとして機能する場合)

このフィールドは、EN = 0 のときにソフトウェアによって最初書き込まれ、SETMRL コマンドを受信したときに、ハードウェアによって更新されます (場合によっては IBIP[2:0] も更新されます)。

ソフトウェアは、MRLUPF による MRL 更新について、MRLUPF によって、また対応する割込みが有効な場合はそれによって、通知されます。

このフィールドは、ターゲットが GETMRL CCC を受信したときに、I3C バスの値を返すためにハードウェアによって使用されます。

49.16.19 I3C 最大書込み長レジスタ (I3C_MAXWLR)

アドレスオフセット : 0x094

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは GETMWL または SETMWL CCC の間にコントローラと交換された最大書込み長の値を、それぞれセットまたは取得するために使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによって書き込むことができます。

プライベート書込みメッセージの受信時に、受信データ数が I3C_MAXWLR レジスタの MWL[15:0] に達すると、ターゲットはデータ受信を停止します (余分な受信データは RX-FIFO には書き込まれません)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MWL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **MWL[15:0]** : 最大データ書込み長 (I3C がターゲットとして機能する場合)

このフィールドは、EN = 0 のときにソフトウェアによって最初書き込まれ、SETMWL コマンドを受信したときに、ハードウェアによって更新されます。

ソフトウェアは、MWLUPF による MWL 更新について、I3C_EVR レジスタの MRLUPF によって、また対応する割込みが有効な場合はそれによって、通知されます。

このフィールドは、ターゲットが GETMWL CCC を受信したときに、I3C バスの値を返すためにハードウェアによって使用されます。

49.16.20 I3C タイミングレジスタ 0 (I3C_TIMINGR0)

アドレスオフセット : 0x0A0

リセット値 : 0x0000 0000

I3C がコントローラとして機能する場合、このレジスタは SCL クロック信号波形を設定するために使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによって書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCLH_I2C[7:0]								SCLL_OD[7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH_I3C[7:0]								SCLL_PP[7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 **SCLH_I2C[7:0]** : 従来の I²C メッセージのカーネルクロックサイクル数に使用される SCL ハイ時間 :

$$t_{\text{SCLH_I2C}} = (\text{SCLH_I2C} + 1) \times t_{\text{I3CCLK}}$$

注 : **SCLH_I2C** は、I²C デバイスとの通信時に $t_{\text{DIG_H}}$ (I²C) タイミングを生成するために使用されます。

注 : I²C fm+ デバイス $t_{\text{DIG_Hmin}} = 260 \text{ ns}$ 、I²C fm デバイス $t_{\text{DIG_Hmin}} = 600 \text{ ns}$ 。

ビット 23:16 **SCLL_OD[7:0]** : オープンドレイン・フェーズの SCL ロー時間は、従来の I²C メッセージと I3C オープンドレイン・フェーズ（開始後のアドレスフェーズ、コントローラ開始メッセージ中の ACK フェーズ、および直接/プライベート/IBI ペイロード中の T ビット・フェーズ）のカーネルクロックサイクル数に使用されます。

$$t_{\text{SCLL_OD}} = (\text{SCLL_OD} + 1) \times t_{\text{I3CCLK}}$$

注 : **SCLL_OD** は、 $t_{\text{DIG_L}}$ (I²C) と $t_{\text{DIG_OD_L}}$ (I3C) の両方のタイミングを生成するために使用されます。

注 : I²C fm+ デバイス $t_{\text{DIG_Lmin}} = 500 \text{ ns}$ 、I²C fm デバイス $t_{\text{DIG_Lmin}} = 1320 \text{ ns}$ 。

注 : I3C メッセージ : $t_{\text{DIG_OD_Lmin}} = 200 \text{ ns}$ 。

注 : I²C と I3C のメッセージを 1 つの I3C フレームがまとめている場合、I3C オープンドレイン・フェーズ中の SCL ロー時間は I²C タイミングに合うように延長されます。

ビット 15:8 **SCLH_I3C[7:0]** : I3C メッセージ（プッシュプルとオープンドレインの両方のフェーズ）のカーネルクロックサイクル数に使用される SCL ハイ時間 :

$$t_{\text{SCLH_I3C}} = (\text{SCLH_I3C} + 1) \times t_{\text{I3CCLK}}$$

注 : **SCLH_I3C** は、 $t_{\text{DIG_H}}$ (I3C) と $t_{\text{DIG_H_MIXED}}$ の両方のタイミングを生成するために使用されます。

注 : バスが混在している場合（少なくとも 1 つの I²C ターゲットで） : $t_{\text{DIG_H_MIXEDmin}} = 32 \text{ ns}$ および $t_{\text{DIG_H_MIXEDmax}} = 45 \text{ ns}$ (I²C 50 ns のスパイクフィルタによる)。

注 : I3C バスのみの場合 (I²C ターゲットなし) : $t_{\text{DIG_Hmin}} = 32 \text{ ns}$ 。

ビット 7:0 **SCLL_PP[7:0]** : I3C プッシュプル・フェーズの SCL ロー時間（カーネルクロックサイクル数） :

$$t_{\text{SCLL_PP}} = (\text{SCLL_PP} + 1) \times t_{\text{I3CCLK}}$$

注 : **SCLL_PP** は、 $t_{\text{DIG_L}}$ (I3C は PP 内) のタイミングを生成するために使用されます。

注 : $t_{\text{DIG_Lmin}} = 32 \text{ ns}$ (12.5 MHz で最大 40/60 デューティサイクル)。

49.16.21 I3C タイミングレジスタ 1 (I3C_TIMINGR1)

アドレスオフセット : 0x0A4

リセット値 : 0x0000 0000

I3C がコントローラとして機能する場合、このレジスタは I3C タイミング設定を設定するために使用されます。

I3C がターゲットとして機能し、コントローラ対応である場合、このレジスタは、コントローラ機能のハンドオフ手順のタイミングを設定するために使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによって書き込むことができません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SDA_HD	Res.	Res.	Res.	Res.	Res.	FREE[6:0]						
			rw						rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	ASNCR[1:0]		AVAL[7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **SDA_HD** : SDA ホールド時間 (I3C がコントローラとして機能する場合) のカーネルクロックサイクル数 (プッシュプルでの MIPI のタイミングの SDA ホールド時間 $_{HD_PP}$ を参照) :

$$\text{SDA ホールド時間} = (\text{SDA_HD} + 0.5) \times t_{3CCCLK}$$

注 : コントローラが $t_{HD_PPmin} = \min(t_{CR}, t_{CF}) + 3 \text{ ns}$ である場合。

ビット 27:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **FREE[6:0]** : バスフリー条件時間などの MIPI のタイミングをセットするために使用されるカーネルクロックサイクル数 (I3C がコントローラとして機能する場合)

I3C がコントローラとして機能する場合 :

1. I3C の START のタイミング : STOP 後、START 前に (バスフリー条件) 時間が経過するのを待つ必要があります。MIPI のタイミング (I3C) t_{CAS} および (I^2C) t_{BUF} を参照してください。タイミングの定義方法 : $t_{BUF} = t_{CAS} = [(FREE[6:0] + 1) \times 2 - (0.5 + SDA_HD)] \times t_{3CCCLK}$

注 : I3C バスのみの場合 : $t_{CASmin} = 38.4 \text{ ns}$ 、および $t_{CASmax} = 1 \mu\text{s}$ 、 $100 \mu\text{s}$ 、 2 ms 、 50 ms (それぞれ ENTAS0、1、2、および 3 に対して)。

注 : $I^2C \text{ fm+}$ デバイス $t_{BUFmin} = 0.5 \mu\text{s}$ で混在しているバスの場合、 $I^2C \text{ fm}$ デバイス $t_{BUFmin} = 1.3 \mu\text{s}$ で混在しているバスの場合。

2. I3C の反復 START のタイミング : 反復 START 後 (SDA がデアサートされる)、SCL をローに駆動する前に、時間が経過するのを待つ必要があります。MIPI のタイミング t_{CASr} を参照してください。タイミングの定義方法 : $t_{CASr} = [(FREE[6:0] + 1) \times 2 - (0.5 + SDA_HD)] \times t_{3CCCLK}$

注 : $t_{CASr, min} = 19.2 \text{ ns}$ 。

3. I3C の STOP のタイミング : SCL クロックをハイに駆動した後、STOP コンディションの前に (SDA がアサートされる)、時間が経過するのを待つ必要があります。タイミングの定義方法 : $t_{CBP} = (FREE[6:0] + 1) \times t_{3CCCLK}$

注 : $t_{CBPmin} = 19.2 \text{ ns}$ 。

4. I3C の反復 START のタイミング (反復 START の後に STOP を続けて、コントローラが読出しを終了した場合は T ビット) : SCL クロックをハイに駆動した後、反復 START コンディションの前に (SDA がデアサートされる)、時間が経過するのを待つ必要があります。タイミングの定義方法 : $t_{CBSr} = (FREE[6:0] + 1) \times t_{3CCCLK}$

注 : $t_{CBSr, min} = 19.2 \text{ ns}$ 。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **ASNCR[1:0]** : 新しいコントローラの動作状態 (I3C がアクティブなコントローラとして機能する場合)

このフィールドは、新しいターゲットとしてアクセスされる前の待機時間を示します。AVAL[7:0] を参照してください。

このレジスタは、I3C がコントローラとして機能する場合にのみ変更できます。

ビット 7:0 **AVAL[7:0]** : 1 μ s の時間単位をセットするためのカーネルクロックサイクル数 (I3C がコントローラ/ターゲットのどちらで機能する場合でも)。

この時間単位は、以下の MIPI I3C タイミングに対応する一部の内部タイマを構築するために、ハードウェアによって使用されます。

I3C がターゲットとして機能する場合 :

1. バス使用可能条件時間 : STOP 後、IBI またはコントローラ機能リクエストの開始リクエストを発行する前に、(バス使用可能条件) 時間が経過するのを待つ必要があります (バス使用可能条件は、少なくとも t_{AVAL} までは維持されます)。MIPI のタイミング $t_{\text{AVAL}} = 1 \mu\text{s}$ を参照してください。タイミングの定義方法 : $t_{\text{AVAL}} = (\text{AVAL}[7:0] + 2) \times t_{\text{I3CCCLK}}$
2. バスアイドル条件時間 : SDA と SCL の両方が、ホットジョイン・イベントを発行する前にハイで安定した状態が継続した後で、(バスアイドル条件) 時間が経過するのを待つ必要があります。MIPI v1.1 のタイミング $t_{\text{IDLE}} = 200 \mu\text{s}$ を参照してください。タイミングの定義方法 : $t_{\text{IDLE}} = (\text{AVAL}[7:0] + 2) \times 200 \times t_{\text{I3CCCLK}}$

I3C がコントローラとして機能する場合、以下のように、最大ストール時間を超えたクロックのストールはできません (SCL クロックロー)。

1. 動的なアドレスの割り当て時に割り当てられたアドレスの最初のビット : MIPI のタイミング $t_{\text{STALLDAA}} = 15 \text{ ms}$ を超えたクロックのストールはできません。タイミングの定義方法 : $t_{\text{STALLDAAmax}} = (\text{AVAL}[7:0] + 1) \times 15000 \times t_{\text{I3CCCLK}}$
2. I3C/I²C 転送の ACK/NACK フェーズ、書込みデータ転送のパリティビット、I3C 読出し転送の遷移ビット : MIPI のタイミング $t_{\text{STALL}} = 100 \mu\text{s}$ を超えたクロックのストールはできません。タイミングの定義方法 : $t_{\text{STALLmax}} = (\text{AVAL}[7:0] + 1) \times 100 \times t_{\text{I3CCCLK}}$

I3C がコントローラ/ターゲット (コントローラ対応) のどちらで機能する場合でも、コントローラ機能のハンドオフ手順中に以下を行います。

1. 新しいコントローラは、GETACCR CCC 完了後、SDA ローにプルダウン (START を発行) する前に $t_{\text{NEWCRLOCK}}$ を待つ必要があります。その後、新しいコントローラは SCL クロックを有効にするために t_{CAS} 以内に SCL ローにプルダウンすることができます。有効なコントローラは、新しいコントローラが SDA ローにプルダウンすることでバスを制御できるようになったかをテストする前に、同じ $t_{\text{NEWCRLOCK}}$ 時間、または少なくとも $100 \mu\text{s}$ 待つ必要があります。待機時間は I3C_TIMINGR1 レジスタの ANSCR[1:0] によって異なります。
 - ASNCR[1:0] = 00 : $t_{\text{NEWCRLOCK}} = (\text{AVAL}[7:0] + 1) \times t_{\text{I3CCCLK}}$
 - ASNCR[1:0] = 01 : $t_{\text{NEWCRLOCK}} = (\text{AVAL}[7:0] + 1) \times 100 \times t_{\text{I3CCCLK}}$
 - ASNCR[1:0] = 10 : $t_{\text{NEWCRLOCK}} = (\text{AVAL}[7:0] + 1) \times 2000 \times t_{\text{I3CCCLK}}$
 - ASNCR[1:0] = 11 : $t_{\text{NEWCRLOCK}} = (\text{AVAL}[7:0] + 1) \times 50000 \times t_{\text{I3CCCLK}}$

49.16.22 I3C タイミングレジスタ 2 (I3C_TIMINGR2)

アドレスオフセット : 0x0A8

リセット値 : 0x0000 0000

I3C がコントローラとして機能する場合、このレジスタはアドレス指定された I3C または従来の I²C ターゲットが SCL クロックローのストールの有効化を必要とするときに、SCL クロックのストールを設定し、有効にするために使用されます。

このレジスタは、I3C がコントローラとして機能する場合にのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STALL[7:0]								Res.	Res.	Res.	Res.	STALLA	STALLC	STALLD	STALLT
rW	rW	rW	rW	rW	rW	rW	rW					rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **STALL[7:0]** : コントローラクロックのストール時間（カーネルクロックサイクル数）

$$t_{\text{SCLL_STALL}} = \text{STALL} \times t_{\text{I3CCLK}}$$

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **STALLA** : ACK フェーズでのコントローラクロックのストールの有効化

SCL は、アドレス ACK/NACK フェーズ（9 番目のビットの前）で（STALL で定義された $t_{\text{SCLL_STALL}}$ 中に）ストールします。これにより、ターゲットはデータを準備できます。ターゲットによるリクエスト（IBI/HJ/CR）でコントローラ自体によって駆動された ACK は、この制御ビットの影響を受けません。

0 : ストールなし

1 : ストールを有効化

ビット 2 **STALLC** : CCC の PAR フェーズでのコントローラクロックのストールの有効化

SCL は、コモン・コマンド・コードの T ビット・フェーズの STALL $\times t_{\text{SCLL_PP}}$ 中（9 番目のビットの前）にストールします。これにより、ターゲットはコマンドをデコードできます。

0 : ストールなし

1 : ストールを有効化

注 : $t_{\text{SCLL_PP}} = (\text{I3C_TIMINGR0.SCLL_PP}[7:0] + 1) \times t_{\text{I3CCLK}}$

ビット 1 **STALLD** : データの PAR フェーズでのコントローラクロックのストールの有効化

SCL は、T ビット・フェーズの STALL $\times t_{\text{SCLL_PP}}$ 中（9 番目のビットの前）にストールします。これにより、ターゲットは受信データを読み出せます。

0 : ストールなし

1 : ストールを有効化

注 : $t_{\text{SCLL_PP}} = (\text{I3C_TIMINGR0.SCLL_PP}[7:0] + 1) \times t_{\text{I3CCLK}}$

ビット 0 **STALLT** : データの T ビット・フェーズ（および従来の I²C 読出しのデータバイトの ACK/NACK フェーズ）でのコントローラクロックのストールの有効化

SCL は、T ビット・フェーズの STALL $\times t_{\text{SCLL_PP}}$ 中（9 番目のビットの前）にストールします。これにより、ターゲットは送信データを準備できます。

0 : ストールなし

1 : ストールを有効化

注 : $t_{\text{SCLL_PP}} = (\text{I3C_TIMINGR0.SCLL_PP}[7:0] + 1) \times t_{\text{I3CCLK}}$

49.16.23 I3C バス特性レジスタ (I3C_BCR)

アドレスオフセット : 0x0C0

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは GETBCR または ENTDAACCC の受信時にデータバイト BCR[7:0] を返すためにハードウェアが使用する 3 つのビットを設定するために使用されます。I3C バスの返されたバイト BCR[7:0] は、次のとおりです。

- BCR[7] = 0 (予約済み)
- BCR[6] = I3C_BCR6 (コントローラ対応)
- BCR[5] = 1 (先進機能、GETCAPS CCC を使用して選択)
- BCR[4] = 0 (仮想ターゲットではない)
- BCR[3] = 1 (オフライン対応)
- BCR[2] = I3C_BCR2 (IBI ペイロード)
- BCR[1] = 1 (IBI リクエスト対応)
- BCR[0] = I3C_BCR0 (最大データ速度制限)

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCR6	Res.	Res.	Res.	BCR2	Res.	BCR0
									rw				rw		rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **BCR6** : コントローラ対応

0 : I3C ターゲット (コントローラ非対応)

1 : I3C コントローラに対応

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **BCR2** : 帯域内割込み (IBI) ペイロード

0 : 許可された IBI に続くデータバイトなし

1 : 許可された IBI に続く 1 以上の必須のデータバイト (最大 4 データバイト)

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **BCR0** : 最大データ速度制限

0 : 制限なし

1 : 制限あり、I3C_GETMXDSR を参照

49.16.24 I3C デバイス特性レジスタ (I3C_DCR)

アドレスオフセット : 0x0C4

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは GETDCR、ENTDAA、または DEFTGTS CCC の受信時にデータバイト DCR[7:0] を返すためにハードウェアが使用するデバイス特性 ID を設定するために使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DCR[7:0]** : デバイス特性 ID

0X00 : 汎用デバイス (v1.0 デバイス用)

上記以外 : I3C センサ/デバイスのタイプについて説明する ID

注 : 最新の MIPI DCR ID の割り当てについては、<https://www.mipi.org> をご確認ください。

49.16.25 I3C 取得のケーパビリティ・レジスタ (I3C_GETCAPR)

アドレスオフセット : 0x0C8

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタはペンディング中の読出し通知サポートに IBI MDB サポートをセットするために使用され、またフォーマット 1 の GETCAPS CCC の受信時に GETCAP3 バイトを返すためにハードウェアによって使用されます。

I3C バスの返されたバイト GETCAP1[7:0] は、次のとおりです。

- GETCAP1[7:0] = 0 (HDR なし)

I3C バスの返されたバイト GETCAP2[7:0] は、次のとおりです。

- GETCAP2[7:6] = 00 (HDR なし)
- GETCAP2[5:4] = 00 (グループアドレスなし)
- GETCAP2[3:0] = 0001 (MIPI 仕様 v1.1 と一致)

I3C バスの返されたバイト GETCAP3[7:0] は、次のとおりです。

- GETCAP3[7] = 0 (予約済み)
- I3C_GETCAPR レジスタで GETCAP3[6] = CAPPEND (ペンディング中の読出し通知の IBI MDB をサポート)
- GETCAP3[5] = 0 (HDR なし)
- GETCAP3[4] = 1 (GETSTATUS で定義バイトをサポート)
- GETCAP3[3] = 1 (GETCAPS で定義バイトをサポート)
- GETCAP3[2] = 0 (デバイス間転送なし)
- GETCAP3[1] = 0 (デバイス間転送なし)
- GETCAP3[0] = 0 (マルチレーン・データ転送なし)

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CAPP END	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw														

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **CAPPEND** : ペンディング中の読出し通知のための IBI MDB サポート

このビットはバスの初期化中 (EN = 0) にソフトウェアによって書き込まれ、IBI MDB[7:0] 値によるペンディング中の読出し通知のサポートの有無を示します。

このビットは、GETCAPS CCC のフォーマット 1 に応じて GETCAP3 バイトを返すために使用されます。

0 : この I3C は、ターゲットとして機能する場合、必須のデータバイト値なしで IBI リクエストを送信し、読出し通知がペンディング中であることを示します。

1 : この I3C は、ターゲットとして機能する場合、必須のデータバイト値 (MDB[7:5] = 101) とともに IBI リクエストを送信し、読出し通知がペンディング中であることを示します。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

49.16.26 I3C コントローラ機能のケーパビリティ・レジスタ (I3C_CRCAPR)

アドレスオフセット : 0x0CC

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは、コントローラ機能のハンドオフ後に、2 次コントローラとして I3C がサポートする機能をセットするために使用されます。また、定義バイトが CRCAPS (0x91) のフォーマット 2 の GETCAPS CCC を受信したときに、CRCAP1 バイトと CRCAP2 バイトを返すために、ハードウェアによって使用されます。

I3C バスの返された CRCAP1[7:0] は、次のとおりです。

- CRCAP1[7:3] = 00000 (予約済み)
- CRCAP1[2] = 0 (マルチレーンなし)
- I3C_CRCAPR レジスタで CRCAP1[1] = CAPGRP (グループ管理)
- CRCAP1[0] = 1 (ホットジョイン)

I3C バスの返された CRLCAP2[7:0] は、次のとおりです。

- CRCAP2[7:4] = 0000 (予約済み)
- I3C_CRCAPR レジスタで CRCAP2[3] = CAPDHOFF (コントローラ機能ハンドオフの遅延)
- CRCAP2[2] = 1 (ディープスリープ可能)
- CRCAP2[1] = 0 (コントローラ機能の返送なし)
- CRCAP2[0] = 1 (IBI ack 可能)

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CAP GRP	Res.	Res.	Res.	Res.	Res.	CAP DHOFF	Res.	Res.	Res.
						rw						rw			

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CAPGRP** : グループ管理サポート (コントローラとして機能する場合)

このビットはバスの初期化中 (EN = 0) にソフトウェアによって書き込まれ、発行された DEFGRPA、RSTGRPA、および SETGRPA CCC により、I3C がコントローラとして機能する場合にグループ管理をサポートできるかどうかを示します (コントローラ機能のハンドオフ後)。

このビットは、GETCAPS CCC のフォーマット 2 に応じて CRCAP1 バイトを返すために使用されます。

0 : この I3C ではグループアドレス機能をサポートしない

1 : この I3C ではグループアドレス機能をサポートする (コントローラになる場合)

ビット 8:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CAPDHOFF** : 遅延したコントローラ機能のハンドオフ

このビットはバスの初期化中 (EN = 0) にソフトウェアによって書き込まれ、現在使用中のコントローラによってリクエストされたコントローラ機能のハンドオフを処理するために、ターゲット I3C が追加の時間を必要とするかどうかを示します。

このビットは、GETCAPS CCC のフォーマット 2 に応じて CRCAP2 バイトを返すために使用されます。

0 : この I3C では、コントローラ機能のハンドオフを処理するための追加の時間を必要としません。

1 : この I3C では、コントローラ機能のハンドオフを処理するための追加の時間が必要です。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

49.16.27 I3C 最大データ速度の取得レジスタ (I3C_GETMXDSR)

アドレスオフセット : 0x0D0

リセット値 : 0x0000 0000

I3C がターゲットとして機能する場合、このレジスタは機能や制限をセットするために使用されます (存在する場合)。このレジスタは、フォーマット 1 (2 バイト、maxRdTurn なし)、フォーマット 2 (5 バイト、MaxRdTurn あり)、またはフォーマット 3 (定義バイト WRRDTURN = 0x00 あり : フォーマット 2 と同じ 5 バイト、または定義バイト CRHDLY = 0x91 あり : シングルバイト CRHDLY1) の GETMXDS CCC を受信したときに、データバイトを返すためにハードウェアによって使用されます。

I3C バスの返されたバイト MaxWr[7:0] は、次のとおりです。

- MaxWr[7:4] = 0000 (予約済み)
- MaxWr[3] = 1 (定義バイト WRRDTURN および CRHDLY をサポート)
- MaxWr[2:0] = 000 (コントローラによってターゲットに送信される非 CCC メッセージの最大維持データレートは 12.5 MHz で動作するように設計されている)

I3C バスの返されたバイト MaxRd[7:0] は、次のとおりです。

- MaxRd[7] = 0 (予約済み)
- MaxRd[6] = 1 (書込みと読出しの間で停止可能)
- I3C_GETMXDSR レジスタで TSCO = 0 の場合 MaxRd[5:3] = 100 (クロックからデータへのターンアラウンド時間は $t_{SCO} \leq 12$ ns)、それ以外の場合は 111 ($t_{SCO} > 12$ ns、詳細についてはデータシートを参照)
- MaxRd[2:0] = 000 (ターゲットによってコントローラに送信される非 CCC メッセージの最大維持データレートは 12.5 MHz で動作するように設計されている)

I3C_GETMXDSR レジスタで FMT[1:0] = 00 の場合、I3C バスで返される 3 バイト MaxRdTurn[23:0] は、次のとおりです。

- MaxRdTurn[23:0] は、I3C_GETMXDSR レジスタの RDTURN[7:0] で、最上位ビット (65 ms ~ 16 s)、中間バイト (256 μ s ~ 65 ms)、または最下位ビット (256 μ s 未満) のいずれかで (他のビットは 0)、同じレジスタの FMT[1:0] に依存します。

I3C バスの返されたバイト CRHDLY1[7:0] は、次のとおりです。

- CRHDLY1[7:3] = 00000 (予約済み)
- I3C_GETMXDSR レジスタで HOFFAS[1:0] = 00 の場合は CRHDLY1[2] = 0、それ以外の場合は 1 (バス動作状態をセット)
- CRHDLY1[1:0] = HOFFAS[1:0] (コントローラ機能のハンドオフ動作状態)

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCO	RDTURN[7:0]							
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FMT[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	HOFFAS[1:0]	
						rw	rw							rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TSCO** : クロックからデータへのターンアラウンド時間 (t_{SCO})

このビットはバスの初期化中 (I3C_CFGR レジスタで EN = 0) にソフトウェアによって書き込まれ、クロックからデータへのターンアラウンド時間 t_{SCO} (およびそれに対する値 12 ns) を指定するために使用されます。このビットは、返された MaxRd[5:3] ビット経由でエンコードされたロックからデータへのターンアラウンド時間を返すために、ハードウェアによって GETMXDS CCC に応じて使用されます。

0 : $t_{SCO} \leq 12$ ns

1 : $t_{SCO} > 12$ ns (詳細についてはデータシートを参照)

ビット 23:16 **RDTURN[7:0]**: MaxRdTurn の 3 バイトのプログラムされたバイト (最大読出しターンアラウンド・バイト)

このビットはバスの初期化中 (EN = 0) にソフトウェアによって書き込まれ、3 バイトの MaxRdTurn の選択されたバイトの値を書き込みます (FMT[1:0] フィールド経由)。これは、最大読出しターンアラウンド時間をエンコードするために、GETMXDS CCC フォーマット 2 に応じて返されます。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **FMT[1:0]** : GETMXDS CCC フォーマット

このフィールドはバスの初期化 (EN = 0) 中にソフトウェアによって書き込まれ、フォーマット 1 (MaxRdTurn なし) およびフォーマット 2 (MaxRdTurn あり) の GETMXDS がどのように返されるかを示します。

このフィールドは、GETCAPS CCC に応じて 2 バイトのフォーマット 1 (MaxWr, MaxRd) または 5 バイトのフォーマット 2 (MaxWr, MaxRd, 3 バイトの MaxRdTurn) バイトを返すために使用されます。

00 : フォーマット 1 (2 バイト、定義バイトなしの MaxWr, MaxRd)

01 : フォーマット 2 : (5 バイト、定義バイトなしの MaxWr, MaxRd, MaxRdTurn)

- 3 バイトの MaxRdTurn が、最上位ビット = 0、中間バイト = 0、および最下位ビット = RDTURN[7:0] で返されます。

- 最大読出しターンアラウンド時間は 256 μ s 未満です。

10 : フォーマット 2 (5 バイト、定義バイトなしの MaxWr, MaxRd, および MaxRdTurn の中間バイト)

- 3 バイトの MaxRdTurn が、最上位ビット = 0、中間バイト = RDTURN[7:0]、および最下位ビット = 0 で返されます。

- 最大読出しターンアラウンド時間は 256 ~ 65535 μ s です。

11 : フォーマット 2 (5 バイト、定義バイトなしの MaxWr, MaxRd, および MaxRdTurn の最上位ビット)

- 3 バイトの MaxRdTurn が、最上位ビット = RDTURN[7:0]、中間バイト = 0、および最下位ビット = 0 で返されます。

- 最大読出しターンアラウンド時間は 65535 μ s ~ 16 s です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **HOFFAS[1:0]** : コントローラのハンドオフ動作状態

このフィールドは、バスの初期化 (EN = 0) 中に、ソフトウェアによって書き込まれます。定義バイト CRHDLY (0x91) を GETMXDS CCC に返す際に、(他の) 現在使用中のコントローラが、このコントローラ対応 I3C へのコントローラ機能のハンドオフ後に、I3C バスに期待する初期動作状態がどれかを示します。

この 2 ビットフィールドは、I3C がコントローラ機能のハンドオフ後にコントローラになった場合の動作状態に戻すために、また結果的に I3C の所有権を確認するテストの前に旧コントローラが待機する必要がある時間に戻すために、フォーマット 3 の GETCAPS CCC に応じて CRHDLY1 バイトを返すために使用されます。

00 : 動作状態 0 は、コントローラになる前となったときの、I3C の最初の動作状態です。

01 : 動作状態 1 は、コントローラになったときの、I3C の最初の動作状態です。

10 : 動作状態 2 は、コントローラになったときの、I3C の最初の動作状態です。

11 : 動作状態 3 は、コントローラになったときの、I3C の最初の動作状態です。

49.16.28 I3C 拡張された専用 ID レジスタ (I3C_EPIDR)

アドレスオフセット : 0xD4

リセット値 : 0x0208 0000

I3C がターゲットとして機能する場合、このレジスタは、ソフトウェアによる 4 ビットの MIPI インスタンス ID のセット、および 48 ビットの専用 ID に使用する他の固定ビットのセットに使用されます。これは、GETPID および ENTDAACCC の受信時に、ハードウェアによって 48 ビットの専用 ID に 6 バイトを返す際にも使用されます。

このレジスタは、I3C_CFGR レジスタで EN = 0 のとき、ソフトウェアによってのみ書き込むことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIPIMID[14:0]															IDTSEL
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIPIID[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw												

ビット 31:17 **MIPIMID[14:0]** : 15 ビットの MIPI の製造業者 ID

この読み出しフィールドは、15 ビットの STMicroelectronics MIPI ID (0x0104) です。

このフィールドは、48 ビットの専用 ID のビット [47:33] を表します。

ビット 16 **IDTSEL** : 専用 ID タイプ・セクタ

このフィールドは 0 にセットされます (ベンダ固定値)。

このフィールドは、48 ビットの専用 ID のビット [32] を表します。

注 : 専用 ID のビット [31:16] は 0 である場合があります。

ビット 15:12 **MIPIID[3:0]** : 4 ビットの MIPI のインスタンス ID

このフィールドは、1 つの I3C バスで、特定の番号を持つこの I3C IP の各インスタンスを個別にセットし、識別するために、ソフトウェアによって書き込まれます。

このフィールドは、48 ビットの専用 ID のビット [15:12] を表します。

注 : 専用 ID のビット [11:0] は 0 である場合があります。

ビット 11:0 予約済みであり、リセット値に保持する必要があります。

49.16.29 I3C レジスタマップ

表 529. I3C レジスタマップとリセット値

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	I3C_CR	MEND	MTYPE[3:0]				Res.	Res.	Res.	ADD[6:0]				CCQ[7:0]				RNW	DCNT[15:0]															
	リセット値	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	I3C_CFGR	Res.	TSFSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFLUSH	CDMAEN	TMODE	SMODE	SFLUSH	SDMAEN	Res.	TXTHRES	TXFLUSH	TXDMAEN	Res.	RXTHRES	RXFLUSH	RXDMAEN	HJACK	Res.	HKSDAEN	EXITPTRN	RSTPTRN	NOARBH	CRINIT	EN	
	リセット値		0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x010	I3C_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDB0 [7:0]									
	リセット値																								0	0	0	0	0	0	0	0	0	0
0x014	I3C_RDWR	RDB3 [7:0]					RDB2 [7:0]					RDB1 [7:0]					RDB0 [7:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	I3C_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDB0 [7:0]										
	リセット値																								0	0	0	0	0	0	0	0	0	0
0x01C	I3C_TDWR	TDB3 [7:0]					TDB2 [7:0]					TDB1 [7:0]					TDB0 [7:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	I3C_IBIDR	IBIDB3 [7:0]					IBIDB2 [7:0]					IBIDB1 [7:0]					IBIDB0 [7:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	I3C_TGTTDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRELOAD	TGTTDCNT[15:0]																	
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	I3C_SR	MID[7:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIR	ABT	Res.	XDCNT[15:0]																
	リセット値	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	I3C_SER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DERR	DNACK	ANACK	COVR	DOVR	STALL	PERR	CODERR[3:0]				
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	I3C_RMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RADD[6:0]				Res.	RCODE[7:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	I3C_EVR	GRPF	DEFF	INTUPDF	ASUPDF	RSTF	MRLUPDF	MWLUPDF	DAUPDF	STAF	GETF	WKPF	Res.	HJF	CRUPDF	CRF	IBENDF	IBIF	Res.	Res.	Res.	Res.	ERRF	RXTGTENDF	FCF	Res.	RXLASTF	TXLASTF	RXFNEF	TXFNEF	SFNEF	CFNEF	TXFEF	CFEF
	リセット値	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	1	1

表 529. I3C レジスタマップとリセット値 (続き)

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x054	I3C_IER	GRPIE	DEFIE	INTUPDIE	ASUPDIE	RSTIE	MRLUPDIE	MWLUPDIE	DAUPDIE	STAI	GETIE	WKPIE	Res.	HJIE	CRUPDIE	CRIE	IBIENDIE	IBIE	Res.	Res.	Res.	Res.	ERRIE	RXTGTENDIE	FCIE	Res.	Res.	Res.	RXFNEIE	TXFNIE	SFNEIE	CFNIE	Res.	Res.			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	0	0	0	Res.	Res.	Res.	0	0	0	0	0	0	0		
0x058	I3C_CEV	CGRPF	CDEFF	CINTUPDF	CASUPDF	CRSTF	CMRLUPDF	CMWLUPDF	CDAUPDF	CSTAF	CGETF	CWKPF	Res.	CHJF	CRUPDF	CCRF	CIBIENDF	CIBIF	Res.	Res.	Res.	Res.	CERRF	CRXTGTENDF	CFCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	0	0	0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
0x05C	予約済みです。																																				
0x060	I3C_DEVR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTVAL	RSTACT	[1:0]	AS	[1:0]	HJEN	Res.	CREN	IBIEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DA[6:0]										DAVAL
	リセット値								0	0	0	0	0	0		0	0	0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0		
0x060 + 4 * x、 (x = 1~4)	I3C_DEVRx	DIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	IBIDEN	CRACK	IBIACK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DA[6:0]										DA[6:0]
	リセット値	0												0	0	0	0	0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0		
0x090	I3C_MAXRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IBIP	[2:0]	MRL[15:0]																		
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x094	I3C_MAXWLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MWL[15:0]																		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0A0	I3C_TIMINGR0	SCLH_I2C [7:0]					SCLL_OD[7:0]					SCLH_I3C [7:0]					SCLL_PP[7:0]																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0A4	I3C_TIMINGR1	Res.	Res.	Res.	SDA_HD	Res.	Res.	Res.	Res.	Res.	FREE[6:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ASNCR[1:0]	AVAL[7:0]												
	リセット値				0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0A8	I3C_TIMINGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AVAL[7:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0C0	I3C_BCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																									0											
0x0C4	I3C_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCR[7:0]											
	リセット値																								0	0	0	0	0	0	0	0	0	0	0		
0x0C8	I3C_GETCAPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPPEND	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																		0																		
0x0CC	I3C_CRCAPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAPGR	Res.	Res.	Res.	Res.	Res.	CAPDHOFF	Res.	Res.	Res.			
	リセット値																							0						0							

表 529. I3C レジスタマップとリセット値 (続き)

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D0	I3C_GETMXDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCO	RDTURN[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	FMT[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	HOFFAS [1:0]	
	リセット値								0	0	0	0	0	0	0	0	0							0	0							0	0
0x0D4	I3C_EPIDR	MIPIMID[14:0]															IDTSEL	MIPIID[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0												Res.

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。



50 ユニバーサル同期／非同期レシーバトランスミッタ (USART/UART)

このセクションでは、ユニバーサル同期非同期レシーバトランスミッタ (USART/UART) について説明します。

50.1 概要

USART を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟性の高い全二重データ交換を行うことができます。フラクショナルボーレートジェネレータによって、非常に広範囲のボーレートが達成できます。

USART は同期式単方向通信と半二重単線通信の両方のほか、LIN (Local Interconnection Network)、スマートカードプロトコル、IrDA (infrared data association) SIR ENDEC 仕様、およびモデム動作 (CTS/RTS) もサポートします。マルチプロセッサ通信もサポートされています。

マルチバッファ設定で DMA (直接メモリアクセス) を使用することによって、ハイスピードデータ通信が可能です。

50.2 USART の主な機能

- 全二重非同期通信
- NRZ 標準フォーマット (マーク／スペース)
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロック誤差の間の最適な調整を達成
- ボーレートジェネレータシステム
- データ送信および受信用の 2 つの内部 FIFO
各 FIFO はソフトウェアで有効／無効にすることができ、ステータスフラグを装備
- 共通にプログラム可能な送信および受信ボーレート
- PCLK から独立したペリフェラル専用のカーネルクロックによるデュアルクロックドメイン
- 自動ボーレート検出
- プログラム可能なデータワード長 (7 または 8 または 9 ビット)
- データ順序をプログラム可能 (MSB ファースト／LSB ファーストのシフト)
- 設定可能なストップビット (1 または 2 個のストップビット)
- 同期通信のための同期マスタ／スレーブモードとクロック出力／入力
- SPI スレーブ送信アンダーランエラーフラグ
- 単線半二重通信
- DMA を使用した連続通信
- 送受信バイトは集中型 DMA を使用して予約済み SRAM にバッファ
- トランスミッタとレシーバ用に個別の有効ビット
- 送信と受信の信号極性を個別に制御
- スワップ可能な Tx/Rx ピン設定
- モデムと RS-485 トランシーバのハードウェアフロー制御
- 通信制御／エラー検出フラグ

- パリティ制御 :
 - パリティビットの送信
 - 受信したデータバイトのパリティ検査
- フラグ付き割込みソース
- マルチプロセッサ通信: アイドルライン検出またはアドレスマーク検出によるミュートモードからのウェイクアップ
- STOP モードからのウェイクアップ

50.3 USART の拡張機能

- LIN マスタの同期ブレーク送信機能と LIN スレーブのブレーク検出機能
 - USART が LIN 用にハードウェア設定されている場合、13 ビットのブレーク生成と 10/11 ビットのブレーク検出
- 通常モードで 3/16 ビット期間をサポートする IrDA SIR エンコーダデコーダ
- スマートカードモード
 - ISO/IEC 7816-3 標準で定義されているスマートカードの T=0 および T=1 非同期プロトコルをサポート
 - スマートカード動作に 0.5 個および 1.5 個のストップビット
- Modbus 通信のサポート
 - タイムアウト機能
 - CR/LF キャラクタ認識

50.4 USART の実装

以下の表では、USART の実装について説明します。比較のため、表には LPUART も含めています。

表 530. STM32H563/H573および STM32H562 機能

USART のモード／機能	STM32H563/H573
USART1	フル
USART2	フル
USART3	フル
USART6	フル
USART10	フル
USART11	フル
UART4	基本
UART5	基本
UART7	基本
UART8	基本
UART12	基本
LPUART1	低消費電力

表 531. USART/LPUART の機能

USART のモード／機能 ⁽¹⁾	フル機能セット	基本機能セット	低消費電力機能セット
モデムのハードウェアフロー制御	X	X	X
DMA を使用した連続通信	X	X	X
マルチプロセッサ通信	X	X	X
同期モード（マスタ／スレーブ）	X	-	-
スマートカードモード	X	-	-
単線半二重通信	X	X	X
Ir SIR ENDEC ブロック	X	X	-
LIN モード	X	X	-
デュアルクロックドメイン	X	X	X
レシーバタイムアウト割込み	X	X	-
Modbus 通信	X	X	-
自動ボーレート検出	X	X	-
ドライバイネーブル	X	X	X
USART データ長	7、8、および 9 ビット		
Tx/Rx FIFO	X	X	X
Tx/Rx FIFO サイズ	8		
低消費電力モードからのウェイクアップ	X ⁽²⁾	X ⁽²⁾	X ⁽²⁾

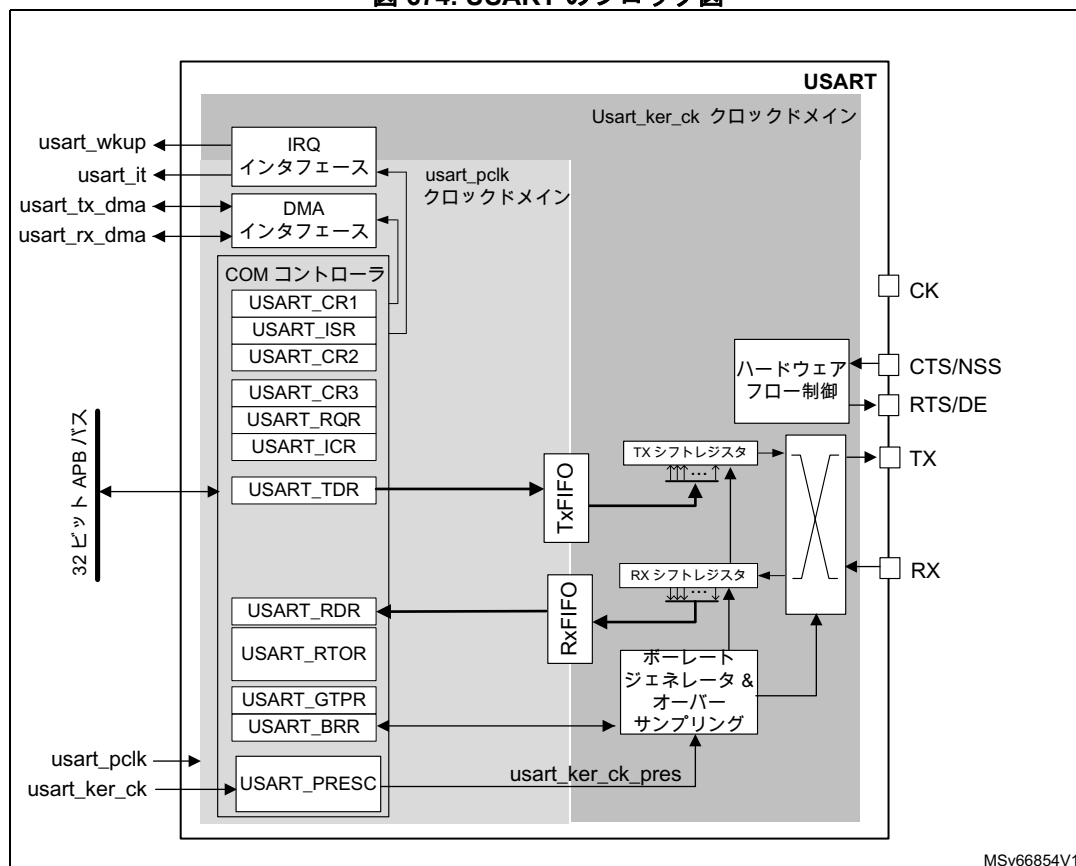
1. X : サポートされています。

2. STOP モードからのウェイクアップに対応

50.5 USART の機能説明

50.5.1 USART ブロック図

図 674. USART のブロック図



50.5.2 USART ピンおよび内部信号

USART の入出力ピンの説明

- USART 双方向通信

USART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力 (RX) と送信データ出力 (TX) です。

- RX (受信データ入力)

RX はシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用されています。それによって、有効な入力データとノイズを区別しています。

- TX (送信データ出力)

トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、データを送信する必要がないとき、TX ピンはハイになります。単線およびスマートカードモードでは、この I/O はデータの送受信に使用されます。

- RS232 ハードウェアフロー制御モード
RS232 ハードウェアフロー制御モードでは、以下のピンが必要です。
 - **CTS** (Clear To Send)
ハイレベルのとき、この信号は現在の転送の終わりにデータ送信をブロックします。
 - **RTS** (Request To Send)
ローレベルのとき、この信号は USART がデータを受信する準備ができたことを示します。
- RS485 ハードウェア制御モード
RS485 ハードウェア制御モードでは、**DE** (Driver Enable) ピンが必要です。この信号は、外部トランシーバの送信モードを有効にします。
- 同期マスタ／スレーブモードおよびスマートカードモード
同期マスタ／スレーブモードおよびスマートカードモードでは、以下のピンが必要です。
 - **CK**
このピンは同期マスタモードおよびスマートカードモードで、クロック出力として機能します。
同期スレーブモードでは、クロック入力として機能します。
同期マスタモードでは、このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します（スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します）。これと並行して、RX ピンでデータを同期受信できます。このメカニズムを使用して、シフトレジスタを持つペリフェラル（LCD ドライバなど）を制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。
スマートカードモードでは、CK 出力はスマートカードにクロックを供給します。
 - **NSS**
このピンは、同期スレーブモードでスレーブ選択入力として機能します。

USART 入出力ピンおよび内部信号のリストについては、表 532 および表 533 を参照してください。

表 532. USART の入出力ピン

ピン名	信号タイプ	説明
USART_RX	入力	シリアルデータ受信入力。
USART_TX	出力	送信データ出力。
USART_CTS	入力	送信可否
USART_RTS	出力	送信要求
USART_DE ⁽¹⁾	出力	ドライバイネーブル
USART_CK	出力	同期マスタモードおよびスマートカードモードでのクロック出力。
USART_NSS ⁽²⁾	入力	同期スレーブモードでのスレーブ選択入力。

1. USART_DE と USART_RTS は同じピンを共有します。

2. USART_NSS と USART_CTS は同じピンを共有します。

USART の入出力信号の説明

表 533. USART 内部入力／出力信号

ピン名	信号タイプ	説明
usart_pclk	入力	APB クロック
usart_ker_ck	入力	USART カーネルクロック
usart_wkup	出力	USART はウェイクアップ割込みに対応
usart_it	出力	USART グローバル割込み
usart_tx_dma	入力／出力	USART 送信 DMA リクエスト
usart_rx_dma	入力／出力	USART 受信 DMA リクエスト

50.5.3 USART クロック

図 674 に示した簡易ブロック図には、以下の完全に独立した2つのクロックドメインが示されています。

- **usart_pclk** クロックドメイン
usart_pclk クロック信号はペリフェラルバスインタフェースにクロック供給します。USART レジスタへアクセスが必要なとき、アクティブになっていなければなりません。
- **usart_ker_ck** カーネルクロックドメイン
usart_ker_ck は USART クロックソースです。usart_pclk とは独立で、RCC によって供給されます。したがって、usart_ker_ck クロックが停止しているときでも、USART レジスタは読み書きできます。
デュアルクロックドメイン機能がサポートされていないときは、usart_ker_ck クロックは usart_pclk クロックと同じになります。

usart_pclk と usart_ker_ck の間には何の制約也没有ありません。usart_ker_ck は usart_pclk より速くすることも遅くすることもできます。唯一の限界は、十分に速く通信を管理できるソフトウェアの能力です。

USART が SPI スレーブモードで動作するときは、USART は、外部マスタ SPI デバイスによって提供される外部 CK 信号から導き出されるシリアルインタフェースクロックを使用してデータフローを処理します。usart_ker_ck クロックは CK 入力のクロックより少なくとも 3 倍は速くなければなりません。

50.5.4 USART キャラクタの説明

ワード長は、USART_CR1 レジスタの M ビット (M0 : ビット 12 および M1 : ビット 28) をプログラムすることによって、7、8、または 9 ビットに設定できます (図 675 を参照)。

- 7 ビットのキャラクタ長 : M[1:0] = 10
- 8 ビットのキャラクタ長 : M[1:0] = 00
- 9 ビットのキャラクタ長 : M[1:0] = 01

注 : 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ボーレート (0x7F および 0x55 フレーム検出) はサポートされません。

デフォルトでは、信号 (TX または RX) はスタートビットの処理中ではロー状態です。また、ストップビットの処理中にはハイ状態です。

これらの値は、極性設定制御により、各信号について個別に反転できます。

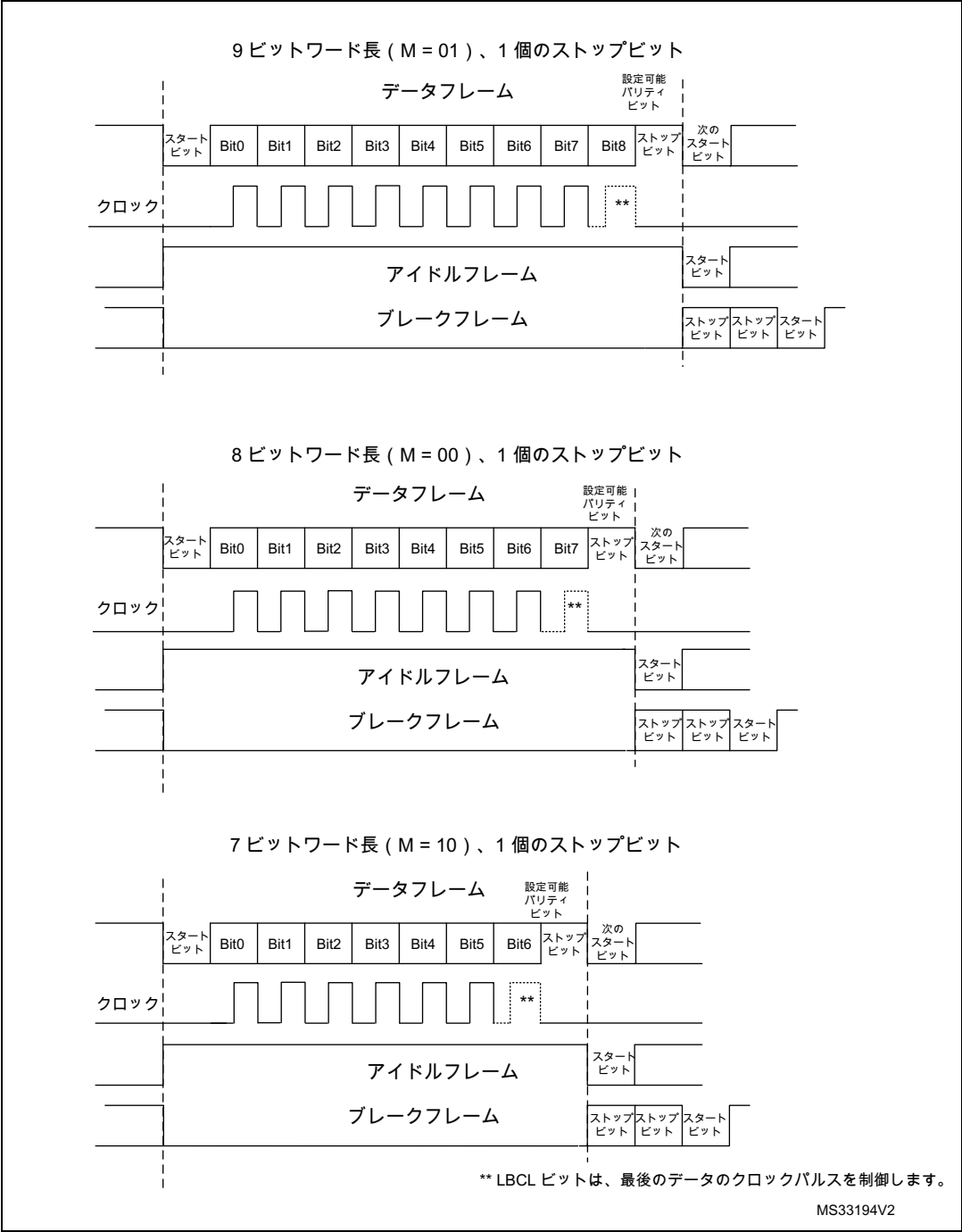
アイドルキャラクタは、すべてが「1」のフレームとして解釈されます (「1」の数にはストップビットの数が含まれます)。

ブ레이크キャラクタは、フレーム周期中に「0」を受信することと解釈されます。ブ레이크フレームの終了時、トランスミッタは 2 個のストップビットを挿入します。

送信と受信は共通ボーレートジェネレータによって駆動されます。送信および受信クロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 675. ワード長のプログラミング



50.5.5 USART の FIFO と閾値

USART は FIFO モードで動作できます。

USART は送信 FIFO (TXFIFO) と受信 FIFO (RXFIFO) を備えています。FIFO モードは、USART_CR1 レジスタの FIFOEN (ビット 29) をセットすることによって有効になります。このモードは、UART、SPI、およびスマートカードモードでのみ、サポートされます。

最大のデータワード長が 9 ビットなので、TXFIFO は 9 ビット幅です。しかし、RXFIFO のデフォルト幅は 12 ビットです。この理由は、レシーバは FIFO にデータを格納するだけでなく、各キャラクタに伴うエラーフラグ (パリティエラー、ノイズエラー、およびフレーミングエラーフラグ) も格納するためです。

注： 受信データは、対応するフラグとともに RXFIFO に格納されます。ただし、RDR を読み出すときは、データのみが読み出されます。

ステータスフラグは、USART_ISR レジスタで入手可能です。

Tx および Rx 割込みがトリガされる TXFIFO および RXFIFO のレベルを設定することができます。これらの閾値は、USART_CR3 制御レジスタの RXFTCFG および TXFTCFG ビットフィールドによってプログラムされます。

このとき、

- RXFIFO に受信したデータの数 RXFTCFG ビットフィールドでプログラムされた閾値に達すると、Rx 割込みが生成されます。
この場合、USART_ISR レジスタの RXFT フラグがセットされます。これは、RXFTCFG 分のデータが受信されたことを意味します。つまり、1 データが USART_RDR にあり、(RXFTCFG - 1) データが RXFIFO に入っています。例として、RXFTCFG が "101" にプログラムされている場合、FIFO サイズに相当する量のデータが受信された (FIFO サイズ - 1 のデータが RXFIFO に入り、1 データが USART_RDR に入った) 時に、RXFT フラグがセットされます。その結果、次に受信されるデータによってオーバーランフラグがセットされることはありません。
- TXFIFO 内の空き場所の数 TXFTCFG ビットフィールドでプログラムされた閾値に達すると、Tx 割込みが生成されます。

50.5.6 USART トランスミッタ

トランスミッタは、M ビットのステータスに応じて、7 または 8 または 9 ビットのデータワードを送信できます。トランスミッタ機能を有効にするには、送信イネーブルビット (TE) をセットする必要があります。送信シフトレジスタ内のデータは TX ピンで出力され、対応するクロックパルスは CK ピンで出力されます。

キャラクタ送信

USART 送信時、データは LSB ファースト (デフォルト設定) で TX ピンにシフトアウトされます。このモードでは、USART_TDR レジスタは、内部バスと送信シフトレジスタの間のバッファ (TDR) で構成されます。

FIFO モードが有効なとき、送信データレジスタ (USART_TDR) に書き込まれたデータは TXFIFO のキューに入ります。

各キャラクタの前には、スタートビット (1 ビット周期、ロー論理レベル) があります。キャラクタは、設定可能な数のストップビットで終端されます。

ストップビットの数は 0.5、1、1.5、または 2 に設定できます。

注： 送信データを USART_TDR に書き込む前に、TE ビットをセットする必要があります。
データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ボーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中の現在のデータは失われます。

TE ビットが有効になると、アイドルフレームが送信されます。

設定可能なストップビット

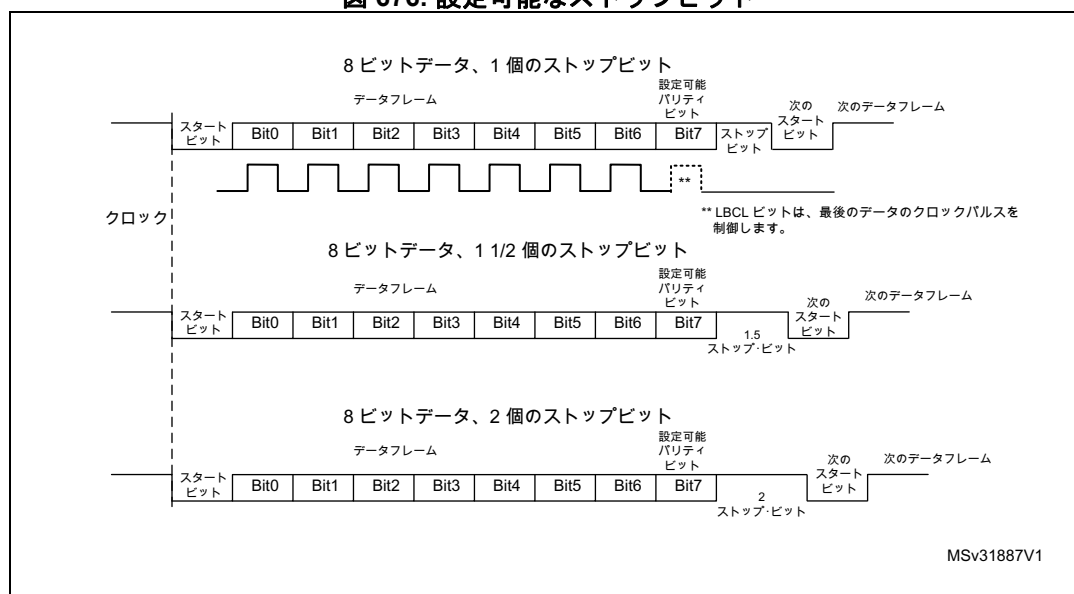
各キャラクタとともに送信されるストップビットの数は、USART_CR2 のビット 13、12 でプログラミングできます。

- **1 個のストップビット**：ストップビット数のデフォルト値です。
- **2 個のストップビット**：通常の USART モード、単線モード、およびモデムモードでサポートされます。
- **1.5 個のストップビット**：スマートカードモードで使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信は、10個のロービット (M[1:0] = "00" のとき) または 11個のロービット (M[1:0] = "01" のとき) または 9個のロービット (M[1:0] = "10" のとき) の後に 2個のストップビットが続きます (図 676 を参照)。長いブ레이크 (9/10/11 個のロービットを超える長さのブ레이크) を送信することはできません。

図 676. 設定可能なストップビット



キャラクタ送信手順

キャラクタを送信するには、次の手順に従います。

1. USART_CR1 の M ビットをプログラムして、ワード長を定義します。
2. USART_BRR レジスタを使用して、希望するボーレートを選択します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USART_CR3 レジスタの DMA 有効 (DMAT) を選択します。セクション 50.5.20 : USART および DMA を使用した連続通信の説明に基づいて、DMA レジスタを設定します。

6. USART_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
 7. 送信するデータを USART_TDR レジスタに書き込みます。シングルバッファの場合、送信される各データにこれを繰り返します。
 - FIFO モードが無効になっている場合は、USART_TDR に 1 つのデータを書き込むと、TXE フラグがクリアされます。
 - FIFO モードが有効になっている場合は、USART_TDR に 1 つのデータを書き込むと、1 つのデータが TXFIFO に追加されます。TXFNF フラグがセットされたとき、USART_TDR への書き込み動作が行われます。このフラグは TXFIFO がフルになるまでセットされたままになります。
 8. USART_TDR レジスタに最後のデータを書き込んだら、TC=1 になるまで待ちます。
 - FIFO モードが無効になっている場合、これは、最後のフレームの送信が完了したことを示します。
 - FIFO モードが有効になっている場合、これは、TXFIFO とシフトレジスタの両方が空になっていることを示します。
- このチェックは、USART が無効になったり、停止モードに入ったりするときに、最後の送信が壊れないようにするために必要です。

1 バイト通信

- FIFO モードが無効の場合

送信データレジスタに書き込むと、必ず TXE ビットがクリアされます。TXE フラグがハードウェアによってセットされます。同フラグは以下を示します。

 - データは USART_TDR レジスタからシフトレジスタへ移動され、データ送信が開始しています。
 - USART_TDR レジスタは空です。
 - 次のデータを、前のデータに上書きせずに、USART_TDR レジスタに書き込みます。

TXEIE ビットがセットされている場合、このフラグは割込みを生成します。

送信が行われているとき、USART_TDR レジスタへの書き込み命令によってデータが TDR バッファに格納されます。そして、データは現在の送信の最後にシフトレジスタにコピーされます。送信が行われていないときには、USART_TDR レジスタへの書き込み命令によってデータがシフトレジスタに格納され、データ送信が開始され、TXE ビットがセットされます。
- FIFO モードが有効になっている場合は、以下を示すためにハードウェアによって TXFNF (TXFIFO はフルではない) フラグがセットされます。
 - TXFIFO はフルではありません。
 - USART_TDR レジスタは空です。
 - 次のデータを、前のデータに上書きせずに、USART_TDR レジスタに書き込みます。送信が行われているとき、USART_TDR レジスタへの書き込み命令によってデータが TXFIFO に格納されます。現在の送信の最後にデータが TXFIFO からシフトレジスタにコピーされます。

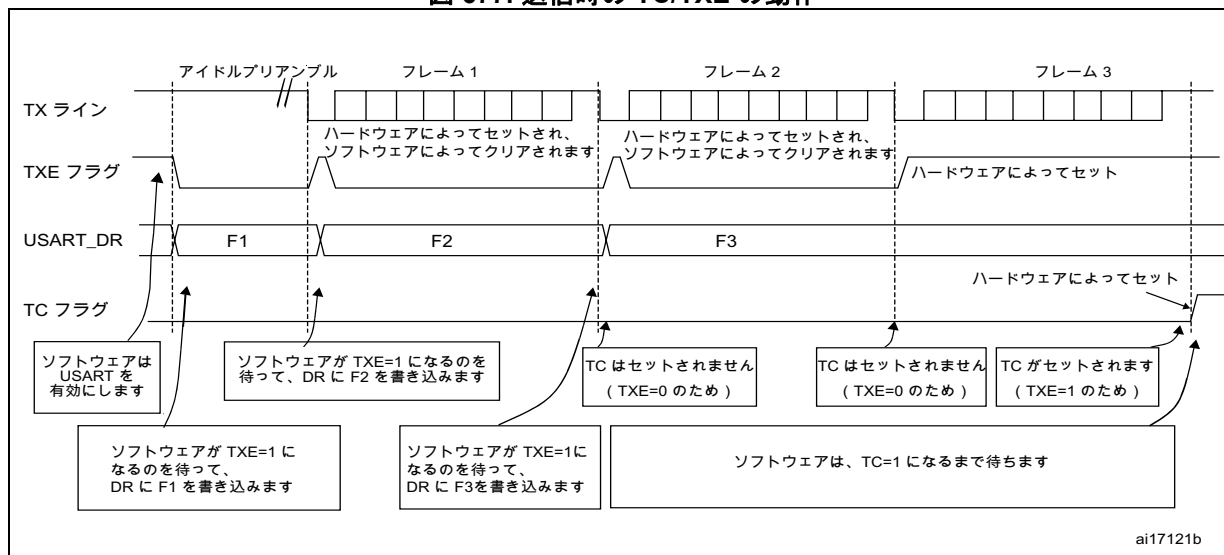
TXFIFO がフルではない場合、USART_TDR レジスタへの書き込み動作が終わった後も、TXFNF フラグは1のままに留まります。TXFIFO がフルになると、クリアされます。TXFNFIE ビットがセットされている場合、このフラグは割込みを生成します。

あるいは、TXFIFO 閾値に達した時、割込みが生成され、データを FIFO に書き込むことができます。この場合、CPU は、プログラムされたトリガレベルによって定義されたデータのブロックを書き込むことができます。

フレームが送信され (ストップビットの後)、TXE フラグ (FIFO モードの場合は TXFE) がセットされると、TC フラグはハイレベルになります。USART_CR1 レジスタの TCIE ビットがセットされると、割込みが生成されます。

USART_TDR レジスタに最後のデータを書き込んだ後は、USART を無効にしたり、マイクロコントローラを低消費電力モードにする前に TC がセットされるまで待つ必要があります (図 677: 送信時の TC/TXE の動作を参照)。

図 677. 送信時の TC/TXE の動作



注： FIFO 管理が有効になっているときは、TXFNF フラグがデータ送信のために使用されます。

ブレイクキャラクタ

SBKRQ ビットをセットすると、ブレイクキャラクタが送信されます。ブレイクフレーム長は、M ビットに依存します (図 675 を参照)。

SBKRQ ビットに 1 が書き込まれた場合、現在のキャラクタ送信の完了後、TX ラインにブレイクキャラクタが送信されます。SBKF ビットは書き込み操作によってセットされ、ブレイクキャラクタが完了すると (ブレイクキャラクタの後のストップビット時に) ハードウェアによってリセットされます。USART は、次のフレームのスタートビットの認識を保証するために、ブレイクフレームの終わりに 2 ビットの長さの間、ロジック 1 信号 (ストップ) を挿入します。

SBKRQ ビットがセットされると、現在のキャラクタ送信の最後に、ブレイクキャラクタが送信されます。

FIFO モードが有効になっている場合、TXFIFO がフルになっていても、ブレイクキャラクタの送信はデータ送信より優先されます。

アイドルキャラクタ

TE ビットをセットすると、USART は、最初のデータフレームの前にアイドルフレームを送信します。

50.5.7 USART レシーバ

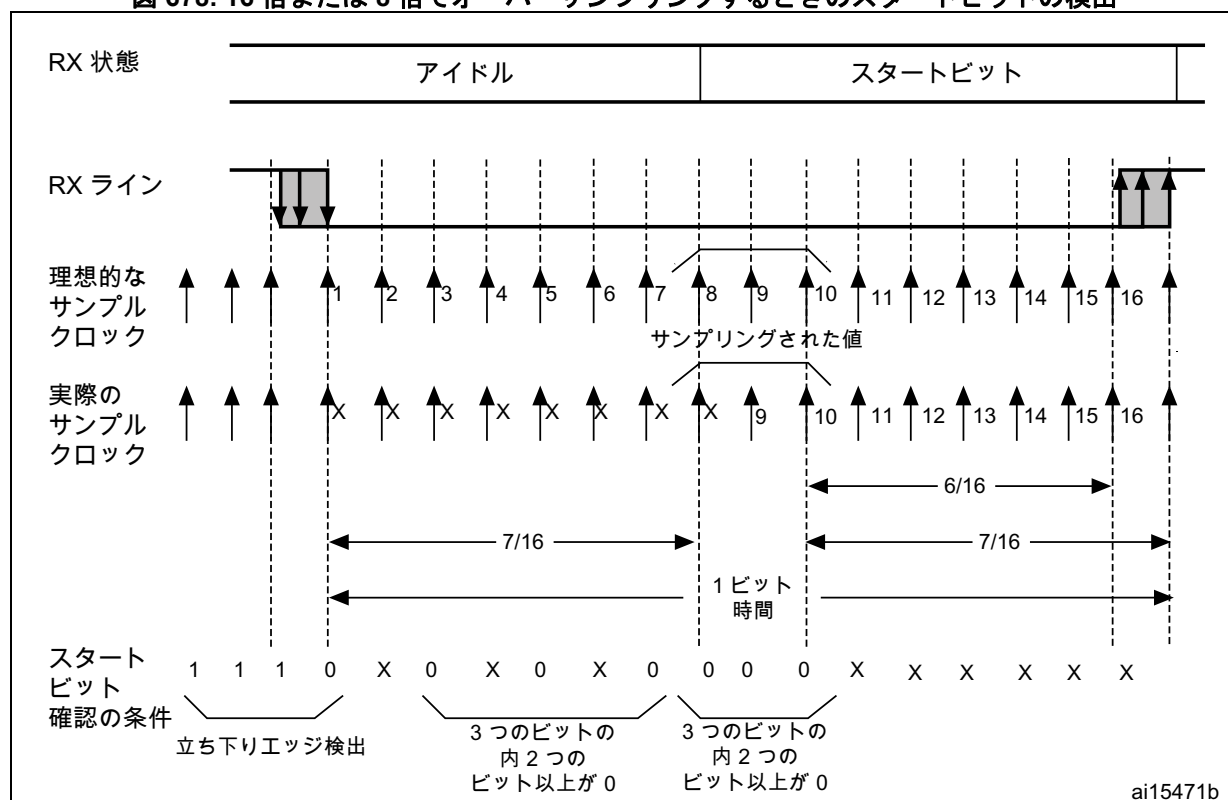
USART は、USART_CR1 レジスタの M ビットに応じて、7 または 8 または 9 ビットのデータワードを受信できます。

スタートビット検出

スタートビット検出シーケンスは、16 倍または 8 倍でオーバーサンプリングするときと同じです。

USART では、サンプルの特定シーケンスが認識されると、スタートビットが検出されます。このシーケンスは、1110X0X0X0X0X0X0です。

図 678. 16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出



注： シーケンスが完了していない場合、スタートビットの検出は中止され、レシーバはアイドル状態に
戻って（フラグはセットされません）、立ち下がりエッジを待ちます。

サンプリングされた 3 つのビットが“0”の場合（第 3、第 5、および第 7 ビットでの最初のサンプリングで 3 つのビットが“0”であり、第 8、第 9、および第 10 ビットでの 2 回目のサンプリングでも 3 つのビットが“0”）、スタートビットが確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割込みが生成されるか、または FIFO モードが有効な場合は、RXFNE フラグがセットされ、RXFNEIE=1 の場合は割込みが生成されます）。

次の場合、スタートビットは確認されますが、NE ノイズフラグがセットされます。

- a) 両方のサンプリングについて、サンプリングされた 3 ビットのうちの 2 ビットが 0 の場合 (第 3、第 5、および第 7 ビットのサンプリングと、第 8、第 9、および第 10 ビットのサンプリング)。

または

- b) いずれか一方のサンプリングで (第 3、第 5、および第 7 ビットでのサンプリングまたは第 8、第 9、および第 10 ビットでのサンプリング)、3 つのビットのうち 2 つが 0 の場合。

上記のいずれの条件も満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります (フラグはセットされません)。

キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファースト (デフォルトの設定) でシフトアウトされます。

キャラクタ受信手順

キャラクタを受信するには、次の手順に従います。

1. USART_CR1 の M ビットをプログラムして、ワード長を定義します。
2. ボーレートレジスタ USART_BRR を使用して、希望するボーレートを選択します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USART_CR3 レジスタの DMA 有効 (DMAR) を選択します。[セクション 50.5.20 : USART および DMA を使用した連続通信](#)の説明に基づいて、DMA レジスタを設定します。
6. USART_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- FIFO モードが無効の場合、RXNE ビットは、シフトレジスタの内容が RDR レジスタに転送されたことを示すためにセットされます。言い換えると、データは受信され、読み出し可能です (関連するエラーフラグも同様です)。
- FIFO モードが有効な場合、RXFIFO が空ではないことを示すために RXFNE ビットがセットされます。USART_RDR を読み出すと、RXFIFO に入力された最も古いデータが返されます。データが受信されると、対応するエラービットとともに RXFIFO に格納されます。
- RXNEIE (FIFO モードが有効な場合は RXFNEIE) ビットがセットされていた場合、割込みが生成されます。
- 受信中にフレームエラー、ノイズ、パリティ、またはオーバーランエラーが検出された場合、エラーフラグをセットできます。
- マルチバッファ通信モードでは、
 - FIFO モードが無効な場合は、RXNE フラグはバイトを受信するたびにセットされます。DMA が受信データレジスタを読み出すと、同フラグはクリアされます。
 - FIFO モードが有効な場合は、RXFIFO が空ではないときに RXFNE フラグがセットされます。DMA リクエストのたびに、RXFIFO から 1 データが取り出されます。DMA リクエストは、RXFIFO が空ではないとき、すなわち、RXFIFO から読み出されるべきデータがあるときに、トリガされます。
- シングルバッファモードでは、
 - FIFO モードが無効の場合、RXNE フラグのクリアは、ソフトウェアによる USART_RDR レジスタからの読み出しによって行われます。RXNE フラグは、USART_RQR レジスタの RXFRQ ビットに“1”をプログラムすることによってクリアすることもできます。オーバー

ランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE フラグをクリアする必要があります。

- FIFO モードが有効な場合は、RXFIFO が空ではないときに RXFNE がセットされます。USART_RDR からの読出し動作のたびに、RXFIFO から 1 つのデータが取り出されます。RXFIFO が空になると、RXFNE フラグがクリアされます。RXFNE フラグは、USART_RQR の RXFRQ ビットに“1”をプログラムすることによってクリアすることもできます。RXFIFO がフルのとき、オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXFIFO 内の最初のエントリを読み出す必要があります。RXFNEIE ビットがセットされている場合、RXFNE フラグは割込みを生成します。あるいは、RXFIFO 閾値に達した時、割込みが生成され、データを RXFIFO から読み出すことができます。この場合、CPU は、プログラムされた閾値によって定義されたデータのブロックを読み出すことができます。

ブレークキャラクタ

ブレークキャラクタを受信すると、USART はブレークキャラクタをフレーミングエラーとして処理します。

アイドルキャラクタ

アイドルフレームが検出された場合、データキャラクタ受信と同じように処理されますが、違いは、IDLEIE ビットがセットされている場合に割込みが生成されることです。

オーバーランエラー

- FIFO モードが無効の場合
RXNE ビットがリセットされていなくてキャラクタを受信すると、オーバーランエラーが発生します。
RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。RXNE フラグは、バイトを受信するたびにセットされます。
次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、
 - ORE ビットがセットされます。
 - RDR の内容は失われません。前回のデータは USART_RDR レジスタを読み出すことによって、入手可能です。
 - シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
 - RXNEIE ビットまたは EIE ビットがセットされている場合、割込みが生成されます。
- FIFO モードが有効な場合
シフトレジスタが転送される準備ができていて、受信 FIFO がフルのとき、オーバーランエラーが発生します。
RXFIFO に 1 つの空き場所ができるまで、データをシフトレジスタから USART_RDR レジスタに転送することはできません。RXFIFO が空でないとき、RXFNE フラグがセットされます。
RXFIFO がフルで、シフトレジスタが転送される準備ができている場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、
 - ORE ビットがセットされます。
 - RXFIFO の最初のエントリは失われません。それは、USART_RDR レジスタを読み出すことによって入手可能です。

- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXFNEIE ビットまたは EIE ビットがセットされている場合、割込みが生成されます。

ORE ビットは、USART_ICR レジスタの ORECF ビットをセットすることによってリセットされます。

注： ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。

FIFO モードが無効になっているとき、2 つの可能性があります。

- RXNE=1 の場合、有効な最後のデータは、受信レジスタ RDR に格納され、読出しが可能です。
- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR レジスタには読み出すべきものが残っていないことを意味します。このケースは、有効な最後のデータが RDR レジスタで読み出されると同時に新しい（そして失われた）データが受信されると発生します。

クロックソースと適切なオーバーサンプリング方式の選択

クロックソースの選択は、クロック制御システムを通じて行われます（セットおよびクロック制御 (RCC) のセクションを参照）。クロックソースは、UE ビットのセットによって USART を有効にする前に選ぶ必要があります。

クロックソースは、次の 2 つの基準に従って選択する必要があります。

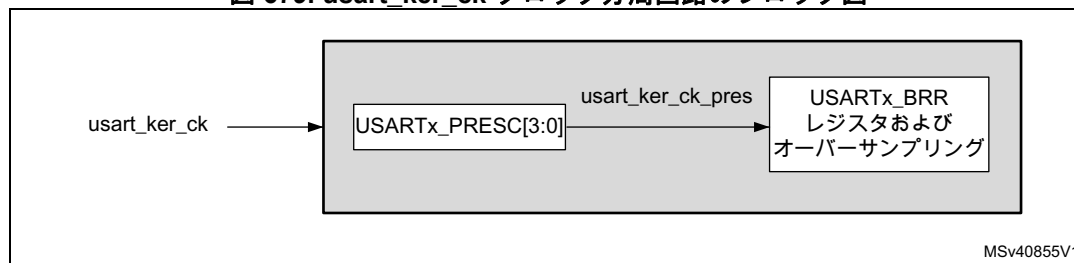
- USART を低消費電力モードで使用できること
- 通信速度

クロックソース周波数は、usart_ker_ck です。

デュアルクロックドメインと低消費電力モードからのウェイクアップ機能がサポートされるときには、usart_ker_ck クロックソースは RCC で設定できます（リセットおよびクロック制御 (RCC) のセクションを参照）。そうしない場合、usart_ker_ck クロックは usart_pclk と同じになります。

usart_ker_ck クロックは、USART_PRESC レジスタで定義されたプログラム可能な係数によって分周できます。

図 679. usart_ker_ck クロック分周回路のブロック図



usart_ker_ck ソースによっては、USART は MCU が低消費電力モードのときにデータを受信することができます。受信データと選択されたウェイクアップモードに応じて、USART より必要なときに MCU をウェイクアップし、ソフトウェアによる USART_RDR レジスタの読出しによって、または DMA によって受信データを転送できます。

他のクロックソースの場合、USART 通信を可能にするためには、システムをアクティブにする必要があります。

通信速度の範囲（特に最大通信速度）もクロックソースによって決まります。

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています（同期モードの場合を除く）。これにより、最大通信速度とノイズ／クロック精度の耐性との間の最適なトレードオフが図られます。

オーバーサンプリング方式は、USART_CR1 レジスタの OVER8 ビットをボーレートクロックの 16 倍または 8 倍にプログラムすることによって選択できます (図 680 および 図 681 を参照)。

アプリケーションに応じて、

- 高速 (最大 $\text{usart_ker_ck_pres}/8$) を達成するには 8 倍 (OVER8=1) のオーバーサンプリングを選択します。この場合、クロック偏差に対するレシーバの最大許容誤差は軽減されます (セクション 50.5.9 : 2219 ページのクロック偏差に対する USART レシーバの許容誤差 を参照)。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍 (OVER8=0) のオーバーサンプリングを選択します。この場合、最大速度は $\text{最大 usart_ker_ck_pres}/16$ に制限されます (ここで、usart_ker_ck_pres は USART 入力クロックをプリスケールで分周したものです)。

論理レベルの評価方法を選択するには、USART_CR3 レジスタの ONEBIT ビットをプログラミングします。次の 2 つのオプションが利用できます。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NE ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。

アプリケーションに応じて、

- ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式 (ONEBIT=0) を選択します。そしてノイズが検出された場合 (表 534 を参照) は、サンプリング中にグリッチが発生していることとなりますので、そのデータを除去します。
- ラインがノイズフリーであるときは、1 つのサンプルによる方式 (ONEBIT=1) を選択し、クロック偏差に対するレシーバの許容誤差を増やします (セクション 50.5.9 : 2219 ページのクロック偏差に対する USART レシーバの許容誤差 を参照)。この場合、NE ビットはセットされません。

フレーム内でノイズが検出された場合 :

- RXNE ビット (FIFO モードが有効な場合は RXFNE ビット) の立ち上がりエッジで、NE ビットがセットされます。
- 無効なデータがシフトレジスタから USART_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割込みは生成されません。ただし、このビットは、割込みを生成する RXNE ビット (FIFO モードが有効な場合は RXFNE ビット) と同時に立ち上がります。マルチバッファ通信の場合、USART_CR3 レジスタの EIE ビットがセットされている場合に割込みが発行されます。

NE ビットは、ICR レジスタの NCFE ビットをセットすることによってリセットされます。

注 : ノイズエラーは SPI モードではサポートされていません。

8 倍のオーバーサンプリングは、スマートカード、IrDA、および LIN の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって "0" に固定されています。

図 680. データサンプリング（16 倍のオーバーサンプリング）

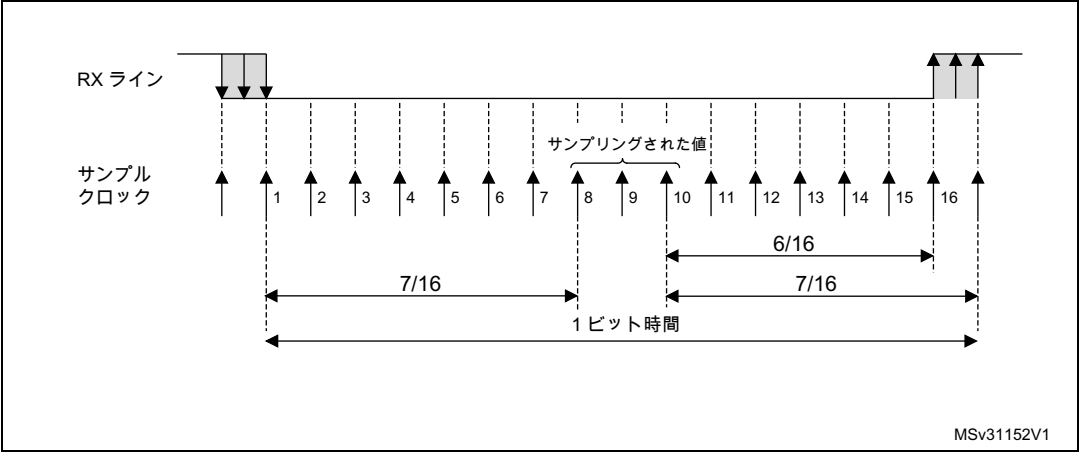


図 681. データサンプリング（8 倍のオーバーサンプリング）

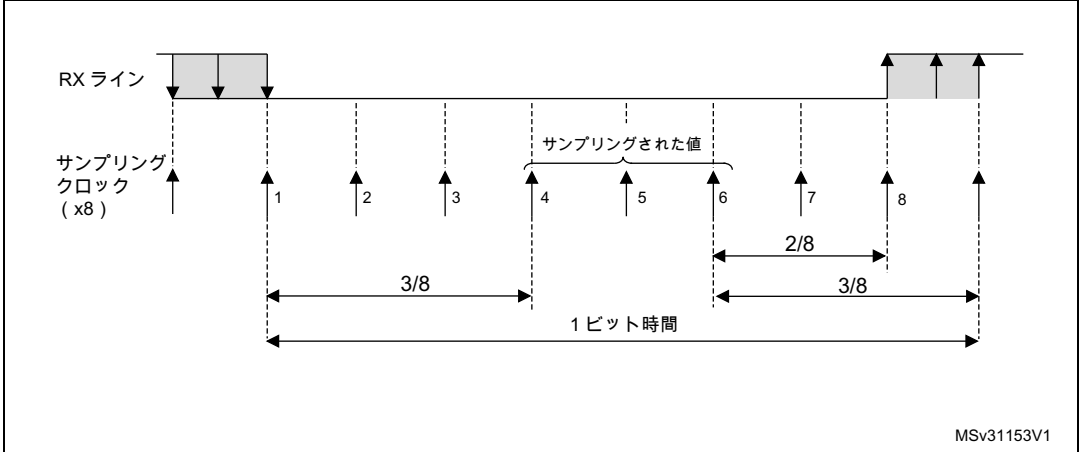


表 534. サンプリングされたデータからのノイズ検出

サンプリングされた値	NE ステータス	受信ビットの値
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

フレーミングエラー

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合、フレーミングエラーが検出されます。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USART_RDR レジスタ (FIFO モードが有効な場合は RXFIFO) へ転送されます。
- 1 バイト通信の場合、割込みは生成されません。ただし、このビットは、割込みを生成する RXNE ビット (FIFO モードが有効な場合は RXFNE ビット) と同時に立ち上がります。マルチバッファ通信の場合、USART_CR3 レジスタの EIE ビットがセットされている場合に割込みが発行されます。

USART_ICR レジスタの FECF に 1 を書き込むことによって、FE ビットがリセットされます。

注： フレーミングエラーは SPI モードではサポートされていません。

受信時の設定可能なストップビット

受信するストップビット数は、USART_CR の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

- **0.5 個のストップビット (スマートカードモードでの受信)：**0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
- **1 個のストップビット：**ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
- **1.5 個のストップビット (スマートカードモード)**

スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USART_CR1 の RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。

パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE フラグ (FIFO モードが有効な場合は RXFNE フラグ) を通して FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボークロック周期後に) 行われます。1.5 個のストップビットは、2 つの部分に分解できます。すなわち、何も起こらない 0.5 ボークロック周期と、途中でサンプリングが行われる通常の 1 ストップビット周期です (詳細は [セクション 50.5.17：2231 ページの USART レシーバタイムアウト](#) を参照)。

- **2 個のストップビット**
ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。
最初のストップビット中にフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。
2 番目のストップビットでは、フレーミングエラーの検査は行われません。最初のストップビットの最後に、RXNE フラグ (FIFO モードが有効な場合は RXFNE フラグ) がセットされます。

50.5.8 USART ボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートは、USART_BRR レジスタでプログラムされた値に設定されます。

式 1 : 標準 USART のボーレート (SPI モードを含む) (OVER8 = “0”または“1”)

16 倍のオーバーサンプリングの場合、ボーレートは次の計算式で与えられます。

$$\text{Tx/Rx ボー} = \frac{\text{usart_ker_ck_pres}}{\text{USARTDIV}}$$

8 倍のオーバーサンプリングの場合、ボーレートは次の計算式で与えられます。

$$\text{Tx/Rx ボー} = \frac{2 \times \text{usart_ker_ck_pres}}{\text{USARTDIV}}$$

式 2 : スマートカード、LIN、および IrDA モードのボーレート (OVER8 = 0)

ボーレートは次の計算式で与えられます。

$$\text{Tx/Rx ボー} = \frac{\text{usart_ker_ck_pres}}{\text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USART_BRR レジスタでコード化されます。

- OVER8 = 0 のとき、BRR = USARTDIV です。
- OVER8 = 1 のとき、
 - BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。
 - BRR[3] は、クリアされたままにする必要があります。
 - BRR[15:4] = USARTDIV[15:4]

注 : ボーカウンタは、USART_BRR への書き込み後、ボーレジスタの新しい値によって更新されます。したがって、通信中はボーレートレジスタの値を変更しないでください。

16 倍および 8 倍のオーバーサンプリングの場合、USARTDIV は 16 以上である必要があります。

USART_BRR レジスタの値から USARTDIV を得る方法

例 1

usart_ker_ck_pres = 8 MHz で 9600 ボーを得るには、

- 16 倍のオーバーサンプリングの場合 :
 - USARTDIV = 8 000 000/9600
 - BRR = USARTDIV = 0d833 = 0x0341
- 8 倍のオーバーサンプリングの場合 :
 - USARTDIV = 2 * 8 000 000/9600
 - USARTDIV = 1666,66 (0d1667 = 0x683)
 - BRR[3:0] = 0x3 >> 1 = 0x1
 - BRR = 0x681

例 2

usart_ker_ck_pres = 48 MHz で 921.6 キロボーを得るには、

- 16 倍のオーバーサンプリングの場合 :
 $\text{USARTDIV} = 48\,000\,000 / 921\,600$
 $\text{BRR} = \text{USARTDIV} = 0x52 = 0x34$
- 8 倍のオーバーサンプリングの場合 :
 $\text{USARTDIV} = 2 * 48\,000\,000 / 921\,600$
 $\text{USARTDIV} = 104$ (0d104 = 0x68)
 $\text{BRR}[3:0] = \text{USARTDIV}[3:0] >> 1 = 0x8 >> 1 = 0x4$
 $\text{BRR} = 0x64$

50.5.9 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバは、クロックシステムの合計偏差が USART レシーバの許容誤差未満の場合のみ、正しく動作します。

合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタのローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$$\text{DTRA} + \text{DQUANT} + \text{DREC} + \text{DTCL} + \text{DWU} < \text{USART receiver 許容誤差}$$

ここで、

DWU は、低消費電力モードからのウェイクアップが使用されたときのサンプリングポイントの偏差によるエラーです。

M[1:0] = 01 の場合 :

$$\text{DWU} = \frac{t_{\text{WUUSART}}}{11 \times \text{Tbit}}$$

M[1:0] = 00 の場合 :

$$\text{DWU} = \frac{t_{\text{WUUSART}}}{10 \times \text{Tbit}}$$

M[1:0] = 10 の場合 :

$$\text{DWU} = \frac{t_{\text{WUUSART}}}{9 \times \text{Tbit}}$$

t_{WUUSART} は、スタートビットの立ち下がりエッジが検出され、クロック (ペリフェラルによるリクエスト) が動作可能状態になってペリフェラルに達し、レギュレータが動作可能状態になるまでの時間です。

USART レシーバは、次の設定に応じて、表 535 および 表 536 で指定された最大許容偏差まで、データを正しく受信できます。

- USART_CR1 レジスタの M ビットによって定義された 9、10 または 11 ビットのキャラクタ長
- USART_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- USART_BRR レジスタのビット BRR[3:0] が 0000 に等しいかどうか。
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USART_CR3 レジスタの ONEBIT ビットの値に依存)。

表 535. BRR [3:0] = 0000 のときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT=1	ONEBIT = 0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 536. BRR[3:0] が 0000 でないときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT=1	ONEBIT = 0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注： 表 535 および 表 536 で指定されたデータは、M ビット = 00 のとき、受信フレームに正確に 10 ビット時間のアイドルフレームが含まれる特殊なケースで、若干異なることがあります (M = 01 のときには 11 ビット時間、または M = 10 のときには 9 ビット時間)。

50.5.10 USART 自動ボーレート検出

USART は、1 キャラクタの受信に基づいて、USART_BRR レジスタ値を検出し、自動的にセットすることができます。自動ボーレート検出は、2 つの状況で便利です。

- システムの通信速度が事前に分かっていないとき。
- システムが比較的低い精度のクロックソースを使用している場合、このメカニズムによって、クロック偏差を測定しなくても、正しいボーレートを求めることができます。

クロックソース周波数は、予期される通信速度と互換性がなければなりません。

- 16 倍のオーバーサンプリングのとき、ボーレートは usart_ker_ck_pres/65535 から usart_ker_ck_pres/16 までの範囲内です。
- 8 倍のオーバーサンプリングのとき、ボーレートは usart_ker_ck_pres/65535 から usart_ker_ck_pres/8 までの範囲内です。

自動ボーレート検出を有効にする前に、USART_CR2 レジスタの ABRMOD[1:0] フィールドによって自動ボーレート検出モードを選択する必要があります。キャラクタパターンに基づいて、4 つのモードがあります。これらの自動ボーレートモードでは、同期データ受信中にボーレートが数回測定され、各測定値が前回の測定値と比較されます。

これらのモードは以下のとおりです。

- **モード 0** : 1 のビットで始まるキャラクタ。
この場合、USART はスタートビットの時間を測定します（立ち下がりエッジから立ち上がりエッジまで）。
- **モード 1** : 10xx ビットパターンで始まるキャラクタ。
この場合、USART はスタートと最初のデータビットの時間を測定します。低速な信号スロープの場合の精度を高めるために、測定は立ち下がりエッジから立ち下がりエッジまでで行われます。
- **モード 2** : 0x7F キャラクタフレーム (LSB ファーストモードでは 0x7F キャラクタ、または MSB ファーストモードでは 0xFE)。
この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 6 の終了時に更新されます（立ち下がりエッジから立ち下がりエッジまで行われた測定に基づいて、BR6）。ビット 0 からビット 6 までは BRs でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。
- **モード 3** : 0x55 キャラクタフレーム。
この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 0 の終了時に更新されます（立ち下がりエッジから立ち下がりエッジまで行われた測定に基づいて、BR0）、最後にビット 6 (BR6) の終了時に更新されます。ビット 0 は BRs でサンプリングされ、ビット 1 からビット 6 までは BR0 でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。並行して、RX ラインの中間遷移ごとに別のチェックが行われます。RX の遷移がレシーバと十分に同期していない場合はエラーが生成されます（レシーバは、ビット 0 で計算されたボーレートに基づきます）。

自動ボーレート検出を有効にする前に、ゼロ以外のボーレート値を書き込むことによって、USART_BRR レジスタを初期化する必要があります。

自動ボーレート検出を有効にするには、USART_CR2 レジスタの ABREN ビットをセットします。USART は、RX ラインの最初のキャラクタを待ちます。自動ボーレート動作の完了は、USART_ISR レジスタの ABRF フラグのセットによって示されます。ラインにノイズが多い場合、正しいボーレート検出を保証できません。この場合、BRR 値が破損して、ABRE エラーフラグがセットされることがあります。また、通信速度が自動ボーレート検出の範囲と互換性がない場合にも発生します（ビット時間が 16 から 65536 までのクロック周期でなく（16 倍のオーバーサンプリング）、8 から 65536 までのクロック周期でない（8 倍のオーバーサンプリング））。

その後、ABRF フラグをリセットすることによって（"0" を書き込むことによって）、自動ボーレート検出を再開できます。

FIFO マネージメントが無効になっていて、自動ボーレートエラーが発生した場合、RXNE および FE ビットを通して ABRE フラグがセットされます。

FIFO マネージメントが有効になっていて、自動ボーレートエラーが発生した場合、RXFNE および FE ビットを通して ABRE フラグがセットされます。

FIFO モードが有効な場合、自動ボーレート検出は、RXFIFO の最初の場所のデータを使用して行う必要があります。したがって、自動ボーレート検出を開始する前に、USART_ISR レジスタの RXFNE フラグをチェックして RXFIFO が空であることを確認してください。

注 : 自動ボーレート操作中に USART が無効化された場合 (UE=0)、BRR 値が破損することがあります。

50.5.11 USART マルチプロセッサ通信

USART のマルチプロセッサ通信が可能です（ネットワーク内で複数の USART を接続して）。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。一方、他の USART は、それぞれの TX 出力の論理和をとってマスタの RX 入力に接続して、スレーブとなります。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードに入ります。ミュートモード機能を使用するためには、USART_CR1 レジスタの MME ビットをセットする必要があります。

注： FIFO マネージメントが有効になっていて MME がすでにセットされている場合は、MME ビットはクリアしてはなりません。クリアした場合はすぐに（2 usart_ker_ck サイクル以内に）再セットしてください。そうしないとミュートモードはアクティブのままになることがあります。

ミュートモードが有効な場合、

- 受信ステータスビットはセットできません。
- 受信割込みはすべて禁止されます。
- USART_ISR レジスタの RWU ビットは“1”にセットされます。特定の条件下では、USART_RQR レジスタの MMRQ ビットを通じて、RWU をハードウェアまたはソフトウェアによって自動的に制御できます。

USART は、USART_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

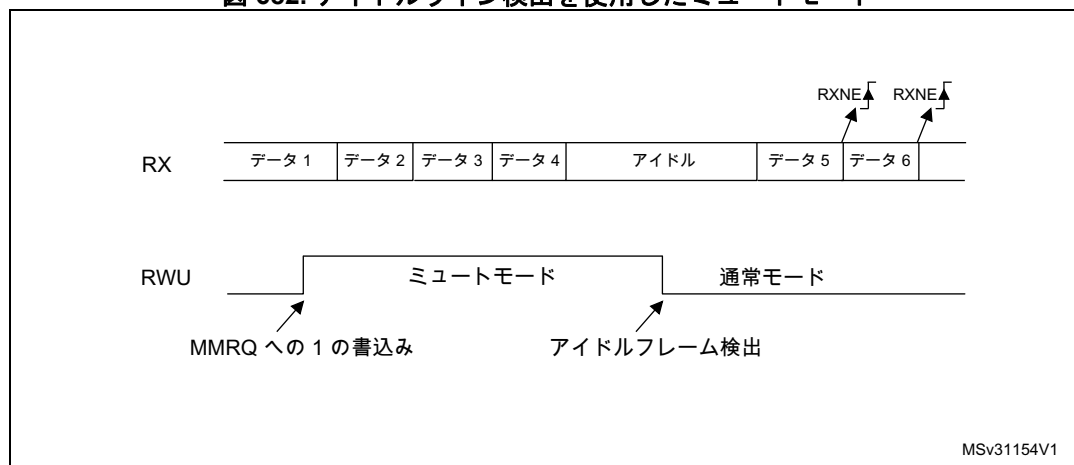
- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

アイドルライン検出 (WAKE=0)

MMRQ ビットに 1 が書き込まれ、RWU が自動的にセットされたときには、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USART_ISR レジスタの IDLE ビットはセットされません。アイドルライン検出を使用したミュートモードの動作例を [図 682](#) に示します。

図 682. アイドルライン検出を使用したミュートモード



注： IDLE キャラクタがすでに経過しているときに MMRQ がセットされた場合は、ミュートモードに入りません (RWU はセットされません)。

ラインが IDLE のときに USART が有効にされた場合、1 IDLE フレーム後にアイドル状態が検出されます (1 キャラクタフレームの受信後だけでなく)。

4 ビット／7 ビットアドレスマーク検出 (WAKE=1)

このモードでは、MSB が "1" の場合、バイトはアドレスとして認識され、そうでない場合はデータとみなされます。アドレスバイトのうち、ターゲットレシーバのアドレスは 4 または 7 LSB です。7 または 4 ビットアドレス検出の選択は、ADDM7 ビットを使用して行われます。この 4 ビット／7 ビットワードは、レシーバによって、USART_CR2 レジスタの ADD ビットでプログラムされたレシーバの自己アドレスと比較されます。

注： 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) で行われます。

プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART がミュートモードに入ったときには、このアドレスバイトに対して RXNE フラグはセットされず、割込みも DMA リクエストも発行されません。FIFO 管理が有効になっている場合、ミュートモードに入る前に、RXFIFO に少なくとも 1 つの空き場所があることをソフトウェアによって確認する必要があります。

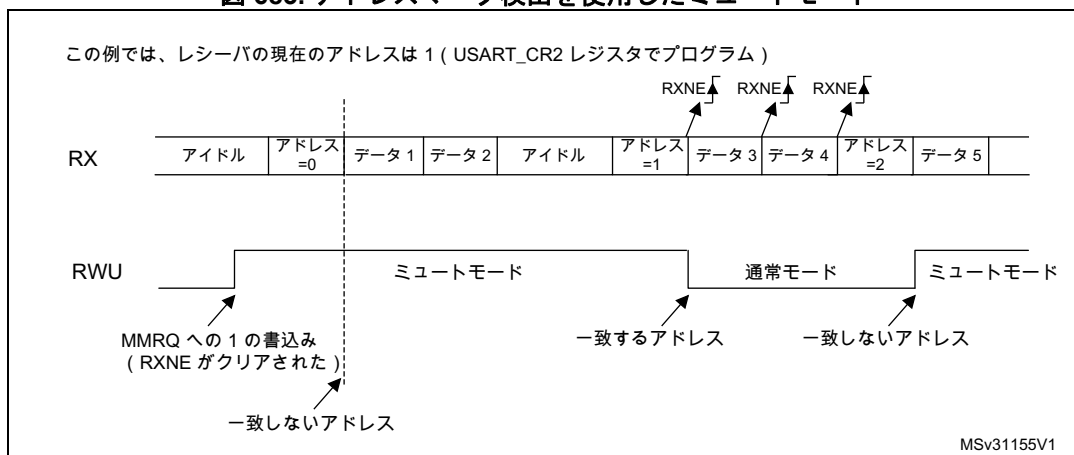
MMRQ ビットに 1 が書き込まれたときにも、USART はミュートモードに入ります。この場合、RWU ビットも自動的にセットされます。

プログラムされたアドレスに一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE/RXFNE ビットがセットされます。

注： FIFO 管理が有効になっている場合、レシーバがデータの最後のビットをサンプリングしている間に MMRQ がセットされると、ミュートモードに実際に移行する前にこのデータが受信されることがあります。

アドレスマーク検出を使用したミュートモードの動作例を 図 683 に示します。

図 683. アドレスマーク検出を使用したミュートモード



50.5.12 USART Modbus 通信

USART は、Modbus/RTU および Modbus/ASCII プロトコルの実装に対する基本的サポートを備えています。Modbus/RTU は、半二重のブロック転送プロトコルです。プロトコルの制御部分（アドレス認識、ブロック整合性制御、およびコマンド解釈）は、ソフトウェアで実装する必要があります。

USART は、ソフトウェアに負荷をかけず、他のリソースを使用せずに、ブロック検出の終了に対する基本的サポートを提供します。

Modbus/RTU

このモードでは、1 つのブロックの終了は 2 キャラクタ時間を超える「サイレンス」（アイドルライン）によって認識されます。この機能は、プログラム可能なタイムアウト機能を通じて実装されます。

タイムアウト機能と割込みは、USART_CR2 レジスタの RTOEN ビットと、USART_CR1 レジスタの RTOIE を通じて有効にする必要があります。2 キャラクタ時間に対応する値（たとえば、22 x ビット時間）を RTO レジスタでプログラムする必要があります。最後のストップビットの受信後、この時間にわたって受信ラインがアイドルのときには、割込みが生成されて、現在のブロック受信が完了したことをソフトウェアに知らせます。

Modbus/ASCII

このモードでは、ブロックの終了は特定の（CR/LF）キャラクタシーケンスによって認識されます。USART は、キャラクタ一致機能を使用して、このメカニズムを管理します。

ADD[7:0] フィールドで LF ASCII コードをプログラムし、キャラクタ一致割込みを有効にすることによって（CMIE=1）、LF が受信されたときにソフトウェアに通知し、ソフトウェアは DMA バッファの CR/LF をチェックできます。

50.5.13 USART パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティチェック）を有効にするには、USART_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な USART フレームフォーマットを表 537 に示します。

表 537. USART フレームのフォーマット

M ビット	PCE ビット	USART フレーム ⁽¹⁾
00	0	SB 8 ビットデータ STB
00	1	SB 7 ビットデータ PB STB
01	0	SB 9 ビットデータ STB
01	1	SB 8 ビットデータ PB STB
10	0	SB 7 ビットデータ STB
10	1	SB 6 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。データレジスタでは、PB は常に MSB 位置を取ります（M ビットの値に応じて、8 または 7 番目）。

偶数パリティ

パリティビットは、6、7、または 8 LSB ビット（M ビットの値に応じて）とパリティビットから構成されるフレーム内で「1」の数が偶数になるように計算されます。

たとえば、データ = 00110101 であり、4bit がセットされた場合、偶数パリティが選択された場合（USART_CR1 の PS ビット = 0）、パリティビットは 0 になります。

奇数パリティ

パリティビットは、6、7、または 8 LSB ビット (M ビットの値に応じて) とパリティビットで構成されるフレーム内で「1」の数が奇数になるように計算されます。

たとえば、データ = 00110101 であり、4bit がセットされた場合、奇数パリティが選択された場合 (USART_CR1 の PS ビット = 1)、パリティビットは 1 になります。

受信中のパリティチェック

パリティチェックに失敗した場合、USART_ISR レジスタの PE フラグがセットされ、USART_CR1 レジスタの PEIE ビットがセットされている場合は割込みが生成されます。PE フラグは、USART_ICR レジスタの PECF に 1 を書き込むことによってクリアされます。

送信中のパリティ生成

USART_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます (偶数パリティが選択された場合 (PS=0) は偶数個の「1」、奇数パリティが選択された場合 (PS=1) は奇数個の「1」)。

50.5.14 USART LIN (Local Interconnection Network) モード

このセクションは、LIN モードがサポートされるときにのみ適用されます。[セクション 50.4 : 2200 ページのUSARTの実装](#)を参照してください。

LIN モードを選択するには、USART_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの CLKEN
- USART_CR3 レジスタの STOP[1:0]、SCEN、HDSEL、および IREN

LIN 送信

LIN マスタ送信の場合、[セクション 50.5.5](#) に記述されている手順を適用する必要があります。通常の USART 送信と同じですが、次のような違いがあります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBKRQ ビットをセットすると、13 個の「0」ビットがブレークキャラクタとして送信されます。その後、値「1」の 2 ビットが送信され、次の START 検出が可能になります。

LIN 受信

LIN モードが有効になると、ブレーク検出回路が有効になります。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生たびにブレークが検出できます。

レシーバが有効になると (USART_CR1 レジスタの RE=1)、RX 入力の START 信号を探します。スタートビットの検出方法は、ブレークキャラクタやデータの検索方法と同じです。スタートビットが検出された後、データの場合とまったく同様に次のビットがサンプリングされます (8、9、および 10 番目のサンプル)。10 個 (USART_CR2 レジスタの LBDL=0) または 11 個 (USART_CR2 レジスタの LBDL=1) の連続したビットが「0」として検出され、その後にデリミタキャラクタが続く場合、USART_ISR レジスタの LBDF フラグがセットされます。LBDIE ビットが 1 の場合、割込みが生成されます。ブレークを確認する前に、RX ラインがハイレベルに戻ったことを知らせるデリミタが検査されます。

この 10 または 11 が発生する前に「1」がサンプリングされた場合、ブレーク検出回路は現在の検出をキャンセルし、再びスタートビットを検索します。

LIN モードが無効にされた場合 (LINEN=0)、レシーバは、ブレーク検出を考慮することなく、通常の USART として機能し続けます。

LIN モードが有効にされた場合 (LINEN=1)、フレーミングエラーが発生 (つまり、ブレークフレームの場合と同様に、“0” の位置でストップビットを検出) すると、レシーバは停止し、ブレーク検出回路が “1” (ブレークワードが完全でなかった場合) またはデリミタキャラクタ (ブレークが検出された場合) を受信するまで停止状態を維持します。

ブレーク検出回路ステートマシンの動作とブレークフラグを [図 684 : 2226 ページの LIN モードでのブレーク検出 \(11 ビットブレーク長、LBDL=1\)](#) に示します。

ブレークフレームの例を [図 685: 2227 ページの LIN モードでのブレーク検出とフレーミングエラー検出](#) に示します。

図 684. LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1)

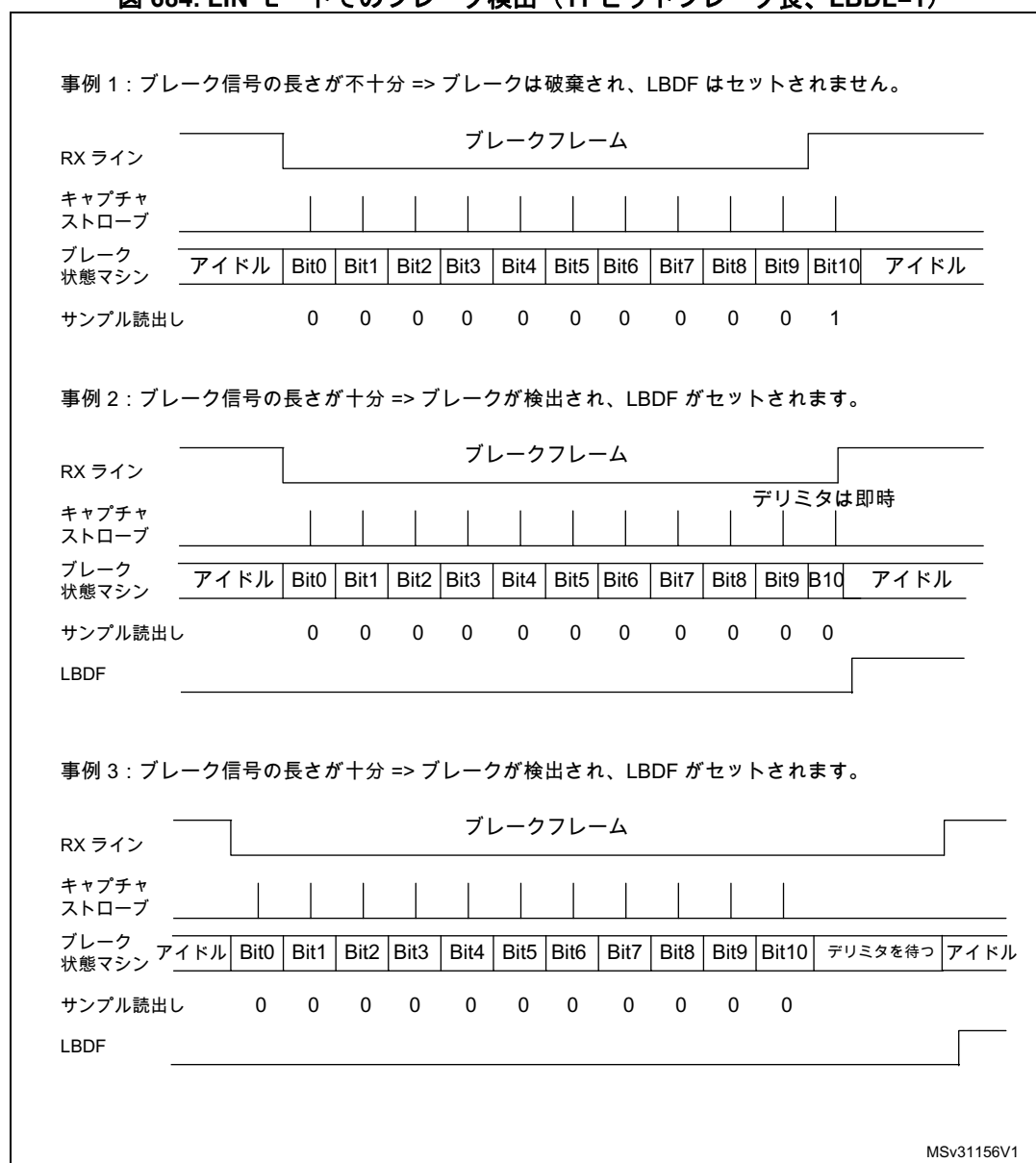
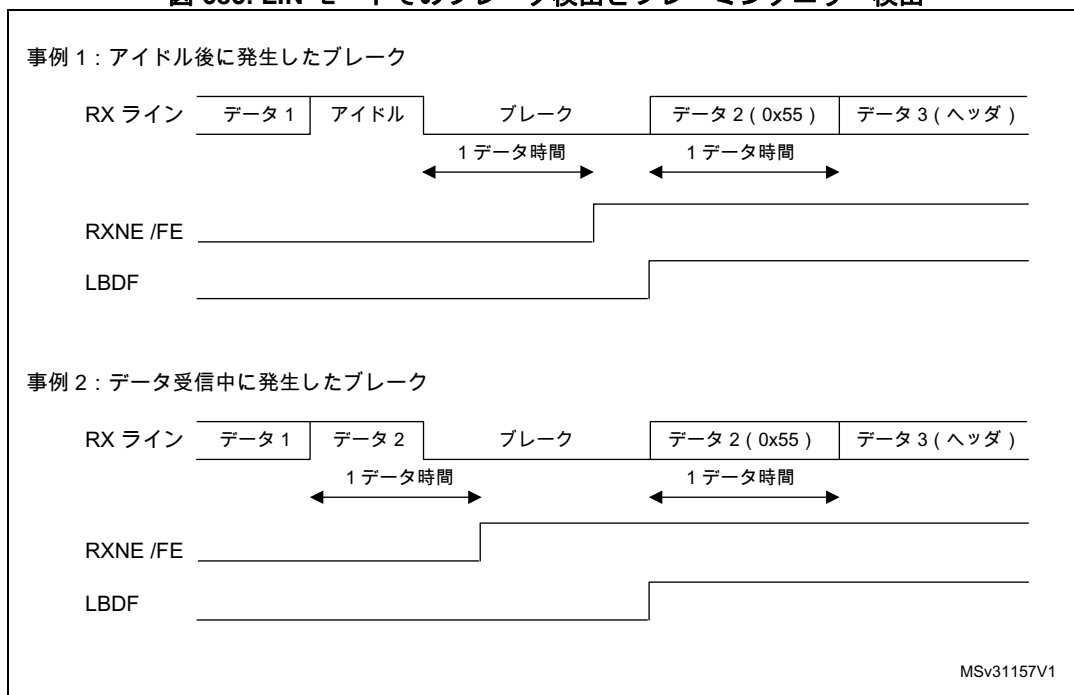


図 685. LIN モードでのブレーク検出とフレーミングエラー検出



50.5.15 USART 同期モード

マスタモード

同期マスタモードを選択するには、USART_CR2 レジスタの CLKEN ビットを“1”にプログラムします。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

このモードでは、USART を使用して、双方向同期シリアル通信をマスタモードで制御できます。CK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、CK ピンにクロックパルスは送信されません。USART_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USART_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用され、USART_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用されます（図 686、図 687、および図 688 を参照）。

アイドル時、プリアンブル処理時、およびブレーク送信時には、外部 CK クロックは起動されません。

同期マスタモードでは、USART トランスミッタは非同期モードの場合とまったく同じように動作します。しかし、CPOL と CPHA に基づいて CK が TX と同期するので、TX 上のデータが同期します。

同期マスタモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE が 1 にセットされた場合、データはオーバーサンプリングなしで、CK（CPOL と CPHA に応じて立ち上がりまたは立ち下がリエッジ）でサンプリングされます。ボーレート（1/16 ビット時間）に応じたセットアップ時間とホールド時間に従う必要があります。1/16 ビットタイミング）。

注： マスタモードでは、CK ピンは TX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (USART_TDR データレジスタへの書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。

図 686. USART の同期マスタ送信の例

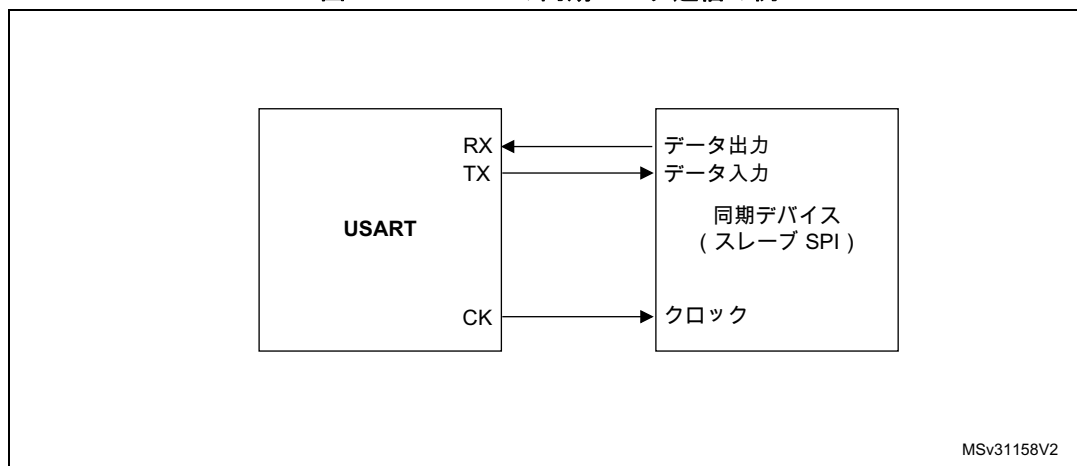


図 687. 同期マスタモードでの USART データクロックタイミング図 (M ビット = 00)

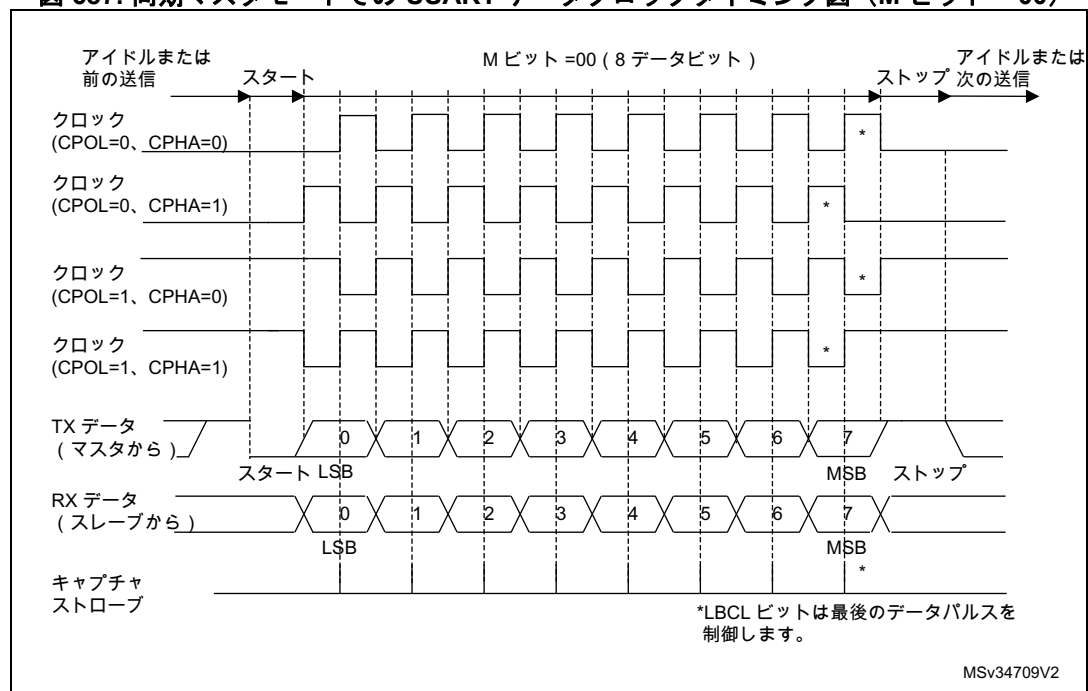
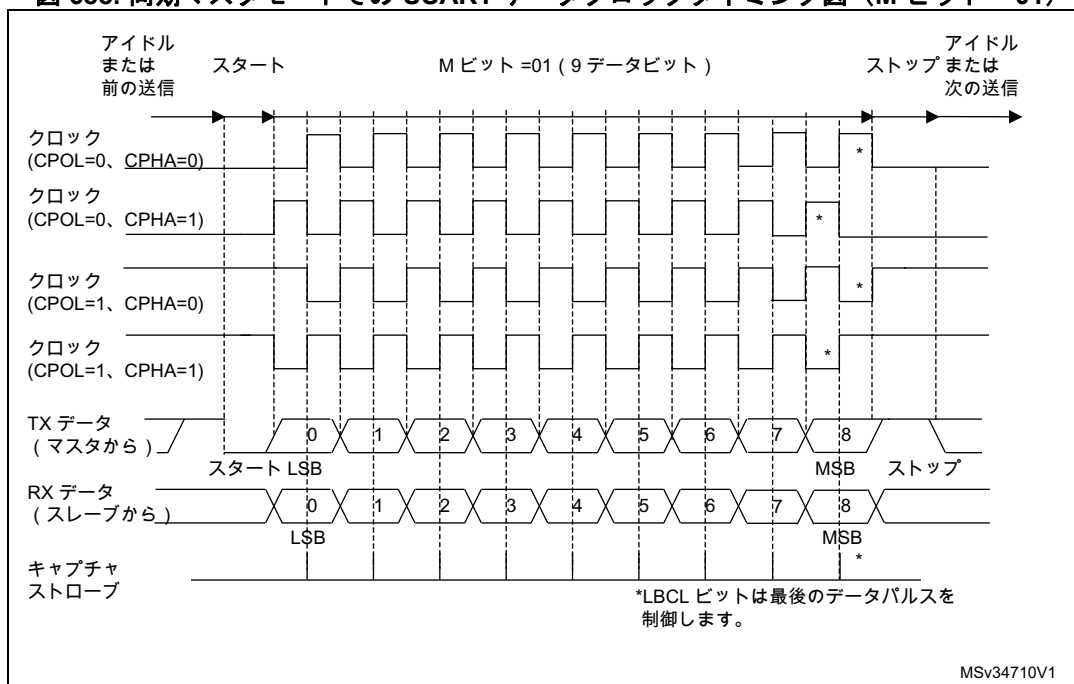


図 688. 同期マスタモードでの USART データクロックタイミング図 (M ビット = 01)



スレーブモード

同期スレーブモードを選択するには、USART_CR2 レジスタの SLVEN ビットを "1" にプログラムします。同期スレーブモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN および CLKEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

このモードでは、USART を使用して、双方向同期シリアル通信をスレーブモードで制御できます。CK ピンは、スレーブモードの USART の入力です。

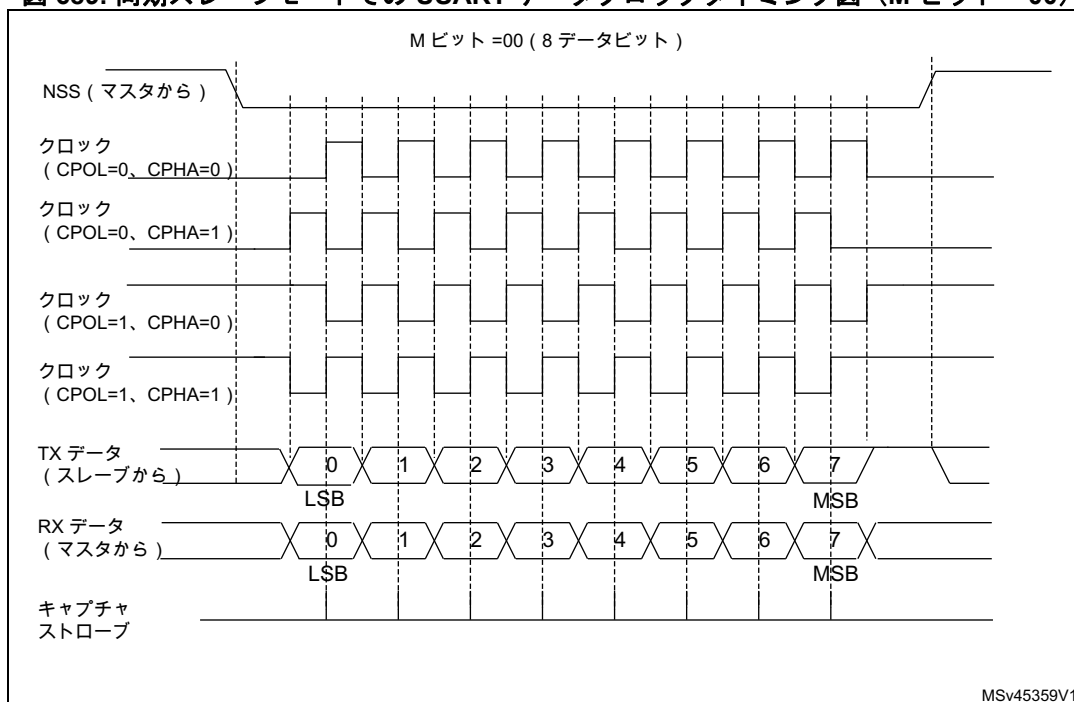
注： ペリフェラルが SPI のスレーブモードで使用されるとき、ペリフェラルのクロックソースの周波数 (usart_ker_ck_pres) は CK 入力周波数の 3 倍より大きくなければなりません。

USART_CR2 レジスタの CPOL ビットと CPHA ビットは、それぞれクロック極性と外部クロックの位相を選択するために使用されます (図 689 を参照)。

アンダーランエラーフラグはスレーブ送信モードで使用できます。ソフトウェアが USART_TDR にまだ値をロードしていない間に、データ送信用の最初のクロックパルスが現れると、このフラグがセットされます。

スレーブはハードウェアおよびソフトウェアの NSS 管理をサポートしています。

図 689. 同期スレーブモードでの USART データクロックタイミング図 (M ビット = 00)



スレーブ選択 (NSS) ピンの管理

ハードウェアまたはソフトウェアのスレーブ選択管理は、USART_CR2 レジスタの DIS_NSS ビットを使用して設定することができます。

- ソフトウェア NSS 管理 (DIS_NSS = 1)
SPI スレーブが常に選択され、NSS 入力ピンは無視されます。
外部NSS ピンは他のアプリケーションで使用できます。
- ハードウェア NSS 管理 (DIS_NSS = 0)
SPI スレーブ選択は NSS 入力ピンに依存します。NSS がローレベルのときにスレーブが選択され、NSS がハイレベルのときには選択されません。

注： USART が無効にされたときには (UE="0")、クロックパルスが正常に機能するように、LBCL (SPI マスタモードでのみ使用)、CPOL、および CPHA ビットを選択する必要があります。

SPI スレーブモードでは、マスタ通信を開始する前に (またはクロックが安定しているときのフレーム間で)、USART を有効にする必要があります。そうしないと、マスタがフレームの中央に位置しているときに USART スレーブが有効化されると、スレーブはマスタとの同期がとれなくなります。スレーブのデータレジスタは、通信クロックの最初のエッジまたは現在の通信の終了より前に、準備ができていない必要があります。そうしないと、SPI スレーブはゼロを送信します。

SPI スレーブのアンダーランエラー

アンダーランエラーが起きると、USART_ISR レジスタの UDR フラグがセットされ、ソフトウェアによってアンダーランエラーフラグがクリアされるまで SPI スレーブは最後のデータを送信し続けます。

アンダーランフラグはフレームの開始時にセットされます。USART_CR3 レジスタの EIE ビットがセットされている場合、アンダーランエラー割込みがトリガされます。

アンダーランエラーフラグは、USART_ICR レジスタのビット UDRCF をセットすることによってクリアされます。

アンダーランエラーの場合、TDR レジスタに書き込むことは可能です。アンダーランエラーをクリアすると、新しいデータを送信できます。

アンダーランエラーが起きて、TDR に書き込まれる新しいデータがない場合は、フレームの終りに TC フラグがセットされます。

注： データが USART_TDR に書き込まれた時点が CK の最初の送信エッジに近すぎると、アンダーランエラーが起きることがあります。このアンダーランエラーを避けるために、USART_TDR への書き込みは、最初の CK エッジより usart_ker_ck の 3 サイクル前に行う必要があります。

50.5.16 USART 単線半二重通信

単線半二重モードを選択するには、USART_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN および CLKEN ビット
- USART_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、USART_CR3 レジスタの制御ビット HDSEL で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が外部プルアップ付きの代替機能オープンドレインとして設定されるように、I/O を設定する必要があります。

この点を除くと、通信プロトコルは通常の USART モードと同じです。ラインの競合はソフトウェアによって管理する必要があります（たとえば、集中型アービタを使用して）。特に、送信がハードウェアによってブロックされることはなく、TE ビットがセットされている間は、データレジスタにデータが書き込まれるとすぐに、送信が続行されます。

50.5.17 USART レシーバタイムアウト

レシーバタイムアウト機能を有効にするには、USART_CR2 制御レジスタの RTOEN ビットをセットします。

タイムアウト経過時間は USART_RTOR レジスタの RTO ビットフィールドを使用してプログラムします。

レシーバタイムアウトカウンタは次の時点からカウントを開始します。

- STOP = 00 または STOP = 11 の場合、ストップビットの最後から
- STOP = 10 の場合、2 番目のストップビットの最後から。
- STOP = 01 の場合、ストップビットの最初から。

タイムアウト経過時間が過ぎると、USART_ISR レジスタの RTOF フラグがセットされます。USART_CR1 レジスタの RTOIE ビットがセットされている場合、タイムアウトが生成されます。

50.5.18 USART スマートカードモード

このセクションは、スマートカードモードがサポートされるときにのみ適用されます。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

スマートカードモードを選択するには、USART_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの HDSEL および IREN ビット

スマートカードにクロックを供給するために CLKEN ビットもセットすることができます。

スマートカードインタフェースは、ISO 7816-3 標準で定義された非同期スマートカードプロトコルをサポートするように設計されています。T=0（キャラクタモード）と T=1（ブロックモード）の両方がサポートされます。

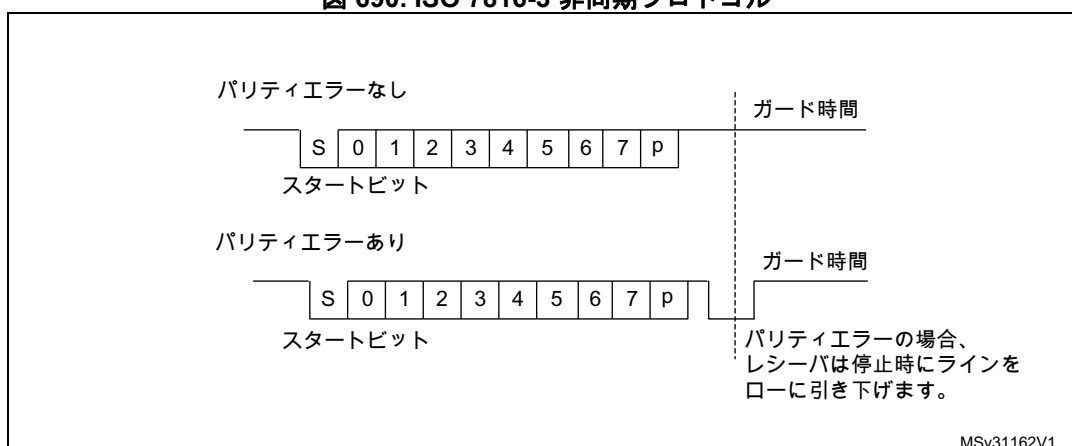
USART は次のように設定してください。

- 8 ビット+パリティ：USART_CR1 レジスタの M = 1 および PCE = 1
- データ送受信ではストップビット 1.5 個：USART_CR2 レジスタ STOP=11。受信にはストップビット 0.5 個を選択することも可能です。

T=0（キャラクタ）モードでは、パリティエラーはガードタイム中の各キャラクタの終わりに示されます。

[図 690](#) に、パリティエラーの有無によるデータラインの状況の変化の例を示します。

図 690. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、やはりスマートカードによって駆動される双方向ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードモードは、単線半二重通信プロトコルを実装します。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されます。通常動作では、フルの送信シフトレジスタは、次のボークロックエッジでシフト動作を開始します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。
- 送信時、スマートカードがパリティエラーを検出した場合には、ラインをローに駆動することによって (NACK)、この条件を USART に知らせます。この NACK 信号 (1 ボークロックの間、送信ラインをローに引き下げ) は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。USART は、プロトコルに従って、データの自動再送信を処理できます。再試行回数は、SCARCNT ビットフィールドでプログラムされます。プログ

ラムされた再試行回数後も USART が NACK を受信し続けた場合は、送信を停止して、エラーをフレーミングエラーとして通知します。TXE ビット (FIFO モードが有効な場合は TXFNF ビット) は、USART_RQR レジスタの TXFRQ ビットを使用してセットできます。

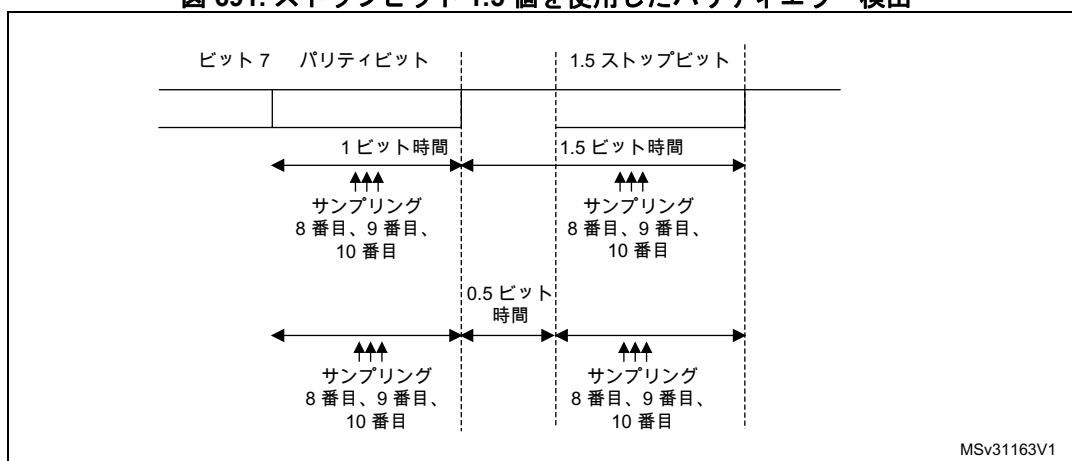
- 送信時のスマートカード自動再試行: USART による NACK の検出と反復キャラクタのスタートビットの間に 2.5 ボー周期の遅延が挿入されます。最後の反復キャラクタの受信終了時、ただちに TC ビットがセットされます (ガードタイムはありません)。ソフトウェアで再び繰り返したい場合は、規格によって指定されている 2 ボー周期以上を確保する必要があります。
- 1.5 個のストップビット周期でプログラムされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボー周期周期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって "NACK" され、そうでない場合、NACK は送信されません (T=1 モードで使用されます)。受信したキャラクタにエラーがあった場合、RXNE (FIFO モードが有効な場合は RXFNE) / 受信 DMA リクエストは有効になりません。プロトコルの仕様に従って、スマートカードは同じキャラクタを再送信する必要があります。SCARCNT ビットフィールドで指定された最大試行回数後も、受信したキャラクタにエラーがあった場合、USART は NACK の送信を停止して、エラーをパリティエラーとして通知します。
- 受信時のスマートカード自動再試行: USART がカードを NACK したが、カードがキャラクタを繰り返さなかった場合、BUSY フラグはセットされたままです。
- 送信時、USART は 2 つの連続するキャラクタの間にガードタイム (ガードタイムレジスタでプログラム) を挿入します。ガードタイムは前のキャラクタのストップビット後に測定されるので、GT[7:0] レジスタを目的の CGT (7816-3 仕様で定義されている Character Guard Time) から 12 (1 キャラクタの時間) を引いた値にプログラムする必要があります。
- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。TCBGT フラグは、ガード時間完了を待たずにデータ転送の終わりを検出するために使用できます。このフラグは、フレーム送信終了後およびカードから NACK を受信しなかった場合にのみセットされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ端でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボー周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

注 : スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

[図 691](#) に、USART による NACK 信号のサンプリング方法を示します。この例では、USART はデータを送信中であり、ストップビットが 1.5 個組み込まれています。データと NACK 信号の整合性を検査するために、USART のレシーバ部が有効にされます。

図 691. ストップビット 1.5 個を使用したパリティエラー検出



USART は、CK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、CK は通信に関係せず、5 ビットのプリスケラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、USART_GTPR レジスタで設定されます。CK の周波数は $\text{usart_ker_ck_pres}/2$ から $\text{usart_ker_ck_pres}/62$ の間でプログラミングできます。ここで、 usart_ker_ck_pres は、プログラムされたプリスケラで分周されたペリフェラル入力クロックです。

ブロックモード (T=1)

T=1 (ブロック) モードでは、パリティエラー送信は、UART_CR3 レジスタの NACK ビットをクリアすることによって無効化されます。

ブロックモードでスマートカードからの読出しをリクエストするときには、ソフトウェアは RTOR レジスタを BWT (ブロックウェイトタイム) - 11 の値にプログラムする必要があります。この期間が終了する前にカードからの応答が受信されなかった場合、タイムアウト割込みが生成されます。この期間が終了する前に最初のキャラクタが受信された場合は、RXNE/RXFNE 割込みによって通知されます。

注： ブロックモードのスマートカードからの読出しに USART を DMA モードで使用するときでも、RXNE/RXFNE 割込みを有効にする必要があります。並行して、DMA は最初の受信バイトの後でのみ有効にする必要があります。

2つの連続するキャラクタの間で最大ウェイトタイムの自動チェックを行うには、最初のキャラクタの受信後 (RXNE/RXFNE 割込み)、RTO レジスタを CWT (キャラクタウェイトタイム - 11) の値にプログラムする必要があります。この時間は、ポータイム単位で表されます。スマートカードが前のキャラクタの終了後、CWT 未満の時間内に新しいキャラクタを送信しなかった場合、USART は RTOF フラグと割込み (RTOIE ビットがセットされているとき) によって、これをソフトウェアに通知します。

注： スマートカードプロトコルの定義にあるように、BWT/CWT 値は最後のキャラクタの開始 (スタートビット) から定義する必要があります。RTO レジスタは、最後のキャラクタ自体の長さを考慮して、それぞれ BWT - 11 または CWT - 11 にプログラムする必要があります。

ブロック長カウンタは、USART が受信するすべてのキャラクタをカウントするために使用されます。このカウンタは、USART の送信時にリセットされます。ブロックの長さは、スマートカードによってブロックの 3 番目のバイト (プロローグフィールド) で伝えられます。この値を USART_RTOR レジスタの BLEN フィールドでプログラムする必要があります。DMA モードを使用するときには、ブロックの開始前に、このレジスタフィールドを最小値 (0x0) にプログラムする必要があります。この値では、4 番目の受信キャラクタの後に割込みが生成されます。ソフトウェアは LEN フィールド (3 番目のバイト) を読み出す必要があり、その値は受信バッファから読み出される必要があります。

割り込み駆動受信モードでは、ブロックの長さはソフトウェアによって、または BLEN 値をプログラムすることによってチェックできます。ただし、ブロックの開始前に、BLEN の最大値 (0xFF) をプログラムすることができます。実際の値は、3 番目のキャラクタの受信後にプログラムされます。

ブロックが LRC 水平冗長検査 (1 エピログバイト) を使用している場合は、BLEN=LEN です。ブロックが CRC メカニズム (2 エピログバイト) を使用している場合は、BLEN=LEN+1 をプログラムする必要があります。合計ブロック長 (プロログ、エピログ、および情報フィールドを含む) は、BLEN+4 に等しくなります。ブロックの終わりは EOBFF フラグと割り込み (EOBIE ビットがセットされているとき) によってソフトウェアに通知されます。

ブロック長エラーの場合、ブロックの終わりは RTO 割り込みによって通知されます (キャラクタウェイトタイムオーバーフロー)。

注： エラーチェックコード (LRC/CRC) は、ソフトウェアによって計算／確認されなければなりません。

ダイレクトおよびインバースコンベンション

スマートカードプロトコルは、ダイレクトとインバースの 2 つコンベンションを定義しています。

ダイレクトコンベンションは、LSB ファースト、論理ビットの値 1 がラインの H 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=0, DATAINV=0 (デフォルト値) をプログラムする必要があります。

インバースコンベンションは、MSB ファースト、論理ビットの値 1 が単線の L 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=1, DATAINV=1 をプログラムする必要があります。

注： 論理データ値が反転されると (0=H、1=L)、パリティビットも同じように反転されます。

カードのコンベンションを認識するために、カードは最初のキャラクタ TS を ATR (Answer To Reset) フレームの最初のキャラクタとして送信します。TS には、LHHL LLL LLH と LHHL HHH LLH の 2 つのパターンが使えます。

- (H) LHHL LLL LLH は、インバースコンベンションをセットアップします。状態 L が値 1 にエンコードされ、モーメント 2 は最上位ビットを含みます (MSB ファースト)。インバースコンベンションによってデコードされると、送受信されたバイトは 3F に等しくなります。
- (H) LHHL HHH LLH は、ダイレクトコンベンションをセットアップします。状態 H が値 1 にエンコードされ、モーメント 2 は最下位ビットを含みます (LSB ファースト)。ダイレクトコンベンションによってデコードされると、送受信されたバイトは 3B に等しくなります。

キャラクタパリティは、2 から 10 までの 9 個のモーメントに 1 にセットされた偶数個のビットがあったときに正しいとみなされます。

USART はカードが使用するコンベンションを知らないなので、いずれのパターンであるかを認識して、それに応じて動作できる必要があります。パターン認識はハードウェアでは行われず、ソフトウェアシーケンスによって行われます。さらに、USART がダイレクトコンベンション (デフォルト) で設定され、カードがインバースコンベンションで応答した場合、TS = LHHL LLL LLH になり、USART が受信したキャラクタは 03 になり、パリティは奇数です。

したがって、TS パターン認識には 2 つの方式を使用できます。

方式 1

USART は、標準スマートカードモード／ダイレクトコンベンションでプログラムされます。この場合、TS パターンの受信によってパリティエラー割込みと、カードに対するエラー信号が生成されます。

- パリティエラー割込みは、カードがダイレクトコンベンションで正しく応答しなかったことをソフトウェアに知らせます。ソフトウェアは、USART をインバースコンベンションで再プログラムします。
- エラー信号に対して、カードは同じ TS キャラクタを再試行し、再プログラムされた USART によって今度は正しく受信されます。

または、パリティエラー割込みに応答して、ソフトウェアは USART を再プログラムし、カードに対して新しいリセットコマンドを生成してから、TS を再び待つこともできます。

方式 2

USART は 9 ビット／パリティなしモード、ビット反転なしでプログラムされます。このモードでは、次のように 2 つの TS パターンのいずれかを受信します。

(H) LHHH LLL LLH = 0x103 -> インバースコンベンションを選択

(H) LHHH HHH LLH = 0x13B -> ダイレクトコンベンションを選択

ソフトウェアは受信されたキャラクタをこの 2 つのパターンと照合して、いずれかに一致した場合は、それに応じて、次のキャラクタ受信に備えて USART をプログラムします。

2 つのうちのどちらも認識されなかった場合、ネゴシエーションを再開するためにカードリセットが生成されます。

50.5.19 USART IrDA SIR ENDEC ブロック

このセクションは、IrDA モードがサポートされるときにのみ適用されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

IrDA モードを選択するには、USART_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USART_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します (図 692 を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbaud のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力がローレベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである場合 (USART が IrDA エンコーダにデータを送信しているとき)、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。レシーバがビジーである (USART がデコードされたデータを受信している) 場合、IrDA は USART から IrDA への TX 上のデータをエンコードしません。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。

- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます (図 693 を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。
- IrDA 仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ側のグリッチ検出回路は、PSC 2 周期 (PSC は USART_GTPR でプログラムされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上 2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低消費電力トランスミッタと通信できます。
- IrDA モードでは、USART_CR2 レジスタのストップビットを“1 ストップビット”に設定する必要があります。

IrDA 低消費電力モード

トランスミッタ

低消費電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低消費電力ボーレート (最小で 1.42 MHz) の 3 倍となります。一般に、この値は 1.8432 MHz ($1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$) です。低消費電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

レシーバ

低消費電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は $1/\text{PSC}$ よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低消費電力ボーレート (USART_GTPR の PSC 値) の 2 周期分を超える場合にのみ受け入れられます。

注 :

PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。

レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 692. IrDA SIR ENDEC ブロック図

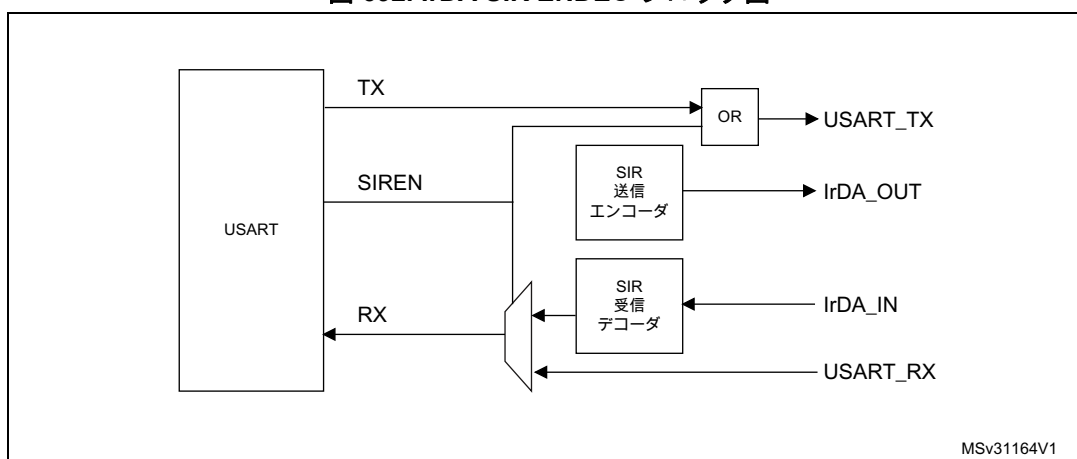
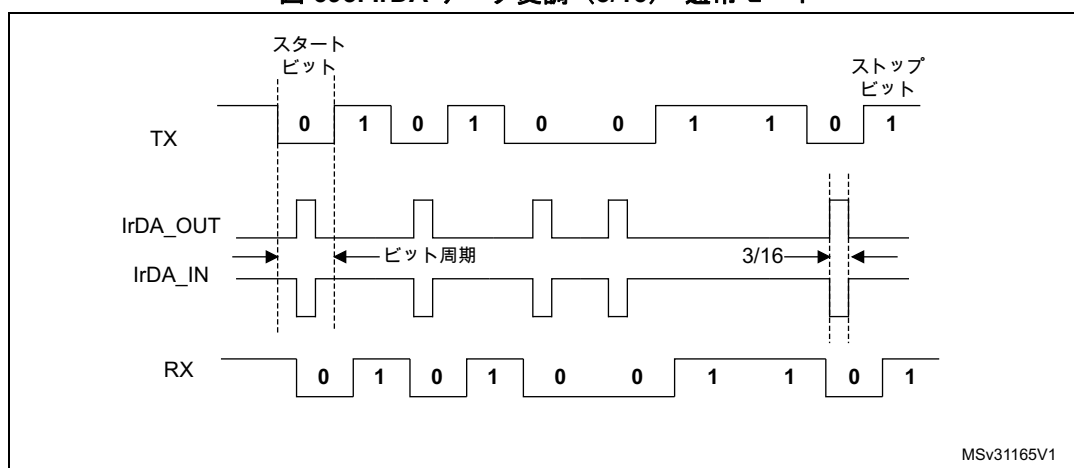


図 693. IrDA データ変調 (3/16) - 通常モード



50.5.20 USART および DMA を使用した連続通信

USART は、DMA を使用して連続通信を行うことができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

注： DMA モードがサポートされるかどうかについては、[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。DMA がサポートされない場合は、[セクション 50.5.7](#)の説明に従って USART を使用してください。FIFO が無効の場合、連続通信を行うには、USART_ISR レジスタの TXE/ RXNE フラグをクリアします。

DMA を使用した送信

DMA モードでの送信を有効にするには、USART_CR3 レジスタの DMAT ビットをセットします。TXE フラグ (FIFO モードが有効な場合は TXFNF フラグ) がセットされるたびに、データは、DMA ペリフェラル (ダイレクトメモリアクセスコントローラ (DMA) セクション) を使用して設定された SRAM 領域から USART_TDR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します (x はチャンネル番号を示します)

1. DMA 制御レジスタに USART_TDR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE (または FIFO モードが有効な場合は TXFNF) イベント後に、メモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE (または FIFO モードが有効な場合は TXFNF) イベント後に、このメモリ領域から USART_TDR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. USART_ICR レジスタの TCCF ビットをセットすることによって、USART_ISR レジスタの TC フラグをクリアします。
7. DMA レジスタのチャンネルを有効にします。

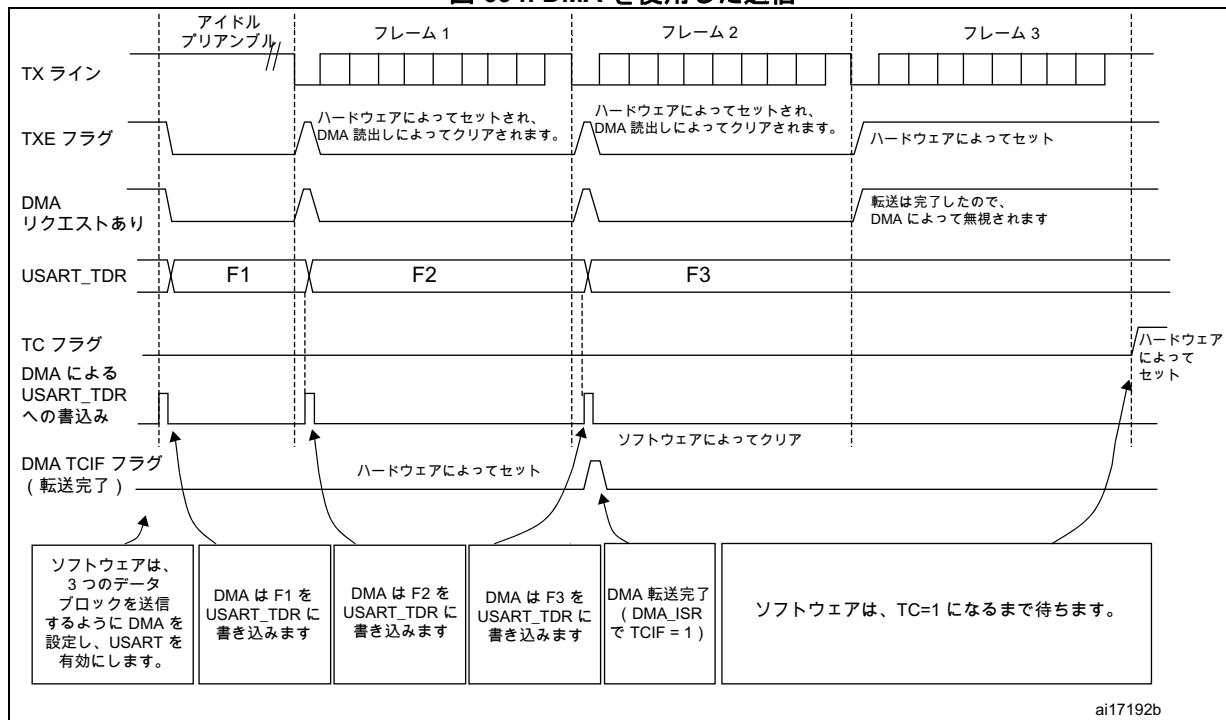
DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと (DMA_ISR レジスタの TCIF フラグがセットされます)、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にする前に、またはペリフェラルクロックが無効のときシステムが低消費電力モード

に入る前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、すべてのデータ転送中、クリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

注: DMA の転送終了前に DMAT ビットをクリアしないでください。

図 694. DMA を使用した送信



注： FIFO 管理が有効になっているときは、DMA リクエストは、送信 FIFO ノットフル（すなわち、TXFNF = 1）によってトリガされます。

DMA を使用した受信

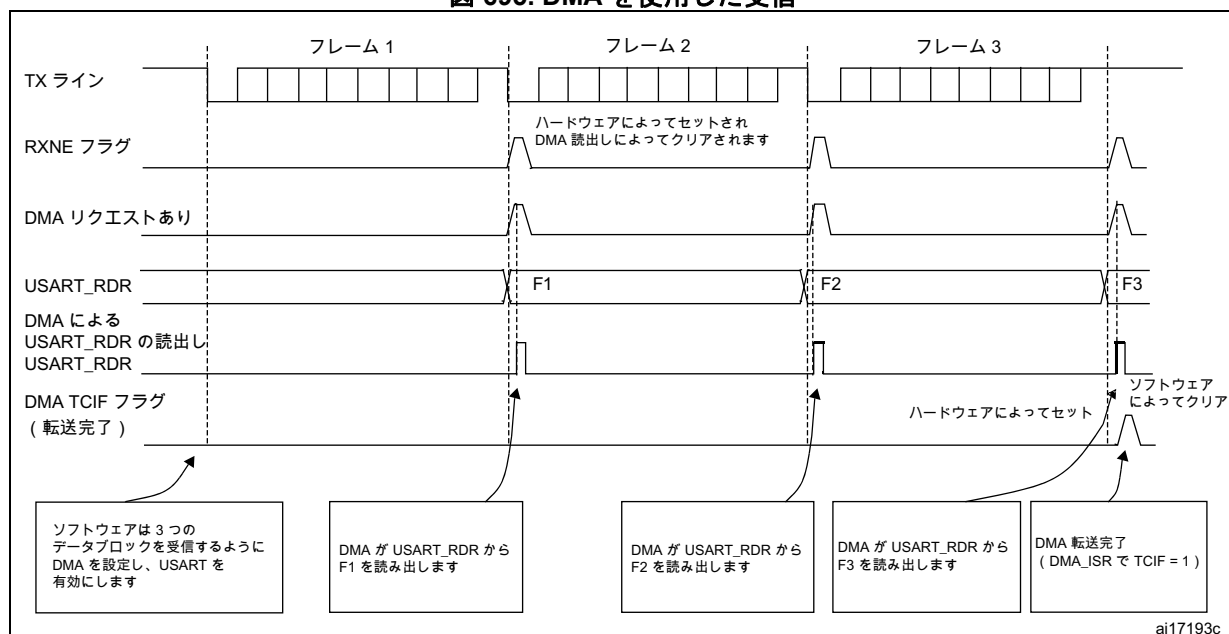
DMA モードでの受信を有効にするには、USART_CR3 レジスタの DMAR ビットをセットします。データバイトが受信されるたびに、データは、USART_RDR レジスタから DMA ペリフェラル（ダイレクトメモリアクセスコントローラ (DMA) セクション）を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USART_RDR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE (FIFO モードが有効な場合は RXFNE) イベント後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE (FIFO モードが有効な場合は RXFNE) イベント後に、USART_RDR からこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

注： DMA の転送終了前に DMAR ビットをクリアしないでください。

図 695. DMA を使用した受信



注： FIFO 管理が有効になっているときは、DMA リクエストは、受信 FIFO ノットエンプティ（すなわち、RXFNE = 1）によってトリガされます。

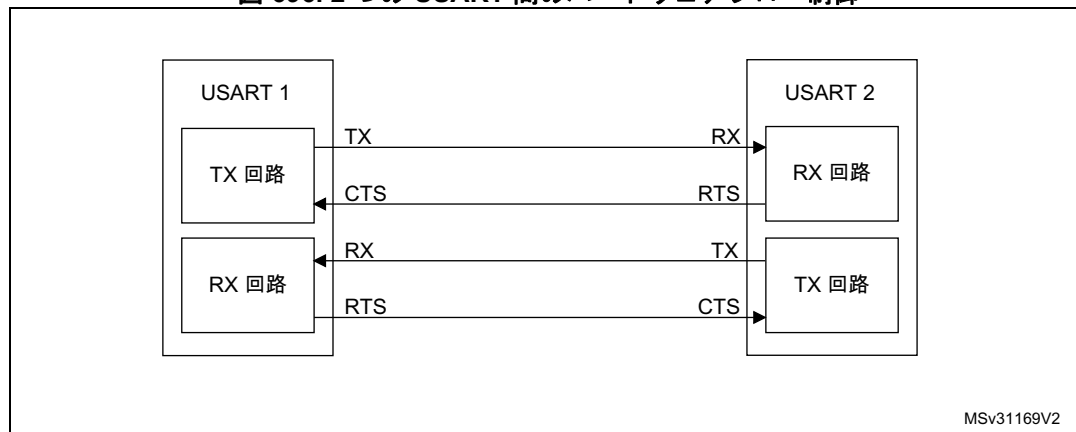
マルチバッファ通信における割込み生成とエラーフラグ

マルチバッファ通信モードでトランザクション中にエラーが発生した場合、現在のバイトの後でエラーフラグがアサートされます。割込み有効フラグがセットされている場合、割込みが生成されます。1 バイト受信において RXNE（FIFO モードが有効な場合は RXFNE）とともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別のエラーフラグ割込み有効ビット（USART_CR3 レジスタの EIE ビット）があり、これがセットされている場合、いずれかのエラーが発生すると、現在のバイトの後で割込みが有効になります。

50.5.21 RS232 ハードウェアフロー制御および RS485 ドライバ有効

CTS 入力と RTS 出力を使用すると、2 つのデバイス間でシリアルデータフローを制御できます。
図 696 に、このモードで 2 つのデバイスを接続する方法を示します。

図 696. 2 つの USART 間のハードウェアフロー制御

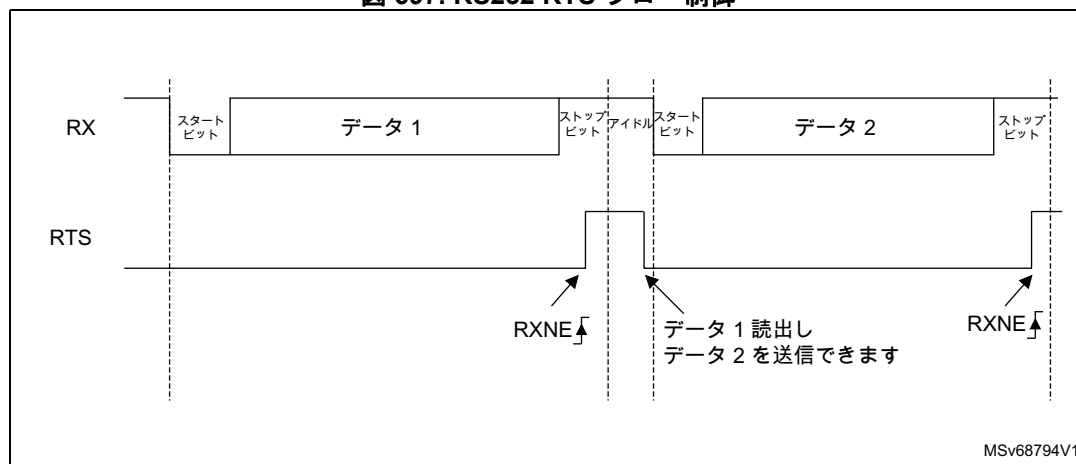


RS232 RTS と CTS のフロー制御は、USART_CR3 レジスタの RTSE ビットと CTSE ビットに "1" を書き込むことによって、個別に有効にできます。

RS232 RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、RTS がネゲートされます (ローレベル接続)。受信レジスタがフルになると RTS がアサートされ、現在のフレームの終わりに送信が停止する予定であることを示します。図 697 に、RTS フロー制御が有効な場合の通信例を示します。

図 697. RS232 RTS フロー制御



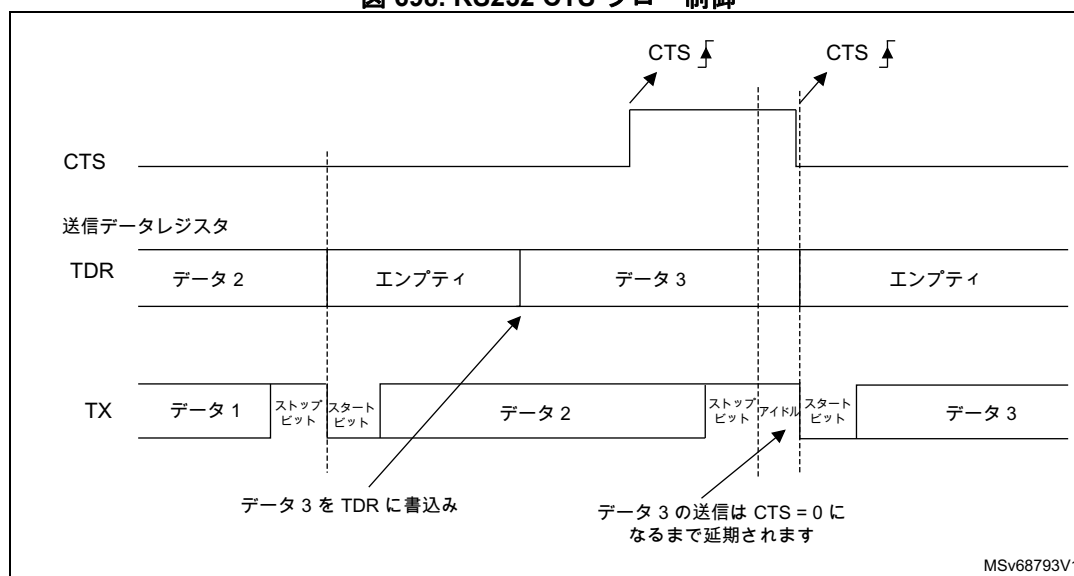
注： FIFO モードが有効な場合は、RXFIFO がフルのときにのみ、RTS がアサートされます。

RS232 CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、CTS 入力をチェックしてから、次のフレームを送信します。CTS がネグートされた場合 (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE/TXFE=0 の場合)。そうでない場合、送信は行われません。送信中に CTS がアサートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、CTS 入力がトグルすると、CTSIF ステータスビットはハードウェアによって自動的にセットされます。このビットは、レシーバの通信準備ができているかどうかを示します。USART_CR3 レジスタの CTSIE ビットがセットされている場合、割込みが生成されます。図 698 に、CTS フロー制御が有効な場合の通信例を示します。

図 698. RS232 CTS フロー制御



注： 正しい動作のために、CTS は、現在のキャラクタの終了の少なくとも 3 USART クロックソース周期前にネグートする必要があります。さらに、2 x PCLK 周期より短いパルスでは CTSCF フラグがセットされない場合があることに注意してください。

RS485 ドライバ有効

ドライバ有効機能を有効にするには、USART_CR3 制御レジスタのビット DEM をセットします。これにより、DE (Driver Enable) 信号によって外部トランシーバ制御を有効にできます。アサーション時間は、DE 信号の有効化からスタートビットの開始までの時間です。USART_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。ネグート時間は、送信メッセージの最後のストップビットの終了から DE 信号の無効化までの時間です。USART_CR1 制御レジスタの DEDT [4:0] ビットフィールドを使用してプログラムされます。DE 信号の極性は、USART_CR3 制御レジスタの DEP ビットを使用して設定できます。

USART では、DEAT および DEDT はサンプル時間単位 (オーバーサンプリングレートに応じて 1/8 または 1/16 ビット時間) で表されます。

50.5.22 USART 低消費電力管理

USART には、高度な低消費電力モード機能があり、`usart_pclk` クロックが無効になっているときでもデータを適切に転送することができます。

USART は、UESM ビットがセットされているとき、MCU を低消費電力モードからウェイクアップできます。

`Usart_pclk` がゲートされているとき、**`usart_pclk`** クロックの有効化を必要とする特定の動作が必要になった場合、USART はウェイクアップ割込み (**`usart_wkup`**) を生成します。

- FIFO モードが無効の場合

USART データレジスタを空にするために `Usart_pclk` クロックを有効にする必要があります。

この場合、`usart_wkup` 割込みのソースは "1" にセットされた `RXNE` です。低消費電力モードに入る前に `RXNEIE` ビットをセットする必要があります。

- FIFO モードが有効な場合

次のために `Usart_pclk` クロックを有効にする必要があります。

- TXFIFO を満たすため
- または RXFIFO を空にするため

この場合、`usart_wkup` 割込みのソースになる可能性のあるものは以下の通りです。

- RXFIFO ノットエンプティ。この場合、低消費電力モードに入る前に `RXFNEIE` ビットをセットする必要があります。
- RXFIFO フル。この場合、低消費電力モードに入る前に `RXFFIE` ビットをセットする必要があります。受信データの数 `RXFIFO` のサイズに一致し、`RXFF` フラグはセットされません。
- TXFIFO エンプティ。この場合、低消費電力モードに入る前に `TXFEIE` ビットをセットする必要があります。

これによって、低消費電力モード中にデータを TXFIFO/RXFIFO に送信／受信することができます。

低消費電力モードで、オーバーラン／アンダーランエラーを避けてデータを送信／受信するために、`usart_wkup` 割込みソースになり得るのは次のイベントのうちの 1 つです。

- TXFIFO 閾値に達した。この場合、低消費電力モードに入る前に `TXFTIE` ビットをセットする必要があります。
- RXFIFO 閾値に達した。この場合、低消費電力モードに入る前に `RXFTIE` ビットをセットする必要があります。

たとえば、ウェイクアップ時間が、ラインを経て 1 バイトを受信するのに必要な時間より少ない場合は、アプリケーションは閾値を `RXFIFO` の最大サイズに設定できます。

MCU を低消費電力モードからウェイクアップするための `RXFIFO` フル、`TXFIFO` エンプティ、`RXFIFO` ノットエンプティ、および `RXFIFO`/`TXFIFO` 閾値割込みを使用すれば、低消費電力モード中にできるだけ多くの USART 転送を行うことができ、電力消費を最適化できるメリットがあります。

あるいは、`WUS` ビットフィールドによって、特定の **`usart_wkup`** 割込みを選択することもできます。

ウェイクアップイベントが検出されると、ハードウェアによって `WUF` フラグがセットされ、`WUFIE` ビットがセットされていた場合は **`usart_wkup`** 割込みが生成されます。

- 注： 低消費電力モードに移行する前に、USART 転送が進行中ではないことを確認してください。BUSY フラグをチェックすることでは、データ受信中に低消費電力モードに入らないことを保証できません。
- WUF フラグは、MCU が低消費電力モードか、アクティブモードかに関係なく、ウェイクアップイベントが検出されたときにセットされます。
- 初期化とレシーバの有効化の直後に低消費電力モードに入るときには、REACK ビットをチェックして、USART が有効であることを確認する必要があります。
- 受信に DMA が使用されるときには、低消費電力モードに入る前に無効化し、低消費電力モードの終了時に再び有効にする必要があります。
- FIFO が有効なときには、アドレス一致時の低消費電力モードからのウェイクアップはミュートモードが有効な場合のみ可能です。

低消費電力モードでのミュートモードの使用

低消費電力モードに入る前に USART がミュートモードになった場合は、

- アイドル検出は低消費電力モードでは機能しないので、アイドル検出時にミュートモードからウェイクアップすることはできません。
- アドレス一致によるミュートモードからのウェイクアップが使用される場合、低消費電力モードからのウェイクアップのソースもアドレス一致でなければなりません。低消費電力モードに入るときに RXNE フラグがセットされた場合、アドレス一致によって低消費電力モードからウェイクアップしても、インタフェースはミュートモードのままです。

- 注： FIFO 管理が有効なとき、ミュートモードは何の制約もなく低消費電力モードからのウェイクアップとともに使用できます（すなわち、ミュートおよび低消費電力モードについて上に述べた 2 点は、FIFO 管理が無効なときのみ有効です）。

低消費電力モードで USART カーネルクロック (usart_ker_ck) がオフのときの低消費電力モードからのウェイクアップ

低消費電力モード中、usart_ker_ck クロックがオフになっている場合、USART 受信ラインの立ち下がりがエッジが検出されると、usart_ker_ck_req 信号によって USART インタフェースが usart_ker_ck クロックをオンにするようリクエストします。その後、usart_ker_ck がフレーム受信に使用されます。

ウェイクアップイベントが確認された場合、MCU は低消費電力モードからウェイクアップし、データ受信が正常に続行します。

ウェイクアップイベントが確認されない場合、usart_ker_ck クロックが再度オフになり、MCU がウェイクアップせずに低消費電力モードに留まり、カーネルクロックリクエストが解除されます。

以下の例は、ウェイクアップイベントが「アドレス一致検出」にプログラムされ、FIFO 管理が無効になっている場合を示しています。

図 699 に、ウェイクアップイベントが確認された時の USART の動作を示します。

図 699. 確認されたウェイクアップイベント（ウェイクアップイベント = アドレス一致、FIFO 無効）

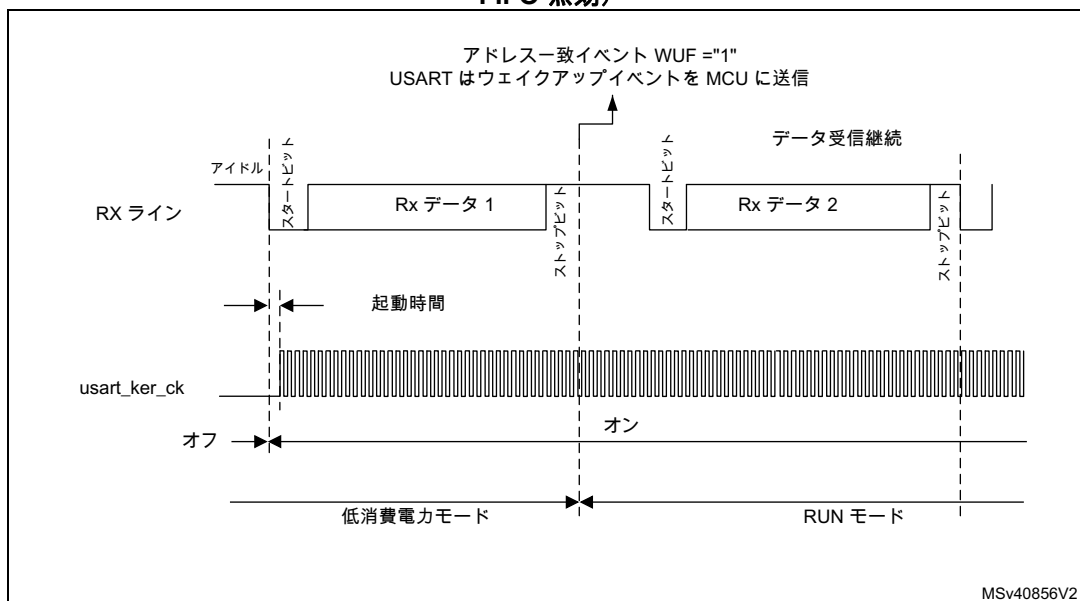
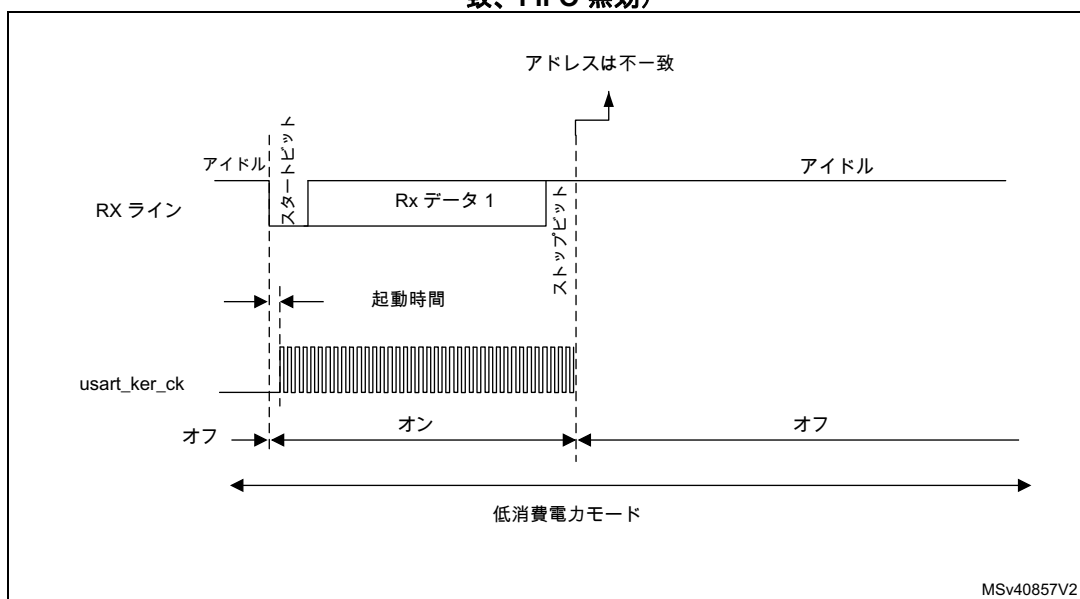


図 700 に、ウェイクアップイベントが確認されていない時の USART の動作を示します。

図 700. 確認されていないウェイクアップイベント（ウェイクアップイベント = アドレス一致、FIFO 無効）



注： 上図は、アドレス一致または受信フレームがウェイクアップイベントとして使用されたとき、有効です。ウェイクアップイベントがスタートビット検出である場合、USART はスタートビットの終端にウェイクアップイベントを MCU に送ります。

マイクロコントローラの低消費電力モードからの正常なウェイクアップを可能にする最大 USART ボーレートの決定

マイクロコントローラの低消費電力モードからの正常なウェイクアップを可能にする最大 USART ボーレートは、ウェイクアップ時間パラメータ（デバイスのデータシートを参照）と USART レシーバの許容誤差（[セクション 50.5.9: クロック偏差に対する USART レシーバの許容誤差](#)を参照）に依存します。

例として、OVER8 = 0、M ビット = 01、ONEBIT = 0、BRR [3:0] = 0000 とします。

この条件では、[表 535: BRR \[3:0\] = 0000 のときの USART レシーバの許容誤差](#)によると、USART レシーバの許容誤差は 3.41% です。

$$DTRA + DQUANT + DREC + DTCL + DWU < \text{USART レシーバの許容誤差}$$

$$D_{WU\max} = t_{WU\text{USART}} / (11 \times T_{\text{bitmin}})$$

$$T_{\text{bitmin}} = t_{WU\text{USART}} / (11 \times D_{WU\max})$$

ここでは、 $t_{WU\text{USART}}$ が低消費電力モードからのウェイクアップ時間です。

パラメータ DTRA、DQUANT、DREC、および DTCL が 0% であるときに理想的なケースを考えた場合、DWU の最大値は 3.41% です。実際には、最低 usart_ker_ck 精度を考慮する必要があります (DREC)。

たとえば HSI が usart_ker_ck として使用され、HSI 精度が 1% の場合、以下を得ることができます。

$t_{WU\text{USART}} = 3 \mu\text{s}$ （この値は 1 例です。正確な値については、デバイスのデータシートを参照してください）

$$D_{WU\max} = \text{USART レシーバの許容誤差} - \text{DREC} = 3.41\% - 1\% = 2.41\%$$

$$T_{\text{bitmin}} = 3 \mu\text{s} / (11 \times 2.41\%) = 11.32 \mu\text{s}$$

その結果、低消費電力モードからの正しいウェイクアップを有効にする最大ボーレートは、次のようになります。1/11.32 μs = 88.36 kbaud

50.6 低消費電力モードの USART

表 538. 低消費電力モードが USART に与える影響

モード	説明
SLEEP	影響はありません。USART 割込みによって、デバイスは SLEEP モードを終了します。
STOP ⁽¹⁾	USART レジスタの内容は保持されます。 STOP モードで使用可能なオシレータによってクロック供給されている場合、USART ではマイクロコントローラを STOP モードからウェイクアップさせることができます。
STANDBY	USART ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. STOP モードからのウェイクアップが指定のペリフェラルインスタンスでサポートされているかどうかについては、[セクション 50.4: USART の実装](#)を参照してください。インスタンスが指定の STOP モードで機能しない場合、この STOP モードを開始する前に無効化する必要があります。

50.7 USART 割込み

すべての USART 割込みリクエストの詳細な説明については、表 539 を参照してください。

表 539. USART 割込みリクエスト

割込み ベクタ	割込みイベント	イベント フラグ	イネーブル 制御ビット	割込みのクリア方法	SLEEP モードの 終了	STOP ⁽¹⁾ モードの 終了	STANDBY モードの 終了
USART または UART	送信データレジスタエン プティ	TXE	TXEIE	TDR を書き込む	可能	不可	不可
	送信 FIFO ノットフル	TXFNF	TXFNFIE	TXFIFO フル		不可	
	送信 FIFO エンプティ	TXFE	TXFEIE	TXFRQ に TDR または 1 を書き込む		可能	
	送信 FIFO 閾値到達	TXFT	TXFTIE	TDR を書き込む		可能	
	CTS 割込み	CTSIF	CTSIE	CTSCF に 1 を書き込む		不可	
	送信完了	TC	TCIE	TCCF に TDR または 1 を 書き込む		不可	
	ガード時間前に送信完了	TCBGT	TCBGTIE	TCBGT に TDR または 1 を書き込む		不可	
USART または UART	受信データレジスタノッ トエンプティ (データの 読み出し可能)	RXNE	RXNEIE	RXFRQ で RDR または 1 を読み出す	可能	はい	不可
	受信 FIFO ノットエンブ ティ	RXFNE	RXFNEIE	RXFIFO エンプティまで RDR を読み出すか、 RXFRQ に 1 を書き込む		可能	
	受信 FIFO フル	RXFF ⁽²⁾	RXFFIE	読み出しは RDR です。		可能	
	受信 FIFO 閾値到達	RXFT	RXFTIE	読み出しは RDR です。		可能	
	オーバーランエラー検出	ORE	RXNEIE/R XFNEIE	ORECF に 1 を書き込む		不可	
	アイドルライン検出	IDLE	IDLEIE	IDLECF に 1 を書き込む		不可	
	パリティエラー	PE	PEIE	PECF に 1 を書き込む		不可	
	LIN ブレーク	LBDF	LBDIE	LBDCF に 1 を書き込む		不可	
	マルチバッファ通信でノ イズエラー。	NE	EIE	NFCF に 1 を書き込む		不可	
	マルチバッファ通信で オーバーランエラー。	ORE ⁽³⁾		ORECF に 1 を書き込む		不可	
	マルチバッファ通信でフ レーミングエラー。	FE		FECF に 1 を書き込む		不可	
	キャラクター致	CMF	CMIE	CMCF に 1 を書き込む		不可	
	レシーバタイムアウト	RTOF	RTOFIE	RTOCCF に 1 を書き込む		不可	
	ブロックの終了	EOBF	EOBIE	EOBCF に 1 を書き込む		不可	
	低消費電力モードからの ウェイクアップ	WUF	WUFIE	WUC に 1 を書き込む		可能	
	SPI スレーブのアンダー ランエラー	UDR	EIE	UDRCF に 1 を書き込む		不可	

1. USART は、ペリフェラルインスタンスが STOP モードからのウェイクアップ機能をサポートしている場合にのみデバイスを STOP モードからウェイクアップすることができます。サポートされている STOP モードのリストについては、[セクション 50.4 : USART の実装](#)を参照してください。
2. RXFF フラグは、USART が 次のように n+1 個のデータを受信した場合にアサートされます (n は RXFIFO のサイズ)。RXFIFO に n 個のデータ、USART_RDR に 1 個のデータ。STOP モードでは、USART_RDR はクロック供給されません。その結果として、このレジスタは書き込まれず、n 個のデータが受信されて RXFIFO に書き込まれた後に、RXFF 割込みがアサートされます (RXFF フラグはセットされません)。
3. OVRDIS = 0 の場合。

50.8 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。

50.8.1 USART 制御レジスタ 1 (USART_CR1)

アドレスオフセット：0x00

リセット値：0x0000 0000

同じレジスタが FIFO モード有効（このセクション）でも、FIFO モード無効（次のセクション）でも使用できます。

FIFO モードが有効な場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXF FIE	TXFEIE	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 RXFFIE：RXFIFO フル割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0：割込みは禁止されています。

1：USART_ISR レジスタの RXFF = 1 のときに、USART 割込みが生成されます。

ビット 30 TXFEIE：TXFIFO エンプティ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0：割込みは禁止されています。

1：USART_ISR レジスタの TXFE = 1 のときに、USART 割込みが生成されます。

ビット 29 FIFOE：FIFO モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0：FIFO モードは無効です。

1：FIFO モードは有効です。

このビットフィールドは、USART が無効（UE = 0）のときのみ書き込むことができます。

注： FIFO モードは、SPI マスタ／スレーブモードおよびスマートカードモードでのみ、標準 UART 通信で使用できます。IrDA および LIN のモードでは有効にはできません。

ビット 28 M1：ワード長

このビットはビット 12（M0）と併せて使用して、ワード長を決定する必要があります。ソフトウェアによってセット／クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、USART が無効（UE=0）のときのみ書き込むことができます。

注： 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート（0x7F および 0x55 フレーム検出）はサポートされません。

ビット 27 EOBIE : ブロック終了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの EOBIF フラグがセットされると、USART 割込みが生成されます。

注 : USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 26 RTOIE : レシーバタイムアウト割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの RTOF ビットがセットされると、USART 割込みが生成されます。

注 : USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 25:21 DEAT[4:0] : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット時間) で表されます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注 : ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 20:16 DEDT[4:0] : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット時間) で表されます。

DEDT 時間中に USART_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注 : ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 15 OVER8 : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : LIN、IrDA、およびスマートカードモードでは、このビットは常にクリア状態に保つ必要があります。

ビット 14 CMIE : キャラクター一致割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの CMF ビットがセットされると、USART 割込みが生成されます。

ビット 13 MME : ミュートモード有効

このビットは USART ミュートモード機能を有効にします。セットされると、USART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット／クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

ビット 12 M0 : ワード長

このビットはビット 28 (M1) と併せて使用して、ワード長を決定します。ソフトウェアによってセット／クリアされます (ビット 28 (M1) の説明を参照)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 WAKE : レシーバのウェイクアップ方式

このビットによって、ミュートモードからの USART のウェイクアップ方式が決まります。ソフトウェアによってセット／クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 10 PCE : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合は 9 番目のビット、M=0 の場合は 8 番目のビット) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 9 PS : パリティ選択

このビットは、パリティの生成／検出が有効である (PCE ビットがセットされている) とき、奇数パリティ／偶数パリティを選択します。ソフトウェアによってセット／クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 8 PEIE : PE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの PE=1 のときには、USART 割込みが生成されます。

ビット 7 TXFNFIE : TXFIFO ノットフル割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの TXFNF=1 のときには、USART 割込みが生成されます。

ビット 6 TCIE : 転送完了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの TC=1 のときには、USART 割込みが生成されます。

ビット 5 RXFNEIE : RXFIFO ノットエンプティ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの ORE=1 または RXFNE=1 のときには、USART 割込みが生成されます。

ビット 4 IDLEIE : IDLE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの IDLE=1 のときには、USART 割込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : スマートカードモードの場合を除いて、送信中に TE ビットにローパルスを与える (“0”に続けて “1”を書き込む) と、現在のワードの後にプリアンプル (アイドルライン) が送信されます。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは USART_ISR レジスタの TEACK ビットをポーリングできます。スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 UESM : USART 低消費電力モードで有効

このビットがクリアされると、USART は MCU を 低消費電力モードからウェイクアップできません。

このビットがセットされると、USART は MCU を 低消費電力モードからウェイクアップできます。

このビットは、ソフトウェアによってセット／クリアされます。

0 : USART は低消費電力モードから MCU をウェイクアップできません。

1 : USART は低消費電力モードから MCU をウェイクアップできます。

注 : 低消費電力モードに入る直前に UESM ビットをセット、低消費電力モードの終了時にクリアすることが推奨されます。

ビット 0 UE : USART 有効

このビットがクリアされると、USART プリスケアラと出力はただちに停止され、現在のすべての動作は破棄されます。USART の設定は保たれますが、USART_ISR のステータスフラグはすべてリセットされます。このビットは、ソフトウェアによってセット／クリアされます。

0 : USART プリスケアラと出力は無効であり、低消費電力モードです。

1 : USART は有効です。

注 : ラインにエラーを生成せずに低消費電力モードに入るためには、TE ビットを事前にリセットする必要があり、ソフトウェアは USART_ISR の TC ビットがセットされるのを待ってから、UE ビットをリセットする必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャネルを無効にする必要があります。

スマートカードモードでは (SCEN = 1)、CK は UE ビット値にかかわらず、CLKEN = 1 の場合に常に使用可能です。

50.8.2 USART 制御レジスタ 1[オルタネート] (USART_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

同じレジスタが FIFO モード有効（前のセクション）でも、FIFO モード無効（このセクション）でも使用できます。

FIFO モードが無効の場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **FIFOEN** : FIFO モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : FIFO モードは無効です。

1 : FIFO モードは有効です。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注 : FIFO モードは、SPI マスタ／スレーブモードおよびスマートカードモードでのみ、標準 UART 通信で使用できます。IrDA および LIN のモードでは有効にはできません。

ビット 28 **M1** : ワード長

このビットはビット 12 (M0) と併せて使用して、ワード長を決定する必要があります。ソフトウェアによってセット／クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート (0x7F および 0x55 フレーム検出) はサポートされません。

ビット 27 **EOBIE** : ブロック終了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの EOBIF フラグがセットされると、USART 割込みが生成されます。

注 : USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 26 **RTOIE** : レシーバタイムアウト割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの RTOF ビットがセットされると、USART 割込みが生成されます。

注 : USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 25:21 DEAT[4:0] : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット時間) で表されます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 20:16 DEDT[4:0] : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット時間) で表されます。

DEDT 時間中に USART_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 15 OVER8 : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： LIN、IrDA、およびスマートカードモードでは、このビットは常にクリア状態に保つ必要があります。

ビット 14 CMIE : キャラクター一致割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの CMF ビットがセットされると、USART 割込みが生成されます。

ビット 13 MME : ミュートモード有効

このビットは USART ミュートモード機能を有効にします。セットされると、USART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット／クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

ビット 12 M0 : ワード長

このビットはビット 28 (M1) と併せて使用して、ワード長を決定します。ソフトウェアによってセット／クリアされます (ビット 28 (M1) の説明を参照)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 WAKE : レシーバのウェイクアップ方式

このビットによって、ミュートモードからの USART のウェイクアップ方式が決まります。ソフトウェアによってセット／クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 10 PCE : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合は 9 番目のビット、M=0 の場合は 8 番目のビット) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 9 PS : パリティ選択

このビットは、パリティの生成／検出が有効である (PCE ビットがセットされている) とき、奇数パリティ／偶数パリティを選択します。ソフトウェアによってセット／クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 8 PEIE : PE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの PE=1 のときには、USART 割込みが生成されます。

ビット 7 TXEIE : 送信データレジスタエンプティ

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの TXE=1 のときには、USART 割込みが生成されます。

ビット 6 TCIE : 転送完了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの TC=1 のときには、USART 割込みが生成されます。

ビット 5 RXNEIE : 受信データレジスタノットエンプティ

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの ORE=1 または RXNE=1 のときには、USART 割込みが生成されます。

ビット 4 IDLEIE : IDLE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの IDLE=1 のときには、USART 割込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : スマートカードモードの場合を除いて、送信中に TE ビットにローパルスを与える ("0" に続けて "1" を書き込む) と、現在のワードの後にプリアンブル (アイドルライン) が送信されます。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは USART_ISR レジスタの TEACK ビットをポーリングできます。スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 UESM : USART 低消費電力モードで有効

このビットがクリアされると、USART は MCU を低消費電力モードからウェイクアップできません。

このビットがセットされると、USART は MCU を低消費電力モードからウェイクアップできます。

このビットは、ソフトウェアによってセット／クリアされます。

0 : USART は低消費電力モードから MCU をウェイクアップできません。

1 : USART は低消費電力モードから MCU をウェイクアップできます。

注 : 低消費電力モードに入る直前に UESM ビットをセット、低消費電力モードの終了時にクリアすることが推奨されます。

ビット 0 **UE** : USART 有効

このビットがクリアされると、USART プリスケアラと出力はただちに停止され、現在のすべての動作は破棄されます。USART の設定は保たれますが、USART_ISR のステータスフラグはすべてリセットされます。このビットは、ソフトウェアによってセット／クリアされます。

0 : USART プリスケアラと出力は無効であり、低消費電力モードです。

1 : USART は有効です。

注 : ラインにエラーを生成せずに低消費電力モードに入るためには、TE ビットを事前にリセットする必要があります、ソフトウェアは USART_ISR の TC ビットがセットされるのを待ってから、UE ビットをリセットする必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

スマートカードモードでは (SCEN = 1)、CK は UE ビット値にかかわらず、CLKEN = 1 の場合に常に使用可能です。

50.8.3 USART 制御レジスタ 2 (USART_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

ビット 31:24 **ADD[7:0]** : USART ノードのアドレス

これらのビットは、ミュートモードの USART ノードのアドレス、または低消費電力モードまたは RUN モードで認識されるキャラクタコードを指定します。

- ミュートモードでは、マルチプロセッサ通信で 4 ビット／7 ビットアドレスマーク検出によってミュートモードからウェイクアップするために使用されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。4 ビットアドレスマーク検出では、ADD[3:0] ビットのみが使用されます。
- 低消費電力モードでは、キャラクター一致時に低消費電力モードからのウェイクアップに使用されず。
WUS[1:0] が 0b00 にプログラムされている場合 (アドレス一致時に WUF がアクティブ)、低消費電力モードからのウェイクアップは、受信したキャラクタが ADD[6:0] または ADD[3:0] ビットフィールドを通じて (ADDM7 ビットに応じて) プログラムされたキャラクタに対応している場合に実行され、WUF 割込みは WUFIE ビットをセットすることで有効化されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。
- ミュートモードが無効化された RUN モードでは (ModBus プロトコルでのブロックの終了の検出など)、受信したキャラクタ全体 (8 ビット) が ADD[7:0] 値と比較され、一致している場合は CMF フラグがセットされます。CMIE ビットがセットされている場合、割込みが生成されます。

これらのビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

ビット 23 RTOEN : レシーバタイムアウト有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : レシーバタイムアウト機能は無効です。

1 : レシーバタイムアウト機能は有効です。

この機能が有効なとき、RTOR (レシーバタイムアウトレジスタ) でプログラムされた時間にわたって RX ラインがアイドル (受信なし) であった場合、USART_ISR レジスタの RTOF フラグがセットされます。

注 : USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 22:21 ABRMOD[1:0] : 自動ボーレートモード

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : スタートビットの測定がボーレートの検出に使用されます。

01 : 立ち下がりがエッジから立ち下がりがエッジまでの測定 (受信されたフレームはシングルビット = 1 で始まらなければならない、その場合、フレーム = Start10xxxxxx)

10 : 0x7F フレーム検出

11 : 0x55 フレーム検出

このビットフィールドは、ABREN=0 または USART が無効 (UE=0) のときのみ書き込むことができます。

注 : DATAINV=1 および／または MSBFIRST=1 の場合、パターンはライン上で同じである必要があります (たとえば、MSBFIRST の場合は 0xAA)。

USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 20 ABREN : 自動ボーレート有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 自動ボーレート検出は無効です。

1 : 自動ボーレート検出は有効です。

注 : USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 19 MSBFIRST : MSB ファースト

このビットは、ソフトウェアによってセット／クリアされます。

0 : スタートビットに続いて、データはデータビット 0 から順に送受信されます。

1 : スタートビットに続いて、データは MSB (ビット 7/8) から順に送受信されます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 18 DATAINV : バイナリデータ反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : データレジスタからの論理データは正／ダイレクトロジックで送受信されます。(1=H、0=L)

1 : データレジスタからの論理データは、負／インバースロジックで送受信されます。(1=L、0=H) パリティビットも反転されます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 17 TXINV : TX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD} = 1$ /アイドル、Gnd = 0/マーク)。

1 : TX ピン信号値は反転されます。($V_{DD} = 0$ /マーク、Gnd = 1/アイドル)。

これにより、TX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 16 RXINV : RX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : RX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD} = 1$ /アイドル、Gnd = 0/マーク)。

1 : RX ピン信号値は反転されます。($V_{DD} = 0$ /マーク、Gnd = 1/アイドル)。

これにより、RX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 15 SWAP : TX/RX ピンのスワップ

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX/RX ピンは標準ピンアウトでの定義に従って使用されます。

1 : TX および RX ピンの機能はスワップされます。これにより、別の UART へのクロスワイヤ接続の場合に動作できます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 14 LINEN : LIN モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : LIN モードが無効の場合

1 : LIN モードが有効な場合

LIN モードでは、USART_CR1 レジスタの SBKRQ ビットを使用して LIN 同期ブレーク (下位 13 ビット) を送信し、LIN 同期ブレークを検出することができます。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注 : USART が LIN モードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 13:12 STOP[1:0] : ストップビット

このビットは、ストップビットのプログラミングに使用します。

00 : 1 個のストップビット

01 : 0.5 個のストップビット

10 : 2 個のストップビット

11 : 1.5 個のストップビット

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 11 CLKEN : クロック有効

このビットによって、CK ピンを有効にできます。

0 : CK ピンは無効です。

1 : CK ピンは有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 同期モードもスマートカードモードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

スマートカードモードでは、スマートカードに CK クロックを正しく供給するには、次のステップを順守する必要があります。

UE = 0

SCEN = 1

GTPR 設定

CLKEN = 1

UE = 1

ビット 10 CPOL : クロック極性

このビットによって、同期モードにおける CK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック／データ関係になるようにします。

0 : 送信ウィンドウの外で、CK ピンはローレベルを維持します。

1 : 送信ウィンドウの外で、CK ピンはハイレベルを維持します。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 同期モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 9 CPHA : クロック位相

このビットは、同期モードでの CK ピンのクロック出力の位相を選択するために使用されます。CPOL ビットと連携して動作し、希望するクロック／データ関係になるようにします (図 680 および図 681 を参照)。

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 同期モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 8 LBCL : 最終ビットのクロックパルス

このビットは、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、CK ピンに出力する必要があるかどうかを選択するために使用されます。

0 : 最終データビットのクロックパルスは、CK ピンに出力されません。

1 : 最終データビットのクロックパルスは、CK ピンに出力されます。

注意 : 最終ビットは、USART_CR1 レジスタの M ビットによって選択された 7 または 8 または 9 ビットフォーマットに応じて送信された 7 番目または 8 番目または 9 番目のデータビットです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 同期モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 LBDIE : LIN ブレーク検出割込み有効

ブレーク割込みマスクです (ブレークデリミタを使用したブレーク検出)。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの LBDF=1 のときには、割込みが生成されます。

注 : LIN モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 5 LBDL : LIN ブレーク検出長

このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。

0 : 10 ビットブレーク検出

1 : 11 ビットブレーク検出

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : LIN モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 4 ADDM7 : 7 ビットアドレス検出／4 ビットアドレス検出

このビットは、4 ビットアドレス検出と 7 ビットアドレス検出の選択に使用されます。

0 : 4 ビットアドレス検出

1 : 7 ビットアドレス検出 (8 ビットデータモード)

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) に対して行われます。

ビット 3 DIS_NSS :

DIS_NSS ビットがセットされているとき、NSS ピンの入力は無視されます。

0 : SPI スレーブ選択は NSS 入力ピンに依存します。

1 : SPI スレーブが常に選択され、NSS 入力ピンは無視されます。

注 : SPI スレーブモードがサポートされていないときは、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 2:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **SLVEN** : 同期スレーブモード有効

SLVEN ビットがセットされると、同期スレーブモードが有効になります。

0 : スレーブモードは無効です。

1 : スレーブモードは有効です。

注 : SPI スレーブモードがサポートされていないときは、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

注 : トランスミッタが有効なときには、CPOL、CPHA、LBCL のビットに書き込まないでください。

50.8.4 USART 制御レジスタ 3 (USART_CR3)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG[2:0]			RXF TIE	RXFTCFG[2:0]			TCBG TIE	TXFTIE	WUFIE	WUS1	WUS0	SCARCNT[2:0]			Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OV R DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:29 **TXFTCFG[2:0]** : TXFIFO 閾値設定

000 : TXFIFO はその深さの 1/8 に達します。

001 : TXFIFO はその深さの 1/4 に達します。

010 : TXFIFO はその深さの 1/2 に達します。

011 : TXFIFO はその深さの 3/4 に達します。

100 : TXFIFO はその深さの 7/8 に達します。

101 : TXFIFO は空になります。

残りの組み合わせ : 予約済み

ビット 28 **RXF TIE** : RXFIFO 閾値割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : 受信 FIFO が RXFTCFG でプログラムされた閾値に達すると、USART 割込みが生成されます。

ビット 27:25 **RXFTCFG[2:0]** : 受信 FIFO 閾値設定

000 : 受信 FIFO はその深さの 1/8 に達します。

001 : 受信 FIFO はその深さの 1/4 に達します。

010 : 受信 FIFO はその深さの 1/2 に達します。

011 : 受信 FIFO はその深さの 3/4 に達します。

100 : 受信 FIFO はその深さの 7/8 に達します。

101 : 受信 FIFO はフルになります。

残りの組み合わせ : 予約済み

ビット 24 **TCBG TIE** : ガード時間前送信完了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの TCBGT=1 のときには、USART 割込みが生成されます。

注 : USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 23 **TXFTIE** : TXFIFO 閾値割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : TXFIFO が TXFTCFG でプログラムされた閾値に達すると、USART 割込みが生成されます。

ビット 22 **WUFIE** : 低消費電力モードからのウェイクアップ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの WUF=1 のときには、USART 割込みが生成されます。

注 : **WUFIE** は、低消費電力モードに入る前にセットする必要があります。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 21:20 **WUS[1:0]** : 低消費電力モードからのウェイクアップ割込みフラグ選択

このビットフィールドは、WUF（低消費電力モードからのウェイクアップフラグ）を有効にするイベントを指定します。

00 : WUF はアドレス一致時に有効になります (ADD[7:0] および ADDM7 による定義に従って)。

01 : 予約済み。

10 : WUF はスタートビット検出時に有効になります。

11 : WUF は RXNE/RXFNE 時に有効になります。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注 : **USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。**

ビット 19:17 **SCARCNT[2:0]** : スマートカード自動再試行カウント

このビットフィールドは、スマートカードモードにおける送受信の再試行回数を指定します。

送信モードでは、送信エラーが生成されるまでの送信の自動再試行回数を指定します (FE ビットをセット)。

受信モードでは、受信エラーが生成される (RXNE/RXFNE および PE ビットのセット) までの受信の試行エラー回数を指定します。

このビットフィールドは、USART が無効 (UE=0) のときのみプログラムする必要があります。

USART が有効になると (UE=1)、このビットフィールドは再送信を停止するために 0x0 にのみ書き込み可能です。

0x0 : 再送信無効 - 送信モードでの自動再送信禁止

0x1 から 0x7 : 自動再送信試行回数 (信号エラーの生成前)

注 : **スマートカードモードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。**

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DEP** : ドライバ有効極性選択

0 : DE 信号はアクティブハイです。

1 : DE 信号はアクティブローです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。**

ビット 14 **DEM** : ドライバ有効モード

このビットにより、DE 信号によって外部トランシーバ制御を有効にできます。

0 : DE 機能は無効です。

1 : DE 機能は有効です。DE 信号は RTS ピンで出力されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。**

ビット 13 **DDRE** : 受信エラー時 DMA 無効

0 : 受信エラーの場合、DMA は無効になりません。対応するエラーフラグはセットされますが、RXNE は 0 に保たれ、オーバーランを防ぎます。結果として、DMA リクエストはアサートされないで、エラーのあるデータは転送されず (DMA リクエストなし)、次の正しい受信データが転送されます。(スマートカードモードで使用されます)

1 : 受信エラーの後、DMA は無効化されます。対応するエラーフラグと RXNE がセットされます。エラーフラグがクリアされるまで、DMA リクエストはマスクされます。つまり、ソフトウェアは、まず、DMA リクエストを無効にするか (DMAR=0)、RXNE (FIFO モードが有効な場合は RXFNE) をクリアしてから、エラーフラグをクリアする必要があります。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： 受信エラーは、パリティエラー、フレーミングエラー、またはノイズエラーです。

ビット 12 **OVRDIS** : オーバーラン無効

このビットは、受信オーバーラン検出を無効にするために使用されます。

0 : オーバーランエラーフラグ、ORE は、受信データが読み出される前に新しいデータを受信したときにセットされます。

1 : オーバーラン機能は無効です。RXNE フラグがまだセットされている間に新しいデータを受信した場合、

ORE フラグはセットされず、新しく受信されたデータが USART_RDR レジスタの前の内容に上書きされます。FIFO モードが有効なとき、RXFIFO はバイパスされ、データは直接、USART_RDR レジスタに書き込まれます。FIFO 管理が有効になっているときでも、RXNE フラグが使用されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： この制御ビットにより、データを読み出さずに通信フローをチェックできます。

ビット 11 **ONEBIT** : 1 サンプルビット方式有効

このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NE) が無効になります。

0 : 3 サンプルビット方式

1 : 1 サンプルビット方式

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 10 **CTSIE** : CTS 割込み有効

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの CTSIF=1 のときには、割込みが生成されます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 9 **CTSE** : CTS 有効化

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは CTS 入力にネゲート (0 に関係) されている場合にのみ転送されます。データの送信中に CTS 入力のアサートされた場合、送信は停止前に完了します。CTS がアサートされている間にデータがデータレジスタに書き込まれた場合、CTS がネゲートされるまで送信は延期されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 8 **RTSE** : RTS 有効化

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 出力は有効であり、レシーババッファにスペースがあるときにのみ、データがリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。データを受信できるとき、RTS 出力がネゲートされます (0 にプルされます)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 7 **DMAT** : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードは送信に有効です。

0 : DMA モードは送信に無効です。

ビット 6 **DMAR** : DMA 有効レシーバ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

ビット 5 **SCEN** : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

0 : スマートカードモードが無効です。

1 : スマートカードモードが有効です。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 4 **NACK** : スマートカード NACK 有効

0 : パリティエラーの際の NACK 転送が無効です。

1 : パリティエラー時の NACK 転送が有効です。

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 3 **HDSEL** : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 2 IRLP : IrDA 低消費電力

このビットは、通常と低消費電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低消費電力モード

このビットは、USART が無効（UE=0）のときのみ書き込むことができます。

注： IrDA モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 1 IREN : IrDA モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

このビットは、USART が無効（UE=0）のときのみ書き込むことができます。

注： IrDA モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 0 EIE : エラー割込み有効化

エラー割込み有効ビットは、フレーミングエラー、オーバーランエラー、ノイズフラグ、または SPI スレーブアンダーランエラー（USART_ISR レジスタの FE=1 または ORE=1 または NE=1 または UDR=1）の場合に割込み生成を有効にするために必要です。

0 : 割込みは禁止されています。

1 : USART_ISR レジスタの FE=1 または ORE=1 または NE=1 または UDR=1（SPI スレーブモード）のときに、割込みが生成されます。

50.8.5 USART ボーレートレジスタ（USART_BRR）

このレジスタは、USART が無効（UE=0）のときのみ書き込むことができます。自動ボーレート検出モードでハードウェアによって自動的に更新されます。

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 BRR[15:0] : USART ボーレート

BRR[15:4]

BRR[15:4] は USARTDIV[15:4] に対応します。

BRR[3:0]

OVER8 = 0 のとき、BRR[3:0] = USARTDIV[3:0]。

OVER8 = 1 のとき、

BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。

BRR[3] は、クリアされたままにする必要があります。

50.8.6 USART ガード時間およびプリスケアラレジスタ (USART_GTPR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **GT[7:0]** : ガード時間値

このビットフィールドは、ガード時間値をバースクロック周期数でプログラムするために使用します。
これはスマートカードモードで使用します。このガード時間値の後は転送完了フラグがセットされます。
このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： スマートカードモードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 7:0 **PSC[7:0]** : プリスケアラ値

状態 : IrDA 低消費電力および IrDA 通常モード

PSC[7:0] = IrDA 通常および低消費電力ポーレート

このビットフィールドは、USART クロックソースを分周して低消費電力周波数を得るためのプリスケアラのプログラミングに使用します。

クロックソースは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

.....

状態 : スマートカードモード

PSC[4:0] : プリスケアラ値

このビットフィールドは、USART クロックソースを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、クロックソース周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

.....

このビットフィールドは、USART が無効 (UE = 0) のときのみ書き込むことができます。

注： スマートカードモードが使用される場合、ビット [7:5] はクリアされたままにする必要があります。
スマートカードモードや IrDA モードがサポートされない場合、このビットフィールドは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

50.8.7 USART レシーバタイムアウトレジスタ (USART_RTOR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **BLEN[7:0]** : ブロック長

このビットフィールドは、受信時のスマートカード T=1 のブロック長を指定します。この値は、情報文字の数 + エピログフィールドの長さ (1-LEC/2-CRC) - 1 と等しくなります。

例 :

BLEN = 0 -> 0 情報文字 + LEC

BLEN = 1 -> 0 情報文字 + CRC

BLEN = 255 -> 254 情報文字 + CRC (合計 256 文字)

スマートカードモードでは、TXE=0 (FIFO モードが有効な場合は TXFE =0) のときにブロック長カウンタがリセットされます。

このビットフィールドは、他のモードでも使用できます。この場合、RE=0 (受信無効) のとき、および/または EOBCF ビットが 1 に書き込まれたときに、ブロック長カウンタがリセットされます。

注 : この値は、ブロック受信の開始後にプログラムできます (プロログフィールドの LEN 文字のデータを使用)。必ず受信したブロックにつき一度だけプログラムするようにしてください。

ビット 23:0 **RTO[23:0]** : レシーバタイムアウト値

このビットフィールドは、レシーバタイムアウト値をビット期間数で指定します。

標準モードでは、最後の受信キャラクタの後、RTO 値を超える間、新しいスタートビットが検出されなかった場合、RTOF フラグがセットされます。

スマートカードモードでは、この値は CWT および BWT を実装するために使用されます。詳細については、スマートカードの章を参照してください。この標準では、CWT/BWT 測定は最後の受信キャラクタのスタートビットから開始して行われます。

注 : この値は、受信キャラクタごとにプログラムされる必要があります。

注 : RTOR は、動作中に書き込むことができます。新しい値がカウンタ以下の場合、RTOF フラグがセットされます。

レシーバタイムアウト機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって 0x00000000 に強制的に設定されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

50.8.8 USART リクエストレジスタ (USART_RQR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 TXFRQ : 送信データ一掃リクエスト

FIFO モードが無効になっている場合は、このビットに "1" を書き込むと、TXE フラグがセットされます。これにより、送信データを破棄できます。このビットは、エラー (NACK) によりデータが送信されなかった場合、および USART_ISR レジスタで FE フラグがアクティブである場合に、スマートカードモードでのみ使用する必要があります。USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。

FIFO モードが有効な場合、FIFO 全体を一掃するために TXFRQ ビットがセットされます。これによって TXFE フラグ (送信 FIFO エンプティ、USART_ISR レジスタのビット 23) がセットされます。送信 FIFO の一掃は、UART モードおよびスマートカードモードの両方でサポートされています。

注： FIFO モードでは、一掃リクエスト中にデータがデータレジスタに書き込まれないようにするために、TxFIFO が空になるまで TXFNF フラグはリセットされます。

ビット 3 RXFRQ : 受信データ一掃リクエスト

このビットに 1 を書き込むと、受信 FIFO 全体を空にします (すなわち、RXFNE ビットをクリアします)。これにより、受信したデータを読み出さずに破棄して、オーバーラン条件を避けることができます。

ビット 2 MMRQ : ミュートモードリクエスト

このビットに 1 を書き込むと、USART はミュートモードになり、RWU フラグはリセットされます。

ビット 1 SBKRQ : ブレーク送信リクエスト

このビットに 1 を書き込むと、SBKF フラグがセットされ、送信マシンが使用可能になるとすぐに、ラインで BREAK を送信するリクエストが発行されます。

注： アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレークキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

ビット 0 ABRRQ : 自動ボーレートリクエスト

このビットに 1 を書き込むと、USART_ISR の ABRF フラグと ABRE フラグがリセットされ、次の受信データフレームでの自動ボーレート測定をリクエストします。

注： USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

50.8.9 USART 割込みおよびステータスレジスタ (USART_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0XX0 00C0

XX = 28 (FIFO／スマートカードモードが有効の場合)

XX = 08 (FIFO が有効で、スマートカードモードが無効の場合)

同じレジスタが FIFO モード有効 (このセクション) でも、FIFO モード無効 (次のセクション) でも使用できます。

FIFO モードが有効な場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **TXFT** : TXFIFO 閾値フラグ

このビットは、TXFIFO が USART_CR3 レジスタの TXFTCFG でプログラムされた閾値に達したとき、すなわち、TXFIFO に TXFTCFG の空き場所ができたときに、ハードウェアによってセットされます。USART_CR3 レジスタの TXFTIE ビット = 1 (ビット 31) の場合、割込みが生成されます。

0 : TXFIFO はプログラムされた閾値に達していません。

1 : TXFIFO はプログラムされた閾値に達しました。

ビット 26 **RXFT** : RXFIFO 閾値フラグ

このビットは、USART_CR3 レジスタの RXFTCFG でプログラムされた閾値に達したときに、ハードウェアによってセットされます。これは、受信 FIFO に (RXFTCFG - 1) 個のデータがあり、USART_RDR レジスタに 1 個のデータがあることを意味します。USART_CR3 レジスタの RXFTIE ビット = 1 (ビット 27) の場合、割込みが生成されます。

0 : 受信 FIFO はプログラムされた閾値に達していません。

1 : 受信 FIFO はプログラムされた閾値に達しました。

注 : RXFTCFG 閾値が 101 に設定されているとき、16 個のデータが利用可能になった場合 (すなわち、RXFIFO に 15 個のデータが入り、USART_RDR に 1 個のデータが入った場合)、RXFT フラグがセットされます。したがって、17 番目の受信データによってオーバーランエラーが起きることはありません。オーバーランエラーは 18 番目のデータを受信した後に発生します。

ビット 25 **TCBGT** : ガード時間前送信完了フラグ

このビットは、USART_TDR に書き込まれた最後のデータがシフトレジスタから正しく送信されたときにセットされます。

スマートカードモードで、データを含むフレームの送信が完了し、スマートカードが NACK を返信しなかった場合、このビットはハードウェアによってセットされます。USART_CR3 レジスタの TCBGTIE=1 の場合、割込みが生成されます。

このビットは、USART_ICR レジスタの TCBGTCF に 1 を書き込むことによって、または USART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

0 : 送信が完了していないか、または送信が正常に完了していません (つまり、カードから NACK を受信)。

1 : 送信は正常に完了しました (ガード時間完了前で、スマートカードから NACK なし)。

注 : USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。USART がスマートカードモードをサポートする場合でこのモードが有効な場合、TCBGT のリセット値は 1 です。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 24 **RXFF** : RXFIFO フル

このビットは、受信したデータの数 RXFIFO サイズ + 1 と一致したときに (RXFIFO がフルで、USART_RDR レジスタに 1 個のデータ)、ハードウェアによってセットされます。

USART_CR1 レジスタの RXFFIE ビット = 1 の場合、割込みが生成されます。

0 : RXFIFO はフルではありません。

1 : RXFIFO はフルです。

ビット 23 **TXFE** : TXFIFO エンプティ

このビットは、TXFIFO が空のとき、ハードウェアによってセットされます。TXFIFO に少なくとも 1 データが入ったとき、このフラグはクリアされます。TXFE フラグは、USART_RQR レジスタのビット TXFRQ (ビット 4) に 1 を書き込むことによってセットすることもできます。

USART_CR1 レジスタの TXFEIE ビット = 1 (ビット 30) の場合、割込みが生成されます。

0 : TXFIFO ノットエンプティ。

1 : TXFIFO エンプティ。

ビット 22 **REACK** : 受信有効確認応答フラグ

このビットは、受信有効値が USART によって考慮されるときに、ハードウェアによってセット／リセットされます。

これを使用して、低消費電力モードに入る前に、USART が受信できる状態であることを確認できます。

注 : USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 21 **TEACK** : 送信有効確認応答フラグ

このビットは、送信有効値が USART によって考慮されるときに、ハードウェアによってセット／リセットされます。

USART_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 **WUF** : 低消費電力モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。USART_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR3 レジスタの WUFIE=1 の場合、割込みが生成されます。

注 : UESM がクリアされると、WUF フラグもクリアされます。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 19 **RWU** : レシーバのミュートモードからのウェイクアップ

このビットは、USART がミュートモードかどうかを示します。ウェイクアップ／ミュートシーケンスが認識されたときに、ハードウェアによってクリア／セットされます。ミュートモード制御シーケンス（アドレスまたは IDLE）は、USART_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは USART_RQR レジスタの MMRQ ビットに "1" を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注 : USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 18 **SBKF** : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。USART_CR3 レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタが送信されません。

1 : ブレークキャラクタが送信されます。

ビット 17 **CMF** : キャラクター致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。USART_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR1 レジスタの CMIE=1 の場合、割込みが生成されます。

0 : キャラクター致は検出されていません。

1 : キャラクター致が検出されました。

ビット 16 **BUSY** : ビジーフラグ

このビットは、ハードウェアによってセット／リセットされます。RX ラインで通信中（スタートビットの検出時）はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : USART はアイドルです（受信なし）。

1 : 受信中です。

ビット 15 **ABRF** : 自動ボーレートフラグ

このビットは、自動ボーレートがセットされたとき（RXFNE もセットされ、RXFNEIE=1 の場合は割込みが生成されます）、または、自動ボーレート操作が成功せずに完了したときにハードウェアによってセットされます（ABRE=1）（この場合、ABRE、RXFNE、および FE もセットされます）。

新しい自動ボーレート検出をリクエストするために、USART_RQR レジスタの ABRRQ に 1 を書き込むことによって、ソフトウェアによってクリアされます。

注 : USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 14 **ABRE** : 自動ボーレートエラー

このビットは、ボーレート測定が失敗した場合に、ハードウェアによってセットされます（範囲外のボーレートまたはキャラクタ比較の失敗）。

USART_RQR レジスタの ABRRQ ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

注 : USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 13 UDR : SPI スレーブアンダーランエラーフラグ

スレーブ送信モードでは、ソフトウェアが USART_TDR にまだ値をロードしていない間に、データ送信用の最初のクロックパルスが現われると、このフラグがセットされます。このフラグは、USART_ICR レジスタの UDRCF ビットをセットすることによってリセットされます。

0 : アンダーランエラーはありません。

1 : アンダーランエラーが発生しました。

注： USART が SPI スレーブモードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 12 EOBFF : ブロック終了フラグ

このビットは、完全なブロックが受信されたときに、ハードウェアによってセットされます（たとえば、T=1 スマートカードモード）。検出は、受信バイト数が BLEN + 4 以上である場合に行われます（ブロックの開始時から、プロローグを含む）。

USART_CR1 レジスタの EOBFIE=1 の場合、割込みが生成されます。

USART_ICR レジスタの EOBCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ブロック終了に達していません。

1 : ブロック終了（文字数）に達しました。

注： スマートカードモードがサポートされない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 11 RTOF : レシーバタイムアウト

このビットは、RTOR レジスタでプログラムされたタイムアウト値が通信なしで経過したときに、ハードウェアによってセットされます。USART_ICR レジスタの RTOCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR2 レジスタの RTOIE=1 の場合、割込みが生成されます。

スマートカードモードでは、タイムアウトは CWT または BWT タイミングに対応します。

0 : タイムアウト値に達していません。

1 : データを受信せずにタイムアウト値に達しました。

注： 時間が RTOR レジスタでプログラムされた値に等しい場合、2 つのキャラクタが分離され、RTOF はセットされません。この時間がこの値に 2 サンプル時間（オーバーサンプリング方式によって 2/16 または 2/8）を加えた値を超える場合、RTOF フラグがセットされます。

カウンタは RE=0 の場合でもカウントしますが、RTOF は RE=1 のときのみセットされます。RE がセットされたときにタイムアウトがすでに経過していた場合、RTOF はセットされます。

USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 10 CTS : CTS フラグ

このビットは、ハードウェアによってセット／リセットされます。CTS 入力ピンのステータスの反転コピーです。

0 : CTS ラインはセットされました。

1 : CTS ラインはリセットされました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 9 CTSIF : CTS 割込みフラグ

このビットは、CTSE ビットがセットされていた場合、CTS 入力がトグルしたときにハードウェアによってセットされます。USART_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR3 レジスタの CTSIE=1 の場合、割込みが生成されます。

0 : CTS ステータスラインでの変更はありません。

1 : CTS ステータスラインでの変更がありました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 8 LBDIF : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。USART_ICR レジスタの LBDIF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR2 レジスタの LBDIE=1 である場合、割込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

注： USART が LIN モードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 7 TXFNF : TXFIFO は空ではありません。

TXFIFO がフルではない、つまり USART_TDR にデータを書き込めるとき、TXFNF はハードウェアによってセットされます。USART_TDR レジスタへの書き込み動作ごとにデータが TXFIFO に格納されます。このフラグは TXFIFO がフルになるまでセットされたままになります。TXFIFO がフルになると、このフラグはクリアされ、データを USART_TDR に書き込むことができないことを示します。

USART_CR1 レジスタの TXFNFIE ビット =1 の場合、割込みが生成されます。

0 : 送信 FIFO はフルです。

1 : 送信 FIFO ノットフル

注： 一掃リクエスト中、TXFIFO が空になるまで、TXFNF はリセットに維持されます。一掃リクエストを (TXFRQ ビットをセットすることによって) 送信した後、TXFIFO に書き込む前に TXFNF フラグをチェックする必要があります (TXFNF および TXFE は同時にセットされます)。

このビットは、シングルバッファ送信時に使用されます。

ビット 6 TC : 送信完了

このビットは、USART_TDR に書き込まれた最後のデータがシフトレジスタから送信されたことを示します。TC フラグの動作は次のようになります。

- TDN =0 の場合、データを含むフレームの送信が完了し、TXE/TXFE がセットされたとき、TC フラグがセットされます。
- TDN が TXFIFO のデータの数と等しい場合、TXFIFO が空で TDN に達すると、TC フラグがセットされます。
- TDN が TXFIFO のデータの数より多い場合、送信するプログラム済みデータの数に達するように TXFIFO が再び満たされるまで、TC はクリアされたままとなります。
- TDN が TXFIFO のデータの数より少ない場合、TXFIFO が空でなくても、TDN に達したときに TC がセットされます。

USART_CR1 レジスタの TCIE=1 の場合、割込みが生成されます。

TC ビットは、USART_ICR レジスタの TCCF に "1" を書き込むことによって、または USART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

ビット 5 RXFNE : RXFIFO は空ではありません。

RXFNE ビットは、RXFIFO が空でないとき、つまりデータが USART_RDR レジスタから読み出せるときに、ハードウェアによってセットされます。USART_RDR からの読み出し動作のたびに、RXFIFO の 1 つの場所が解放されます。

RXFIFO が空になると、RXFNE がクリアされます。RXFNE フラグは、USART_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。

USART_CR1 レジスタの RXFNEIE=1 の場合、割込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 IDLE : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USART_CR1 レジスタの IDLEIE=1 の場合、割込みが生成されます。USART_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注： RXFNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。

ミュートモードが有効な場合 (MME=1)、USART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE="1" (FIFO モードが有効な場合は RXFF="1") のときに、

現在シフトレジスタに受信中のデータを USART_RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USART_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR1 レジスタで RXFNEIE=1、または USART_CR3 レジスタで EIE=1 の場合、割込みが生成されます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

注： このビットがセットされると、USART_RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割込みが生成されます。

USART_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

ビット 2 NE : ノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。USART_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

注： このビットは、割込みを生成する RXFNE ビットと同時に出現するため、割込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NE フラグがセットされると、割込みが生成されます。

ラインがノイズフリーであるとき、NE フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます ([セクション 50.5.9 : 2219 ページのクロック偏差に対する USART レシーバの許容誤差](#)を参照)。

このエラーは USART_RDR 内のキャラクタに関連します。

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。USART_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでデータを送信しているとき、送信時、成功せずに (カードがデータフレームを NACK) 最大送信試行回数に達すると、このビットがセットされます。

USART_CR3 レジスタの EIE=1 の場合、割込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注： このエラーは USART_RDR 内のキャラクタに関連します。

ビット 0 PE : パリティエラー

このビットは、受信モードでパリティエラーが発生したときに、ハードウェアによってセットされます。USART_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR1 レジスタの PEIE=1 である場合、割込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

注： このエラーはUSART_RDR 内のキャラクタに関連します。

50.8.10 USART 割込みおよびステータスレジスタ [オルタネート] (USART_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 00C0

同じレジスタが FIFO モード有効（前のセクション）でも、FIFO モード無効（このセクション）でも使用できます。

FIFO モードが無効の場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 TCBGT : ガード時間前送信完了フラグ

このビットは、USART_TDR に書き込まれた最後のデータがシフトレジスタから正しく送信されたときにセットされます。

スマートカードモードで、データを含むフレームの送信が完了し、スマートカードが NACK を返信しなかった場合、このビットはハードウェアによってセットされます。USART_CR3 レジスタの TCBGTIE=1 の場合、割込みが生成されます。

このビットは、USART_ICR レジスタの TCBGTCF に 1 を書き込むことによって、または USART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

0 : 送信が完了していないか、または送信が正常に完了していません（つまり、カードから NACK を受信）。

1 : 送信は正常に完了しました（ガード時間完了前で、スマートカードから NACK なし）。

注： USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。USART がスマートカードモードをサポートする場合でこのモードが有効な場合、TCBGT のリセット値は 1 です。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 24:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **REACK** : 受信有効確認応答フラグ

このビットは、受信有効値が USART によって考慮されるときに、ハードウェアによってセット／リセットされます。

これを使用して、低消費電力モードに入る前に、USART が受信できる状態であることを確認できます。

注： USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 21 **TEACK** : 送信有効確認応答フラグ

このビットは、送信有効値が USART によって考慮されるときに、ハードウェアによってセット／リセットされます。

USART_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 **WUF** : 低消費電力モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。USART_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR3 レジスタの WUFIE=1 の場合、割込みが生成されます。

注： UESM がクリアされると、WUF フラグもクリアされます。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 19 **RWU** : レシーバのミュートモードからのウェイクアップ

このビットは、USART がミュートモードかどうかを示します。ウェイクアップ／ミュートシーケンスが認識されたときに、ハードウェアによってクリア／セットされます。ミュートモード制御シーケンス（アドレスまたは IDLE）は、USART_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは USART_RQR レジスタの MMRQ ビットに "1" を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注： USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 50.4 : 2200 ページの USART の実装を参照してください。

ビット 18 **SBKF** : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。USART_CR3 レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタが送信されません。

1 : ブレークキャラクタが送信されます。

ビット 17 **CMF** : キャラクター致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。USART_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR1 レジスタの CMIE=1 の場合、割込みが生成されます。

0 : キャラクター致は検出されていません。

1 : キャラクター致が検出されました。

ビット 16 **BUSY** : ビジーフラグ

このビットは、ハードウェアによってセット／リセットされます。RX ラインで通信中（スタートビットの検出時）はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : USART はアイドルです（受信なし）。

1 : 受信中です。

ビット 15 **ABRF** : 自動ボーレートフラグ

このビットは、自動ボーレートがセットされたとき（RXNE もセットされ、RXNEIE=1の場合は割込みが生成されます）、または、自動ボーレート操作が成功せずに完了したときにハードウェアによってセットされます（ABRE=1）（この場合、ABRE、RXNE、およびFE もセットされます）。

新しい自動ボーレート検出をリクエストするために、USART_RQR レジスタのABRRQ に 1 を書き込むことによって、ソフトウェアによってクリアされます。

注： USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 14 **ABRE** : 自動ボーレートエラー

このビットは、ボーレート測定が失敗した場合に、ハードウェアによってセットされます（範囲外のボーレートまたはキャラクタ比較の失敗）。

USART_RQR レジスタのABRRQ ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

注： USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 13 **UDR** : SPI スレーブアンダーランエラーフラグ

スレーブ送信モードでは、ソフトウェアが USART_TDR にまだ値をロードしていない間に、データ送信の最初のクロックパルスが現われると、このフラグがセットされます。このフラグは、USART_ICR レジスタのUDRCF ビットをセットすることによってリセットされます。

0 : アンダーランエラーはありません。

1 : アンダーランエラーが発生しました。

注： USART が SPI スレーブモードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 12 **EOBF** : ブロック終了フラグ

このビットは、完全なブロックが受信されたときに、ハードウェアによってセットされます（たとえば、T=1 スマートカードモード）。検出は、受信バイト数が BLEN + 4 以上である場合に行われます（ブロックの開始時から、プロローグを含む）。

USART_CR1 レジスタのEOBIE=1 の場合、割込みが生成されます。

USART_ICR レジスタのEOBCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ブロック終了に達していません。

1 : ブロック終了（文字数）に達しました。

注： スマートカードモードがサポートされない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページのUSART の実装](#)を参照してください。

ビット 11 RTOF : レシーバタイムアウト

このビットは、RTOR レジスタでプログラムされたタイムアウト値が通信なしで経過したときに、ハードウェアによってセットされます。USART_ICR レジスタの RTOCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR2 レジスタの RTOIE=1 の場合、割込みが生成されます。

スマートカードモードでは、タイムアウトは CWT または BWT タイミングに対応します。

0 : タイムアウト値に達していません。

1 : データを受信せずにタイムアウト値に達しました。

注： 時間が RTOR レジスタでプログラムされた値に等しい場合、2 つのキャラクタが分離され、RTOF はセットされません。この時間がこの値に 2 サンプル時間（オーバーサンプリング方式によって 2/16 または 2/8）を加えた値を超える場合、RTOF フラグがセットされます。

カウンタは RE=0 の場合でもカウントしますが、RTOF は RE=1 のときのみセットされます。RE がセットされたときにタイムアウトがすでに経過していた場合、RTOF はセットされます。

USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 10 CTS : CTS フラグ

このビットは、ハードウェアによってセット／リセットされます。CTS 入力ピンのステータスの反転コピーです。

0 : CTS ラインはセットされました。

1 : CTS ラインはリセットされました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 9 CTSIF : CTS 割込みフラグ

このビットは、CTSE ビットがセットされていた場合、CTS 入力がトグルしたときにハードウェアによってセットされます。USART_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR3 レジスタの CTSIE=1 の場合、割込みが生成されます。

0 : CTS ステータスラインでの変更はありません。

1 : CTS ステータスラインでの変更がありました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 8 LBDF : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。USART_ICR レジスタの LBDCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR2 レジスタの LBDIE=1 である場合、割込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

注： USART が LIN モードをサポートしない場合、このビットは予約済みであり、リセット値に保持されます。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 7 TXE : 送信データレジスタエンプティ

このビットは、USART_TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。このビットは、USART_TDR レジスタへの書き込みによってクリアされます。TXE フラグは、USART_RQR レジスタの TXFRQ に 1 を書き込んでセットし、データを破棄することでもできます（スマートカード T=0 モードでの送信失敗の場合のみ）。

USART_CR1 レジスタの TXEIE ビット =1 の場合、割込みが生成されます。

0 : データレジスタはフルです。

1 : データレジスタはフルです。

ビット 6 TC : 送信完了

このビットは、USART_TDR に書き込まれた最後のデータがシフトレジスタから送信されたことを示します。データを含むフレームの送信が完了し、TXE がセットされたとき、TC フラグがセットされます。USART_CR1 レジスタの TCIE=1 の場合、割り込みが生成されます。
TC ビットは、USART_ICR レジスタの TCCF に "1" を書き込むことによって、または USART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

ビット 5 RXNE : 読出しデータレジスタノットエンプティ

RXNE ビットは、USART_RDR シフトレジスタの内容が USART_RDR レジスタに転送されると、ハードウェアによってセットされます。これは、USART_RDR レジスタからの読出しによってクリアされます。RXNE フラグは、USART_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。
USART_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。
0 : データは受信されていません。
1 : 受信データを読み出すことができます。

ビット 4 IDLE : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USART_CR1 レジスタの IDLEIE=1 の場合、割り込みが生成されます。USART_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。
0 : アイドルラインは検出されていません。
1 : アイドルラインが検出されました。

注： RXNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。

ミュートモードが有効な場合 (MME=1)、USART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE="1" (FIFO モードが有効な場合は RXFF="1") のときに、現在シフトレジスタで受信中のデータを USART_RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USART_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。
USART_CR1 レジスタで RXNEIE=1、または USART_CR3 レジスタで EIE=1 の場合、割り込みが生成されます。
1 : オーバーランエラーが検出されました。

注： このビットがセットされると、USART_RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割り込みが生成されます。

USART_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

ビット 2 NE : ノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。USART_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。
0 : ノイズは検出されていません。
1 : ノイズが検出されました。

注： このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NE フラグがセットされると、割り込みが生成されます。

ラインがノイズフリーであるとき、NE フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます ([セクション 50.5.9 : 2219 ページのクロック偏差に対する USART レシーバの許容誤差](#)を参照)。

このエラーはUSART_RDR 内のキャラクタに関連します。

ビット 1 **FE** : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。USART_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでデータを送信しているとき、送信時、成功せずに（カードがデータフレームを NACK）最大送信試行回数に達すると、このビットがセットされます。

USART_CR3 レジスタの EIE=1 の場合、割込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注： このエラーはUSART_RDR 内のキャラクタに関連します。

ビット 0 **PE** : パリティエラー

このビットは、受信モードでパリティエラーが発生したときに、ハードウェアによってセットされます。USART_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART_CR1 レジスタの PEIE=1 である場合、割込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

注： このエラーはUSART_RDR 内のキャラクタに関連します。

50.8.11 USART 割込みフラグクリアレジスタ (USART_ICR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **WUCF** : 低消費電力モードからのウェイクアップクリアフラグ

このビットに 1 を書き込むと、USART_ISR レジスタの WUF フラグがクリアされます。

注： USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CMCF** : キャラクター致フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの CMF フラグがクリアされます。

ビット 16:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **UDRCF** : SPI スレーブアンダーランフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの UDRF フラグがクリアされます。

注： USART が SPI スレーブモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。

ビット 12 **EOBCF** : ブロック終了クリアフラグ

このビットに 1 を書き込むと、USART_ISR レジスタの EOBF フラグがクリアされます。

注： USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4: 2200 ページの USART の実装](#)を参照してください。

ビット 11 **RTOCF** : レシーバタイムアウトフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの RTOF フラグがクリアされます。

注： USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4: 2200 ページの USART の実装](#)を参照してください。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTSCF** : CTS フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの CTSIF フラグがクリアされます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4: 2200 ページの USART の実装](#)を参照してください。

ビット 8 **LBDCF** : LIN ブレーク検出クリアフラグ

このビットに 1 を書き込むと、USART_ISR レジスタの LBDF フラグがクリアされます。

注： LIN モードがサポートされない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 50.4: 2200 ページの USART の実装](#)を参照してください。

ビット 7 **TCBGTCF** : ガード時間前送信完了フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの TCBGT フラグがクリアされます。

ビット 6 **TCCF** : 送信完了フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの TC フラグがクリアされます。

ビット 5 **TXFECF** : TXFIFO エンプティフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの TXFE フラグがクリアされます。

ビット 4 **IDLECF** : アイドルライン検出フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの IDLE フラグがクリアされます。

ビット 3 **ORECF** : オーバーランエラーフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの ORE フラグがクリアされます。

ビット 2 **NECF** : ノイズ検出フラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの NE フラグがクリアされます。

ビット 1 **FECF** : フレーミングエラーフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの FE フラグがクリアされます。

ビット 0 **PECF** : パリティエラーフラグクリア

このビットに 1 を書き込むと、USART_ISR レジスタの PE フラグがクリアされます。

50.8.12 USARTレシーバデータレジスタ (USART_RDR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **RDR[8:0]** : 受信データ値

受信データキャラクタを含みます。

RDR レジスタは、入カシフトレジスタと内部バスとの間にパラレルインタフェースを提供します (図 674 を参照)。

パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

50.8.13 USARTトランスミッタデータレジスタ (USART_TDR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **TDR[8:0]** : 送信データ値

送信されるデータキャラクタを含みます。

USART_TDR レジスタは、内部バスと出カシフトレジスタとの間にパラレルインタフェースを提供します (図 674 を参照)。

パリティを有効にして (USART_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。

注: このレジスタは、TXE/TXFNF=1 のときのみ書き込む必要があります。

50.8.14 USART プリスケーラレジスタ (USART_PRESC)

このレジスタは、USART が無効 (UE=0) のときのみ書き込むことができます。

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PRESCALER[3:0]** : クロックプリスケーラ

USART 入力クロックは、以下のようにプリスケーラ分周比によって分周できます。

0000 : 入力クロックは分周されません。

0001 : 入力クロックが 2 分周されます。

0010 : 入力クロックが 4 分周されます。

0011 : 入力クロックが 6 分周されます。

0100 : 入力クロックが 8 分周されます。

0101 : 入力クロックが 10 分周されます。

0110 : 入力クロックが 12 分周されます。

0111 : 入力クロックが 16 分周されます。

1000 : 入力クロックが 32 分周されます。

1001 : 入力クロックが 64 分周されます。

1010 : 入力クロックが 128 分周されます。

1011 : 入力クロックが 256 分周されます。

残りの組み合わせ : 予約済み

注 : **PRESCALER** が許容される値と異なる値でプログラムされたとき、プログラムされるプリスケーラ値は 1011 になります。すなわち、入力クロックは 256 分周されます。

50.8.15 USART レジスタマップ

表 540. USART レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	USART_CR1 FIFO モード が有効な場合	RXFIE	TXFIE	FIFOEN	M1	EOBIE	RTOIE			DEAT[4:0]					DEDT[4:0]			OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x00	USART_CR1 FIFO モード が無効の場合	Res.	Res.	FIFOEN	M1	EOBIE	RTOIE			DEAT[4:0]					DEDT[4:0]			OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE			
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x04	USART_CR2	ADD[7:0]								RTOEN	ABRMOD[1:0]			ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	SLVEN			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0			0		
0x08	USART_CR3	TXFTCFG[2:0]		RXFTIE[2:0]		RXFTCFG		TCBGIE		TXFTIE	WUFIE	WUS 1	WUS 0	SCAR CNT[2:0]		Res.		DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]																			
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x10	USART_GTP R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]					PSC[7:0]														
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x14	USART_RTO R	BLEN[7:0]							RTO[23:0]																											
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ			
	リセット値																											0	0	0	0	0	0			
0x1C	USART_ISR FIFO モード が有効な場合	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE			
	リセット値					0	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0			
0x1C	USART_ISR FIFO モード が無効の場合	Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE			
	リセット値							0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF		
	リセット値												0			0					0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]											
	リセット値																								0	0	0	0	0	0	0	0	0	0		

表 540. USART レジスタのマップとリセット値（続き）

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]															
	リセット値																								0	0	0	0	0	0	0	0	0						
0x2C	USART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]									
	リセット値																												0	0	0	0							

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

51 低消費電力ユニバーサル非同期レシーバトランスミッタ (LPUART)

このセクションでは、低消費電力ユニバーサル非同期レシーバトランスミッタ (LPUART) について説明します。

51.1 概要

LPUART は、限られた消費電力で双方向 UART 通信が可能な UART です。わずか 32.768kHz LSE クロックで最大 9600 baud の UART 通信が可能です。LSE クロックとは別のクロックソースによって LPUART にクロック供給すると、さらに高いボーレートを達成できます。

マイクロコントローラが低消費電力モードのときでも、LPUART は UART フレームの着信を待つことができ、その間のエネルギー消費は非常に低くなっています。LPUART には、最小の消費電力で非同期シリアル通信を可能にするために必要なすべてのハードウェアサポートが含まれています。

半二重単線通信とモデム操作 (CTS/RTS) をサポートします。

マルチプロセッサ通信もサポートします。

データの送受信に DMA (直接メモリアクセス) を使用できます。

51.2 LPUART の主な機能

- 全二重非同期通信
- NRZ 標準フォーマット (マーク/スペース)
- プログラム可能なボーレート
- 32.768 kHz クロックソースを使用して、300 baud から 9600 baud まで。
- より高い周波数のクロックソースを使用することにより、より高いボーレートを達成可能
- データを送信および受信する 2 つの内部 FIFO
各 FIFO はソフトウェアで有効/無効にすることができ、FIFO の状態用のステータスフラグを装備
- PCLK から独立したペリフェラル専用のカーネルクロックによるデュアルクロックドメイン
- プログラム可能なデータワード長 (7 または 8 または 9 ビット)
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 設定可能なストップビット (1 または 2 個のストップビット)
- 単線半二重通信
- DMA を使用した連続通信
- 送受信バイトは集中型 DMA を使用して予約済み SRAM にバッファ
- トランスミッタとレシーバ用に個別の有効ビット
- 送信と受信の信号極性を個別に制御
- スワップ可能な Tx/Rx ピン設定
- モデムと RS-485 トランシーバのハードウェアフロー制御
- 転送検出フラグ:
 - 受信バッファフル
 - 送信バッファエンプティ

- ビジーおよび送信終了フラグ
- パリティ制御：
 - パリティビットの送信
 - 受信したデータバイトのパリティ検査
- 4つのエラー検出フラグ：
 - オーバーランエラー
 - ノイズ検出
 - フレームエラー
 - パリティエラー
- フラグ付き割込みソース
- マルチプロセッサ通信: アイドルライン検出またはアドレスマーク検出によるミュートモードからのウェイクアップ
- STOP モードからのウェイクアップ

51.3 LPUART の実装

以下の表では、LPUART の実装について説明します。ここには、比較のために USART と UART も示します。

表 541. STM32H563/H573および STM32H562 機能

USART のモード／機能	STM32H563/H573
USART1	FULL
USART2	FULL
USART3	FULL
USART6	FULL
USART10	FULL
USART11	FULL
UART4	基本
UART5	基本
UART7	基本
UART8	基本
UART12	基本
LPUART1	LP

表 542. USART/LPUART の機能

USART のモード／機能 ⁽¹⁾	フル機能セット	基本機能セット	低消費電力機能セット
モデムのハードウェアフロー制御	X	X	X
DMA を使用した連続通信	X	X	X
マルチプロセッサ通信	X	X	X

表 542. USART/LPUART の機能 (続き)

USART のモード／機能 ⁽¹⁾	フル機能セット	基本機能セット	低消費電力機能セット
同期モード (マスタ／スレーブ)	X	-	-
スマートカードモード	X	-	-
単線半二重通信	X	X	X
Ir SIR ENDEC ブロック	X	X	-
LIN モード	X	X	-
デュアルクロックドメイン	X	X	X
レシーバタイムアウト割込み	X	X	-
Modbus 通信	X	X	-
自動ボーレート検出	X	X	-
ドライバイネーブル	X	X	X
USART データ長	7、8、および 9 ビット		
Tx/Rx FIFO	X	X	X
Tx/Rx FIFO サイズ	8		
低消費電力モードからのウェイクアップ	X ⁽²⁾	X ⁽²⁾	X ⁽²⁾

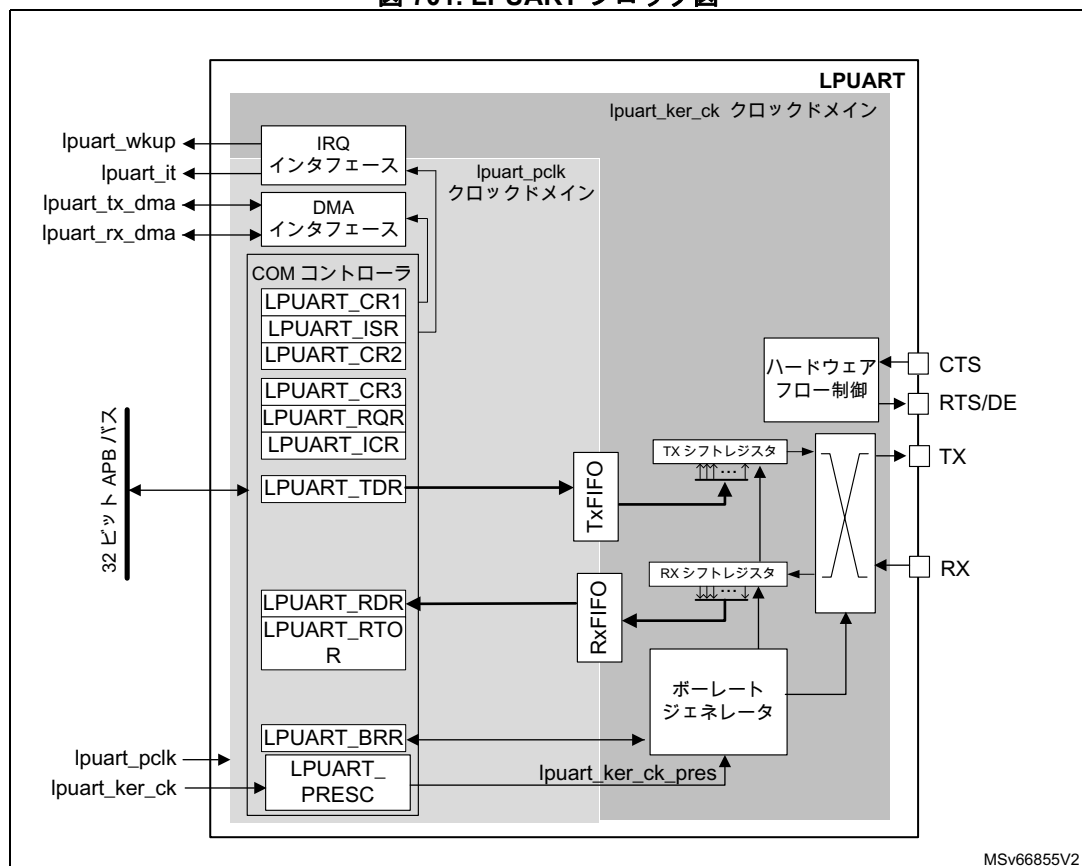
1. X : サポートされています。

2. STOP モードからのウェイクアップに対応

51.4 LPUART の機能詳細

51.4.1 LPUART ブロック図

図 701. LPUART ブロック図



51.4.2 LPUART ピンおよび内部信号

LPUART の入出力ピンの説明

- LPUART 双方向通信
LPUART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力 (RX) と送信データ出力 (TX) です。
 - **RX** (受信データ入力) :
RX はシリアルデータ入力です。
 - **TX** (送信データ出力)
トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TX ピンはハイレベルになります。単線モードでは、この I/O はデータの送受信に使用されます。
- RS232 ハードウェアフロー制御モード
RS232 ハードウェアフロー制御モードでは、以下のピンが必要です。
 - **CTS** (Clear To Send)
ハイレベルのとき、この信号は現在の転送の終わりにデータ送信をブロックします。
 - **RTS** (Request To Send)
ローレベルのとき、この信号は USART がデータを受信する準備ができたことを示します。
- RS485 ハードウェアフロー制御モード
RS485 ハードウェア制御モードでは、**DE** (Driver Enable) ピンが必要です。この信号は、外部トランシーバの送信モードを有効にします。

LPUART 入出力ピンおよび内部信号のリストについては、表 543 および表 544 を参照してください。

表 543. LPUART の入出力ピン

ピン名	信号タイプ	説明
LPUART_RX	入力	シリアルデータ受信入力。
LPUART_TX	出力	送信データ出力。
LPUART_CTS	入力	送信可否
LPUART_RTS	出力	送信要求
LPUART_DE ⁽¹⁾	出力	ドライバイネーブル

1. LPUART_DE と LPUART_RTS は同じピンを共有します。

LPUART の入出力信号の説明

表 544. LPUART 内部入力／出力信号

ピン名	信号タイプ	説明
usart_pclk	入力	APB クロック
lpuart_ker_ck	入力	LPUART カーネルクロック
lpuart_wkup	出力	LPUART はウェイクアップ割込みに対応
lpuart_it	出力	LPUART グローバル割込み
lpuart_tx_dma	入力／出力	LPUART 送信 DMA リクエスト
lpuart_rx_dma	入力／出力	LPUART 受信 DMA リクエスト

51.4.3 LPUART クロック

図 701 に示した簡易ブロック図には、以下の完全に独立した 2 つのクロックドメインが示されています。

- **lpuart_pclk** クロックドメイン
lpuart_pclk クロック信号はペリフェラルバスインタフェースにクロック供給します。LPUART レジスタへアクセスが必要なとき、アクティブになっていなければなりません。
- **lpuart_ker_ck** カーネルクロックドメイン
lpuart_ker_ck は LPUART クロックソースです。**lpuart_pclk** とは独立で、RCC によって供給されます。したがって、**lpuart_ker_ck** クロックが停止しているときでも、LPUART レジスタは読み書きできます。
デュアルクロックドメイン機能がサポートされていないときは、**lpuart_ker_ck** は **lpuart_pclk** クロックと同じになります。

lpuart_pclk と **lpuart_ker_ck** の間には何も制約はありません。**lpuart_ker_ck** は **lpuart_pclk** より速いことも遅いこともあり、十分に速く通信を管理できるソフトウェアの能力以外に制限はありません。

51.4.4 LPUART キャラクタの説明

ワード長は、LPUART_CR1 レジスタの M ビット (M0 : ビット 12 および M1 : ビット 28) をプログラムすることによって、7、8、または 9 ビットに設定できます (図 675 を参照)。

- 7 ビットのキャラクタ長 : M[1:0] = 10
- 8 ビットのキャラクタ長 : M[1:0] = 00
- 9 ビットのキャラクタ長 : M[1:0] = 01

デフォルトでは、信号 (TX または RX) はスタートビットの処理中ではロー状態です。また、ストップビットの処理中にはハイ状態です。

これらの値は、極性設定制御により、各信号について個別に反転できます。

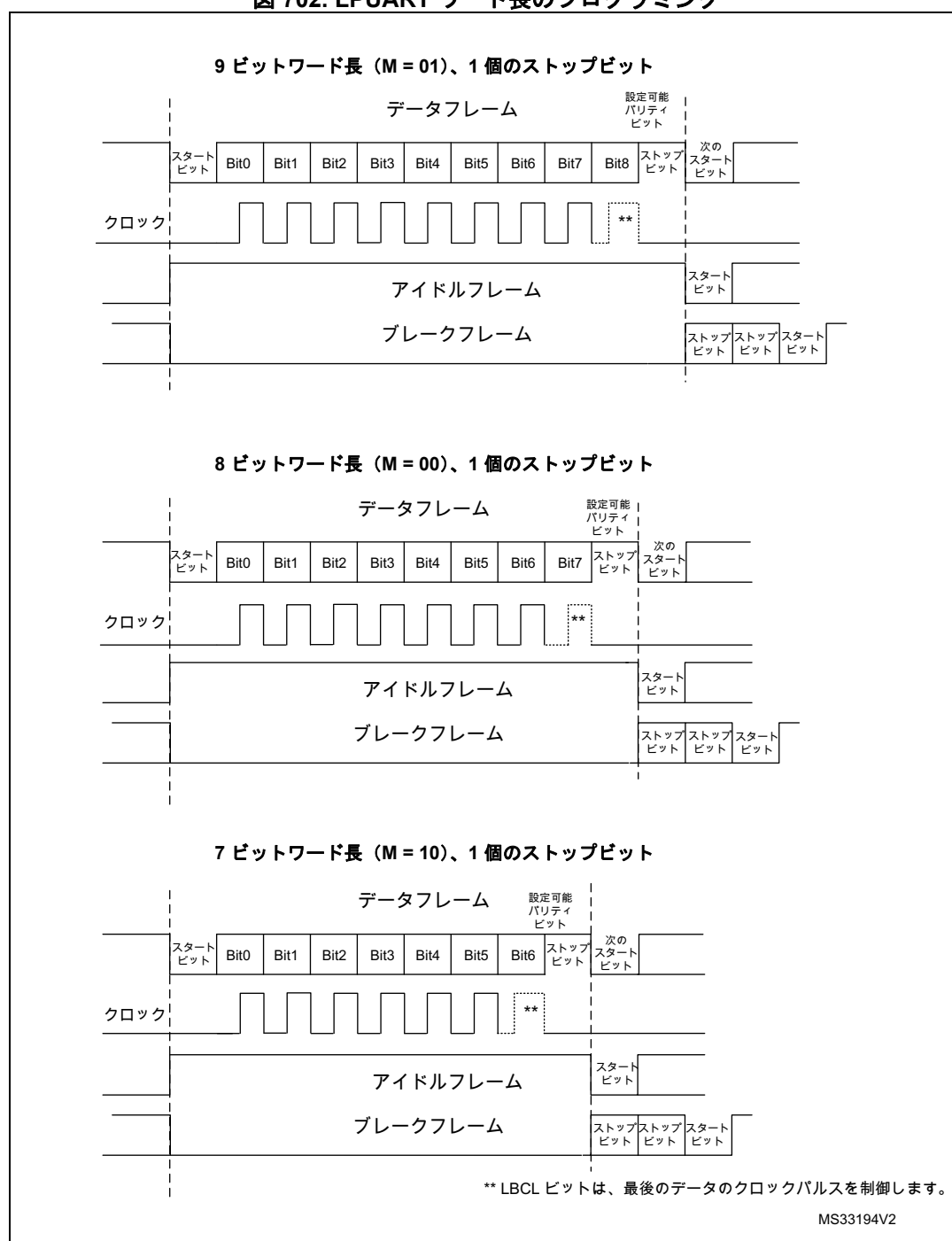
アイドルキャラクタは、すべてが「1」のフレームとして解釈されます (「1」の数にはストップビットの数が含まれます)。

ブレークキャラクタは、フレーム周期中に「0」を受信することと解釈されます。ブレークフレームの終了時、トランスミッタは 2 個のストップビットを挿入します。

送信と受信は共通ボーレートジェネレータによって駆動されます。送信および受信クロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 702. LPUART ワード長のプログラミング



51.4.5 LPUART の FIFO と閾値

LPUART は FIFO モードで動作できます。

LPUART は送信 FIFO (TXFIFO) と受信 FIFO (RXFIFO) を備えています。FIFO モードは、LPUART_CR1 レジスタの FIFOEN ビット (ビット 29) をセットすることによって有効になります。

最大のデータワード長が 9 ビットなので、TXFIFO は 9 ビット幅です。しかし、RXFIFO のデフォルト幅は 12 ビットです。この理由は、レシーバは FIFO にデータを格納するだけでなく、各キャラクタに伴うエラーフラグ (パリティエラー、ノイズエラー、およびフレーミングエラーフラグ) も格納するためです。

注： 受信データは、対応するフラグとともに RXFIFO に格納されます。ただし、RDR を読み出すときは、データのみが読み出されます。

ステータスフラグは、LPUART_ISR レジスタで入手可能です。

Tx および Rx 割込みがトリガされる TXFIFO および RXFIFO のレベルを定義することができます。これらの閾値は、LPUART_CR3 制御レジスタの RXFTCFG および TXFTCFG ビットフィールドによってプログラムされます。

このとき、

- RXFIFO に受信したデータの数 RXFTCFG ビットフィールドでプログラムされた閾値に達すると、Rx 割込みが生成されます。
この場合、LPUART_ISR レジスタの RXFT フラグがセットされます。これは、RXFTCFG 分のデータが受信されたことを意味します。つまり、1 データが LPUART_RDR にあり、(RXFTCFG - 1) データが RXFIFO に入っています。例として、RXFTCFG が "101" にプログラムされている場合、FIFO サイズに相当する量のデータが受信された時に、RXFT フラグがセットされます。つまり、FIFO サイズ - 1 のデータが RXFIFO に入り、1 データが LPUART_RDR に入っています。その結果、次に受信されるデータによってオーバーランフラグがセットされることはありません。
- TXFIFO 内の空き場所の数 TXFTCFG ビットフィールドでプログラムされた閾値に達すると、Tx 割込みが生成されます。

51.4.6 LPUART トランスミッタ

トランスミッタは、M ビットのステータスに応じて、7 または 8 または 9 ビットのデータワードを送信できます。トランスミッタ機能を有効にするには、送信イネーブルビット (TE) をセットする必要があります。送信シフトレジスタ内のデータは、TX ピンで出力されます。

キャラクタ送信

LPUART 送信時、データは LSB ファースト (デフォルト設定) で TX ピンにシフトアウトされます。このモードでは、LPUART_TDR レジスタは、内部バスと送信シフトレジスタの間のバッファ (TDR) で構成されます (図 701 を参照)。

FIFO モードが有効なとき、LPUART_TDR レジスタに書き込まれたデータは TXFIFO のキューに入ります。

各キャラクタの前には、スタートビット (1 ビット周期、ロー論理レベル) があります。キャラクタは、設定可能な数のストップビットで終端されます。

ストップビットの数は 1 または 2 にできます。

注： 送信データを LPUART_TDR に書き込む前に、TE ビットをセットする必要があります。
データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ボーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中の現在のデータは失われます。

TE ビットが有効になると、アイドルフレームが送信されます。

設定可能なストップビット

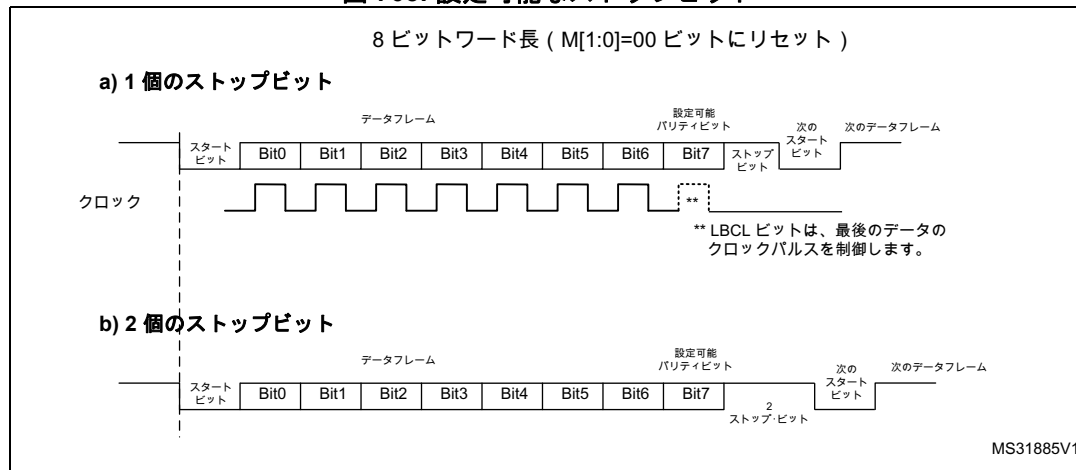
各キャラクタとともに送信されるストップビットの数は、LPUART_CR2（ビット 13、12）でプログラミングできます。

- **1 個のストップビット**：ストップビット数のデフォルト値です。
- **2 個のストップビット**：通常の LPUART モード、単線モード、およびモデムモードでサポートされます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信は、10 個のロービット（M[1:0] = “00” のとき）または 11 個のロービット（M[1:0] = “01” のとき）または 9 個のロービット（M[1:0] = “10” のとき）の後に 2 個のストップビットが続きます。長いブ레이크（9/10/11 個のロービットを超える長さのブ레이크）を送信することはできません。

図 703. 設定可能なストップビット



キャラクタ送信手順

キャラクタを送信するには、次の手順に従います。

1. LPUART_CR1 の M ビットをプログラムして、ワード長を定義します。
2. LPUART_BRR レジスタを使用して、目的のボーレートを選択します。
3. LPUART_CR2 レジスタでストップビットの数をプログラミングします。
4. LPUART_CR1 レジスタの UE ビットに 1 を書き込んで、LPUART を有効にします。
5. マルチバッファ通信を行う場合は、LPUART_CR3 レジスタの DMA 有効 (DMAT) を選択します。セクション 51.4.13: DMA および LPUART を使用した連続通信の説明に基づいて、DMA レジスタを設定します。
6. LPUART_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。

7. 送信するデータを LPUART_TDR レジスタに書き込みますシングルバッファの場合、送信される各データにこの操作を繰り返します。
 - FIFO モードが無効になっている場合は、LPUART_TDR に 1 つのデータを書き込むと、TXE フラグがクリアされます。
 - FIFO モードが有効になっている場合は、LPUART_TDR に 1 つのデータを書き込むと、1 つのデータが TXFIFO に追加されます。TXFNF フラグがセットされたとき、LPUART_TDR への書き込み動作が行われます。このフラグは TXFIFO がフルになるまでセットされたままになります。
8. LPUART_TDR レジスタに最後のデータを書き込んだら、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。
 - FIFO モードが無効になっている場合、これは、最後のフレームの送信が完了したことを示します。
 - FIFO モードが有効になっている場合、これは、TXFIFO とシフトレジスタの両方が空になっていることを示します。

このチェックは、LPUART が無効になったり、停止モードに入ったりするときに、最後の送信が壊れないようにするために必要です。

1 バイト通信

- FIFO モードが無効の場合

送信データレジスタに書き込むと、必ず TXE ビットがクリアされます。TXE フラグがハードウェアによってセットされ、以下のことを示します。

 - データは LPUART_TDR レジスタからシフトレジスタへ移動され、データ送信が開始しています。
 - LPUART_TDR レジスタは空です。
 - 次のデータを、前のデータに上書きせずに、LPUART_TDR レジスタに書き込みます。

TXEIE ビットがセットされている場合、TXE フラグは割込みを生成します。

送信が行われているとき、LPUART_TDR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、さらに、現在の送信の最後にシフトレジスタにコピーされます。

送信が行われていないときには、LPUART_TDR レジスタへの書き込み命令によってデータがシフトレジスタに格納され、データ送信が開始され、TXE ビットがセットされます。
- FIFO モードが有効になっている場合は、以下を示すためにハードウェアによって TXFNF (TXFIFO はフルではない) フラグがセットされます。
 - TXFIFO はフルではありません。
 - LPUART_TDR レジスタは空です。
 - 次のデータを、前のデータに上書きせずに、LPUART_TDR レジスタに書き込みます。送信が行われているとき、LPUART_TDR レジスタへの書き込み命令によってデータが TXFIFO に格納されます。現在の送信の最後にデータが TXFIFO からシフトレジスタにコピーされます。

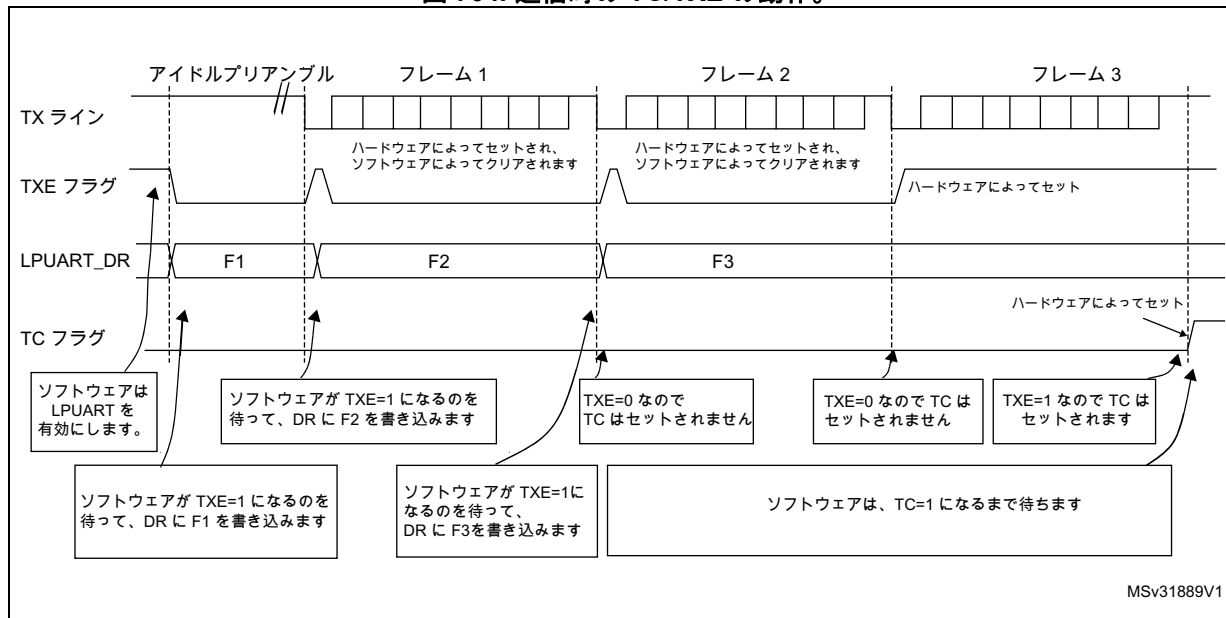
TXFIFO がフルではない場合、LPUART_TDR への書き込み後も、TXFNF フラグは 1 のままに留まります。TXFIFO がフルになると、クリアされます。TXFNEIE ビットがセットされている場合、このフラグは割込みを生成します。

あるいは、TXFIFO 閾値に達した時、割込みが生成され、データを TXFIFO に書き込むことができます。この場合、CPU は、プログラムされた閾値によって定義されたデータのブロックを書き込むことができます。

フレームが送信され (ストップビットの後)、TXE フラグ (FIFO モードの場合は TXFE) がセットされると、TC ビットはハイレベルになります。LPUART_CR1 レジスタの TCIE ビットがセットされると、割込みが生成されます。

LPUART_TDR レジスタに最後のデータを書き込んだ後は、LPUART を無効にしたり、マイクロコントローラを低消費電力モードにする前に TC=1 になるまで待つ必要があります (図 704: 送信時の TC/TXE の動作。を参照)。

図 704. 送信時の TC/TXE の動作。



注： FIFO 管理が有効になっているときは、TXFNF フラグがデータ送信のために使用されます。

ブレークキャラクタ

SBKRQ ビットをセットすると、ブレークキャラクタが送信されます。ブレークフレーム長は、M ビットに依存します (図 702 を参照)。

SBKRQ ビットに 1 が書き込まれた場合、現在のキャラクタ送信の完了後、TX ラインにブレークキャラクタが送信されます。SBKF ビットは書き込み操作によってセットされ、ブレークキャラクタが完了すると (ブレークキャラクタの後のストップビット時に) ハードウェアによってリセットされます。LPUART は、次のフレームのスタートビットの認識を保証するために、ブレークフレームの終わりに 2 ビットの長さの間、ロジック 1 信号 (STOP) を挿入します。

SBKRQ ビットがセットされると、現在のキャラクタ送信の最後に、ブレークキャラクタが送信されます。

FIFO モードが有効になっている場合、TXFIFO がフルになっていても、ブレークキャラクタの送信はデータ送信より優先されます。

アイドルキャラクタ

TE ビットをセットすると、LPUART は、最初のデータフレームの前にアイドルフレームを送信します。

51.4.7 LPUART レシーバ

LPUART は、LPUART_CR1 レジスタの M ビットに応じて、7 または 8 または 9 ビットのデータワードを受信できます。

スタートビット検出

LPUART では、立ち下がりエッジが Rx ラインで発生したときスタートビットが検出され、その後、スタートビットの中央でサンプルが採取され、まだ“0”であることが確認されます。スタートサンプルが“1”であった場合、ノイズエラーフラグ (NE) がセットされ、スタートビットが破棄され、レシーバは新しいスタートビットを待ちます。そうでない場合、レシーバは通常通り、着信するすべてのビットのサンプリングを続けます。

キャラクタの受信

LPUART の受信時には、データは RX ピンを通じて LSB ファースト（デフォルト設定）でシフトインされます。このモードでは、LPUART_RDR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

キャラクタ受信手順

キャラクタを受信するには、次の手順に従います。

1. LPUART_CR1 の M ビットをプログラムして、ワード長を定義します。
2. ボーレートレジスタ LPUART_BRR を使用して、目的のボーレートを選択します。
3. LPUART_CR2 レジスタでストップビットの数をプログラミングします。
4. LPUART_CR1 レジスタの UE ビットに 1 を書き込んで、LPUART を有効にします。
5. マルチバッファ通信を行う場合には、LPUART_CR3 レジスタの DMA 有効 (DMAR) を選択します。[セクション 51.4.13 : DMA および LPUART を使用した連続通信](#)の説明に基づいて、DMA レジスタを設定します。
6. LPUART_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- FIFO モードが無効な場合は、RXNE ビットがセットされます。これは、シフトレジスタの内容が RDR レジスタに転送されたことを示します。言い換えると、データは受信され、読み出し可能です（関連するエラーフラグも同様です）。
- FIFO モードが有効な場合、RXFIFO が空ではないことを示す RXFNE ビットがセットされます。LPUART_RDR を読み出すと、RXFIFO に入力された最も古いデータが返されます。データが受信されると、対応するエラービットとともに RXFIFO に格納されます。
- RXNEIE (FIFO モードの場合は RXFNEIE) ビットがセットされていた場合、割込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。
- マルチバッファ通信モードでは、
 - FIFO モードが無効のとき、RXNE フラグはバイト受信ごとにセットされ、受信データレジスタの DMA 読み出しによってクリアされます。
 - FIFO モードが有効な場合は、RXFIFO が空ではないときに RXFNE フラグがセットされます。DMA リクエストのたびに、RXFIFO から 1 データが取り出されます。DMA リクエストは、RXFIFO が空ではないとき、すなわち、RXFIFO から読み出されるべきデータがあるときに、トリガされます。

- シングルバッファモードでは、
 - FIFO モードが無効の場合、RXNE フラグのクリアは、ソフトウェアによる LPUART_RDR レジスタからの読出しによって行われます。RXNE フラグは、LPUART_RQR レジスタの RXFRQ に 1 を書き込むことによってもクリアすることもできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。
 - FIFO モードが有効な場合は、RXFIFO が空ではないときに RXFNE フラグがセットされます。LPUART_RDR レジスタからの読出し動作のたびに、RXFIFO から 1 つのデータが取り出されます。RXFIFO が空になると、RXFNE フラグがクリアされます。RXFNE フラグは、LPUART_RQR レジスタの RXFRQ ビットに 1 を書き込むことによってもクリアすることもできます。RXFIFO がフルのとき、オーバーランエラーを避けるため、次のキャラクタの受信が終了する前に、RXFIFO 内の最初のエントリを読み出す必要があります。RXFNEIE ビットがセットされている場合、RXFNE フラグは割込みを生成します。あるいは、RXFIFO 閾値に達した時、割込みが生成され、データを RXFIFO から読み出すことができます。この場合、CPU は、プログラムされた閾値によって定義されたデータのブロックを読み出すことができます。

ブレークキャラクタ

ブレークキャラクタを受信すると、USART はブレークキャラクタをフレーミングエラーとして処理します。

アイドルキャラクタ

アイドルフレームが検出された場合、データキャラクタ受信と同じように処理されますが、違いは、IDLEIE ビットがセットされている場合に割込みが生成されることです。

オーバーランエラー

- FIFO モードが無効の場合

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。

RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。RXNE フラグは、バイトを受信するたびにセットされます。

次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

 - ORE ビットがセットされます。
 - RDR の内容は失われません。LPUART_RDR の読出しが行われると、前のデータが使用できます。
 - シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
 - RXNEIE ビットまたは EIE ビットがセットされている場合、割込みが生成されます。
- FIFO モードが有効な場合

受信 FIFO がフルのときにシフトレジスタが転送される準備ができると、オーバーランエラーが発生します。

RXFIFO に 1 つの空き場所ができるまで、データをシフトレジスタから LPUART_RDR レジスタに転送することはできません。RXFIFO が空でないとき、RXFNE フラグがセットされます。

RXFIFO がフルで、シフトレジスタが転送される準備ができている場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RXFIFO の最初のエントリは失われません。LPUART_RDR の読出しを行うと、そのデータが入手できます。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXFNEIE ビットまたは EIE ビットがセットされている場合、割込みが生成されます。

ORE ビットは、ICR レジスタの ORECF ビットをセットすることによってリセットされます。

注： ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。T

FIFO モードが無効になっているとき、2 つの可能性があります。

- RXNE=1 の場合、有効な最後のデータは、受信レジスタ RDR に格納され、読出しが可能です。
- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものが残ってないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。

クロックソースの選択

クロックソースの選択は、クロック制御システムを通じて行われます（リセットおよびクロック制御 (RCC) のセクションを参照）。クロックソースは、UE ビットのセットによって LPUART を有効にする前に選ぶ必要があります。

クロックソースは、次の 2 つの基準に従って選択する必要があります。

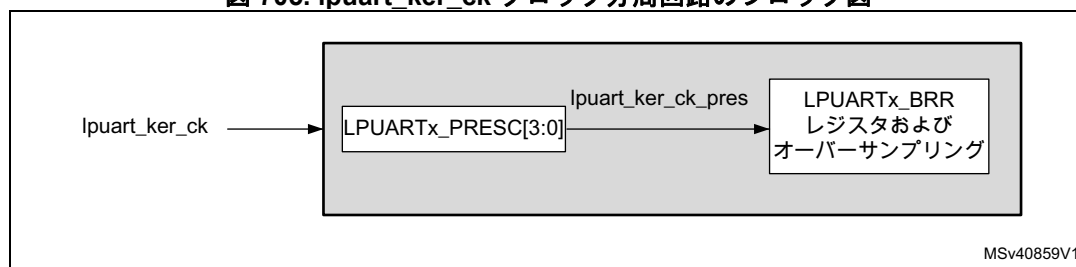
- LPUART を低消費電力モードで使用できること
- 通信速度

クロックソース周波数は、lpuart_ker_ck です。

デュアルクロックドメインと低消費電力モードからのウェイクアップ機能がサポートされるときには、lpuart_ker_ck クロックソースは RCC で設定できます（リセットおよびクロック制御 (RCC) のセクションを参照）。そうしない場合、lpuart_ker_ck クロックは lpuart_pclk と同じになります。

lpuart_ker_ck クロックは、LPUART_PRESC レジスタのプログラム可能な係数によって分周できます。

図 705. lpuart_ker_ck クロック分周回路のブロック図



lpuart_ker_ck ソースによっては、LPUART は MCU が低消費電力モードのときにデータを受信することができます。受信データとウェイクアップモードの選択に応じて、LPUART は必要なときに MCU をウェイクアップして、ソフトウェアが LPUART_RDR レジスタを読み出すことによって、または DMA によって受信データを転送します。

他のクロックソースの場合、LPUART 通信を可能にするためには、システムをアクティブにする必要があります。

通信速度の範囲（特に最大通信速度）もクロックソースによって決まります。

レシーバは各受信ビットをビット周期のできるだけ中央でサンプリングします。各受信ビットについて 1 つのサンプルだけが採取されます。

注： データのノイズ検出は行われません。

フレーミングエラー

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合、フレーミングエラーが検出されます。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから LPUART_RDR レジスタへ転送されます。
- 1 バイト通信の場合、割込みは生成されません。ただし、このビットは、割込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、LPUART_CR3 レジスタの EIE ビットがセットされている場合に割込みが発行されます。

FE ビットは、LPUART_ICR レジスタの FECF に 1 を書き込むことによってリセットされます。

受信時の設定可能なストップビット

受信するストップビット数は、LPUART_CR2 の制御ビットを通じて設定でき、通常モードでは 1 または 2 にできます。

- **1 個のストップビット**：ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
- **2 個のストップビット**：2 個のストップビットのサンプリングは、2 番目のストップビットの中央で行われます。RXNE および FE フラグは、このサンプルの直後、すなわち 2 番目のストップビット中にセットされます。最初のストップビットでは、フレーミングエラーの検査は行われません。

51.4.8 LPUART ボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートは、LPUART_BRR レジスタでプログラムされた値に設定されます。

$$\text{Tx/Rx ボー} = \frac{256 \times \text{lpuart_ker_ck_pres}}{\text{LPUARTDIV}}$$

LPUARTDIV は、LPUART_BRR レジスタで定義されます。

注： ボーカウンタは、LPUART_BRR への書き込み操作後、ボーレジスタの新しい値によって更新されます。したがって、通信中はボーレートレジスタの値を変更しないでください。

LPUART_BRR レジスタに 0x300 未満の値を書き込むことは禁じられています。

f_{CK} は、3 x ボーレートから 4096 x ボーレートの範囲内でなければなりません。

LPUART クロックソースが LSE の場合に達成できる最大ボーレートは、9600 baud です。LSE クロックとは別のクロックソースによって LPUART にクロック供給すると、さらに高いボーレートを達成できます。たとえば、LPUART クロックソース周波数が 100 MHz の場合、達成できる最大ボーレートは約 33 Mbaud です。

表 545. lpuart_ker_ck_pres = 32.768 kHz でプログラムされたボーレートのエラー計算

ボーレート		lpuart_ker_ck_pres= 32.768 kHz		
S.No	目標	実際	ボーレートレジスタに プログラミングされている値	誤差(%) = (計算値 - 目標値) B レート / 目標の B レート
1	0.3 Kbaud	0.3 Kbaud	0x6D3A	0
2	0.6 Kbaud	0.6 Kbaud	0x369D	0
3	1200 baud	1200.087 baud	0x1B4E	0.007
4	2400 baud	2400.17 baud	0xDA7	0.007
5	4800 baud	4801.72 baud	0x6D3	0.035
6	9600 Kbaud	9608.94 baud	0x369	0.093

表 546. f_{CK} = 100 MHz でプログラムされたボーレートのエラー計算

ボーレート		f _{CK} = 100 MHz		
S.No	目標	実際	ボーレートレジスタに プログラミングされている値	誤差(%) = (計算値 - 目標値) B レート / 目標の B レート
1	38400 baud	38400.04 baud	A2C2A	0.0001
2	57600 baud	57600.06 baud	6C81C	0.0001
3	115200 baud	115200.12 baud	3640E	0.0001
4	230400 baud	230400.23 baud	1B207	0.0001
5	460800 baud	460804.61 baud	D903	0,001
6	921600 baud	921625.81 baud	6C81	0.0028
7	4000 Kbaud	4000000.00 baud	1900	0
8	10000 Kbaud	10000000.00 baud	A00	0
9	20000 Kbaud	20000000.00 baud	500	0
10	30000 Kbaud	33032258.06 baud	307	0.1

51.4.9 クロック偏差に対する LPUART レシーバの許容誤差

LPUART の非同期レシーバは、クロックシステムの合計偏差が LPUART レシーバの許容誤差未満の場合のみ、正しく動作します。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタのローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$$DTRA + DQUANT + DREC + DTCL + DWU < \text{LPUART レシーバの許容誤差}$$

ここで、

DWU は、低消費電力モードからのウェイクアップが使用されたときのサンプリングポイントの偏差によるエラーです。

M[1:0] = 01 の場合 :

$$DWU = \frac{t_{WULPUART}}{11 \times T_{bit}}$$

M[1:0] = 00 の場合 :

$$DWU = \frac{t_{WULPUART}}{10 \times T_{bit}}$$

M[1:0] = 10 の場合 :

$$DWU = \frac{t_{WULPUART}}{9 \times T_{bit}}$$

$t_{WULPUART}$ は、スタートビットの立ち下がりがエッジが検出され、クロック (ペリフェラルによるリクエスト) が動作可能状態になってペリフェラルに達し、レギュレータが動作可能状態になるまでの時間です。

LPUART レシーバは、[表 547](#) で指定された最大許容偏差まで、データを正しく受信できます。

- LPUART_CR2 レジスタの STOP[1:0] ビットによって定義されたストップビットの数
- LPUART_BRR レジスタの値

表 547. LPUART レシーバの許容誤差

M ビット	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
8 ビット (M=00)、 ストップビット 1 個	1.82%	2.56%	3.90%	4.42%
9 ビット (M=01)、 ストップビット 1 個	1.69%	2.33%	2.53%	4.14%
7 ビット (M=10)、 ストップビット 1 個	2.08%	2.86%	4.35%	4.42%
8 ビット (M=00)、 ストップビット 2 個	2.08%	2.86%	4.35%	4.42%

表 547. LPUART レシーバの許容誤差 (続き)

M ビット	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
9 ビット (M=01)、 ストップビット 2 個	1.82%	2.56%	3.90%	4.42%
7 ビット (M=10)、 ストップビット 2 個	2.34%	3.23%	4.92%	4.42%

注： 表 547 で指定されたデータは、M ビット = “00” のとき、受信フレームに正確に 10 ビット時間のアイドルフレームが含まれる特殊なケースで、若干異なることがあります (M ビット = “01” のときには 11 ビット時間、または M ビット = “10” のときには 9 ビット時間)。

51.4.10 LPUART マルチプロセッサ通信

LPUART のマルチプロセッサ通信が可能です (ネットワーク内で複数の LPUART を接続して)。たとえば、1 つの LPUART をマスタとして、その TX 出力を別の LPUART の RX 入力に接続することができます。別の LPUART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する LPUART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモード機能を使用するためには、LPUART_CR1 レジスタの MME ビットをセットする必要があります。

注： FIFO マネージメントが有効になっていて MME がすでにセットされている場合は、MME ビットはクリアしてはなりません。クリアした場合はすぐに (2 lpuart_ker_ck サイクル以内に) 再セットしてください。そうしないとミュートモードはアクティブのままになることがあります。

ミュートモードが有効な場合、

- 受信ステータスビットはセットできません。
- 受信割込みはすべて禁止されます。
- LPUART_ISR レジスタの RWU ビットは“1”にセットされます。特定の条件下では、LPUART_RQR レジスタの MMRQ ビットを通じて、RWU をハードウェアまたはソフトウェアによって自動的に制御できます。

LPUART は、LPUART_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

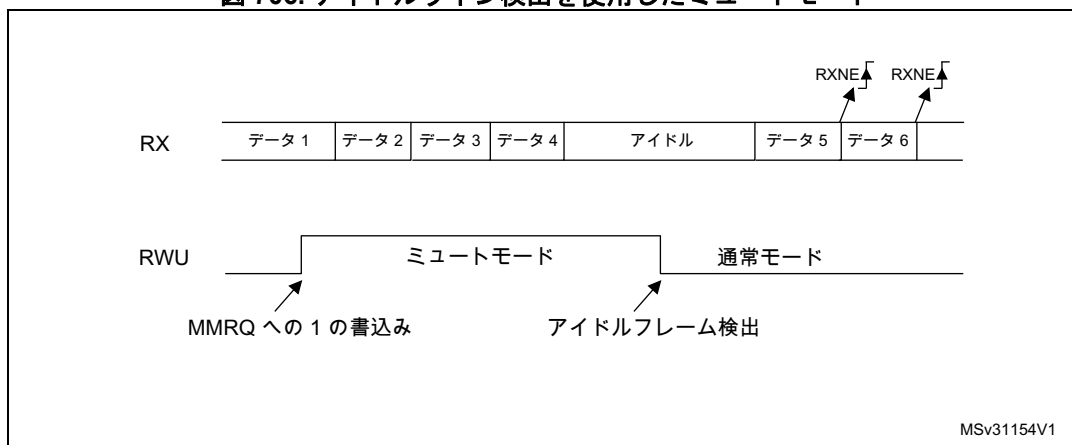
- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

アイドルライン検出 (WAKE=0)

MMRQ ビットに 1 が書き込まれ、RWU が自動的にセットされたときには、LPUART はミュートモードに入ります。

LPUART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、LPUART_ISR レジスタの IDLE ビットはセットされません。アイドルライン検出を使用したミュートモードの動作例を [図 706](#) に示します。

図 706. アイドルライン検出を使用したミュートモード



注： IDLE キャラクタがすでに経過しているときに MMRQ がセットされた場合は、ミュートモードに入りません (RWU はセットされません)。

ラインが IDLE のときに LPUART が有効にされた場合、1 IDLE フレーム後にアイドル状態が検出されます (1 キャラクタフレームの受信後だけでなく)。

4 ビット/7 ビットアドレスマーク検出 (WAKE=1)

このモードでは、MSB が "1" の場合、バイトはアドレスとして認識され、そうでない場合はデータとみなされます。アドレスバイトのうち、ターゲットレシーバのアドレスは 4 または 7 LSB です。7 または 4 ビットアドレス検出の選択は、ADDm7 ビットを使用して行われます。この 4 ビット/7 ビットワードは、レシーバによって、LPUART_CR2 レジスタの ADD ビットでプログラムされたレシーバの自己アドレスと比較されます。

注： 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) で行われます。

プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、LPUART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。LPUART がミュートモードに入ったときには、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

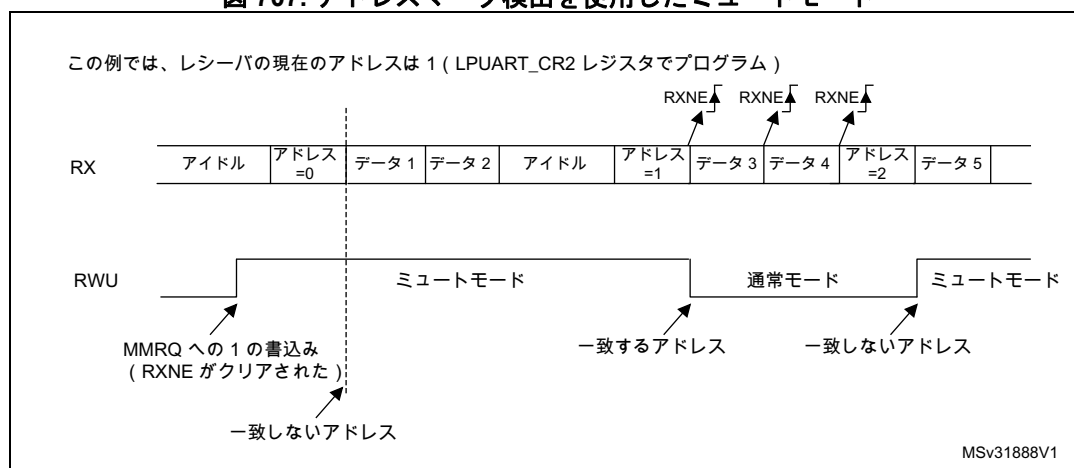
MMRQ ビットに 1 が書き込まれたときにも、LPUART はミュートモードに入ります。この場合、RWU ビットも自動的にセットされます。

プログラムされたアドレスに一致するアドレスキャラクタを受信すると、LPUART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE/RXFNE ビットがセットされます。

注： FIFO 管理が有効になっている場合、レシーバがデータの最後のビットをサンプリングしている間に MMRQ ビットがセットされると、ミュートモードに実際に移行する前にこのデータが受信されることがあります。

アドレスマーク検出を使用したミュートモードの動作例を [図 707](#) に示します。

図 707. アドレスマーク検出を使用したミュートモード



51.4.11 LPUART パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティチェック）を有効にするには、LPUART_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な LPUART フレームフォーマットを表 548 に示します。

表 548. LPUART フレームのフォーマット

M ビット	PCE ビット	LPUART フレーム ⁽¹⁾
00	0	SB 8 ビットデータ STB
00	1	SB 7 ビットデータ PB STB
01	0	SB 9 ビットデータ STB
01	1	SB 8 ビットデータ PB STB
10	0	SB 7 ビットデータ STB
10	1	SB 6 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。

2. データレジスタでは、PB は常に MSB 位置を取ります (M ビットの値に応じて、8 または 7 番目)。

偶数パリティ

パリティビットは、6、7、または 8 LSB ビット (M ビットの値に応じて) とパリティビットで構成されるフレーム内で「1」の数が偶数になるように計算されます。

たとえば、データ = 00110101 であり、4bit がセットされた場合、偶数パリティが選択された場合 (LPUART_CR1 の PS ビット = 0)、パリティビットは 0 になります。

奇数パリティ

パリティビットは、6、7、または 8 LSB ビット (M ビットの値に応じて) とパリティビットで構成されるフレーム内で「1」の数が奇数になるように計算されます。

たとえば、データ = 00110101 であり、4bit がセットされた場合、奇数パリティが選択された場合 (LPUART_CR1 の PS ビット = 1)、パリティビットは 1 になります。

受信中のパリティチェック

パリティチェックに失敗した場合、LPUART_ISR レジスタの PE フラグがセットされ、LPUART_CR1 レジスタの PEIE ビットがセットされている場合は割込みが生成されます。PE フラグは、LPUART_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによりクリアされます。

送信中のパリティ生成

LPUART_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます (偶数パリティが選択された場合 (PS=0) は偶数個の「1」、奇数パリティが選択された場合 (PS=1) は奇数個の「1」)。

51.4.12 LPUART 単線半二重通信

単線半二重モードを選択するには、LPUART_CR3 レジスタの HDSEL ビットをセットします。

LPUART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、LPUART_CR3 レジスタの制御ビット HDSEL で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が外部プルアップ付きの代替機能オープンドレインとして設定されるように、I/O を設定する必要があります。

この点を除くと、通信プロトコルは通常の LPUART モードと同じです。ラインの競合はソフトウェアによって管理する必要があります (たとえば、集中型アービタを使用して)。特に、送信がハードウェアによってブロックされることはなく、TE ビットがセットされている間は、データレジスタにデータが書き込まれるとすぐに、送信が続行されます。

注： LPUART 通信では、1 個のストップビット設定の場合、ストップビットの中央で RXNE フラグがセットされます。

51.4.13 DMA および LPUART を使用した連続通信

LPUART は、DMA を使用して連続通信を行うことができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

注： DMA モードがサポートされるかどうかについては、[セクション 50.4 : 2200 ページの USART の実装](#)を参照してください。DMA がサポートされない場合は、[セクション 50.5.7](#) の説明に従って LPUSRT を使用してください。連続通信を行うには、FIFO が無効のとき、LPUART_ISR レジスタの TXE/RXNE フラグをクリアします。

DMA を使用した送信

DMA モードでの送信を有効にするには、LPUART_CR3 レジスタの DMAT ビットをセットします。TXE フラグ (FIFO モードが有効な場合は TXFNF フラグ) がセットされるたびに、データは、DMA ペリフェラル (ダイレクトメモリアクセスコントローラのセクションを参照) を使用して SRAM 領域から LPUART_TDR レジスタにロードされます。DMA チャンネルを LPUART 送信用に割り付けるには、次の手順を実行します (x はチャンネル番号を示します)

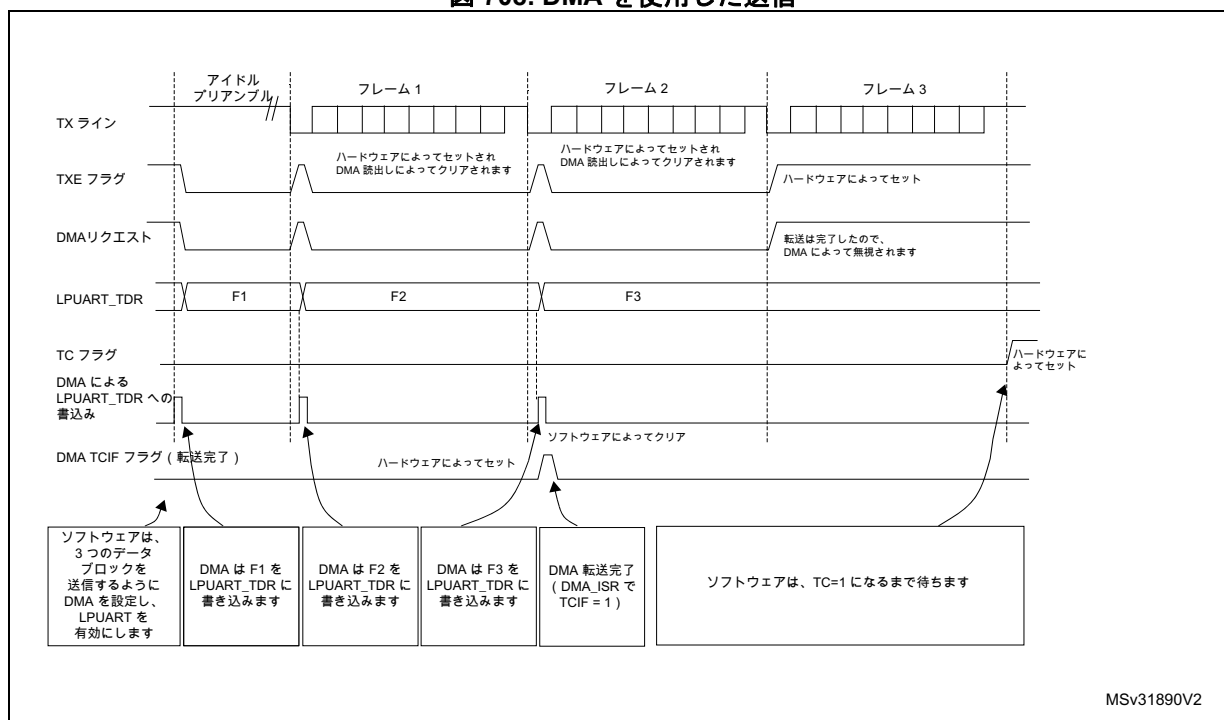
1. DMA 制御レジスタに LPUART_TDR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE (または FIFO モードが有効な場合は TXFNF) イベント後に、メモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE (または FIFO モードが有効な場合は TXFNF) イベント後に、このメモリ領域から LPUART_TDR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. LPUART_ICR レジスタの TCCF ビットをセットすることによって、LPUART_ISR レジスタの TC フラグをクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと (DMA_ISR レジスタの TCIF フラグがセットされます)、TC フラグを観察して LPUART 通信の完了を確認することができます。これは、LPUART を無効にしたり 低消費電力モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、すべてのデータ転送中、クリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

注： DMA の転送終了前に DMAT ビットをクリアしないでください。

図 708. DMA を使用した送信



注： FIFO 管理が有効になっているときは、DMA リクエストは、送信 FIFO ノットフル (すなわち、TXFNF = 1) によってトリガされます。

DMA を使用した受信

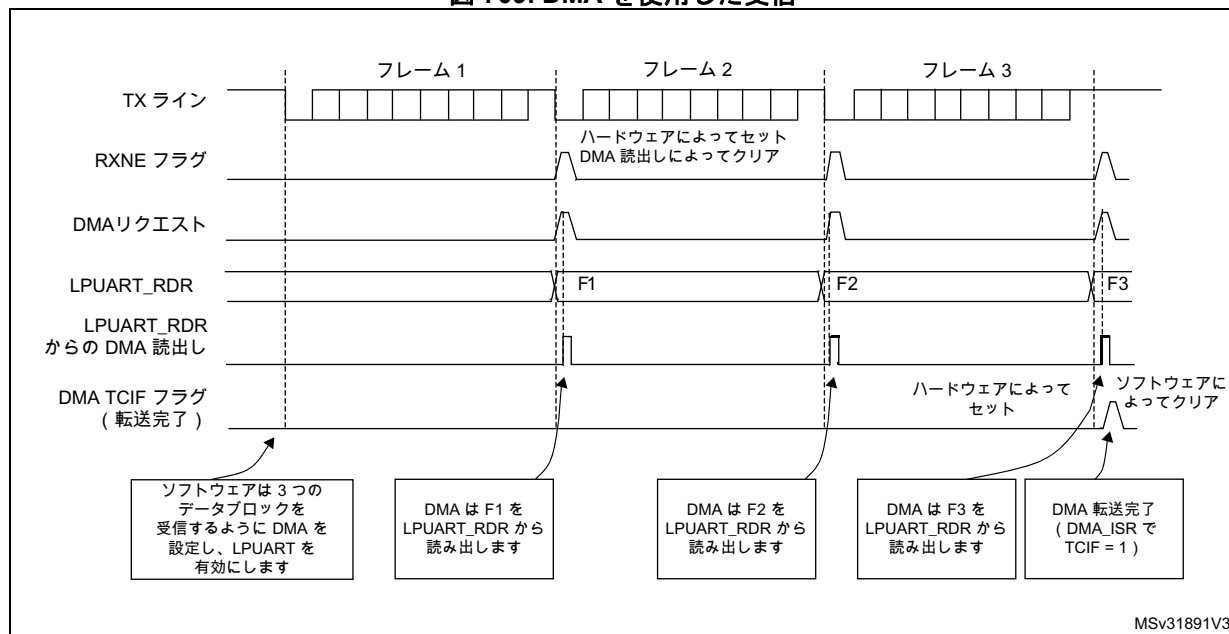
DMA モードでの受信を有効にするには、LPUART_CR3 レジスタの DMAR ビットをセットします。データバイトが受信されるたびに、データは、LPUART_RDR レジスタから DMA ペリフェラル（ダイレクトメモリアクセスコントローラ (DMA) のセクション）を使用して設定された SRAM 領域にロードされます。DMA チャンネルを LPUART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに LPUART_RDR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE (FIFO モードが有効な場合は RXFNE) イベント後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE (FIFO モードが有効な場合は RXFNE) イベント後に、LPUART_RDR からこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

注： DMA の転送終了前に DMAR ビットをクリアしないでください。

図 709. DMA を使用した受信



注： FIFO 管理が有効になっているときは、DMA リクエストは、受信 FIFO ノットエンプティ（すなわち、RXFNE = 1）によってトリガされます。

マルチバッファ通信における割り込み生成とエラーフラグ

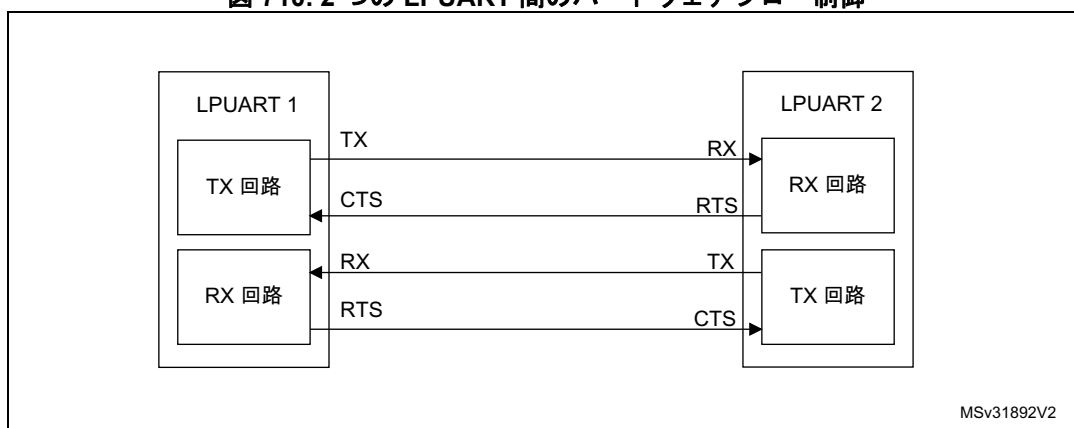
マルチバッファ通信モードでトランザクション中にエラーが発生した場合、現在のバイトの後にエラーフラグがアサートされます。割り込み有効フラグがセットされている場合、割り込みが生成されます。1 バイト受信において RXNE (FIFO モードが有効な場合は RXFNE) とともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別のエラーフラグ割り込み有効

ビット (LPUART_CR3 レジスタの EIE ビット) があり、これがセットされている場合、いずれかのエラーが発生すると、現在のバイトの後で割込みが有効になります。

51.4.14 RS232 ハードウェアフロー制御および RS485 ドライバ有効

CTS 入力と RTS 出力を使用すると、2 つのデバイス間でシリアルデータフローを制御できます。[図 710](#) に、このモードで 2 つのデバイスを接続する方法を示します。

図 710. 2 つの LPUART 間のハードウェアフロー制御

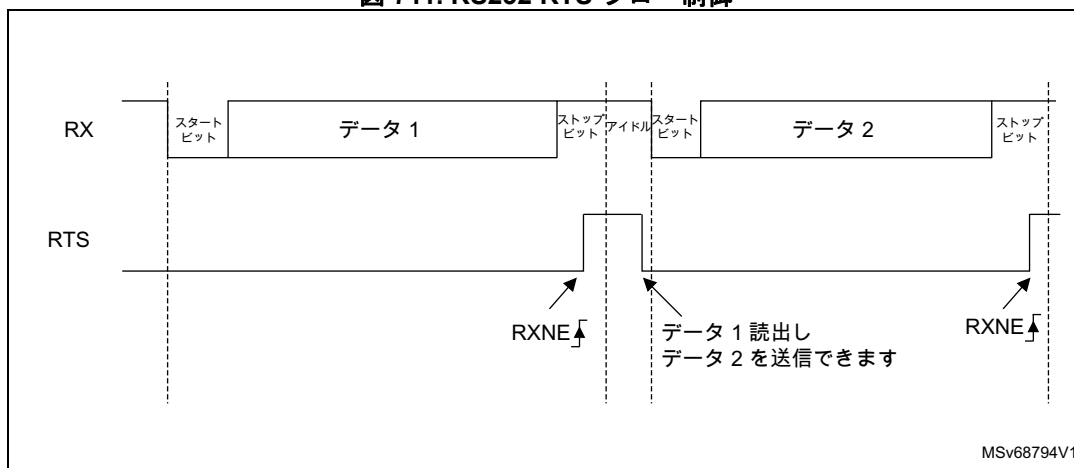


RS232 RTS と CTS のフロー制御は、LPUART_CR3 レジスタの RTSE ビットと CTSE ビットにそれぞれ 1 を書き込むことによって、個別に有効にできます。

RS232 RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、LPUART レシーバが新しいデータを受信可能である限り、RTS がネゲートされます (ローレベル接続)。受信レジスタがフルになると RTS がアサートされ、現在のフレームの終わりに送信が停止する予定であることを示します。[図 711](#) に、RTS フロー制御が有効な場合の通信例を示します。

図 711. RS232 RTS フロー制御



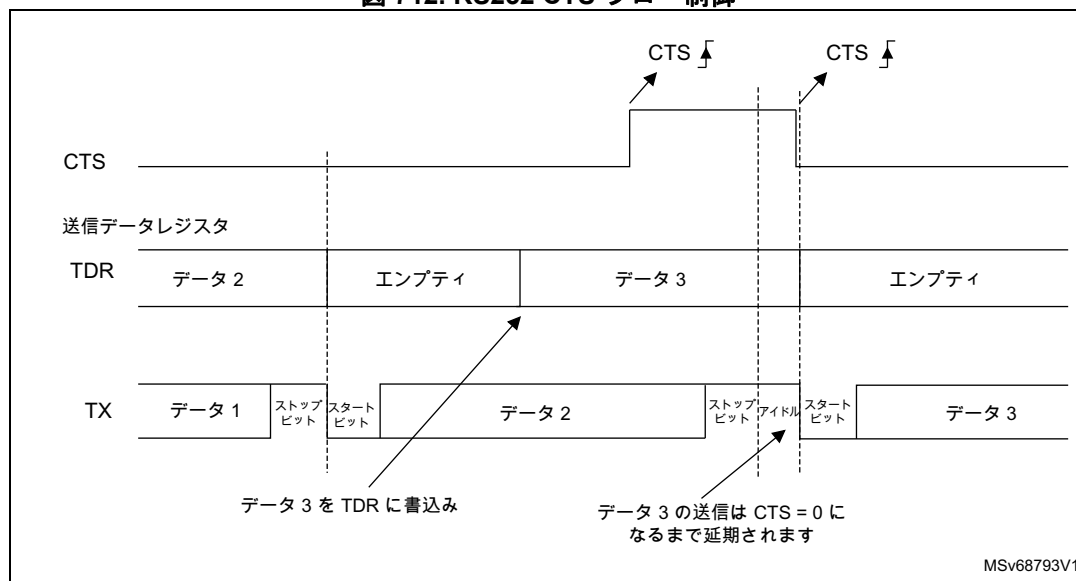
注： FIFO モードが有効な場合は、RXFIFO がフルのときにのみ、RTS がアサートされます。

RS232 CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、CTS 入力をチェックしてから、次のフレームを送信します。CTS がネゲートされた場合 (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE/TXFE=0 の場合)。そうでない場合、送信は行われません。送信中に CTS がアサートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、CTS 入力がトグルすると、CTSIF ステータスビットはハードウェアによってただちに自動的にセットされます。このビットは、レシーバの通信準備ができているかどうかを示します。LPUART_CR3 レジスタの CTSIE ビットがセットされている場合、割込みが生成されます。図 712 に、CTS フロー制御が有効な場合の通信例を示します。

図 712. RS232 CTS フロー制御



注： 正しい動作のために、CTS は、現在のキャラクタの終了の少なくとも 3 LPUART クロックソース周期前にネゲートする必要があります。さらに、2 x PCLK 周期より短いパルスでは CTSCF フラグがセットされない場合があることに注意してください。

RS485 ドライバ有効

ドライバ有効機能を有効にするには、LPUART_CR3 制御レジスタのビット DEM をセットします。これにより、DE (Driver Enable) 信号によって外部トランシーバ制御を有効にできます。アサーション時間は、DE 信号の有効化から スタートビットの開始までの時間です。LPUART_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。ネゲート時間は、送信メッセージの最後のストップビットの終了から DE 信号の無効化までの時間です。LPUART_CR1 制御レジスタの DEDT [4:0] ビットフィールドを使用してプログラムされます。DE 信号の極性は、LPUART_CR3 制御レジスタの DEP ビットを使用して設定できます。

LPUART の DEAT および DEDT は LPUART クロックソース (f_{CK}) サイクルで表されます。

- ドライバ有効アサーション時間は次のようになります。
 - $(1 + (DEAT \times P)) \times f_{CK}$ 、 $P \neq 0$ の場合
 - $(1 + DEAT) \times f_{CK}$ 、 $P = 0$ の場合
- ドライバ有効ネゲート時間は次のようになります。
 - $(1 + (DEDT \times P)) \times f_{CK}$ 、 $P \neq 0$ の場合
 - $(1 + DEDT) \times f_{CK}$ 、 $P = 0$ の場合

ここで、P = BRR[20:11]

51.4.15 LPUART 低消費電力管理

LPUART には、高度な低消費電力モード機能があり、lpuart_pclk クロックが無効になっているときでもデータを適切に転送することができます。

LPUART は、UESM ビットがセットされているとき、MCU を低消費電力モードからウェイクアップできます。

Usart_pclk がゲートされているとき、usart_pclk クロックの有効化を必要とする特定の動作が必要になった場合、LPUART はウェイクアップ割込み (usart_wkup) を生成します。

- FIFO モードが無効の場合

LPUART データレジスタを空にするために luart_pclk クロックを有効にする必要があります。

この場合、lpuart_wkup 割込みのソースは "1" にセットされた RXNE です。低消費電力モードに入る前に RXNEIE ビットをセットする必要があります。

- FIFO モードが有効な場合

次のために luart_pclk クロックを有効にする必要があります。

- TXFIFO を満たすため
- または RXFIFO を空にするため

この場合、lpuart_wkup 割込みのソースになる可能性のあるものは以下の通りです。

- RXFIFO ノットエンプティ。この場合、低消費電力モードに入る前に RXFNEIE ビットをセットする必要があります。
- RXFIFO フル。この場合、低消費電力モードに入る前に RXFFIE ビットをセットする必要があります。受信データの数 RXFIFO のサイズに一致し、RXFF フラグはセットされません。
- TXFIFO エンプティ。この場合、低消費電力モードに入る前に TXFEIE ビットをセットする必要があります。

これによって、低消費電力モード中にデータを TXFIFO/RXFIFO に送信/受信することができます。

低消費電力モードで、オーバーラン/アンダーランエラーを避けてデータを送信/受信するため、lpuart_wkup 割込みソースになり得るのは次のイベントのうちの 1 つです。

- TXFIFO 閾値に達した。この場合、低消費電力モードに入る前に TXFTIE ビットをセットする必要があります。
- RXFIFO 閾値に達した。この場合、低消費電力モードに入る前に RXFTIE ビットをセットする必要があります。

たとえば、ウェイクアップ時間が、ラインを経て 1 バイトを受信するための時間より少ない場合は、アプリケーションは閾値を RXFIFO の最大サイズに設定できます。

MCU を低消費電力モードからウェイクアップするための RXFIFO フル、TXFIFO エンプティ、RXFIFO ノットエンプティ、および RXFIFO/TXFIFO 閾値割込みを使用すれば、低消費電力モード中にできるだけ多くの LPUART 転送を行うことができ、電力消費を最適化できるメリットがあります。

あるいは、WUS ビットフィールドによって、特定の lpuart_wkup 割込みを選択することもできます。

MCU を低消費電力モードからウェイクアップするための RXFIFO フル、TXFIFO エンプティ、RXFIFO ノットエンプティ、および RXFIFO/TXFIFO 閾値割込みを使用すれば、低消費電力モード中にできるだけ多くの LPUART 転送を行うことができ、電力消費を最適化できるメリットがあります。

- 注： 低消費電力モードに移行する前に、LPUART 転送が進行中ではないことを確認してください。BUSY フラグをチェックすることでは、データ受信中に低消費電力モードに入らないことを保証できません。
- WUF フラグは、MCU が低消費電力モードか、アクティブモードかに関係なく、ウェイクアップイベントが検出されたときにセットされます。
- 初期化とレシーバの有効化の直後に低消費電力モードに入るときには、REACK ビットをチェックして、LPUART が実際に有効であることを確認する必要があります。
- 受信に DMA が使用されるときには、低消費電力モードに入る前に無効化し、低消費電力モードの終了時に再び有効にする必要があります。
- FIFO が有効なときには、アドレス一致時の低消費電力モードからのウェイクアップはミュートモードが有効な場合のみ可能です。

低消費電力モードでのミュートモードの使用

低消費電力モードに入る前に LPUART がミュートモードになった場合は、

- アイドル検出は低消費電力モードでは機能しないので、アイドル検出時にミュートモードからウェイクアップすることはできません。
- アドレス一致によるミュートモードからのウェイクアップが使用される場合、低消費電力モードからのウェイクアップのソースもアドレス一致でなければなりません。低消費電力モードに入るときに RXNE フラグがセットされた場合、アドレス一致によって低消費電力モードからウェイクアップしても、インタフェースはミュートモードのままです。

- 注： FIFO 管理が有効なとき、ミュートモードは何の制約もなく低消費電力モードからのウェイクアップとともに使用されます（すなわち、ミュートおよび低消費電力モードについて上に述べた 2 点は、FIFO 管理が無効なときのみ有効です）。

低消費電力モードで LPUART カーネルクロック lpuart_ker_ck がオフのときの低消費電力モードからのウェイクアップ

低消費電力モード中、lpuart_ker_ck クロックがオフになっている場合、LPUART 受信ラインの立ち下がりエッジが検出されると、lpuart_ker_ck_req 信号によって LPUART インタフェースが lpuart_ker_ck クロックをオンにするようリクエストします。その後、lpuart_ker_ck がフレーム受信に使用されます。

ウェイクアップイベントが確認された場合、MCU は低消費電力モードからウェイクアップし、データ受信が正常に続行します。

ウェイクアップイベントが確認されない場合、usart_ker_ck クロックが再度オフになり、MCU がウェイクアップせずに低消費電力モードに留まり、カーネルクロックリクエストが解除されます。

以下の例は、ウェイクアップイベントが「アドレス一致検出」にプログラムされ、FIFO 管理が無効になっている場合を示しています。

図 713 に、ウェイクアップイベントが確認された時の動作を示します。

図 713. 確認されたウェイクアップイベント (ウェイクアップイベント = アドレス一致、FIFO 無効)

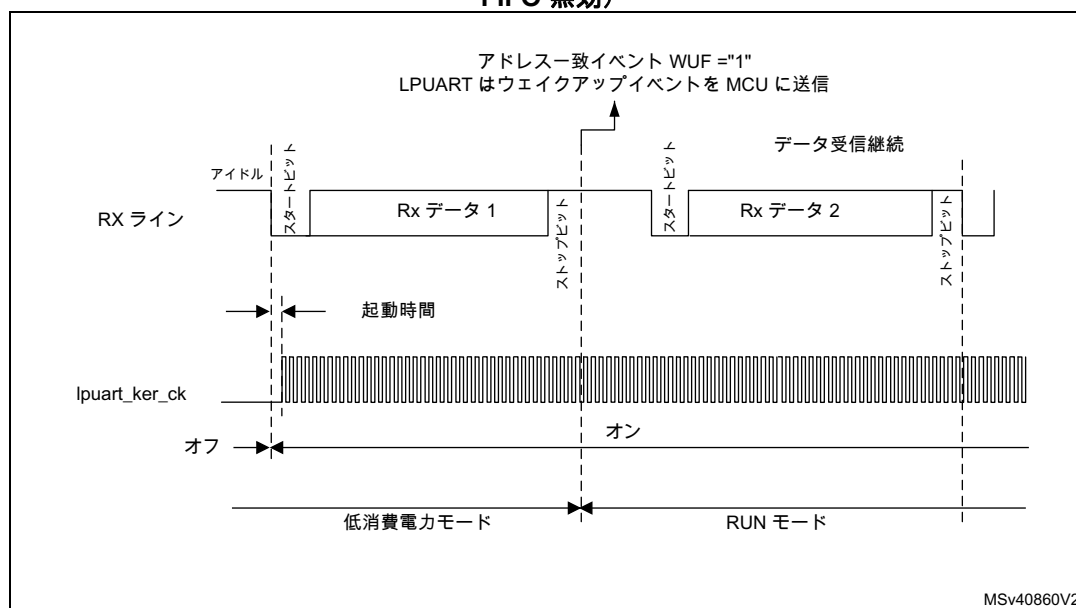
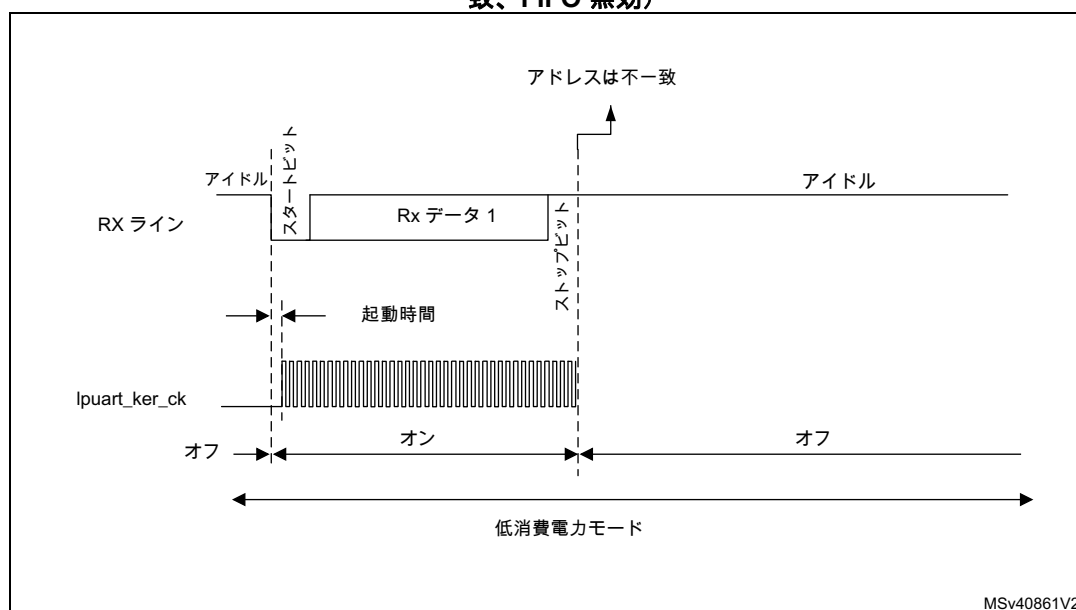


図 714 に、ウェイクアップイベントが確認されていない時の動作を示します。

図 714. 確認されていないウェイクアップイベント (ウェイクアップイベント = アドレス一致、FIFO 無効)



注： 上図は、アドレス一致または受信フレームがウェイクアップイベントとして使用されたとき、有効です。ウェイクアップイベントがスタートビット検出である場合、LPUART はスタートビットの終端にウェイクアップイベントを MCU に送ります。

MCU の低消費電力モードからの正常なウェイクアップを可能にする最大 LPUART ボーレートの決定

マイクロコントローラの低消費電力モードからの正常なウェイクアップを可能にする最大 LPUART ボーレートは、ウェイクアップ時間パラメータ（デバイスのデータシートを参照）と LPUART レシーバの許容誤差（[セクション 51.4.9：クロック偏差に対する LPUART レシーバの許容誤差](#)を参照）に依存します。

例として、OVER8 = 0、M ビット = 01、ONEBIT = 0、BRR [3:0] = 0000 とします。

この条件では、[表 547：LPUART レシーバの許容誤差](#)によると、LPUART レシーバの許容誤差は 3.41% です。

$$DTRA + DQUANT + DREC + DTCL + DWU < \text{LPUART レシーバの許容誤差}$$

$$D_{WUmax} = t_{WULPUART} / (11 \times T_{bit \text{ Min}})$$

$$T_{bit \text{ Min}} = t_{WULPUART} / (11 \times D_{WUmax})$$

ここでは、 $t_{WULPUART}$ が低消費電力モードからのウェイクアップ時間です。

パラメータ DTRA、DQUANT、DREC、および DTCL が 0% であるときに理想的なケースを考えた場合、DWU の最大値は 3.41% です。実際には、最低 lpuart_ker_ck 精度を考慮する必要があります。

たとえば HSI が lpuart_ker_ck として使用され、HSI 精度が 1% の場合、以下を得ることができます。

$t_{WULPUART} = 3 \mu\text{s}$ （値は例です。正確な値については、デバイスのデータシートを参照してください）。

$$D_{WUmax} = 3.41\% - 1\% = 2.41\%$$

$$T_{bit \text{ min}} = 3 \mu\text{s} / (11 \times 2.41\%) = 11.32 \mu\text{s}$$

その結果、低消費電力モードからの正しいウェイクアップを有効にする最大ボーレートは、次のようになります。 $1/11.32 \mu\text{s} = 88.36 \text{ kbaud}$

51.5 低消費電力モードの LPUART

表 549. 低消費電力モードが LPUART に与える影響

モード	説明
SLEEP	影響はありません。LPUART 割込みによって、デバイスは SLEEP モードを終了します。
STOP ⁽¹⁾	LPUART レジスタの内容は保持されます。 STOP モードで使用可能なオシレータによってクロック供給されている場合、LPUART ではマイクロコントローラを STOP モードからウェイクアップさせることができます。
STANDBY	LPUART ペリフェラルはパワーダウンされ、STANDBY モード終了後に再初期化する必要があります。

1. STOP モードからのウェイクアップが指定のペリフェラルインスタンスでサポートされているかどうかについては、[セクション 51.3：LPUART の実装](#)を参照してください。インスタンスが指定の STOP モードで機能しない場合、この STOP モードを開始する前に無効化する必要があります。

51.6 LPUART 割込み

すべての LPUART 割込みリクエストの詳細な説明については、表 550 を参照してください。

表 550. LPUART 割込みリクエスト

割込みベクタ	割込みイベント	イベントフラグ	イネーブル制御ビット	割込みのクリア方法	SLEEP モードの 終了	STOP ⁽¹⁾ モードの 終了	STANDBY モードの 終了
LPUART	送信データレジスタエンプティ	TXE	TXEIE	TDR を書き込む	可能	不可	不可
	送信 FIFO ノットフル	TXFNF	TXFNFIE	TXFIFO フル		不可	
	送信 FIFO エンプティ	TXFE	TXFEIE	TXFRQ に TDR または 1 を書き込む		可能	
	送信 FIFO 閾値到達	TXFT	TXFTIE	TDR を書き込む		可能	
	CTS 割込み	CTSIF	CTSIE	CTSCF に 1 を書き込む		不可	
	送信完了	TC	TCIE	TCCF に TDR または 1 を書き込む		不可	
	受信データレジスタ ノットエンプティ (データの読出し可能)	RXNE	RXNEIE	RXFRQ で RDR または 1 を読み出す	可能	はい	
	受信 FIFO ノットエンプティ	RXFNE	RXFNEIE	RXFIFO エンプティまで RDR を読み出すか、RXFRQ に 1 を書き込む		可能	
	受信 FIFO フル	RXFF ⁽²⁾	RXFFIE	読出しは RDR です。		可能	
	受信 FIFO 閾値到達	RXFT	RXFTIE	読出しは RDR です。		可能	
	オーバーランエラー検出	ORE	RXNEIE/RXFNEIE	ORECF に 1 を書き込む		不可	
	アイドルライン検出	IDLE	IDLEIE	IDLECF に 1 を書き込む		不可	
	パリティエラー	PE	PEIE	PECF に 1 を書き込む		不可	
	マルチバッファ通信で ノイズエラー。	NE	EIE	NFCF に 1 を書き込む		不可	
	マルチバッファ通信で オーバーランエラー。	ORE ⁽³⁾		ORECF に 1 を書き込む		不可	
	マルチバッファ通信で フレーミングエラー。	FE		FECF に 1 を書き込む		不可	
	キャラクター致	CMF	CMIE	CMCF に 1 を書き込む		不可	
	低消費電力モードからの ウェイクアップ	WUF	WUFIE	WUC に 1 を書き込む		可能	

- LPUART は、ペリフェラルインスタンスが STOP モードからのウェイクアップ機能をサポートしている場合にのみデバイスを STOP モードからウェイクアップすることができます。サポートされている STOP モードのリストについては、[セクション 51.3: LPUART の実装](#)を参照してください。
- RXFF フラグは、LPUART が次のように n+1 個のデータを受信した場合にアサートされます (n は RXFIFO のサイズ)。RXFIFO に n 個のデータ、LPUART_RDR に 1 個のデータ。STOP モードでは、LPUART_RDR はクロック供給されません。その結果として、このレジスタは書き込まれず、n 個のデータが受信されて RXFIFO に書き込まれた後に、RXFF 割込みがアサートされます (RXFF フラグはセットされません)。
- OVRDIS = 0 の場合。

51.7 LPUART レジスタ

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#) を参照してください。

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。

51.7.1 LPUART 制御レジスタ 1 (LPUART_CR1)

アドレスオフセット：0x00

リセット値：0x0000 0000

同じレジスタが FIFO モード有効（このセクション）でも、FIFO モード無効（次のセクション）でも使用できます。

FIFO モードが有効な場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXF FIE	TXFEIE	FIFO EN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFN FIE	TCIE	RXFN EIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **RXFFIE**：RXFIFO フル割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0：割込みは禁止されています。

1：LPUART_ISR レジスタで RXFF=1 のときに LPUART 割込みが生成されます。

ビット 30 **TXFEIE**：TXFIFO エンプティ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0：割込みは禁止されています。

1：LPUART_ISR レジスタで TXFE=1 のときに LPUART 割込みが生成されます。

ビット 29 **FIFOEN**：FIFO モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0：FIFO モードは無効です。

1：FIFO モードは有効です。

ビット 28 **M1**：ワード長

このビットはビット 12（M0）と併せて使用して、ワード長を決定する必要があります。ソフトウェアによってセット／クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、LPUART が無効（UE=0）のときのみ書き込むことができます。

注： 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート（0x7F および 0x55 フレーム検出）はサポートされません。

ビット 27:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:21 **DEAT[4:0]** : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。lpuart_ker_ck クロックサイクルで表されます。詳細については、[セクション 50.5.21 : RS232 ハードウェアフロー制御および RS485 ドライバ有効](#)を参照してください。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 20:16 **DEDT[4:0]** : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。lpuart_ker_ck クロックサイクルで表されます。詳細については、[セクション 51.4.14 : RS232 ハードウェアフロー制御および RS485 ドライバ有効](#)を参照してください。DEDT 時間中に LPUART_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **CMIE** : キャラクター一致割込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの CMF ビットがセットされると、LPUART 割込みが生成されます。

ビット 13 **MME** : ミュートモード有効

このビットは、LPUART のミュートモード機能を有効にします。セットされると、LPUART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット/クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

ビット 12 **M0** : ワード長

このビットはビット 28 (M1) と併せて使用して、ワード長を決定します。ソフトウェアによってセット/クリアされます (ビット 28 (M1) の説明を参照)。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 **WAKE** : レシーバのウェイクアップ方式

このビットによって、ミュートモードからの LPUART のウェイクアップ方式が決まります。ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 10 **PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 9 **PS** : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 8 PEIE : PE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで PE=1 のときには LPUART 割込みが生成されます。

ビット 7 TXFNFIE : TXFIFO ノットフル割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで TXFNF=1 のときには LPUART 割込みが生成されます。

ビット 6 TCIE : 転送完了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで TC=1 のときには LPUART 割込みが生成されます。

ビット 5 RXFNEIE : RXFIFO ノットエンプティ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの ORE=1 または RXFNE=1 のときには、LPUART 割込みが生成されます。

ビット 4 IDLEIE : IDLE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで IDLE=1 のときには LPUART 割込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : スマートカードモードの場合を除いて、送信中に TE ビットにローパルスを与える（“0”に続けて“1”を書き込む）と、現在のワードの後にプリアンプル（アイドルライン）が送信されます。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは LPUART_ISR レジスタの TEACK ビットをポーリングできます。

スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 UESM : LPUART 低消費電力モードで有効

このビットがクリアされると、LPUART は MCU を低消費電力モードからウェイクアップできません。

このビットがセットされると、LPUART は MCU を低消費電力モードからウェイクアップできます。

このビットは、ソフトウェアによってセット／クリアされます。

0 : USART は低消費電力モードから MCU をウェイクアップできません。

1 : USART は低消費電力モードから MCU をウェイクアップできます。

注 : 低消費電力モードに入る直前に UESM ビットをセット、低消費電力モードの終了時にクリアすることが推奨されます。

ビット 0 **UE** : LPUART 有効

このビットがクリアされると、LPUART プリスケアラと出力はただちに停止され、現在の操作は破棄されます。LPUART の設定は保たれますが、LPUART_ISR のステータスフラグはすべてリセットされます。このビットは、ソフトウェアによってセット/クリアされます。

0 : LPUART プリスケアラと出力は無効であり、低消費電力モードです。

1 : LPUART は有効です。

注 : ラインにエラーを生成せずに低消費電力モードに入るためには、TE ビットをその前にリセットする必要があります。ソフトウェアは、UE ビットをリセットする前に LPUART_ISR の TC ビットがセットされるのを待つ必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

51.7.2 LPUART 制御レジスタ 1[オルタネート] (LPUART_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

同じレジスタが FIFO モード有効（前のセクション）でも、FIFO モード無効（このセクション）でも使用できます。

FIFO モードが無効の場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
		rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **FIFOEN** : FIFO モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : FIFO モードは無効です。

1 : FIFO モードは有効です。

ビット 28 **M1** : ワード長

このビットはビット 12 (M0) と併せて使用して、ワード長を決定する必要があります。ソフトウェアによってセット/クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10 : スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

注 : 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート (0x7F および 0x55 フレーム検出) はサポートされません。

ビット 27:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:21 **DEAT[4:0]** : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。lpuart_ker_ck クロックサイクルで表されます。詳細については、[セクション 50.5.21: RS232 ハードウェアフロー制御および RS485 ドライバ有効](#)を参照してください。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 20:16 DEDT[4:0] : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。lpuart_ker_ck クロックサイクルで表されます。詳細については、[セクション 51.4.14 : RS232 ハードウェアフロー制御および RS485 ドライバ有効](#)を参照してください。

DEDT 時間中に LPUART_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 CMIE : キャラクター一致割込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの CMF ビットがセットされると、LPUART 割込みが生成されます。

ビット 13 MME : ミュートモード有効

このビットは、LPUART のミュートモード機能を有効にします。セットされると、LPUART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット/クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

ビット 12 M0 : ワード長

このビットはビット 28 (M1) と併せて使用して、ワード長を決定します。ソフトウェアによってセット/クリアされます (ビット 28 (M1) の説明を参照)。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 WAKE : レシーバのウェイクアップ方式

このビットによって、ミュートモードからの LPUART のウェイクアップ方式が決まります。ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 10 PCE : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 9 PS : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 8 PEIE : PE 割込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで PE=1 のときには LPUART 割込みが生成されます。

ビット 7 TXEIE : 送信データレジスタエンプティ

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで TXE=1 のときには LPUART 割込みが生成されます。

ビット 6 TCIE : 転送完了割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで TC=1 のときには LPUART 割込みが生成されます。

ビット 5 RXNEIE : 受信データレジスタノットエンプティ

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの ORE=1 または RXNE=1 のときには、LPUART 割込みが生成されます。

ビット 4 IDLEIE : IDLE 割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで IDLE=1 のときには LPUART 割込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : スマートカードモードの場合を除いて、送信中に TE ビットにローパルスを与える（“0”に続けて“1”を書き込む）と、現在のワードの後にプリアンプル（アイドルライン）が送信されます。アイドルキャラクタを生成するためには、すぐには TE に“1”を書き込まないでください。必要な時間を確保するために、ソフトウェアは LPUART_ISR レジスタの TEACK ビットをポーリングできます。

スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 UESM : LPUART 低消費電力モードで有効

このビットがクリアされると、LPUART は MCU を低消費電力モードからウェイクアップできません。

このビットがセットされると、LPUART は MCU を低消費電力モードからウェイクアップできます。

このビットは、ソフトウェアによってセット／クリアされます。

0 : USART は低消費電力モードから MCU をウェイクアップできません。

1 : USART は低消費電力モードから MCU をウェイクアップできます。

注 : 低消費電力モードに入る直前に UESM ビットをセット、低消費電力モードの終了時にクリアすることが推奨されます。

ビット 0 **UE** : LPUART 有効

このビットがクリアされると、LPUART プリスケアラと出力はただちに停止され、現在の操作は破棄されます。LPUART の設定は保たれますが、LPUART_ISR のステータスフラグはすべてリセットされます。このビットは、ソフトウェアによってセット/クリアされます。

0 : LPUART プリスケアラと出力は無効であり、低消費電力モードです。

1 : LPUART は有効です。

注 : ラインにエラーを生成せずに低消費電力モードに入るためには、TE ビットをその前にリセットする必要があります、ソフトウェアは、UE ビットをリセットする前に LPUART_ISR の TC ビットがセットされるのを待つ必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

51.7.3 LPUART 制御レジスタ 2 (LPUART_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:0]								Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
rw		rw	rw								rw				

ビット 31:24 **ADD[7:0]** : LPUART ノードのアドレス

これらのビットは、ミュートモードの LPUART ノードのアドレス、または低消費電力モードまたは RUN モードで認識されるキャラクタコードを指定します。

- ミュートモードでは、マルチプロセッサ通信で 4 ビット/7 ビットアドレスマーク検出によってミュートモードからウェイクアップするために使用されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。4 ビットアドレスマーク検出では、ADD[3:0] ビットのみが使用されます。
- 低消費電力モードでは、キャラクター一致時に低消費電力モードからのウェイクアップに使用されます。WUS[1:0] が 0b00 にプログラムされている場合 (アドレス一致時に WUF がアクティブ)、低消費電力モードからのウェイクアップは、受信したキャラクタが ADD[6:0] または ADD[3:0] ビットフィールドを通じて (ADDM7 ビットに応じて) プログラムされたキャラクタに対応している場合に実行され、WUF 割込みは WUFIE ビットをセットすることで有効化されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。
- ミュートモードが無効化された RUN モードでは (ModBus プロトコルでのブロックの終了の検出など)、受信したキャラクタ全体 (8 ビット) が ADD[7:0] 値と比較され、一致している場合は CMF フラグがセットされます。CMIE ビットがセットされている場合、割込みが生成されます。

これらのビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **MSBFIRST** : MSB ファースト

このビットは、ソフトウェアによってセット/クリアされます。

0 : スタートビットに続いて、データはデータビット 0 から順に送受信されます。

1 : スタートビットに続いて、データは MSB (ビット 7/8) から順に送受信されます。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 18 **DATAINV** : バイナリデータ反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : データレジスタからの論理データは正／ダイレクトロジックで送受信されます。(1=H、0=L)

1 : データレジスタからの論理データは、負／インバースロジックで送受信されます。(1=L、0=H) パリティビットも反転されます。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 17 **TXINV** : TX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD} = 1$ /アイドル、Gnd = 0/マーク)。

1 : TX ピン信号値は反転されます。($V_{DD} = 0$ /マーク、Gnd = 1/アイドル)。

これにより、TX ラインで外部インバータを使用できます。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 16 **RXINV** : RX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : RX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD} = 1$ /アイドル、Gnd = 0/マーク)。

1 : RX ピン信号値は反転されます。($V_{DD} = 0$ /マーク、Gnd = 1/アイドル)。

これにより、RX ラインで外部インバータを使用できます。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 15 **SWAP** : TX/RX ピンのスワップ

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX/RX ピンは標準ピンアウトでの定義に従って使用されます。

1 : TX および RX ピンの機能はスワップされます。これにより、別の UART へのクロスワイヤ接続の場合に動作できます。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 **STOP[1:0]** : ストップビット

このビットは、ストップビットのプログラミングに使用します。

00 : 1 個のストップビット

01 : 予約済み。

10 : 2 個のストップビット

11 : 予約済み

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

ビット 11:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **ADDM7** : 7 ビットアドレス検出／4 ビットアドレス検出

このビットは、4 ビットアドレス検出と 7 ビットアドレス検出の選択に使用されます。

0 : 4 ビットアドレス検出

1 : 7 ビットアドレス検出 (8 ビットデータモード)

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

注： 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) に対して行われます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

51.7.4 LPUART 制御レジスタ 3 (LPUART_CR3)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			Res.	TXFTIE	WUFIE	WUS1	WUS0	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

ビット 31:29 **TXFTCFG[2:0]** : TXFIFO 閾値設定

000 : TXFIFO はその深さの 1/8 に達します。

001 : TXFIFO はその深さの 1/4 に達します。

110 : TXFIFO はその深さの 1/2 に達します。

011 : TXFIFO はその深さの 3/4 に達します。

100 : TXFIFO はその深さの 7/8 に達します。

101 : TXFIFO は空になります。

残りの組み合わせ : 予約済み。

ビット 28 **RXFTIE** : RXFIFO 閾値割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : 受信 FIFO が RXFTCFG でプログラムされた閾値に達すると、LPUART 割込みが生成されます。

ビット 27:25 **RXFTCFG[2:0]** : 受信 FIFO 閾値設定

000 : 受信 FIFO はその深さの 1/8 に達します。

001 : 受信 FIFO はその深さの 1/4 に達します。

110 : 受信 FIFO はその深さの 1/2 に達します。

011 : 受信 FIFO はその深さの 3/4 に達します。

100 : 受信 FIFO はその深さの 7/8 に達します。

101 : 受信 FIFO はフルになります。

残りの組み合わせ : 予約済み。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TXFTIE** : TXFIFO 閾値割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : TXFIFO が TXFTCFG でプログラムされた閾値に達すると、LPUART 割込みが生成されます。

ビット 22 **WUFIE** : 低消費電力モードからのウェイクアップ割込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタで WUF=1 のときには、USART 割込みが生成されます。

注 : **WUFIE** は、低消費電力モードに入る前にセットする必要があります。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 51.3 : 2286 ページの LPUART の実装](#)を参照してください。

ビット 21:20 **WUS[1:0]** : 低消費電力モードからのウェイクアップ割込みフラグ選択

このビットフィールドは、WUF（低消費電力モードからのウェイクアップフラグ）を有効にするイベントを指定します。

00 : WUF はアドレス一致時に有効になります (ADD[7:0] および ADDM7 による定義に従って)。

01 : 予約済み。

10 : WUF はスタートビット検出時に有効になります。

11 : WUF は RXNE/RXFNE 時に有効になります。

このビットフィールドは、LPUART が無効 (UE = 0) のときのみ書き込むことができます。

注 : **USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 51.3 : 2286 ページの LPUART の実装を参照してください。**

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DEP** : ドライバ有効極性選択

0 : DE 信号はアクティブハイです。

1 : DE 信号はアクティブローです。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 14 **DEM** : ドライバ有効モード

このビットにより、DE 信号によって外部トランシーバ制御を有効にできます。

0 : DE 機能は無効です。

1 : DE 機能は有効です。DE 信号は RTS ピンで出力されます。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 13 **DDRE** : 受信エラー時 DMA 無効

0 : 受信エラーの場合、DMA は無効になりません。対応するエラーフラグはセットされますが、RXNE は 0 に保たれ、オーバーランを防ぎます。結果として、DMA リクエストはアサートされないの、エラーのあるデータは転送されず (DMA リクエストなし)、次の正しい受信データが転送されます。

1 : 受信エラーの後、DMA は無効化されます。対応するエラーフラグと RXNE がセットされます。エラーフラグがクリアされるまで、DMA リクエストはマスクされます。つまり、ソフトウェアは、まず、DMA リクエストを無効にするか (DMAR=0)、RXNE をクリアしてから、エラーフラグをクリアする必要があります。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

注 : **受信エラーは、パリティエラー、フレーミングエラー、またはノイズエラーです。**

ビット 12 **OVRDIS** : オーバーラン無効

このビットは、受信オーバーラン検出を無効にするために使用されます。

0 : オーバーランエラーフラグ、ORE は、受信データが読み出される前に新しいデータを受信したときにセットされます。

1 : オーバーラン機能は無効です。RXNE フラグがまだセットされている間に新しいデータを受信した場合、

ORE フラグはセットされず、新しく受信されたデータが LPUART_RDR レジスタの前の内容に上書きされます。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

注 : **この制御ビットにより、データを読み出さずに通信フローをチェックできます。**

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CTSIE** : CTS 割込み有効

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの CTSIF=1 のときには、割込みが生成されます。

ビット 9 **CTSE** : CTS 有効化

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは CTS 入力にネゲート (0 に関係) されている場合にのみ転送されます。データの送信中に CTS 入力にアサートされた場合、送信は停止前に完了します。CTS がアサートされている間にデータがデータレジスタに書き込まれた場合、CTS がネゲートされるまで送信は延期されます。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 8 **RTSE** : RTS 有効化

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 出力は有効であり、レシーババッファにスペースがあるときにのみ、データがリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。データを受信できるとき、RTS 出力にネゲートされます (0 にプルされます)。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 7 **DMAT** : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードは送信に有効です。

0 : DMA モードは送信に無効です。

ビット 6 **DMAR** : DMA 有効レシーバ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **HDSEL** : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

このビットは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

ビット 2:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EIE** : エラー割込み有効化

エラー割込み有効ビットは、フレーミングエラー、オーバーランエラー、またはノイズフラグ (LPUART_ISR レジスタの FE=1 または ORE=1 または NE=1) の場合に割込み生成を有効にするために必要です。

0 : 割込みは禁止されています。

1 : LPUART_ISR レジスタの FE=1、ORE=1、または NE=1 の場合、割込みが生成されます。

51.7.5 LPUART ボーレートレジスタ (LPUART_BRR)

このレジスタは、LPUART が無効 (UE=0) のときのみ書き込むことができます。自動ボーレート検出モードでハードウェアによって自動的に更新されます。

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:16]			
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **BRR[19:0]** : LPUART ボーレート分周 (LPUARTDIV)

注 : LPUART_BRR レジスタに 0x300 未満の値を書き込むことは禁じられています。

ただし、LPUART_BRR は $\geq 0x300$ でなければならず、LPUART_BRR は 20 ビットであり、高い fck 値を使用して高いボーレートを生成するときには、注意が必要です。fck は [3 x ボーレート ~ 4096 x ボーレート] の範囲内でなければなりません。

51.7.6 LPUART リクエストレジスタ (LPUART_RQR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
											w	w	w	w	

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **TXFRQ** : 送信データ一掃リクエスト

このビットは FIFO モードが有効なときに使用されます。FIFO 全体を一掃するために TXFRQ ビットがセットされます。これによって TXFE フラグ (TXFIFO エンプティ、LPUART_ISR レジスタのビット 23) がセットされます。

注 : FIFO モードでは、一掃リクエスト中にデータがデータレジスタに書き込まれないようにするために、TxFIFO が空になるまで TXFNF フラグはリセットされます。

ビット 3 **RXFRQ** : 受信データ一掃リクエスト

このビットに 1 を書き込むと、RXNE フラグがクリアされます。

これにより、受信したデータを読み出さずに破棄して、オーバーラン条件を避けることができます。

ビット 2 **MMRQ** : ミュートモードリクエスト

このビットに 1 を書き込むと、LPUART はミュートモードになり、RWU フラグはリセットされます。

ビット 1 **SBKRQ** : ブレーク送信リクエスト

このビットに 1 を書き込むと、SBKF フラグがセットされ、送信マシンが使用可能になるとすぐに、ラインで BREAK を送信するリクエストが発行されます。

注: アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレークキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

ビット 0 予約済みであり、リセット値に保持する必要があります。

51.7.7 LPUART 割込みおよびステータスレジスタ (LPUART_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0080 00C0

同じレジスタが FIFO モード有効 (このセクション) でも、FIFO モード無効 (次のセクション) でも使用できます。

FIFO モードが有効な場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
				r	r		r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **TXFT** : TXFIFO 閾値フラグ

このビットは、TXFIFO が LPUART_CR3 レジスタの TXFTCFG でプログラムされた閾値に達したとき、すなわち、TXFIFO に TXFTCFG の空き場所ができたときに、ハードウェアによってセットされます。LPUART_CR3 レジスタの TXFTIE ビット =1 (ビット 31) の場合、割込みが生成されます。

0 : TXFIFO はプログラムされた閾値に達していません。

1 : TXFIFO はプログラムされた閾値に達しました。

ビット 26 **RXFT** : RXFIFO 閾値フラグ

このビットは、RXFIFO が LPUART_CR3 レジスタの RXFTCFG でプログラムされた閾値に達したとき、すなわち、受信 FIFO に RXFTCFG 分のデータが入ったときに、ハードウェアによってセットされます。LPUART_CR3 レジスタの RXFTIE ビット =1 (ビット 27) の場合、割込みが生成されます。

0 : 受信 FIFO はプログラムされた閾値に達していません。

1 : 受信 FIFO はプログラムされた閾値に達しました。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **RXFF** : RXFIFO フル

このビットは、受信したデータの数 RXFIFO サイズ + 1 と一致したときに (RXFIFO がフルで、LPUART_RDR レジスタに 1 個のデータ)、ハードウェアによってセットされます。

LPUART_CR1 レジスタの RXFFIE ビット =1 の場合、割込みが生成されます。

0 : RXFIFO はフルではありません。

1 : RXFIFO はフルです。

ビット 23 **TXFE** : TXFIFO エンプティ

このビットは、TXFIFO が空のとき、ハードウェアによってセットされます。TXFIFO に少なくとも 1 データが入ったとき、このフラグはクリアされます。TXFE フラグは、LPUART_RQR レジスタのビット TXFRQ (ビット 4) に 1 を書き込むことによってセットすることもできます。

LPUART_CR1 レジスタの TXFEIE ビット =1 (ビット 30) の場合、割込みが生成されます。

0 : TXFIFO は空ではありません。

1 : TXFIFO は空です。

ビット 22 **REACK** : 受信有効確認応答フラグ

このビットは、受信有効値が LPUART によって考慮されるときに、ハードウェアによってセット／リセットされます。

これを使用して、低消費電力モードに入る前に、LPUART が受信できる状態であることを確認できます。

注： LPUART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。

ビット 21 **TEACK** : 送信有効確認応答フラグ

このビットは、送信有効値が LPUART によって考慮されるときに、ハードウェアによってセット／リセットされます。

LPUART_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 **WUF** : 低消費電力モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。LPUART_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR3 レジスタの WUFIE=1 の場合、割込みが生成されます。

注： UESM がクリアされると、WUF フラグもクリアされます。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 51.3 : 2286 ページの LPUART の実装](#)を参照してください。

ビット 19 **RWU** : レシーバのミュートモードからのウェイクアップ

このビットは、LPUART がミュートモードかどうかを示します。ウェイクアップ／ミュートシーケンスが認識されたときに、ハードウェアによってクリア／セットされます。ミュートモード制御シーケンス (アドレスまたは IDLE) は、LPUART_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは LPUART_RQR レジスタの MMRQ ビットに "1" を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注： LPUART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。

ビット 18 **SBKF** : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。LPUART_CR3 レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタが送信されません。

1 : ブレークキャラクタが送信されます。

ビット 17 **CMF** : キャラクター一致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタの CMIE=1 の場合、割込みが生成されます。

0 : キャラクター一致は検出されていません。

1 : キャラクター一致が検出されました。

ビット 16 BUSY : ビジーフラグ

このビットは、ハードウェアによってセット／リセットされます。RX ラインで通信中（スタートビットの検出時）はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : LPUART はアイドルです（受信なし）。

1 : 受信中です。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 CTS : CTS フラグ

このビットは、ハードウェアによってセット／リセットされます。CTS 入力ピンのステータスの反転コピーです。

0 : CTS ラインはセットされました。

1 : CTS ラインはリセットされました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 9 CTSIF : CTS 割込みフラグ

このビットは、CTSE ビットがセットされていた場合、CTS 入力がトグルしたときにハードウェアによってセットされます。LPUART_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR3 レジスタの CTSIE=1 の場合、割込みが生成されます。

0 : CTS ステータスラインでの変更はありません。

1 : CTS ステータスラインでの変更がありました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 TXFNF : TXFIFO は空ではありません。

TXFIFO がフルではない、つまり LPUART_TDR にデータを書き込めるとき、TXFNF はハードウェアによってセットされます。LPUART_TDR への書き込みごとにデータが TXFIFO に格納されます。このフラグは TXFIFO がフルになるまでセットされたままになります。TXFIFO がフルになると、このフラグはクリアされ、データを LPUART_TDR に書き込むことができないことを示します。

一掃リクエスト中、TXFIFO が空になるまで、TXFNF はリセットに維持されます。一掃リクエストを (TXFRQ ビットをセットすることによって) 送信した後、TXFIFO に書き込む前に TXFNF フラグをチェックする必要があります (TXFNF および TXFE は同時にセットされます)。

LPUART_CR1 レジスタの TXFNFIE ビット=1 の場合、割込みが生成されます。

0 : データレジスタはフル／送信 FIFO はフルです。

1 : データレジスタ／送信 FIFO はフルではありません。

注： このビットは、シングルバッファ送信時に使用されます。

ビット 6 TC : 送信完了

このビットは、LPUART_TDR に書き込まれた最後のデータがシフトレジスタから送信されたことを示します。TC フラグの動作は次のようになります。

- TDN =0 の場合、データを含むフレームの送信が完了し、TXFE がセットされたとき、TC フラグがセットされます。
- TDN が TXFIFO のデータの数と等しい場合、TXFIFO が空で TDN に達すると、TC フラグがセットされます。
- TDN が TXFIFO のデータの数より多い場合、送信するプログラム済みデータの数に達するように TXFIFO が再び満たされるまで、TC はクリアされたままとなります。
- TDN が TXFIFO のデータの数より少ない場合、TXFIFO が空でなくても、TDN に達したときに TC がセットされます。

LPUART_CR1 レジスタの TCIE=1 の場合、割込みが生成されます。

TC ビットは、LPUART_ICR レジスタの TCCF に "1" を書き込むことによって、または LPUART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

ビット 5 **RXFNE** : RXFIFO は空ではありません。

RXFNE ビットは、RXFIFO が空でなく、したがってデータが LPUART_RDR レジスタから読み出せるときに、ハードウェアによってセットされます。LPUART_RDR からの読出しのたびに、RXFIFO の 1 つの場所が解放されます。RXFIFO が空になると、クリアされます。

RXFNE フラグは、LPUART_RQR レジスタの RXFRQ に 1 を書き込むことによってもクリアすることもできます。

LPUART_CR1 レジスタの RXFNEIE=1 の場合、割込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 **IDLE** : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。LPUART_CR1 レジスタの IDLEIE=1 の場合、割込みが生成されます。LPUART_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注 : RXFNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。

ミュートモードが有効な場合 (MME=1)、LPUART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。

ビット 3 **ORE** : オーバーランエラー

このビットは、RXNE="1" (FIFO モードが有効な場合は RXFF="1") のときに、

現在シフトレジスタに受信中のデータを LPUART_RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタで RXFNEIE=1、または LPUART_CR3 レジスタで EIE=1 の場合、割込みが生成されます。

1 : オーバーランエラーが検出されました。

注 : このビットがセットされると、LPUART_RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割込みが生成されます。

LPUART_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

ビット 2 **NE** : スタートビットノイズ検出フラグ

このビットは、受信フレームのスタートビットでノイズが検出されるとハードウェアによってセットされます。LPUART_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

注 : このビットは、割込みを生成する RXFNE ビットと同時に出現するため、割込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NE フラグがセットされると、割込みが生成されます。

このエラーは LPUART_RDR 内のキャラクタに関連します。

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでデータを送信しているとき、送信時、成功せずに（カードがデータフレームを NACK）最大送信試行回数に達すると、このビットがセットされます。

LPUART_CR3 レジスタの EIE=1 の場合、割込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注： このエラーはLPUART_RDR 内のキャラクタに関連します。

ビット 0 PE : パリティエラー

このビットは、受信モードでパリティエラーが発生したときに、ハードウェアによってセットされます。LPUART_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタの PEIE=1 である場合、割込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

注： このエラーはLPUART_RDR 内のキャラクタに関連します。

51.7.8 LPUART 割込みとステータスレジスタ [オルタネート] (LPUART_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 00C0

同じレジスタが FIFO モード有効（前のセクション）でも、FIFO モード無効（このセクション）でも使用できます。

FIFO モードが無効の場合

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 REACK : 受信有効確認応答フラグ

このビットは、受信有効値が LPUART によって考慮されるときに、ハードウェアによってセット／リセットされます。

これを使用して、低消費電力モードに入る前に、LPUART が受信できる状態であることを確認できます。

注： LPUART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。

ビット 21 TEACK : 送信有効確認応答フラグ

このビットは、送信有効値が LPUART によって考慮されるときに、ハードウェアによってセット／リセットされます。

LPUART_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 WUF : 低消費電力モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。LPUART_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR3 レジスタの WUFIE=1 の場合、割込みが生成されます。

注： UESM がクリアされると、WUF フラグもクリアされます。

USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。セクション 51.3 : 2286 ページの LPUART の実装を参照してください。

ビット 19 RWU : レシーバのミュートモードからのウェイクアップ

このビットは、LPUART がミュートモードかどうかを示します。ウェイクアップ/ミュートシーケンスが認識されたときに、ハードウェアによってクリア/セットされます。ミュートモード制御シーケンス (アドレスまたは IDLE) は、LPUART_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは LPUART_RQR レジスタの MMRQ ビットに "1" を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注： LPUART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。

ビット 18 SBKF : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。LPUART_CR3 レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタが送信されません。

1 : ブレークキャラクタが送信されます。

ビット 17 CMF : キャラクター一致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタの CMIE=1 の場合、割込みが生成されます。

0 : キャラクター一致は検出されていません。

1 : キャラクター一致が検出されました。

ビット 16 BUSY : ビジーフラグ

このビットは、ハードウェアによってセット/リセットされます。RX ラインで通信中 (スタートビットの検出時) はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : LPUART はアイドルです (受信なし)。

1 : 受信中です。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 CTS : CTS フラグ

このビットは、ハードウェアによってセット/リセットされます。CTS 入力ピンのステータスの反転コピーです。

0 : CTS ラインはセットされました。

1 : CTS ラインはリセットされました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 9 CTSIF : CTS 割込みフラグ

このビットは、CTSE ビットがセットされていた場合、CTS 入力がトグルしたときにハードウェアによってセットされます。LPUART_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR3 レジスタの CTSIE=1 の場合、割込みが生成されます。

0 : CTS ステータスラインでの変更はありません。

1 : CTS ステータスラインでの変更がありました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、リセット値に保持されます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 TXE : 送信データレジスタエンプティ

TXE は、LPUART_TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。LPUART_TDR レジスタへの書き込みによってクリアされます。

LPUART_CR1 レジスタの TXEIE ビット =1 の場合、割込みが生成されます。

0 : データレジスタはフル/送信 FIFO はフルです。

1 : データレジスタ/送信 FIFO はフルではありません。

注： このビットは、シングルバッファ送信時に使用されます。

ビット 6 TC : 送信完了

このビットは、USART_TDR に書き込まれた最後のデータがシフトレジスタから送信されたことを示します。データを含むフレームの送信が完了し、TXE がセットされたとき、TC フラグがセットされます。

LPUART_CR1 レジスタの TCIE=1 の場合、割込みが生成されます。

TC ビットは、USART_ICR レジスタの TCCF に "1" を書き込むことによって、または USART_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

ビット 5 RXNE : 読出しデータレジスタノットエンプティ

RXNE ビットは、LPUART_RDR シフトレジスタの内容が LPUART_RDR レジスタに転送されると、ハードウェアによってセットされます。LPUART_RDR レジスタの読出しによってクリアされます。RXNE フラグは、LPUART_RQR レジスタの RXFRQ に 1 を書き込むことによってもクリアすることもできます。

RXFNE フラグは、LPUART_RQR レジスタの RXFRQ に 1 を書き込むことによってもクリアすることもできます。

LPUART_CR1 レジスタの RXNEIE=1 の場合、割込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 IDLE : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。

LPUART_CR1 レジスタの IDLEIE=1 の場合、割込みが生成されます。LPUART_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注： RXNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。

ミュートモードが有効な場合 (MME=1)、LPUART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE="1" (FIFO モードが有効な場合は RXFF="1") のときに、シフトレジスタで現在受信中のデータを LPUART_RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタで RXNEIE=1、または LPUART_CR3 レジスタで EIE=1 の場合、割込みが生成されます。

1 : オーバーランエラーが検出されました。

注： このビットがセットされると、LPUART_RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割込みが生成されます。

LPUART_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

ビット 2 NE : スタートビットノイズ検出フラグ

このビットは、受信フレームのスタートビットでノイズが検出されるとハードウェアによってセットされます。LPUART_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

注： このビットは、割込みを生成する RXNE/RXFNE ビットと同時に出現するため、割込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NE フラグがセットされると、割込みが生成されます。

FIFO モードでは、このエラーは LPUART_RDR 内のキャラクタに関連します。

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。LPUART_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでデータを送信しているとき、送信時、成功せずに (カードがデータフレームを NACK) 最大送信試行回数に達すると、このビットがセットされます。

LPUART_CR3 レジスタの EIE=1 の場合、割込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注： FIFO モードでは、このエラーは LPUART_RDR 内のキャラクタに関連します。

ビット 0 PE : パリティエラー

このビットは、受信モードでパリティエラーが発生したときに、ハードウェアによってセットされます。LPUART_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

LPUART_CR1 レジスタの PEIE=1 である場合、割込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

注： FIFO モードでは、このエラーは LPUART_RDR 内のキャラクタに関連します。

51.7.9 LPUART 割込みフラグクリアレジスタ (LPUART_ICR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						w			w		w	w	w	w	w

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **WUCF** : 低消費電力モードからのウェイクアップクリアフラグ

このビットに 1 を書き込むと、USART_ISR レジスタの WUF フラグがクリアされます。

注 : USART が STOP モードからのウェイクアップをサポートしない場合、このビットは予約済みであり、リセット値に保持する必要があります。[セクション 51.3 : 2286 ページのLPUART の実装](#)を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CMCF** : キャラクター致フラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの CMF フラグがクリアされます。

ビット 16:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTSCF** : CTS フラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの CTSIF フラグがクリアされます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TCCF** : 送信完了フラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの TC フラグがクリアされます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **IDLECF** : アイドルライン検出フラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの IDLE フラグがクリアされます。

ビット 3 **ORECF** : オーバーランエラーフラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの ORE フラグがクリアされます。

ビット 2 **NECF** : ノイズ検出フラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの NE フラグがクリアされます。

ビット 1 **FECF** : フレーミングエラーフラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの FE フラグがクリアされます。

ビット 0 **PECF** : パリティエラーフラグクリア

このビットに 1 を書き込むと、LPUART_ISR レジスタの PE フラグがクリアされます。

51.7.10 LPUART 受信データレジスタ (LPUART_RDR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **RDR[8:0]** : 受信データ値

受信データキャラクタを含みます。

RDR レジスタは、入力シフトレジスタと内部バスとの間にパラレルインタフェースを提供します (図 701 を参照)。

パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

51.7.11 LPUART 送信データレジスタ (LPUART_TDR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **TDR[8:0]** : 送信データ値

送信されるデータキャラクタを含みます。

TDR レジスタは、内部バスと出力シフトレジスタとの間にパラレルインタフェースを提供します (図 701 を参照)。

パリティを有効にして (LPUART_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。

注 : このレジスタは、TXE/TXFNF=1 のときのみ書き込む必要があります。

51.7.12 LPUART プリスケーラレジスタ (LPUART_PRESC)

このレジスタは、LPUART が無効 (UE=0) のときのみ書き込むことができます。

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PRESCALER[3:0]** : クロックプリスケーラ

LPUART 入力クロックは、以下のようにプリスケーラによって分周できます。

0000 : 入力クロックは分周されません。

0001 : 入力クロックが 2 分周されます。

0010 : 入力クロックが 4 分周されます。

0011 : 入力クロックが 6 分周されます。

0100 : 入力クロックが 8 分周されます。

0101 : 入力クロックが 10 分周されます。

0110 : 入力クロックが 12 分周されます。

0111 : 入力クロックが 16 分周されます。

1000 : 入力クロックが 32 分周されます。

1001 : 入力クロックが 64 分周されます。

1010 : 入力クロックが 128 分周されます。

1011 : 入力クロックが 256 分周されます。

残りの組み合わせ : 予約済み。

注 : **PRESCALER** が許容される値と異なる値でプログラムされたとき、プログラムされるプリスケーラ値は 1011 になります。すなわち、入力クロックは 256 分周されます。

51.7.13 LPUART レジスタマップ

表 551. LPUART レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	LPUART_CR1 FIFO モード 有効	RXFIE	TXFIE	FIFOEN	M1	Res.	Res.			DEAT[4:0]					DEDT[4:0]			Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE	
	リセット値	0	0	0	0			0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00	LPUART_CR1 FIFO モード 無効	Res.	Res.	FIFOEN	M1	Res.	Res.			DEAT[4:0]				DEDT[4:0]				Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE	
	リセット値			0	0			0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	LPUART_CR2	ADD[7:0]								Res.	Res.	Res.	Res.	MSBFIRST	DATINV	TXINV	RXINV	SWAP	Res.	STOP [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.	
	リセット値	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	
0x08	LPUART_CR3	TXFTCFG[2:0]			RXFTIE			RXFTCFG[2:0]			Res.	TXFTIE	WUFIE	WUS 1	WUS 0	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	Res.	Res.	EIE	
	リセット値	0	0	0	0	0	0	0		0	0	0	0	0				0	0	0	0	0		0	0	0	0	0			0		0	
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																				
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10 ~0x14	予約済み																																	
0x18	LPUART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																													0	0	0	0	
0x1C	LPUART_ISR FIFO モード 有効	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFF	TXFF	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	Res.	Res.	Res.	TXFNF	TC	RXFNE	Res.	TXFRQ
	リセット値								0	1	0	0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x1C	LPUART_ISR FIFO モード 無効	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	Res.	Res.	Res.	TXE	TC	RXNE	Res.	TXFRQ
	リセット値										0	0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値												0			0																		
0x24	LPUART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	リセット値																								0	0	0	0	0	0	0	0	0	0
0x28	LPUART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]									
	リセット値																								0	0	0	0	0	0	0	0	0	0

表 551. LPUART レジスタのマップとリセット値（続き）

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	LPUART_ PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3 :0]			
	リセット値																													0	0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

52 シリアルペリフェラルインタフェース (SPI)

52.1 概要

シリアルペリフェラルインタフェース (SPI) を使用して、特定の同期プロトコルを利用して外部デバイスと通信することができます。SPI プロトコルは、外部デバイスとの半二重、全二重、および単方向の同期シリアル通信をサポートしています。このインタフェースはマスタまたはスレーブとして設定することが可能で、マルチスレーブまたはマルチマスタの設定で動作することができます。マスタとして設定されたデバイスは、スレーブデバイスに通信クロック (SCK) を供給します。スレーブ選択 (SS) およびレディ (RDY) 信号は、具体的なスレーブとの通信をセットアップし、データフローが適切に処理されていることを確認するためだけにオプションで適用できます。デフォルトでモトローラのデータフォーマットが使用されますが、その他の特定のモードもサポートされています。

52.2 SPI の主な機能

- 3 本のラインでの全二重同期転送
- 2 本のラインでの半二重同期転送 (双方向データライン有り)
- 2 本のラインでの単方向同期転送 (単方向データライン有り)
- 4 から 32 ビットまでのデータサイズ選択または 8 ビットの倍数に固定
- マルチマスタまたはマルチスレーブモード機能
- デュアルクロックドメイン、ペリフェラルカーネルクロックは APB バスクロックから独立
- ボーレートプリスケアラ、カーネル周波数/2 まで、またはマスタモードで RCC からバイパス
- 構成と設定の保護
- マスタとスレーブの両方に対するハードウェア/ソフトウェアによる SS 管理
- データ間、および SS とデータフロー間の最小遅延を調節可能
- 設定可能な SS 信号の極性とタイミング、MISO と MOSI のスワップ機能
- クロックの極性と位相をプログラム可能
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- SS および CRC を制御するトランザクション内のデータ数をプログラム可能
- 専用の送受信フラグ (割込み機能付き)
- SPI のモトローラおよび TI フォーマットをサポート
- ハードウェアの CRC 機能では、以下によってトランザクションの最後の通信の整合性を確認できます。
 - 送信モードで CRC 値の追加
 - 受信モードで CRC エラーの自動チェック
- 動作モードに応じて、データオーバーラン、CRC エラー、データアンダーラン、モードフォールトおよびフレームエラーの場合は、割込み機能のあるエラー検出
- 8 ビットの内蔵 Rx および Tx FIFO の倍数 (FIFO サイズはインスタンスによって異なる)
- 設定可能な FIFO 閾値 (データパッキング)
- システム DMA コントローラでデータストリームを処理する機能
- スレーブのアンダーラン状態での設定可能な動作 (カスケードサーキュラバッファ)
- データフローを処理できるようになったことをスレーブデバイスに知らせる、オプションのステータスピン RDY

52.3 SPI の実装

表 552 に、SPI の実装を示します。インスタンスには、すべてまたは一部の機能が適用されます。

表 552. SPI の機能

SPI の機能	SPI2S1、SPI2S2、SPI2S3 (全機能がセットされた インスタンス)	SPI4、SPI5、SPI6 (全機能がセットされた インスタンス)
データおよび CRC サイズ	4 ~ 32 ビットに設定可能	4 ~ 16 ビットに設定可能
CRC 計算	CRC 多項式の長さを 5 ~ 33 ビットに設定可能	CRC 多項式の長さを 5 ~ 17 ビットに設定可能
FIFO のサイズ	16x8ビット	8x8ビット
データ制御の数 (TSIZE)	最大 65536	最大 65536
I2S の機能	あり	不可
ウェイクアップ機能により STOP モードで自律	なし	いいえ
ウェイクアップ機能により LP-STOP および STANDBY モードで自律	なし	いいえ

注： STOP および STANDBY モードを終了するためのインスタンスの機能について、詳しくは表 556 : SPI のウェイクアップおよび割込みリクエストを参照してください。

52.4 SPI の機能説明

52.4.1 SPI ブロック図

SPI により、MCU と外部デバイス間の同期シリアル通信が可能です。アプリケーションソフトウェアは、ステータスフラグをポーリングするか、または専用の SPI 割り込みを使用することで、通信を管理することができます。SPI の主要要素およびそれらの相互作用を図 715 に示します。

図 715. SPI/I2S ブロック図

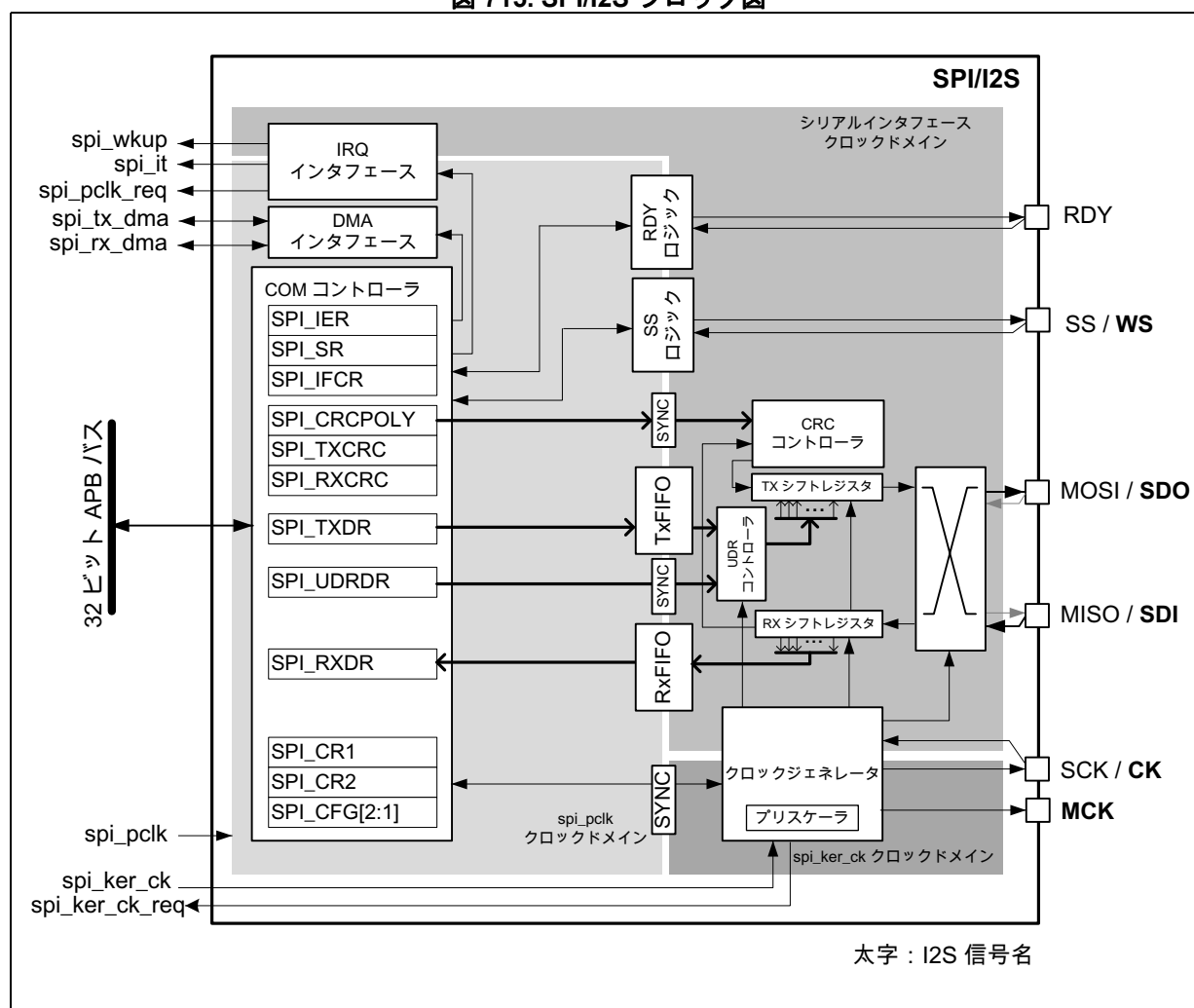


図 715 の簡易体系図には、以下の完全に独立した 3 つのクロックドメインが示されています。

- **spi_pclk** クロックドメイン
- **spi_ker_ck** カーネルクロックドメイン
- シリアルインタフェースクロックドメイン

これらのドメイン間のすべての制御およびステータス信号は厳密に同期されています。これらのクロック信号間の周波数の比に関しては特定の制約はありません。ユーザは、データのアンダーランやオーバーランイベントを避けるために、データフロー速度と互換性のある周波数比を考慮することが必要です。

spi_pclk クロック信号はペリフェラルバスインタフェースにクロックを供給します。SPI レジスタへアクセスが必要なおき、アクティブになっていなければなりません。

スレーブモードで動作している SPI は、外部マスタ SPI デバイスによって提供される外部 SCK 信号から導き出されるシリアルインタフェースクロックを使用してデータフローを処理します。このため、SPI スレーブは、**spi_pclk** および **spi_ker_ck** 信号が非アクティブになっている場合でもデータの送受信ができます。結果的に、シリアルインタフェースクロックドメイン内で動作する特定のスレーブのロジックでは、追加のトラフィックを正しく設定する必要があります（たとえば、アンダーランやオーバーランを評価する場合。詳細については、[セクション 52.5.2](#) を参照）。バスがアイドルになると、これを行うことはできません。一部のケースでは、スレーブはクロックジェネレータの動作も必要とします（[セクション 52.5.1](#) を参照）。

SPI がマスタとして機能する場合は、プリスケラによって分周可能またはオプションでバイパス可能な、クロックジェネレータによってシリアルインタフェースクロックに供給するための通信中に有効な RCC からの **spi_ker_ck** カーネルクロックが必要です。信号はその後、SCK ピンを介してスレーブに供給され、また内部的にマスタのシリアル・インタフェース・ドメインにも供給されます。

52.4.2 SPI ピンおよび内部信号

最大 5 本の I/O ピンが外部デバイスとの SPI 通信専用で使用されます。

- **MISO** : マスタイン/スレーブアウトデータ。一般に、このピンは、スレーブモードではデータの送信に、マスタモードではデータの受信に使用されます。
- **MOSI** : マスタアウト/スレーブインデータ。一般に、このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使用されます。
- **SCK** : このピンは、SPI マスタではシリアルクロックの出力に、SPI スレーブでは入力に使用されます。
- **SS** : スレーブ選択用のピンです。このピンは、SPI および SS の設定に応じて、以下のいずれかに使用できます。
 - 個々の通信用スレーブデバイスを選択する
 - データフレームを同期させる
 - 複数のマスタ間での競合を検出する詳細については、[セクション 52.4.7](#) を参照してください。
- **RDY** : スレーブ FIFO の占有率を示し、データフローを損なうことなくトランザクションを続けるための、オプションのステータスピン。マスタは、現在の通信の一時的な中断を制御するためにこれをチェックできます。

これらのピンはすべて（RDY は除く）I2S モードで共有されます。このモードには、追加の I2S 固有の MCK 信号があります。I2S 信号の詳細については、[セクション 52.9.2](#) を参照してください。

SPI バスを使用することで、1 つのマスタデバイスと 1 つ以上のスレーブデバイスとの間で通信することができます。バスは 2 本以上の線から成り、1 本はクロック信号用、その他はデータの同期転送用です。その他の信号はオプションで、SPI ノード間でのデータ交換とそれらの通信制御管理に応じて、追加することができます。

SPI 入出力ピンおよび内部信号のリストについては、表 553 と表 554 を参照してください。

表 553. SPI/I2S の入出力ピン⁽¹⁾

ピン名	I/O タイプ	説明
MISO/SDI ⁽²⁾	入力／出力	マスタデータ入力/スレーブデータ出力
MOSI/SDO ⁽²⁾	入力／出力	マスタデータ出力/スレーブデータ入力
SCK/CK	入力／出力	マスタクロック出力/スレーブクロック入力
SS/WS	入力／出力	マスタ出力/スレーブ選択入力
RDY	入力／出力	SPI マスタの入力/スレーブ FIFO のステータス占有率の出力
MCK	出力	I2S マスタ周波数出力

1. 詳細については、セクション 52.9.2: SPI 機能とのピンの共用を参照してください。
2. MOSI/SDO および MISO/SDI ピンの機能はスワップできます。これらの方向は、SPI 双方向半二重モードで異なる場合があります。

SPI 入出力信号の説明

表 554. SPI 内部入力／出力信号

信号名	信号タイプ	説明
spi_pclk	入力	SPI クロック信号はペリフェラルバスインタフェースにクロックを供給
spi_ker_ck	入力	SPI カーネルクロック
spi_ker_ck_req	出力	SPI カーネルクロックリクエスト
spi_pclk_req	出力	SPI クロックリクエスト
spi_wkup	出力	SPI はウェイクアップ割込みを提供
spi_it	出力	SPI グローバル割込み
spi_tx_dma	入力／出力	SPI 送信 DMA リクエスト
spi_rx_dma	入力／出力	SPI 受信 DMA リクエスト

SPI の相互作用の説明

52.4.3 SPI 通信の一般的側面

SPI を使用することで、MCU は対象となるデバイスやアプリケーション要件に応じたさまざまな設定で通信ができます。これらの設定には、2 または 3 本の線（ソフトウェア SS 管理あり）、あるいは 3 または 4 本の線（ハードウェア SS 管理あり）が使われます。通信は常にマスタによって開始され制御されます。マスタは SCK ラインにクロック信号を供給し、SS ライン（ハードウェアによって管理される場合）によって通信相手のスレーブを選択したり、同期させたりします。マスタとスレーブ間のデータは MOSI および/または MISO ライン上を流れます。

52.4.4 マスタとスレーブの 1 対 1 の通信

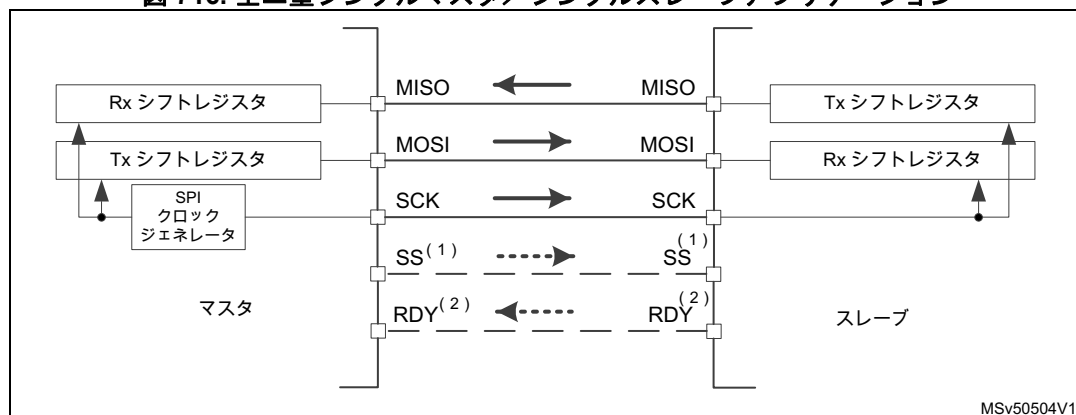
通信の流れは、全二重（3 線）モード、半二重（2 線）モード、単方向（2 線）モードの 3 つのモードのうちの 1 つを使用します。SS 信号は、シングルマスタ-スレーブ設定ではオプションであり、2 つの通信ノード間で接続されないことがしばしばあります。しかしながら、SS 信号は、この設定でデータフローを同期させるのに役立ち、特定の SPI モード（たとえば、TI モード）ではデフォルトで使用されます。

次のオプションの RDY 信号は、スレーブ側でトランザクションされたすべてのデータの正しい管理を保証するのに役立ちます。

全二重通信

SPI は、デフォルトで全二重通信に設定されます（SPI_CFG2 レジスタのビット COMM[1:0] = 00）。この設定では、マスタおよびスレーブのシフトレジスタは、MOSI ピンと MISO ピンの間に 2 本の単方向ラインを介してリンクされます。SPI 通信の間、データはマスタから供給される SCK クロックのエッジに同期してシフトされます。マスタは、送信すべきデータを MOSI ライン経由でスレーブに送信し、同時に MISO ライン経由でスレーブからデータを受信します。データフレーム転送が完了した（すべてのビットがシフトされた）時点で、マスタとスレーブの間で情報が交換されます。

図 716. 全二重シングルマスタ/シングルスレーブアプリケーション

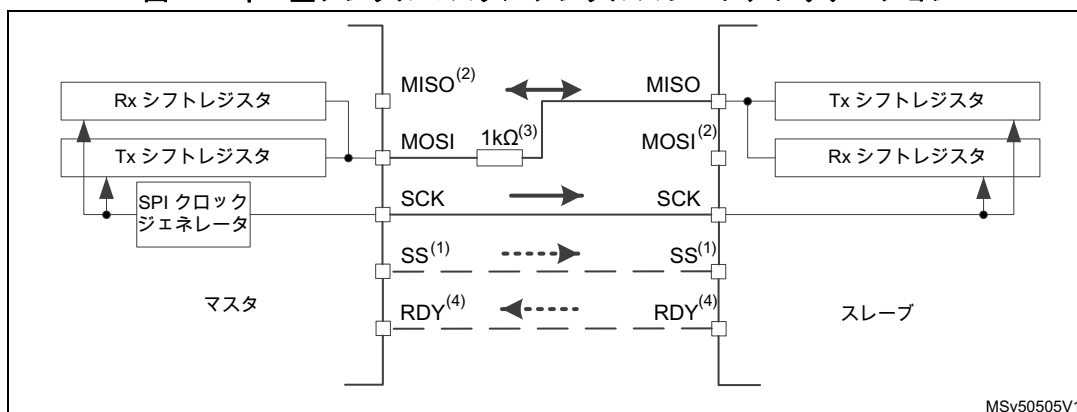


1. SPI インタフェースを機能させるための SS ピンの相互接続の適用は必須ではありません（詳細については、[セクション 52.4.7](#) を参照）。
2. スレーブによって供給された RDY 信号は、オプションでマスタによって読み出すことができます。

半二重通信

SPI_CFG2 レジスタで COMM[1:0] = 11 にセットすることで、SPI は半二重モードで通信できます。この設定では、1 本の交差接続ラインを使用して、マスタとスレーブのシフトレジスタを互いにリンクさせます。この通信中に、データは SCK クロックのエッジに同期して、シフトレジスタ間でシフトされます。シフトの方向は、マスタとスレーブの両方が SPI_CR1 レジスタの HDDIR ビットを使用して相互に選択された転送方向となります。通信の方向を変えるときは SPI を無効にする必要がありますので、注意してください。この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、他のアプリケーションで使用でき、GPIO として機能します。

図 717. 半二重シングルマスタ/シングルスレーブアプリケーション



1. SPI インタフェースを機能させるための SS ピンの相互接続の適用は必須ではありません（詳細については、[セクション 52.4.7](#) を参照）。
2. この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、GPIO として使用できます。
3. 双方向モードで動作している 2 つのノード間で通信方向が同時に変更されなかった場合は、重大な事態が発生する可能性があります。前のトランスミッタが逆の値をライン上に保持し続ける中、新しいトランスミッタは共通のデータラインにアクセスします（値は SPI 設定と通信データによって決まります）。前のトランスミッタがデータ方向設定を変えるまで、ノードは一時的にラインの逆の出力レベルと競合する場合があります。このモードの MISO ピンと MOSI ピンの間に直列抵抗を挿入して競合する出力を保護し、その間の電流の流れを制限することをお奨めします。
4. スレーブによって供給された RDY 信号は、オプションでマスタによって読み出すことができます。

単方向通信

SPI は、SPI_CFG2 レジスタの COMM[1:0] フィールドを使用して送信専用または受信専用に設定することにより、単方向モードで通信できます。この設定では、マスタとスレーブのシフトレジスタ間の転送に使用するのは 1 ラインのみです。残りの MISO ピンまたは MOSI ピンのペアは通信には使用されず、標準の GPIO として使用できます。

送信専用モード : COMM[1:0] = 01

送信専用モードのマスタは、TxFIFO に準備できたデータがあり、マスタの転送が進行中である限り、クロックを生成します。

送信専用モードのスレーブは、SCK ピンにクロックを受信し、SS ピン（またはソフトウェア管理の内部信号）がアクティブである限り、データを送信します（[セクション 52.4.7](#) を参照）。

受信専用モード : COMM[1:0] = 10

マスタモードでは、MOSI 出力が無効化され、GPIO として使用できます。SPI が有効で、SPI_CR1 レジスタの CSTART ビットがセットされている限り、クロック信号が生成され続けます。SPI_CR1 レジスタの CSUSP ビットをセットすることによってソフトウェアが明示的に停止を要求することによって、または SPI_CR1 の MASRX ビットがセットされているとき、RxFIFO がフルになったときに自動的にクロックが停止されます。

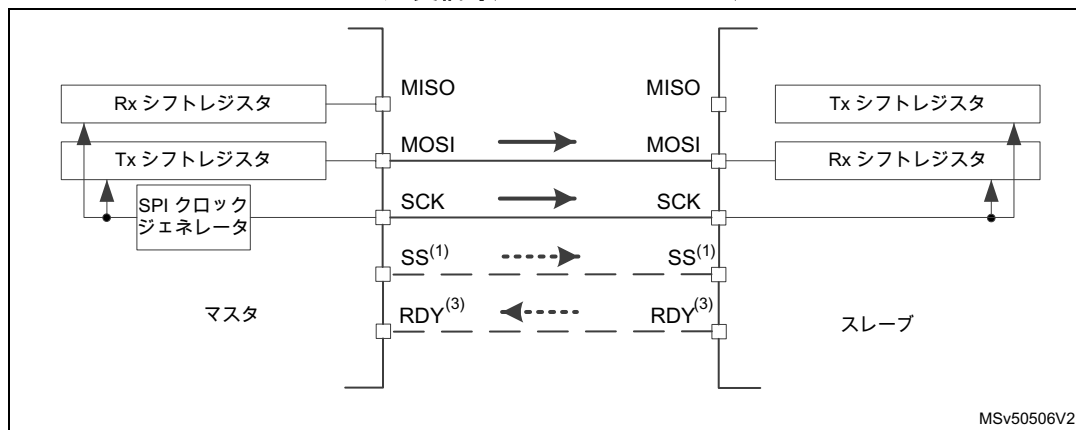
スレーブ設定では、MISO 出力が無効化され、ピンを GPIO として使用することができます。スレーブ選択信号がアクティブな間は、スレーブは MOSI ピンからデータを受信し続けます（[セクション 52.4.7](#) を参照）。

注 :

マスタおよびスレーブモードのどちらであっても、SPI_CFG2 レジスタの IOSWP ビットの値を変更することによって、送信専用のデータピンを受信専用のデータピンで置き換えること、またその逆もできます（このビットは SPI が無効なときにのみ変更できます）。

すべての単方向通信は、トランザクション方向の設定を固定した（双方向モードは有効で HDDIR ビットは決して変更されない）半二重通信の変形型か、未使用データラインと対応するデータフローが無視された場合は全二重制御で置き換えることができます。

図 718. 単方向シングルマスタ/シングルスレーブアプリケーション（送信専用モードのマスタ/受信専用モードのスレーブ）

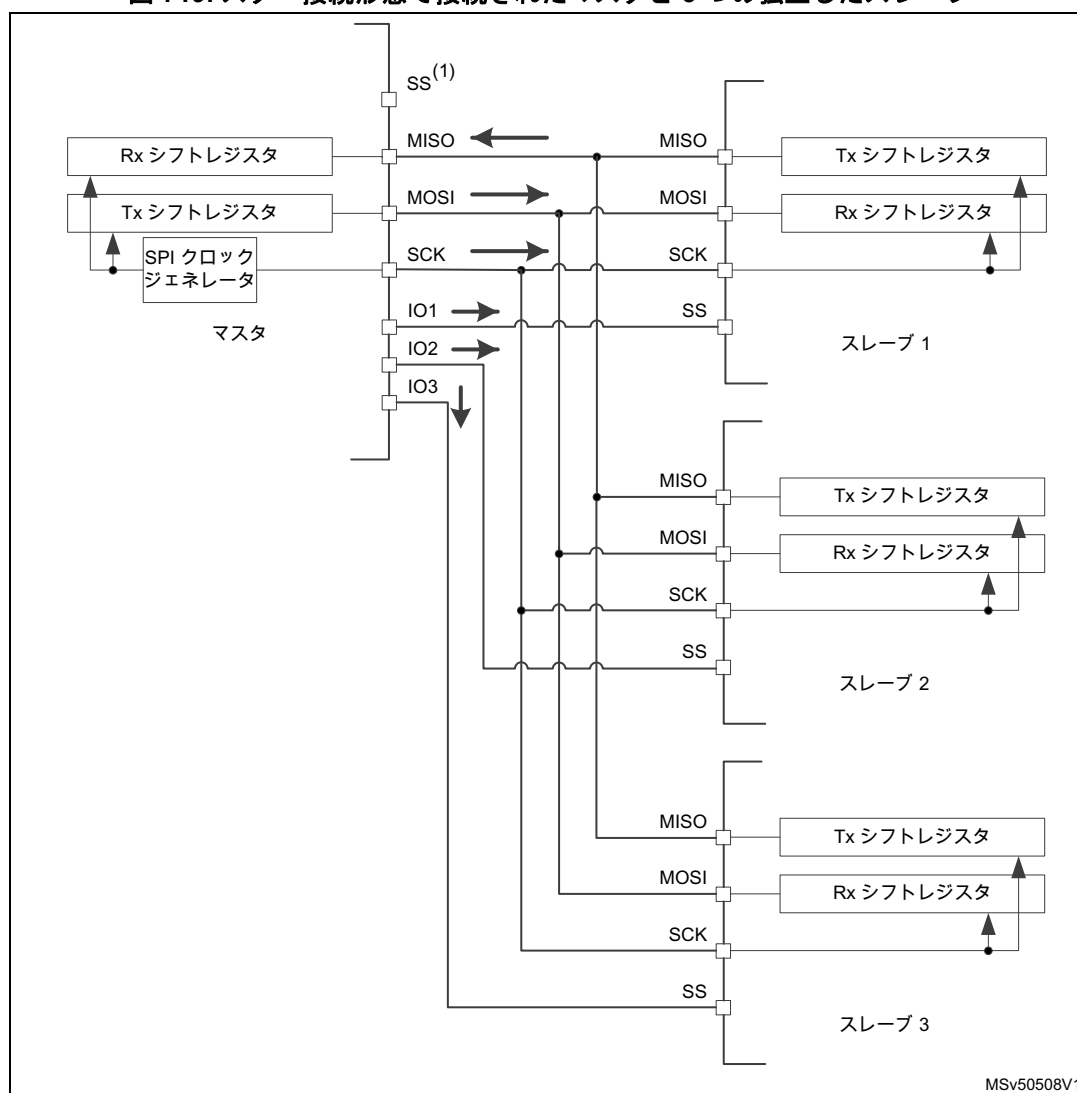


1. SPI インタフェースを機能させるための SS ピンの相互接続は必須ではありません（[セクション 52.4.7](#) を参照）。
2. この設定では、両方の MISO ピンを GPIO として使用できます。
3. スレーブによって供給された RDY 信号は、オプションでマスタによって読み出すことができます。

52.4.5 標準のマルチスレーブ通信

2 つ以上の独立したスレーブがある設定の場合、マスタはスター接続形態を使用し、GPIO ピンを使用して、各スレーブのチップセレクトラインを別々に管理します（[図 719](#) を参照）。マスタは、スレーブの SS 入力に接続されている GPIO をローレベルにプルダウンすることによって、スレーブの 1 つを個別に選択する必要があります（一度に共通 MISO ライン上のデータを制御できるのは 1 つのスレーブのみです）。これを行うことにより、マスタと選択されたスレーブ間の通信が確立します。単純さ以外に、この接続形態の長所は、すべての通信セッションがシングルマスタ-スレーブペア内で別々に行われるので、特定の SPI 設定がスレーブごとに適用できることです。オプションとして、スレーブから情報を読み取る必要がないときは、マスタは同じ情報を複数のスレーブに送信することができます。

図 719. スター接続形態で接続されたマスタと 3 つの独立したスレーブ



1. マスタの 1 つの SS ピンのハードウェア出力機能ではこの接続形態をサポートしていません (ソフトウェア制御されている一連の GPIO に置き換えられます)。ピンでの SPI AF の設定は避けてください (詳細については、[セクション 52.4.7](#) を参照)。
2. アプリケーションが、一度に 1 つの SS アクティブ信号だけがマスタによって供給されていることを確認できない場合、MISO ラインの外部プルアップで MISO ピンをオープンドレイン設定にして、スレーブの相互出力における競合を防ぐことをお勧めします。そうでない場合、外部レジスタなしでプッシュプル設定を適用できます (I/O オルタネート機能入力/出力 (GPIO) 選択を参照)。
3. RDY 信号は、オプションでスレーブからマスタによって読み出すことができます。

サーキュラ接続形態を適用した場合、マスタはすべてのスレーブと一度に SPI 通信を行うことができます (図 720 を参照)。すべてのスレーブは、共通のスレーブ選択 (SS) およびクロック (SCK) 信号の制御のもとで、シリアルチェーンで接続された単純なシフトレジスタのように動作します。すべての情報は同時にサークルを巡ってシフトされて、マスタに戻ります。セッションは長さが固定され、マスタによってトランザクションされるデータフレーム数はスレーブの数と同じです。

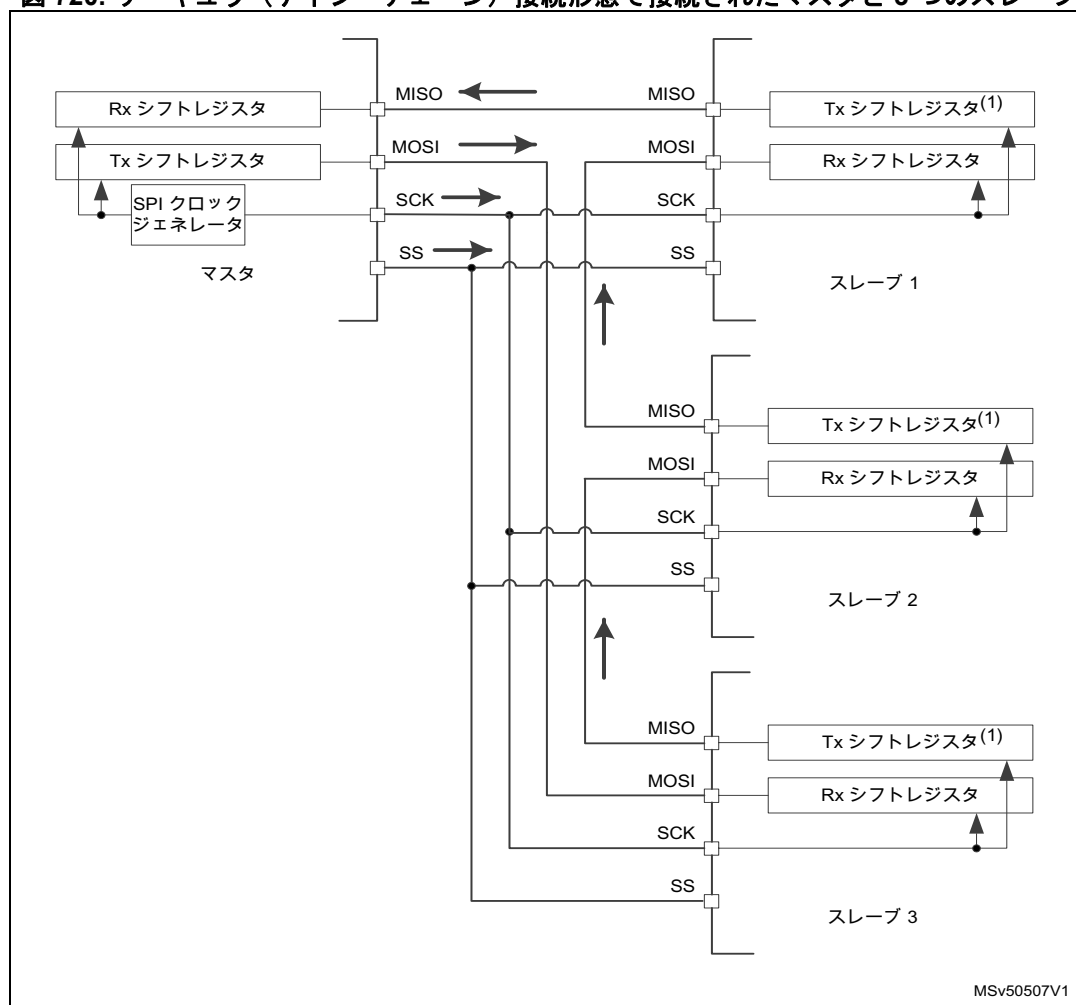
そして、最初のデータフレームがチェーンにトランザクションされる時、マスタはチェーン内の最後のスレーブノードに向けられた情報を最初のスレーブノードの入力を経由して送信し、このとき、マスタによって受信された最初の情報が最後のノードの出力から供給されます。これと同様に、最初の送信データが、セッション中にチェーンの周りを回って他のすべてのスレーブを通過した後で、ちょうどマスタの入力に到達したときに、セッションを終了させる最後のトランザクションデータは

最初のスレーブノード用です。

この接続形態では、データフォーマットの設定とクロックの設定はチェーン内のすべてのノードで同じにする必要があります。受信および送信シフトレジスタは内部的には分離しているため、情報がレシーバとトランスミッタ間でハードウェアによってトランザクションされるとき、TxFIFO スレーブには意図的なアンダーランを伴う手段を適用する必要があります。

この場合、送信アンダーラン機能は、最後に受信したデータフレームを繰り返すモードに設定されます (UDRCFG=1)。セッションはオプションによって、セッション開始前に各スレーブによって TxFIFO に書き込まれた単一のデータパターン (通常、スレーブステータス情報が適用されます) を使用して開始できます。この場合、アンダーランは実際に、この最初のデータフレームがトランザクションされた後に発生します。内部のアンダーラン状態を直ちにクリアして、TxFIFO の内容で再度セッションをリスタートできるようにするには、ユーザはセッション間で SPI を無効化してまた有効化し、TxFIFO に新しい単一データパターンを入れる必要があります (これは、UDRC ビットによる標準の方法でアンダーランをクリアする場合に起きるクリア動作の伝播遅延を回避するためです)。

図 720. サークュラ (デジチェーン) 接続形態で接続されたマスタと 3 つのスレーブ



MSv50507V1

1. スレーブの TxFIFO が空になった後、前に Rx シフトレジスタに受信したデータをスレーブが送信できるときに、アンダーラン機能がこの設定のスレーブによって使用されます。
2. オプションで、RDY 信号は、マスタによって個別に読み出すか、オープンドレイン出力として設定し (RDIOP=0)、低速側デバイスでオーバードライブされた共通チェーンレディステータスとしてプルアップレジスタで接続することができます。

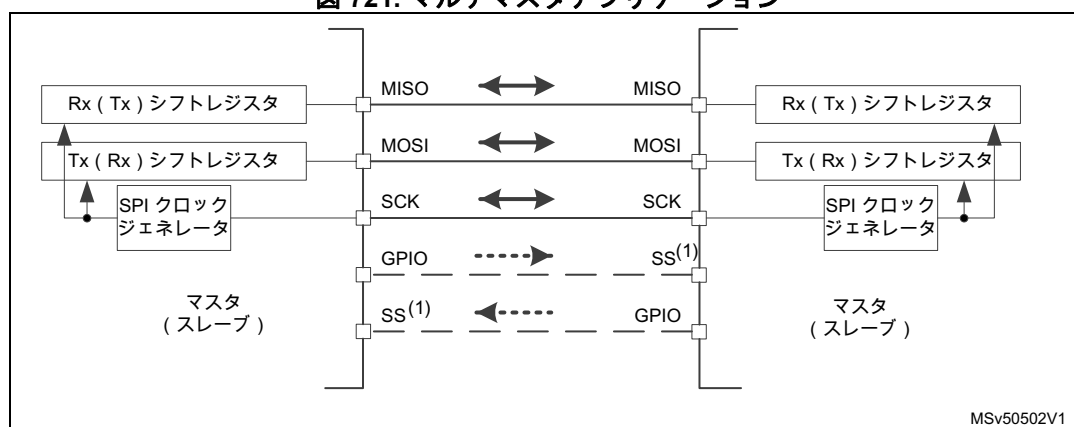
52.4.6 マルチマスタ通信

SPI バスがもともとマルチマスタ機能向けに設計されていない限り、同時にバスを支配しようとする 2 つのノード間での競合の可能性を検出する内蔵機能を使用できます。この検出では、SS ピンをハードウェア入力モードに設定して使用します。共通のデータラインで一度に出力を適用できるノードは 1 つだけであるため、このモードで動作している 2 つ以上の SPI ノードの接続は、不可能です。

ノードが非アクティブであるとき、両ノードはデフォルトでスレーブモードのままとなります。1 つのノードがバスでの制御を得る場合、自身をマスタモードに切り替えて、専用の GPIO ピンを介して他のノードのスレーブ選択入力にアクティブレベルを適用します。セッションが完了すると、アクティブスレーブ選択信号が解除され、バスを一時的に支配するノードがパッシブスレーブモードに戻り、次のセッション開始を待機します。

両ノードのマスタリングリクエストが同時に上がった場合、バス競合イベントが発生します（モードフォールト MODF イベントを参照）。いくつかの簡単なアービトレーションプロセスを適用できます（たとえば、両方のノードに適用する事前定義された異なるタイムアウトによって次の試行を延期します）。

図 721. マルチマスタアプリケーション



1. SS ピンは両ノードともハードウェア入力モードで設定されます。アクティブレベルにすることで、パッシブノードがスレーブとして設定されるため、MISO ライン出力の制御が可能になります。
2. RDY 信号はこの通信では使用されません。

52.4.7 スレーブ選択 (SS) ピンの管理

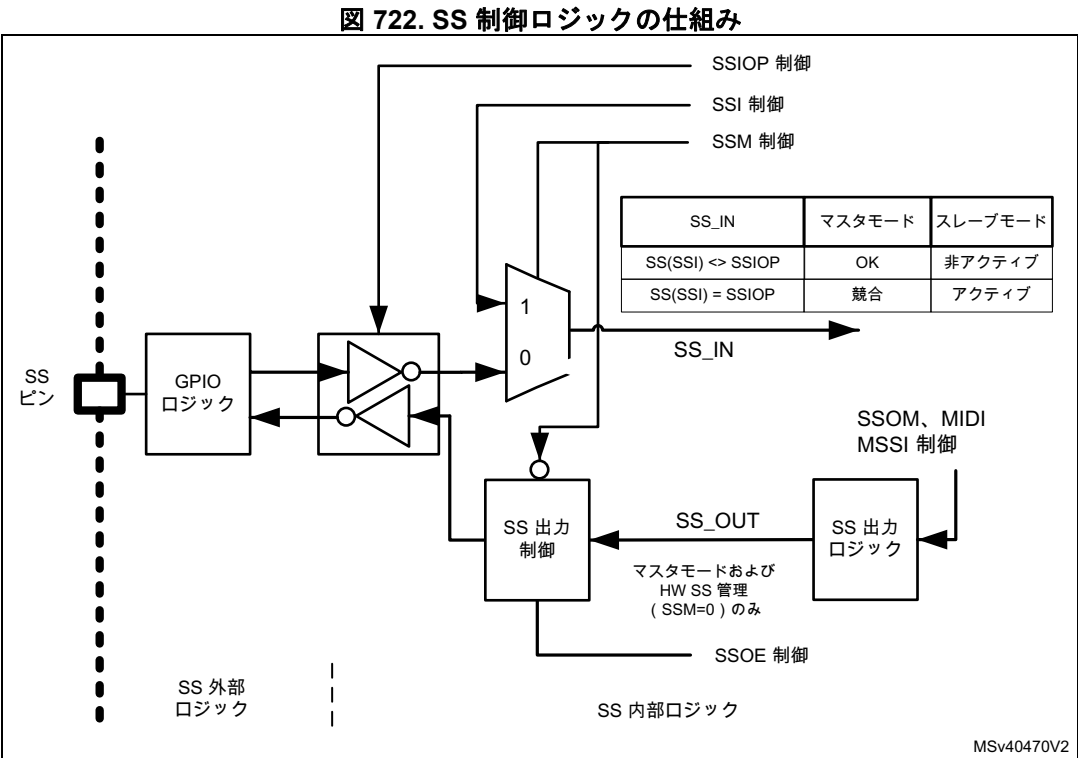
スレーブモードでは、SS は標準の「チップセレクト」入力として機能し、スレーブをマスタと通信させます。マスタモードでは、SS は出力としても入力としても使用できます。入力として使用する場合は、SS はマルチマスタのバスの衝突を未然に防ぎ、出力として使用する場合は 1 つのスレーブのスレーブ選択信号を駆動させることができます。SS 信号は内部的に管理することも（SS 入力のソフトウェアによる管理）、SS 入力および出力の両方が SS ピンに関連付けられているときは外部的に管理することもできます（ハードウェアによる SS 管理）。この入力/出力外部信号（SS ピンに存在）のどちらのレベルがアクティブとみなされるのかを SSIOP ビットのセットによって設定することができます。SS レベルは SSIOP と等しい場合、アクティブとみなされます。

ハードウェアまたはソフトウェアのスレーブ選択管理は、SPI_CFG2 レジスタの SSM ビットを使用して以下のように設定することができます。

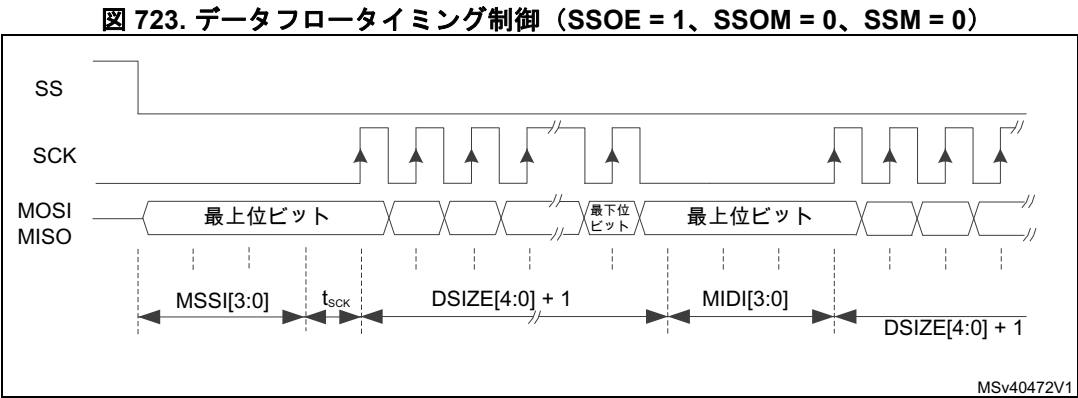
- **ソフトウェア SS 管理 (SSM = 1) :** この設定では、スレーブ選択情報は SPI_CR1 レジスタの SSI ビットの値によって内部で駆動されます。外部 SS ピンは他のアプリケーションで使用できません (GPIO またはその他のオルタネート機能として)。
- **ハードウェア SS 管理 (SSM = 0) :** この場合、2 通りの設定が可能です。次のどちらの設定を使用するかは、SS 出力設定 (SPI_CFG2 レジスタの SSOE ビット) によって決まります。
 - **SS 出力有効 (SSOE = 1) :** この設定は、MCU がマスタとしてセットされている場合にのみ使用します。SS ピンはハードウェアによって管理されます。この機能は CSTART および EOT 制御に接続されています。そのため、マスタは SS 出力を正しく制御するために、適切な TSIZE>0 設定を適用する必要があります。SPI AF が SS ピンで適用されていない場合でも (その場合は標準の GPIO として使用可能)、デフォルトの SS 入力レベルを保証するために SSOE = 1 は保持し、マルチマスタ接続形態にのみ適用可能なマスタ SS 内部ロジック入力時のモードフォールト評価を防いでください。
 - a) SSOM = 0 で、SP = 000 のとき、SS 信号は、マスタの転送が開始されると (CSTART = 1) すぐにアクティブレベルになり、EOT フラグがセットされるか、送信がサスペンドされるまでアクティブに保たれます。
 - b) SP = 001 のとき、TI モードで定義されたパルスが生成されます。
 - c) SSOM = 1、SP = 000、および MIDI > 1 のとき、SS はデータフレーム間で非アクティブにパルス出力され、MIDI 値から 1 を減じた値 (1 ~ 14) によって定義される SPI クロック周期の間、非アクティブに保たれます。
 - d) SS 入力は、モードフォールトを防ぐため、マスタ内部で強制的に非アクティブ状態になります。
 - **SS 出力無効 (SSM = 0、SSOE = 0) :**
 - a) マイクロコントローラがバスでマスタとして機能している場合、この設定によりマルチマスタ機能が可能になります。このモードで、SS ピンがアクティブレベルに引き込まれた場合、SPI はマスタモードのフォールト状態に入り、SPI デバイスは自動的にスレーブモードに再設定 (MASTER = 0) されます。
 - b) スレーブモードでは、SS ピンは標準の「チップセレクト」入力として機能し、SS ラインがアクティブレベルの間はスレーブが選択されます。

注 : モードフォールト状態で自動的にスレーブモードに切り替える目的は、データおよびクロックラインでの競合の可能性を避けるためです。SPE はこの状態で自動的にリセットされるため、Rx と Tx の両方の FIFO は一掃され、現在のデータは失われます。

SPI スレーブがハードウェア SS 管理モードで有効化されたとき、SS がアクティブレベルであることがわかった場合でも、すべてのトラフィックが無視されます。これらは、スレーブをマスタと同期させる SS 信号の開始 (非アクティブからアクティブレベルへのトランザクション) がスレーブによって検出されるまで無視されます。こういうわけで、外部 SS ピンが固定されているときは、ハードウェア管理モードは使用できません。SS のソフトウェア管理ではそのような保護はありません。そこで、この場合に限って、バスにトラフィックが無くて、SCK 信号が転送間でアイドル状態レベルにあるときに、SSI ビットを変更すべきです。



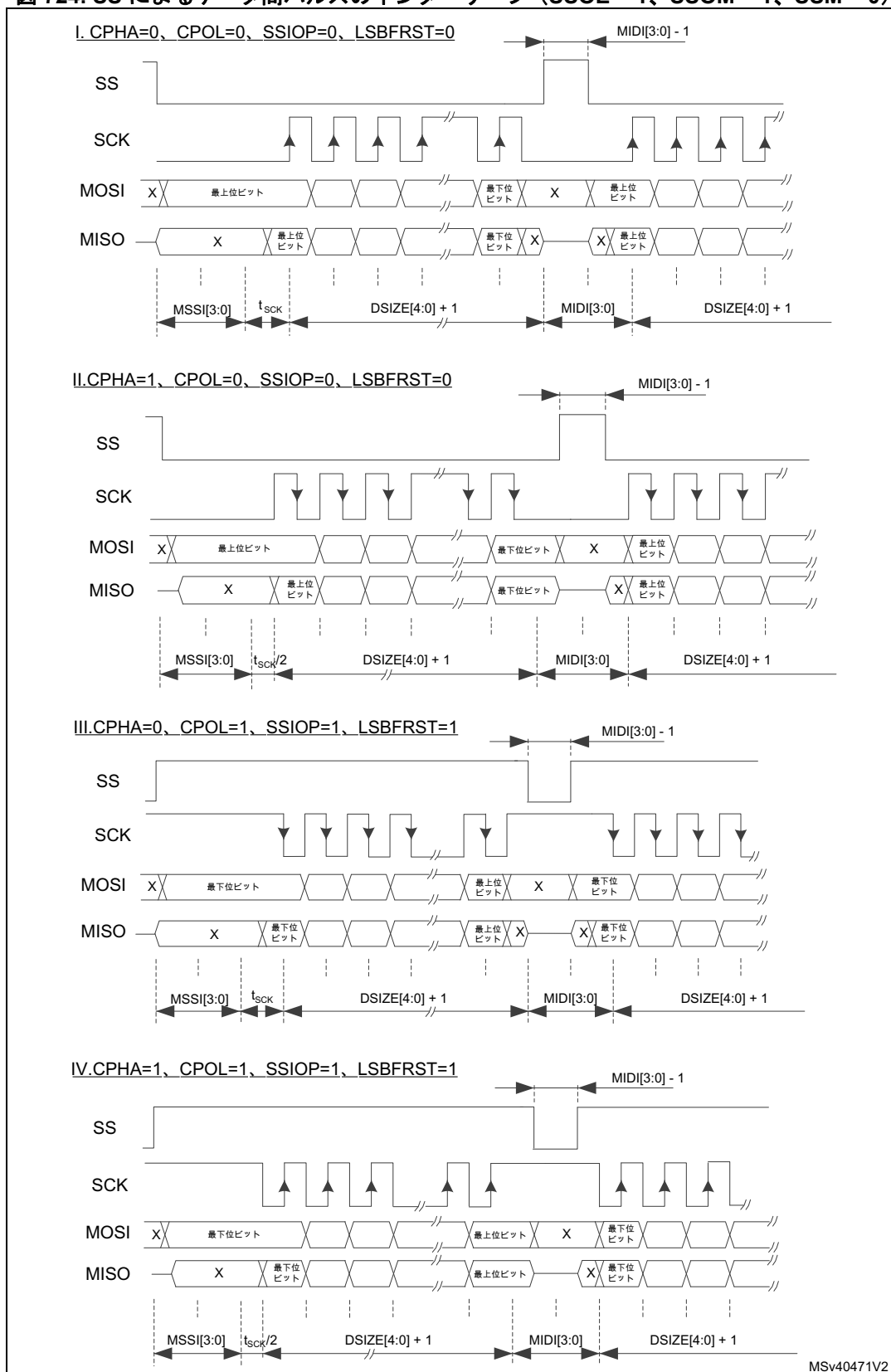
ハードウェア出力 SS 制御が適用されるとき (SSM = 0、SSOE = 1)、MIDI[3:0] および MSSI[3:0] ビットフィールドの設定によって、データフレーム間の SS 信号のタイミングを制御し、(SS とクロック開始を分離するために) すべてのトランザクションの初めに余分の遅延時間を挿入することができます。これは、スレーブが正しいデータ処理のための十分な余裕を得るために流れの速度を落とす必要があるときに有用です (図 723 を参照)。



1. MSSI[3:0] = 0011、MIDI[3:0] = 0011 (MIDI[3:0] = 0 のときは、SCK フローは連続です)。
2. CPHA = 0、CPOL = 0、SSIOP = 0、LSBFRST = 0。

さらに、ビット SSOM = 1 の設定により、パルスを挿入する十分なスペースがあればデータフレーム間にパルスをインターリーブする特定のモードが呼び出されます (MIDI[3:0] は SPI の 1 周期より大きく設定する必要があります)。設定のいくつかの例を図 724 に示します。

図 724. SS によるデータ間パルスのインターリーブ (SSOE = 1、SSOM = 1、SSM = 0)



1. $MSSI[3:0] = 0010$ 、 $MIDI[3:0] = 0010$ 。
2. インターリーブ・パルスの幅が $MIDI[3:0] > 1$ の場合のデータ間の SS インターリーブは、常に (MIDI パラメータによって定義された) フレーム間に設けられたギャップより、1 SCK 周期少なくなります。MIDI がセットされている場合、フレームは 1 つの SCK 周期で分けられますが、SS にインターリーブ・パルスは現れません。

52.4.8 レディピン (RDY) の管理

データを処理するためのスレーブ機能のステータスは、RDY ピンで確認できます。デフォルトで、ローレベルはスレーブのトランザクションの準備ができていないことを示します。理由として、TxFIFO が空であること、RxFIFO がフルであること、または SPI が無効化されていることがあげられます。信号のアクティブレベルは RDIOP ビットで選択できます。ノット・レディ・ステータスを示していても、マスタがスレーブとの通信を続けたり、開始したりする場合、トランザクションはほぼ間違いなく大失敗します。

RDY 出力を制御するロジックはやや複雑で、TSIZE および DSIZE の設定と密接につながっています。RDY の反応は、TxFIFO がほぼ空になったり、フレームのトランザクション中に RxFIFO がほぼフルになったりすることについて悲観的で、影響を受けやすいです。この悲観的なロジックは、RDY がアクティブなままであれば、TxFIFO が完全に空になっても、RxFIFO が完全に占有されても、トランザクションの最後に抑制されます。目的は、データ破損を防ぐことと、次にトランザクションされたデータをまた安全に処理できるようになるまで、一時的にトランザクションをサスペンドする必要があることをマスタに間に合うように通知することです。RDY 信号入力のマスタ側で有効な場合、マスタはスレーブがノット・レディ・ステータスを示したときに通信をサスペンドします。これは、マスタが処理中のフレームのトランザクションを完了するのを防ぎます。単にスレーブの TxFIFO が空になるか、RxFIFO に次のデータが書き込まれるか読み出されるまで完全に満たされます (フレームにかかわらず、制約なしでも完了できます)。スレーブに TSIZE = 0 設定を適用すると、スレーブがトランザクションの終了をまったく評価しなくなるため、問題になる場合があります (これにより、最後のデータの送信時にノット・レディ・ステータスが抑制されます)。その場合、ユーザは RxFIFO を解放し、ソフトウェアによってスレーブ側で TxFIFO に追加の (ダミーも可) データを書き込んでノット RDY 信号を解除し、ST マスタをブロック解除することで、しばしばフレームの中央でサスペンドされる通信で続けられるようにします。

マスタで RDY が使用されない場合は、無効にする必要があります ($RDIOM = 0$)。その後、マスタの内部ロジックがすでにレディ状態のスレーブステータスのシミュレーションを実施します。この場合、RDIOP ビット設定は意味を持ちません。

マスタとスレーブの両側で、クロックドメインと RDY ロジックの評価の間の同期により、RDY ピン機能は信頼できません。また、データフレームのサイズが 8 ビットより短く設定されている場合は使用できません。

52.4.9 通信フォーマット

SPI 通信中は受信と送信の操作が同時に行われます。シリアルクロック (SCK) は、データライン上で行われる情報のシフトとサンプリングを同期させます。通信フォーマットは、クロック位相、クロック極性、およびデータフレームフォーマットに応じて決定されます。マスタデバイスとスレーブデバイスの通信を可能にするには、双方が同じ通信フォーマットに従い、正しく同期される必要があります。

クロックの位相および極性の制御

SPI_CFG2 レジスタの CPOL ビットと CPHA ビットを使用することによって、考えられる 4 つのタイミングの関係をソフトウェアで選択できます。CPOL (クロック極性) ビットは、データが転送されていないときのクロックのアイドル状態の値を制御します。このビットは、マスタモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA ビットがセットされると、SCK ピンの 2 番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがリセットされていれば立下りエッジ、CPOL ビットがセットされていれば立上りエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。CPHA ビットがリセットされている場合、SCK ピンの 1 番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがセットされていれば立下りエッジ、CPOL ビットがリセットされていれば立上りエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。

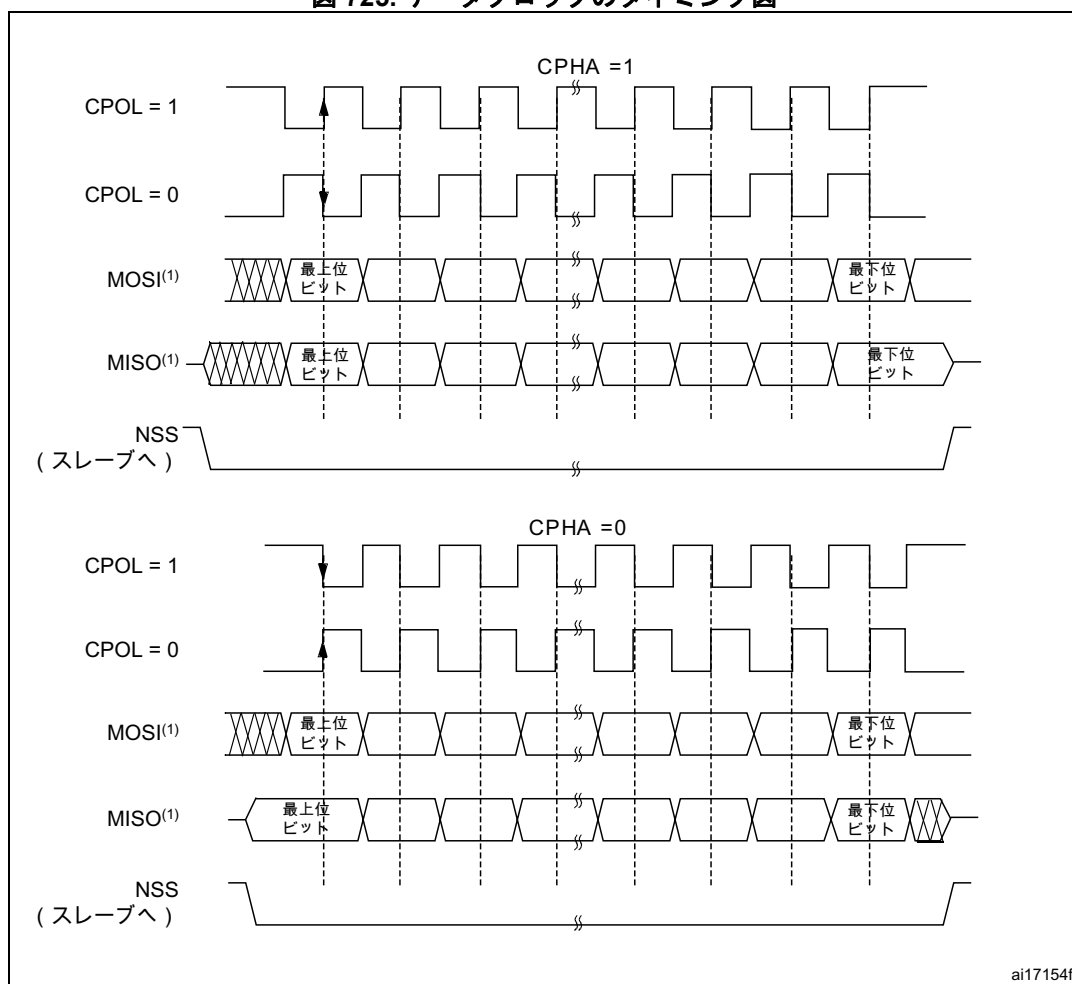
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジ (図 725 の点線) を選択できます。

図 725 は、CPHA ビットと CPOL ビットの 4 つの組み合わせによる SPI 全二重転送を示しています。

注 : CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。

SCK のアイドル状態は、SPI_CFG2 レジスタで (CPOL = 1 なら SCK ピンのプルアップ、CPOL = 0 ならプルダウンによって) 選択された極性に一致する必要があります。

図 725. データクロックのタイミング図



1. データビットの順序は LSBFRST ビットの設定値に依存します。

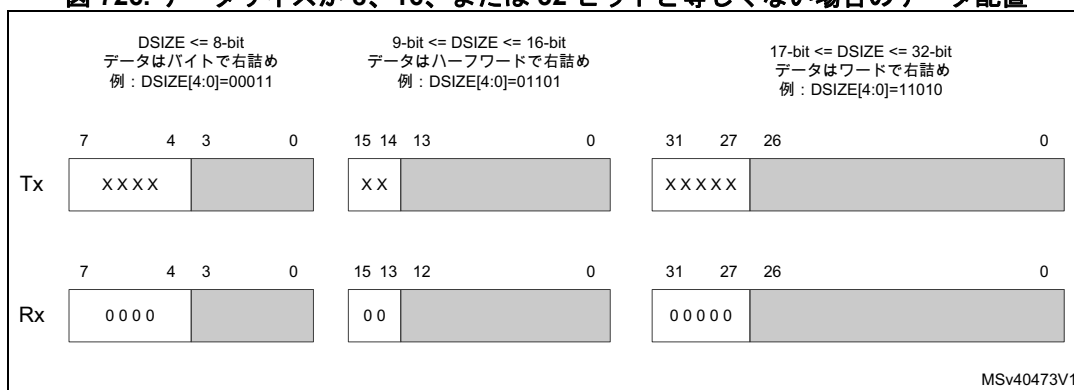
データフレームフォーマット

SPI_CFG2 レジスタの LSBFRST ビットの値に応じて、SPI シフトレジスタを設定することで、シフトを MSB ファーストまたは LSB ファーストに設定することができます。

全機能がセットされたインスタンスで、データフレームサイズは SPI_CFG1 レジスタの DSIZE[4:0] ビットを使用して選択できます。4 ビットから 32 ビット長までの間にセットでき、この設定は送信と受信の両方に適用できます。SPI_TXDR/SPI_RXDR レジスタにアクセスすると、データフレームはバイト（データがバイトに適合する場合）、ハーフワード、またはワードのいずれかに常に右詰めされます（図 726 を参照）。

アクセスが設定されたデータサイズの倍数の場合、データパッキングは自動的に適用されます。通信中、データフレーム内のビットのみがクロック供給され、送信されます。

図 726. データサイズが 8、16、または 32 ビットと等しくない場合のデータ配置



注：最小データ長は 4 ビットです。4 ビット未満のデータ長が選択された場合は、4 ビットのデータフレームサイズに固定されます。

機能が制限されているインスタンスの場合、DSIZE[4:3] のビット値に従って、データサイズは 8 ビットの倍数から最大データ長に固定されます（インスタンスによる）。SPI_TXDR または SPI_RXDR へのアクセスが拡大されている場合（設定されたデータサイズの倍数）、データパッキングは自動的に適用されます。

52.4.10 SPI の設定

設定手順は、マスタとスレーブではほぼ同じです。特定のモードの設定については、それぞれのモードに関する章を参照してください。標準通信を初期化する必要があるときは、以下の手順を実行します。

- 適切な GPIO レジスタの書込み：適用されている場合は、MOSI、MISO、SCK、SS、および RDY ピンで GPIO オルタネート機能を設定します。
- SPI_CFG1 と SPI_CFG2 レジスタに書き込んで、前の SPI が有効な場合、以下を除く「予約済み以外」のビットおよびビットフィールドすべてに適切な値を設定します。
 - SSOM、MASRX、SSOE、RDIOM、MBR[2:0]、BPASS、MIDI[3:0]、MSSI[3:0] ビットはマスタモードでのみ考慮されます。MSSI[3:0] ビットは SSOE ビットがセットされたときのみ有効になり、RDIOP ビットはマスタモードで RDIOM ビットがセットされていない場合は有効になりません。スレーブが TI モードで設定されると、MBR[2:0] 設定も考慮されます。
 - UDRCFG はスレーブモードでのみ考慮されます。
 - CRCSIZE[4:0] は CRCEN がセットされた場合に必要です。
 - CPOL、CPHA、LSBFRST、SSOM、SSOE、SSIOP、SSM、RDIOP、RDIOM、MSSI、および MIDI は TI モードでは不要です。

- e) SPI_CFG2 レジスタで AFCNTR ビットがセットされると、ペリフェラルが有効にもかかわらず、すべての SPI 出力は関連する GPIO ピンに伝播され始めます。したがって、その後の SPI_CFG1 および SPI_CFG2 レジスタの設定変更は、これらのピンの信号のレベルに影響を与えます。
3. 転送の長さを選択するために SPI_CR2 レジスタへ書き込みます。それが未知の場合は、TSIZE はゼロにプログラムしなければなりません。
4. 必要に応じて CRC 多項式と CRC 計算を設定するために、SPI_CRCPOLY と、SPI_CR1 レジスタの TCRCINI、RCRCINI、および CRC33_17 ビットに書き込みます。
5. DMA ストリームが使用されている場合は、DMA レジスタに SPI Tx および Rx 専用の DMA ストリームを設定します (DMA を使用した通信の章を参照)。
6. 必要に応じて、SPI_CR1 レジスタで SSI、HDDIR、および MASRX を設定します。
7. 設定の保護が必要な場合は (安全のため)、SPI_CFG1 レジスタの IOLOCK ビットをプログラムします。

52.4.11 SPI の有効化

マスタがクロックを送信する前に、SPI スレーブを設定し、有効にすることを推奨します。ただし、トラフィックがバスで進行中に設定および有効化手順を実行しても悪影響はありません。このとき、SS 信号はスレーブでハードウェアによって管理されるか、SS 信号がソフトウェアによって管理されているときにスレーブのソフトウェアによって非アクティブ状態で保持されるものとします ([セクション 52.4.7](#) を参照)。スレーブトランスミッタのデータレジスタには、マスタがクロック供給を開始する前に、送信データが含まれていなければなりません。SPI スレーブが SS によって選択される前に、SCK 信号は選択した極性に対応するアイドル状態レベルに安定させる必要があります。そうしないと、続くトランザクションの同期が取れなくなってしまうことがあります。

SPI スレーブがハードウェア SS 管理モードで有効化されたとき、SS がアクティブレベルであることがわかった場合でも、すべてのトラフィックが無視されます。これらは、スレーブをマスタと同期させる SS 信号の開始 (非アクティブからアクティブレベルへのトランザクション) がスレーブによって検出されるまで無視されます。こういうわけで、外部 SS ピンが固定されているときは、ハードウェア管理モードは使用できません。SS のソフトウェア管理ではそのような保護はありません。そこで、この場合に限って、バスにトラフィックが無くて、SCK 信号が転送間でアイドル状態レベルにあるときに、SSI ビットを変更すべきです。

全二重 (または送信専用モード) のマスタは、SPI が有効で CSTART ビットがセットされ、TXFIFO が空でない場合、または次の TXFIFO への書き込み時に通信を開始します。

マスタ受信専用モードでは、SPI が有効化され、CSTART ビットがセットされた後、マスタは通信を開始し、クロックが動作を開始します。

DMA の扱いについては、[セクション 52.4.15](#) を参照してください。

52.4.12 SPI のデータ送受信手順

データ通信フォーマットの設定は、柔軟性の高いサイズの確実なデータ数をセッション内で転送する (トランザクション) という基本原理に従います。一方、オプションで、データ処理は SPI データレジスタの 1 回のアクセス (データパッキング) に効率的に累積したり、一貫したより大きなデータパケットにデータが収集されている場合は、このようなサービスのシーケンスにグループ化したりすることもできます。データ処理サービスは、FIFO パケットの占有イベントに基づいています。このために、完全なデータパケットは、専用パケットフラグによってのみ処理する必要があります。

このセクションの次の詳細な内容についてよりの確に理解するには、最初に以下の項目の設定の影響と意味を把握してください。

データサイズ (DSIZE) - データフレームを定義します (1 つのデータフレームのビット数をセットします)。

FIFO 閾値 (FTHLV) - データパケットを定義し、1 つのデータパケットのデータフレーム数をセットして、ソフトウェアまたは DMA によって処理される SPI データレジスタのパケットの占有イベントの発生をセットします。

データアクセス - パケットイベント時に転送データがアプリケーションと SPI FIFO の間にある場合の SPI データレジスタの内容の処理方法。これは、パケットサイズの設定によって決まります。オプションとして、レジスタに 1 回アクセスすることで (データパッキング)、または一連のそのようなアクセスにより (大きなデータパケットを処理する場合)、複数のデータを処理できます。

FIFO サイズ - 使用可能なデータを吸収する容量またはスペース。これは、データサイズと、データがスペース内でどのように圧縮され、構成されるかという内部ハードウェアの効率によって決定します。FTHLV 設定では、最低 2 つのデータパケットを格納するという FIFO の機能を考慮する必要があります。

トランザクションサイズ (TSIZE) - 複数のデータパケットサービスでカバーされる、全体的なトランザクションセッションに関わるデータフレームの総数を定義します。この数字をパケットサイズに調整する必要はありません (最後の調整されていないデータパケットの処理は、TSIZE が適切にプログラムされている場合にサポートされます)。

RxFIFO および TxFIFO によるデータ処理

すべての SPI データトランザクションは複数バイト ($N \times 8$ ビット) で構成された内蔵 FIFO を通過します。FIFO のサイズ (N) は、製品およびペリフェラルのインスタンスに依存します。これにより、SPI は連続フローで動作できるようになり、またデータフレームサイズが短い場合や、割り込み/DMA 遅延時間が長すぎる場合にオーバーランを防ぐことができます。方向ごとにそれぞれ TxFIFO と RxFIFO と呼ばれる固有の FIFO があります。

FIFO の内容の処理は、専用の FIFO パケットの占有フラグ (TXP、RXP、または DXP) によって発生するデータ・パケット・イベントの処理に基づきます。フラグの発生は、データ変換モード (二重、単方向)、データフレームサイズ (フレーム内のビット数)、およびデータパケットでのデータの構成方法に依存します。パケット・イベントの頻度は、FIFO の閾値を定義して、データをパケットにまとめた場合は大幅に低減できます。これにより、1 つの FIFO 占有パケットイベントに基づいて、1 つの SPI データレジスタアクセスまたはシステム性能の消費量が低いそれらのシーケンスによって、パケットでグループ化されたいくつかのデータフレームを効率的に処理できます。ユーザは、レジスタの読出しまたは書込みに適用された具体的な CPU 命令を強制するために、データレジスタのアドレスを割り当てることで、アクセスタイプを制御できます。これにより、アクセスは 8 ビット、16 ビット、または 32 ビットにすることができますが、少なくとも 1 つのデータフレームに常にアクセスする必要があります。パケットサイズ (FTHVL) およびデータサイズ (DSIZE) の設定は、イベント時に 1 つのデータパケットの処理を適用して完了するためだけに、適用されたデータレジスタのアクセス (1 つのアクセスかそれらのシーケンスが適用されているかにかかわらず) と常にバランスが取れた状態で保持することが重要です。FIFO の占有フラグの原理、発生、およびクリア機能は、DMA、割り込み、またはポーリングのいずれが適用されていても共通です。

SPI_RXDR レジスタへの読出しアクセスからは、RxFIFO に保管された、まだ読み出されていない一番古い値が返されます。SPI_TXDR への書込みアクセスでは、送信キューの最後に TxFIFO に書き込まれるデータを保管します。

SPI_RXDR レジスタへの読出しアクセスは RXP イベントで管理する必要があります。このフラグは、受信がアクティブになっているとき、少なくとも 1 つの完全なデータパケット (SPI_CFG1 レジスタの FTHLV[3:0] ビットによってレシーバの閾値として定義されている) が受信 FIFO で利用可能になったときにハードウェアによってセットされます。RXP は、SPI_RXDR がソフトウェアまたは DMA によって読み出されたとき、RxFIFO に利用可能なデータが、完全な 1 つのパケットよりも少なくなるとすぐにクリアされます。

RXP は、RXPIE ビットがセットされている場合は割込みを、および/または RXDMAEN ビットがセットされている場合は DMA リクエストをトリガします。

RXP フラグがセットされると、アプリケーションは SPI データレジスタの読出しを所定回数行って 1 つのデータパケットの内容をダウンロードします。完全なデータパケットがダウンロードされると、アプリケーションソフトウェアまたは DMA は RXP の値をチェックして受信 FIFO 内に他のパケットが保留されていないかを調べ、もしあれば、RXP が 0 になるまでそれらをパケットごとにダウンロードします。N をバイト単位の FIFO のサイズとして、RxFIFO が格納できる最大データフレーム数は、N 個（フレームサイズ ≤ 8 ビットの場合）、N/2 個（8 ビット < フレームサイズ ≤ 16 ビットの場合）、N/3 個（16 ビット < フレームサイズ ≤ 24 ビットの場合）、または N/4 個（データフレームサイズ > 24 ビットの場合）です。

受信の最後に、FTHLV レベルに達しておらず、いくつかのデータが RxFIFO 内でまだ利用可能になっていて、RXP がセットされないことがあります。この場合、FIFO 内に残っている受信データフレームの数が SPI_SR レジスタの RXWNE および RXPLVL フィールドによって示されます。これは、転送サイズとパケットサイズが整合していない場合に、転送で受信した最後のデータの数で設定パケットサイズを完全に満たすことができないときに発生します。それにもかかわらず、アプリケーションソフトウェアはそれまでの完全なデータパケットに使用された、RxFIFO からの標準の数の読出しを問題なく行うことができます。矛盾のないデータ（完了したデータフレーム）のみが RxFIFO から取り出され、冗長な読み出し（または不完全なデータ）は 0 となります。このおかげで、アプリケーションソフトウェアは転送内のすべてのデータを同じように扱うことができ、転送内の最後のデータの受信を予知することや、RxFIFO から取り出すべき読出し所要数を計算することの負担から解放されます。

同じように、送信するデータフレームの書き込みアクセスは TXP イベントで管理されます。このフラグは、送信がアクティブになっているとき、少なくとも 1 つの完全なデータパケット（SPI_CFG1 レジスタの FTHLV[3:0] ビットで定義されている）をアプリケーションが送信 FIFO にプッシュするのに十分なスペースができたときに、ハードウェアによってセットされます。TXP は、Tx FIFO がソフトウェアおよび/または DMA によって書き込まれるとすぐにクリアされます。次の完全なデータパケットのために現在使用可能なスペースは失われます。これによって、データが Tx FIFO から放出され、新しいパケットがフレームごとに格納されるとき、TXP 信号が振動することになります。少なくとも 1 つのデータフレームを格納するだけの十分なスペースがないとき（TXP イベントは無視されます）、TXTF がセットされているとき、または SPI が無効になっているとき、Tx FIFO への書き込みは無視されます。

TXP は、TXPIE ビットがセットされている場合は割込みを、および/または TXDMAEN ビットがセットされている場合は DMA リクエストをトリガします。TXPIE マスクは、TXTF フラグがセットされると、ハードウェアによってクリアされます。

TXP フラグがセットされると、アプリケーションは SPI データレジスタへの書き込みを所定回数行って 1 つのデータパケット全体の内容をアップロードします。新しい完全なデータパケットがアップロードされると、アプリケーションソフトウェアまたは DMA は TXP の値をチェックして Tx FIFO に他のパケットをプッシュできるかどうかを調べ、もしできるならば、TXP が 0 になるまで、それらをパケットごとにアップロードします。

転送サイズとパケットサイズが整合していない場合、転送の最後のデータの数で設定パケットサイズより短いことがあります。それにもかかわらず、アプリケーションソフトウェアはそれまでのパケットに使用された標準の数のデータレジスタ書き込みを問題なく行うことができます。矛盾のないデータのみが Tx FIFO にプッシュされ、冗長な書き込みは破棄されます。このおかげで、アプリケーションソフトウェアは転送内のすべてのデータを同じように扱うことができ、転送内の最後のデータの送信を予知することや、最後のデータを Tx FIFO にプッシュするための書き込み所要数を計算することの負担から解放されます。最後のデータの場合、現在の転送を完了するための残りのデータを格納するのに十分なスペースが Tx FIFO にあれば、SPI によって TXP イベントがアサートされます。

TXP イベントと RXP イベントの両方を割込みによってポーリングまたは処理できます。全二重モードでは TXP と RXP の共通のイベントとして DXP ビットを監視することができます。

DXP フラグがセットされると、アプリケーションは SPI データレジスタへの書き込みを所要回数行って送信用の 1 つのデータパケット全体の内容をアップロードし、その後、SPI データレジスタから同じ数の読み出しを行い、1 つのデータパケットの内容をダウンロードします。1 つのデータパケットがアップロードされ、1 つがダウンロードされると、アプリケーションソフトウェアまたは DMA は DXP の値をチェックして他のパケットがプッシュおよびポップできないか順に調べ、もしできれば、DXP が 0 になるまでそれらをパケットごとにアップロード/ダウンロードします。

DXPIE ビットがセットされている場合、DXP は割り込みをトリガします。DXPIE マスクは、TXTF フラグがセットされると、ハードウェアによってクリアされます。

DXP は、全二重通信においてデータのアップロード/ダウンロードの性能を最適化するために有用であり、SPI の転送をサポートするのに必要な割り込み回数または DMA シーケンスを削減するか、その結果、特に SPI が STOP モードで動作しているとき、CPU の帯域幅およびシステム電力の要求を最小化します。

リレーが DXP 割り込み専用の場合、TXP サービスが意図的に遅延され、TXP と RXP のイベントが共通手順で処理される場合に、ユーザはそのような簡略化の欠点を考慮する必要があります。これは、TXP イベントが受信 RXP イベントの前に行われるため、これにより通常は最後のフレームの前のトランザクションを TXP で処理して TxFIFO を完全に空にすることができます。それ以外の場合、マスタは連続 SCK クロックの流れを供給できず、スレーブはアンダーラン条件に対応することも可能です。考えられるソリューションとして、セッションを開始する前に TxFIFO をいくつかのデータパケットで事前に書き込むこと、またトランザクションの最後にある EOT によって TXTF イベント後に受信したすべてのデータを処理すること（トランザクションの最後に TXTF が DXP 割り込みを抑制するため）があげられます。CRC 計算が有効である場合、トランザクションの最後の EOT にのみ基づいて、RxFIFO で CRC フレームに対応するため、追加のスペースありで計算する必要があります。

データ交換を管理するもう一つの方法は、DMA を使用することです（[セクション 52.4.15](#) を参照）。

RxFIFO がフルのときに次のデータを受信した場合、オーバーランイベントが発生します（[セクション 52.5.2](#) の OVR フラグの説明を参照）。オーバーランイベントは割り込みによってポーリングまたは処理できます。

これは、MASRX = 0 のときにスレーブモードまたはマスタ受信モードで発生することがあります。マスタスレーバで MASRX ビットがセットされている場合、RxFIFO がフルになったとき、生成されたクロックが自動的に停止し、したがってオーバーランは回避できます。

RxFIFO と TxFIFO の両方の内容は、SPI が無効（SPE = 0）のときには一掃された状態に保たれ、アクセスできません。

トランザクションの処理

いくつかのデータフレームを単一シーケンスに渡してメッセージを完成させることができます。TSIZE に格納された値によって、メッセージ内のデータ数量を処理することができます。原則として、メッセージのトランザクションは CSTART ビットのセットによって SPI が有効になった時に開始し、要求されたデータの総数がトランザクション処理された時に終了します。トランザクションの終わりに、CRC およびハードウェア SS 管理（適用される場合）が制御されます。内部ステートマシンを適切に再開するには、設定を変更しない場合でも、次のトランザクションが始まる前に SPI を無効にして再度有効化することを強く推奨します。

CSTART がセットされる時、TSIZE がゼロに維持されている場合、終わりのないトランザクションが開始されます（転送サイズの制御は適用されません）。終わりのないトランザクションの間、FIFO の閾値と整合が取れているトランザクションされたデータ数のみがサポートされます。データ数（またはパケットにグループ化したもの）は予測不可能です。各データフレームがアプリケーションまたは DMA で処理される独自のパケットイベントを発生させることを確認するために、1 つのデータ（FTHLV = 0）の FIFO 閾値設定（パケットサイズ）を保持する必要があります。トランザクションは、CSTART ビットをクリアする CSUSP によっていつでもサスペンドできます。SPI は、そのようなソ

ソフトウェアの中断後は常に無効にし、次のトランザクションが開始する前に再度有効化する必要があります。

送信が有効な場合、シーケンスは開始し、マスタ側の TxFIFO に何らかのデータが存在する間、続行します。TxFIFO がエンプティになるまで、マスタによってクロック信号が永続的に供給され、その後停止して追加のデータを待ちます。

受信専用モード、半二重 (COMM[1:0] = 11、HDDIR = 0)、または単方向 (COMM[1:0] = 10) モードでは、SPI が有効化され、CSTART ビットによってトランザクションがリリースされると、マスタによってシーケンスが開始されます。マスタによってクロック信号が供給されますが、この信号はマスタが SPI または受信専用モードを無効/サスペンド状態にするまで停止しません。マスタは、クロック信号が停止するまでデータフレームを受信し続けます。ソフトウェア制御によって SPI_CR1 レジスタの CSUSP ビットに 1 を書き込むことで、または MASRX=1 で RxFIFO がフルになったときに自動的に、もしくはこの信号が適用されている場合は RDY ステータス時に、受信がサスペンドされます ([セクション 52.4.8](#) を参照)。また、TSIZE にプログラムされたフレームの数が完了した時にも、受信は自動的に停止します。

マスタのレシーブ専用モードを無効にするには、最初に SPI をサスペンドする必要があります。SPI がサスペンドされると、現在のフレームが完了してから、設定を変更します。

注意 : マスタモードで、受信が中断せずに進行しているときに SPE ビットがクリアされると、現在のフレームを完了することなくクロックが停止し、RxFIFO が一掃されます。

マスタはあらゆるトランザクションを連続モードで供給できる (SCK 信号は連続的) 一方で、データフローおよびその内容をいつでも処理できるスレーブ機能を優先する必要があります。スレーブに RDY 信号オプション機能がある場合、マスタは通信の流れを制御するために、スレーブによって発行された RDY 信号を監視できます。RDY ピンを使用しない場合、スレーブは常にマスタと通信できる状態であるとみなされます。

必要に応じて、マスタは通信速度を下げ、より低速のクロックか、または MIDI[3:0] ビットの設定によって十分な遅延を含む分離したフレームやデータセッションを供給する必要があります。あるいは、スレーブがデータの準備のために十分な余裕を持てるように、トランザクションの開始を延期する MSS[1:0] の設定によって初期遅延時間を提供する必要があります。スレーブからのデータは、たとえスレーブがそれらのデータを時間内に適切に準備できない場合でも、常にマスタによってトランザクション処理されることに注意してください。スレーブが DMA を使用することが望ましく、特に、データフレームが短く、FIFO がバイトでアクセスされ、そして SPI バスが高速の場合に有効です。

スレーブのトランスミッターノードから SPI 通信のフローに何らかのソフトウェア制御を加えるためには、SPI_UDRDR (SPI アンダーランデータレジスタ) に書き込んだ特定の値を使用することができます。スレーブ側で、TxFIFO が空になると、この値が次のデータとして自動的に送出され、マスタレシーバ側のソフトウェアで解釈されます (単に読み捨てられるか、ソフトウェアによってマスタレシーバをサスペンドするために XOFF のようなコマンドとして解釈されます)。

マルチスレーブのスター接続形態では、一度に有効にできるのは出力データの 1 つのスレーブのみです。マスタとの通信用に選択されたスレーブは、マスタとの通信を開始する前に、SS 入力のアクティブレベルへの変化を検出する必要があります。単一のスレーブシステムでは SS によってスレーブを制御する必要はありませんが、スレーブを各データ転送シーケンスの開始と同期させるために、ここにもパルスを提供することをお勧めします。SS は、ソフトウェアとハードウェアの両方で管理できます ([セクション 52.4.7](#) を参照)。

52.4.13 SPI の無効化

SPI を無効にするには、本項に記載されている無効化手順に従ってください。

マスタモードでは、この手順はペリフェラルクロックが停止し、システムが低電力モードに入る前に行うことが重要です。そうしないと、進行中のトランザクションが破壊されることがあります。

スレーブモードでは、**spi_pclk** および **spi_ker_ck** クロックが停止したとき、通信の終了またはデータサービスリクエスト状態に至るまで、SPI 通信は中断することなく続行できます。**spi_pclk** は一般にシステムを STOP モードに設定することによって停止できます。詳細については、RCC のセクションを参照してください。

全二重または送信専用モードでは、マスタは、送信するデータの供給を停止した時点でいかなるトランザクションも終了することができます。この場合、クロックは最後のデータトランザクション後に停止します。最後のデータフレームが送信されるのを待つために、TXC フラグをポーリングする（または EOTIE = 1 によって割込みを有効にする）ことができます。

マスタが受信専用モードになっている場合、ペリフェラルを停止するためには、CSUSP ビットを 1 にセットして、まず SPI 通信をサスペンドする必要があります。

SPI がサスペンドされたとき、受信されているが、まだ読み出されていないデータは RxFIFO に格納されたままになっています。

SPI は、そのようなソフトウェアの中断後は常に無効にし、内部ステートマシンを適切に再開する必要があります。

SPI が無効化されると、RxFIFO が一掃されます。未読のデータが失われないようにするためには、SPI を無効化するとき、残っているすべてのデータ（SPI_SR レジスタの RXP、RXWNE および RXPLVL フィールドに示されています）を読み出して RxFIFO が空であることを確認する必要があります。

標準的な無効化手順は、送信セッションが（完全に）終わったかどうかをチェックするために、EOT および/または TXC ステータスをポーリングすることに基づきます。このチェックは、たとえば以下に示すように、進行中のトランザクションの終わりを識別する必要があるような特別な場合にも行うことができます。

- マスタが SS 信号を SPI に関連しない GPIO で処理する場合（マルチスレーブのスター接続形態の場合など）で、スレーブに適切な SS パルスの終わりを提供する場合、または
- 最後のデータフレームまたは CRC フレームのトランザクションがまだペリフェラルバスで進行している間に、DMA または FIFO からのトランザクションのストリームが完了した場合

TSIZE>0 の場合、EOT と TXC の信号は等しいため、EOT のポーリングは、バス動作の終了を確認するために任意の SPI 通信モードで信頼できます。TSIZE = 0 の場合、適用された SPI モードとデータフローの終了に従って TXC、SUSP、または FIFO の占有フラグを確認する必要があります。

マスタモードでの正しい無効化手順を以下に示します（受信専用モードが使用されている場合を除く）。

1. TXC = 1 および/または EOT = 1（送信するデータがなくなり、最後のデータフレームが送信される）まで待ちます。CRC が使用されている場合、ブロックの最後のデータが処理された後に CRC が自動的に送信されます。この場合、CRC フレームが完了した時に TXC/EOT がセットされます。送信がサスペンドされる場合は、ソフトウェアは CSTART ビットがクリアされるまで待つ必要があります。
2. すべての RxFIFO データを読み出します（RXWNE = 0 および RXPLVL = 00 になるまで）。
3. SPI を無効にします（SPE = 0）。

マスタ受信専用モードの場合の正しい無効化手順を以下に示します。

1. EOT まで待つか、または SPI をサスペンド（CSUSP = 1）することによって受信フローを中断します。

- 受信フローがサスペンドされたら、SUSP = 1（最後のデータフレームが処理される）まで待ちます。
- すべての RxFIFO データを読み出します（RXWNE = 0 および RXPLVL = 00 になるまで）。
- SPI を無効にします（SPE = 0）。

スレーブモードでは、SPI を無効にすると、処理中のデータが失われます。

I/O の制御

SPI が無効化されるとすぐに、AFCNTR 設定に応じて関連する有効化された AF 出力をデバイスによって駆動できます。アクティブな出力制御が適用されていて（AFCNTR = 1）、SPI がちょうど無効化（SPE = 0）されたときに、有効化された出力は SPI 制御信号と関連付けられ（マスタの SS と SCK、スレーブの RDY）、直ちに非アクティブレベルにトグルすることができます（それぞれマスタの SSIOP および CPOL の設定、およびスレーブの RDIOP に応じて）。データライン出力（マスタの場合は MOSI、スレーブの場合は MISO）では、代わりに実際の TxFIFO の内容によって、直ちにレベルを変更することができます。これには、バス上で最後にトランザクションしたビットの値が、場合によっては無効になったり、保証できなくなったりするという影響があります。必要な場合は、ユーザはデータラインでの適切なデータホールド時間について注意する必要があり、最後のデータトランザクション完了後すぐに無効化された、最終的な高速 SPI を避けてください。

注： 最新ビットの安定性がサンプリングクロックエッジを通じて設計保証されているにもかかわらず、一部のデバイスでは、サンプリング中にこのデータビットの安定性の間隔を均等に拡張する必要がある場合があります。これは、たとえば EOT イベント発生と SPI 無効化操作の間に、ソフトウェアディレイをわずかに挿入することで実行できます。

52.4.14 データパッキング

ユーザの観点から、互いに重なり合う以下の 2 つのデータパッキングの方法があります。

- データを TxFIFO に書き込む、または RxFIFO から読み出すときのアクセスタイプ
データサイズが、SPI_TXDR または SPI_RXDR レジスタで実行されるアクセス単位より倍率が小さい場合、複数のデータを 1 回のアクセスによって効率的にプッシュまたはフェッチすることができます。
- 1 回のソフトウェアサービス中に扱われるデータの数
データをフレームごとに別々に扱う代わりに、データをパケットにグループ化し、もっぱらデータパケットの内容全体で FIFO サービスを累積するほうが便利です。ユーザは FIFO 閾値の設定によってパケットを定義できます。そうして、すべての FIFO 占有イベントはこの閾値レベルに関連し、要求されるサービスは割り込みやウェイクアップ機能による適切なフラグによって通知されます。

データフレームサイズが 1 バイト（8 ビット以下）におさまる場合、SPI_RXDR/SPI_TXDR レジスタで 16 ビットまたは 32 ビットの読出しまたは書き込みアクセスが実行されたときに、自動的にデータパッキングが使用されます。この場合、多重データフレームパターンが並行して処理されます。最初に、SPI はアクセスしたワードの LSB に保管されたパターンを使用して動作します。次に、MSB に保管された残りデータを使用して動作します。

[図 727](#) に、全機能がセットされたインスタンスでの、データパッキングモードのシーケンス処理の例を示します。そこでは DSIZE[4:0] が 4 ビットに設定され、トランスミッタの SPI_TXDR レジスタへの 16 ビットまたは 32 ビットの 1 回のアクセスの後、2 つまたは 4 つのデータフレームが TxFIFO に書き込まれます。データフレームサイズが 9 ビットと 16 ビットの間である場合、32 ビットアクセスが実行されたとき、自動的にデータパッキングが使用されます。最下位ハーフワードが最初に使用されます（LSBFRST の値にかかわらず）。

このシーケンスでは、RxFIFO 閾値がフレームに設定されていた（データはパックされずにフレームベースで読み出される）場合、レシーバで 2 または 4 つの RXP イベントが生成される可能性があります。

ます。あるいは、SPI_CFG1 レジスタの FTHLV[3:0] フィールドが、複数フレームをパックモードで読み出す (16 ビットまたは 32 ビット読み出しアクセス) 設定にプログラムされていた場合は、1 つの RXP イベントが生成されます。

データは [図 726](#) に従って配置されています。有効なビットだけがバスで実行されます。未使用のビットはトランスミッタでは考慮されず、レシーバではゼロで埋められます。

短いデータフレーム (8 ビット未満または 16 ビット未満) を、より大きいデータアクセスモード (16 ビットまたは 32 ビット) とともに使用する場合、FTHLV の値はフレームの数/データアクセスの倍数としてプログラムする必要があります (最大 8 ビットのフレームに 32 ビットアクセスを使用する場合は 4 の倍数、あるいは、最大 8 ビットのフレームに 16 ビットアクセスを使用する場合、または最大 16 ビットのフレームに 32 ビットアクセスの場合は 2 の倍数)。

RxFIFO 閾値の設定は、次の読み出しアクセスサイズより必ず大きくなければなりません (最低でも同じでなければなりません)。そうしないと、誤った余分のデータが読み出されるからです。

設定したデータサイズより小さい FIFO データアクセスは禁止されます。必ず最低でも 1 つの完了したデータフレームにアクセスする必要があります。

FTHLV ビットでセットされた閾値未満の不完全なデータパケットが FIFO で利用可能になっている場合は、特定の問題が現れます。

この問題を解決するには、2 つの方法があります。

A. TSIZE フィールドを使用しない方法

トランスミッタ側では、SPI_TXDR への 8 ビット/16 ビットアクセスによって任意の奇数シーケンスの最後のデータフレームを書き込むだけで十分です。

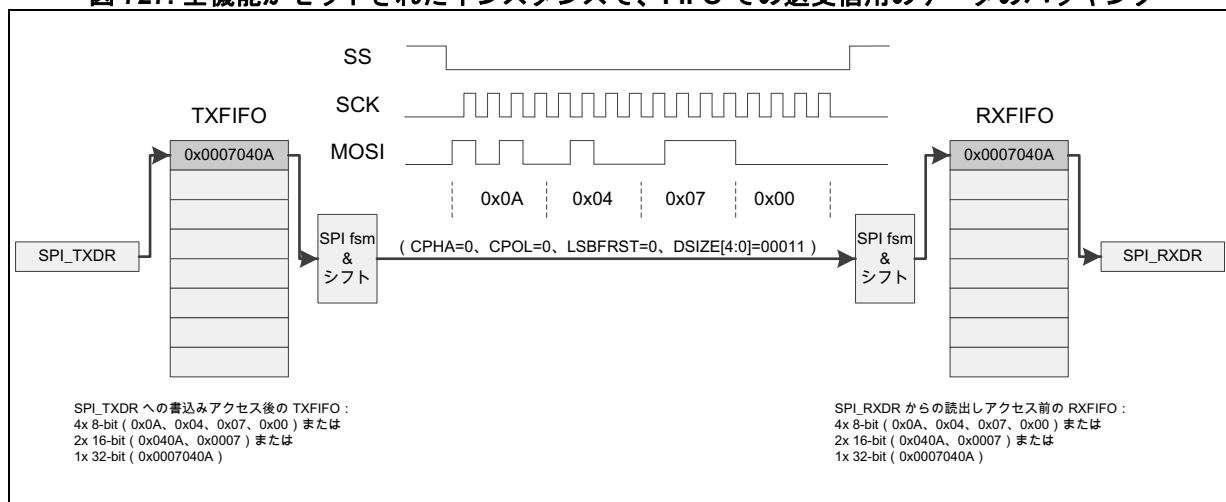
レシーバ側では、残りのデータは任意のアクセスによって読み出すことができます。余分なデータの読み出しはゼロで埋められます。受信データが RxFIFO に残っているかどうかを検出するためには、RXWNE および RXPLVL のポーリングが使用できます (ポーリングを検出するためにシステムレベルではタイムアウトが使用されることがあります)。

B. TSIZE フィールドを使用する方法

トランスミッタ側では、EOT イベントが発生するとトランザクションはマスタによって停止されます。

受信側では、EOT がセットされた時、RXP フラグはセットされません。受信するデータの数 (TSIZE) がパケットサイズの倍数でなかった場合、残っているデータの数が SPI_SR レジスタの RXWNE および RXPLVL フィールドによって示されます。残りのデータは任意のアクセスによって読み出すことができます。余分な読み出しはゼロで埋められます。

図 727. 全機能がセットされたインスタンスで、FIFO での送受信のデータのパッキング



1. DSIZE[4:0] は 4 ビットに設定され、データは右詰め、有効なビットがバス上でのみ実行され、その順序は LSBFRST に依存し、セットされている場合はすべてのデータフレームでその順序が逆になります。

52.4.15 DMA（ダイレクトメモリアクセス）を使用する通信

最高速度で動作し、オーバーランを回避するために必要なデータレジスタの読み出し／書き込み処理を容易にするために、SPI は簡単なリクエスト／確認応答プロトコルを実現する DMA 機能を備えています。

SPI_CFG1 レジスタの TXDMAEN または RXDMAEN 有効ビットをセットすると、DMA アクセスがリクエストされます。Tx バッファと Rx バッファには、定義されたパケットのサービスを満たすために、別々のリクエストを発行する必要があります。

- 送信では、TXP が 1 にセットされるたびに一連の DMA リクエストがトリガされます。その後、DMA は SPI_TXDR レジスタに一連の書き込みを行います。
- 受信では、RXP が 1 にセットされるたびに一連の DMA リクエストがトリガされます。その後、DMA は SPI_RXDR レジスタから一連の読み出しを行います。トランザクションの最後に EOT がセットされ、最後のデータパケットが不完全だった場合、RXWNE および RXPLVL[1:0] の設定に従って DMA リクエストが自動的にアクティブにされ、残りのデータが読み出されます。

SPI が受信専用モードにプログラムされている場合、UDR はセットされません。

SPI が送信モードにプログラムされている場合、送信データが利用できないことがあるので、スレーブ側で最終的に TXP および UDR がセットされることがあります。この場合、UDR 管理の選択によっていくつかのデータが TX ラインに送信されます。

SPI が単方向モードで使用されている場合、相補的な未使用チャネルを無効にしたまま、適切な DMA チャネルのみを有効にする必要があります。

SPI が送信専用モードにプログラムされている場合、RXP および OVR はセットされません。

SPI が全二重モードにプログラムされている場合、受信データが読み出されないため、最終的に RXP および OVR がセットされます。

送信モードで、DMA またはユーザがすべての送信データを書き込んだとき (SPI_SR レジスタの TXTF フラグがセットされます)、EOT (TSIZE = 0 の場合は TXC) フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、またはマスタモードで **spi_clk** を無効にする前にこの操作を行う必要があります。ソフトウェアは、まず EOT = 1 および/または TXC = 1 になるまで待つ必要があります。

DMA を使用して通信を開始する場合、DMA チャンネルの管理によるエラーイベントを防ぐために、以下の手順を順に実行する必要があります。

1. DMA Rx が使用される場合、SPI_CFG1 レジスタの RXDMAEN ビットの DMA Rx バッファを有効にします。
2. DMA が使用される場合、DMA レジスタにて Tx および Rx の DMA リクエストを有効にします。
3. DMA Tx が使用される場合、SPI_CFG1 レジスタの TXDMAEN ビットにて DMA Tx バッファを有効にします。
4. SPE ビットをセットして、SPI を有効にします。

通信を終了するには、以下の手順を順に実行する必要があります。

1. DMA が発行されていた場合、DMA レジスタにて Tx および Rx の DMA リクエストを無効にします。
2. SPI 無効化手順に従って SPI を無効にします。
3. DMA Tx および（または）DMA Rx が使用されている場合、SPI_CFG1 レジスタの TXDMAEN ビットおよび RXDMAEN ビットをクリアすることにより、DMA Tx バッファおよび Rx バッファを無効にします。

DMA によるデータパッキング

DMA (SPI_CFG1 レジスタの TXDMAEN および RXDMAEN をセット) で転送が管理される場合、パッキングモードは SPI TX および SPI RX DMA チャンネルで設定される PSIZE 値によって、自動的に有効/無効になります。

DMA チャンネルの PSIZE 値が 16 ビットと等しい場合、SPI データサイズは 8 ビット以下で、パッキングモードは有効になります。同様に、DMA チャンネルの PSIZE 値が 32 ビットと等しい場合、SPI データサイズは 16 ビット以下で、パッキングモードは有効になります。その後、DMA は SPI_TXDR レジスタへの書き込み動作を管理します。

フレームサイズが小さいとき、データパッキングモードが使用されていて、送信データ数が DMA データサイズ（16 ビットまたは 32 ビット）の倍数になっていない場合、DMA は TSIZE フィールドの設定に従って自動的に転送を完了します。

あるいは、シングル/アンパックモードで最後のデータフレームをソフトウェアで書き込むこともできます。

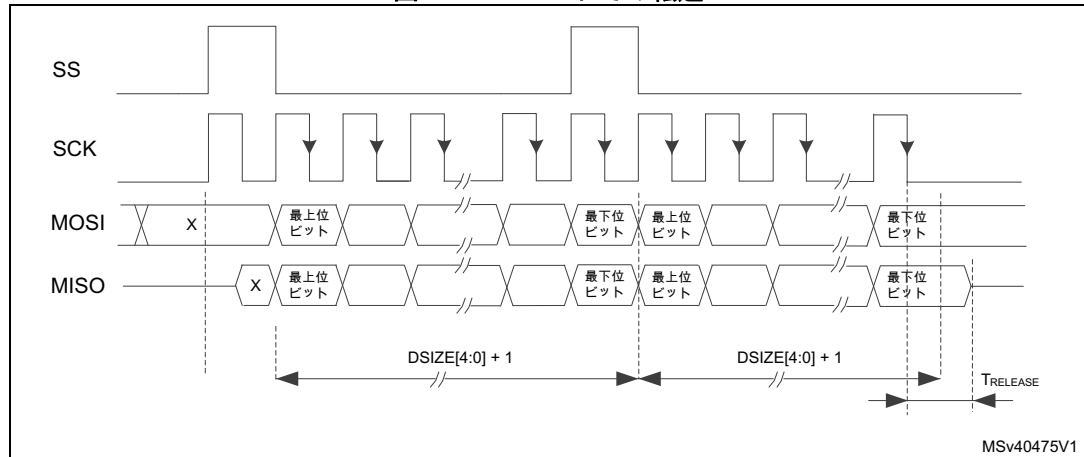
設定したデータサイズより小さい DMA データアクセスを設定することは禁止されます。必ず最低でも 1 つの完了したデータフレームにアクセスする必要があります。

52.5 SPI の特定モードおよび制御

52.5.1 TI モード

SPI_CFG2 レジスタの特定の SP[2:0] ビットフィールド設定で、SPI は TI プロトコルと一致するように設定できます。SPI が TI モードの設定になっているときは、SCK および SS 信号の極性、位相、フロー、およびビット順序は固定されているので、CPOL、CPHA、LSBFRST、SSOM、SSOE、SSIOP、SSM、RDIOP、RDIOM、MSSI、および MIDI の設定は必要ありません。SS 信号は、図 728 に示すように、LSB データビット上のパルスによってプロトコルを同期させます。

図 728. TI モードでの転送



スレーブモードでは、現在のトランザクションが終了した時点で、MISO ピンでのスレーブ出力がハイインピーダンスになるタイミングを定義するためにクロックジェネレータが使用されます。マスタのボーレート設定 (SPI_CFG1 の MBR[2:0]) が適用され、このタイミングを非常に柔軟に決定するために任意のボーレートを使用できます。MISO 信号がハイインピーダンス (TRELEASE) になるまでの遅延も、内部再同期によって変わります。これには、ジェネレータに供給されるクロック信号がさらに 2 ~ 4 周期必要です。この値は次式で求められます：

$$\frac{T_{\text{baud}}}{2} + 2 \times T_{\text{spi_ker_ck}} \leq T_{\text{release}} \leq \frac{T_{\text{baud}}}{2} + 4 \times T_{\text{spi_ker_ck}}$$

スレーブがデータトランザクション中に SS パルスの誤配置を検出すると、TIFRE フラグがセットされます。

52.5.2 SPI エラーフラグ

次のいずれかのエラーフラグがセットされて、対応する割込み有効ビットをセットすることによって割込みが有効になっている場合、SPI 割込みが生成されます。

オーバーランフラグ (OVR)

マスタまたはスレーブでデータを受信し、RxFIFO に受信データを格納するための十分なスペースがない場合に、オーバーラン状態が発生します。これは、ソフトウェアまたは DMA が RxFIFO に格納された前の受信データを読み出すために必要な時間が十分になかった場合に発生する可能性があります。

オーバーラン状態が発生すると、OVR フラグがセットされ、RxFIFO にある前の値を新しく受信した値で上書きすることはありません。新しく受信した値は破棄され、それ以降に送信されるすべてのデータは失われます。OVRIE ビットがセットされている場合、OVR フラグは割込みをトリガします。OVR ビットのクリアは、SPI_IFCR の OVRC ビットに 1 を書き込むことによって行われます。次のオーバーランイベントを防ぐには、ソフトウェアの読み出しによって RxFIFO が空になってからビットのクリアをすべきです。RxFIFO のスペースを最大限解放することをお勧めします。これは、RXP フラグの指示に基づいて、使用可能なすべてのデータパケットを読み出すことを意味します。

マスタモードでは、自動通信サスペンド (MASRX ビット) によって RxFIFO オーバーランを防ぐことができます。

アンダーランフラグ (UDR)

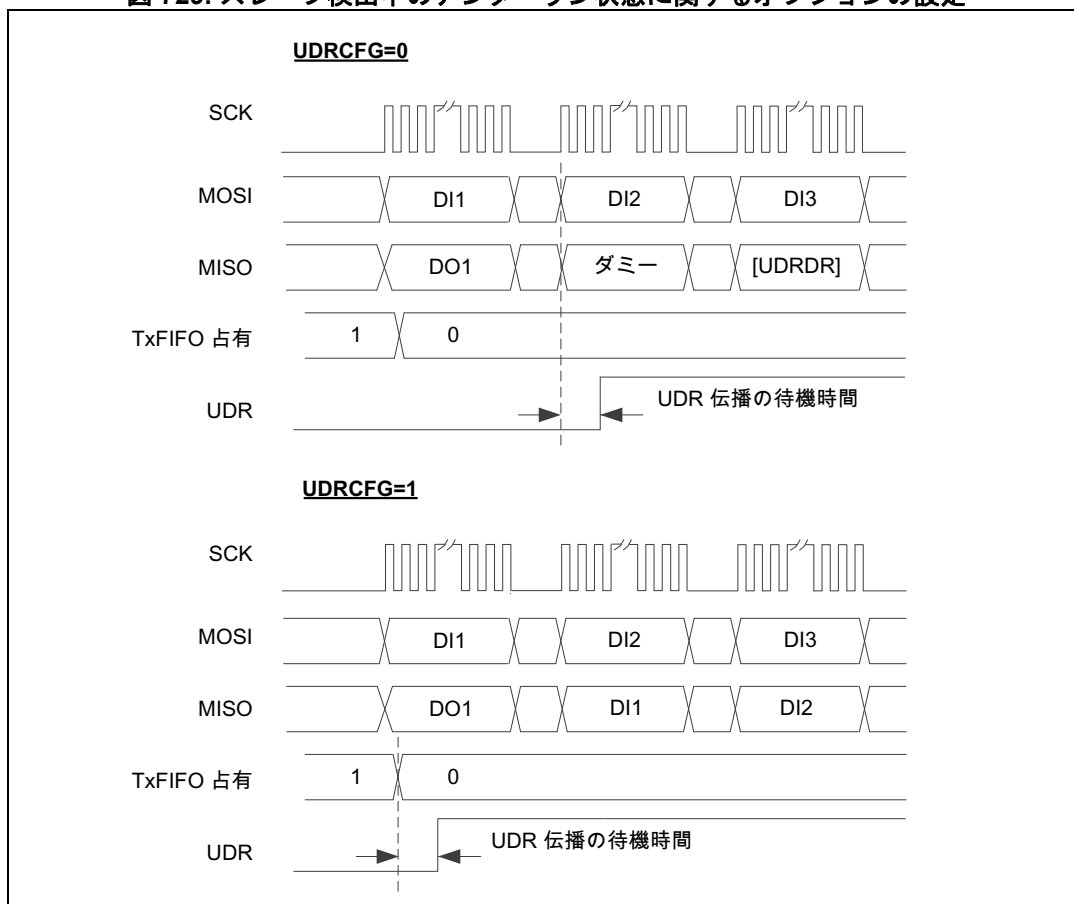
スレーブ送信モードで、一般的にスレーブの TxFIFO に送信できるデータがなかった場合、アンダーラン状態がハードウェアによって内部的にキャプチャされます。その後、UDR フラグの設定はハードウェアによってステータスレジスタに伝えられます (下記の注を参照)。UDRIE ビットがセットされている場合、UDR は割込みをトリガします。

アンダーラン検出ロジックとシステム動作は UDRCFG ビットに依存します。アンダーランがスレーブに検出されると、ユーザによって UDRDR レジスタに格納された一定のパターン、もしくは前にマスタから受信したデータを提供できます。最初の設定 (UDRCFG = 0) が適用されると、TxFIFO が空の時に、マスタが新しいデータフレームの通信を開始したときにいつでもアンダーラン状態が評価されます。次に、最後の有効なデータと、UDRDR レジスタで定義された一定パターンの間に、必ず 1 つの追加のダミー (不測の) データが挿入されます (図 729 を参照)。この 2 番目の設定 (UDRCFG=1) はサーキュラ接続形態 (図 720 を参照) において使用できます。マスタが通信を開始したときに TxFIFO が空でないものとして、アンダーラン状態は次のデータフローで FIFO が空になったときにだけ評価されます。TxFIFO からの有効なデータは、最後の受信データによって直ちに付加されます。

ソフトウェアによって UDR フラグがクリアされると、標準の送信が再有効化され、このクリア情報はハードウェアによって SPI ロジックに伝えられます。UDR フラグをクリアする前に、次のアンダーラン状態発生のカプチャを防ぐために、いくらかのデータを TxFIFO に書き込む必要があります。

スレーブによってトランザクションされるデータは、特に TxFIFO が空で、アンダーラン状態がまだキャプチャされていないか、クリアされたばかりの時に、トランザクションが開始または続行されたときに、予測不可能です。一般的にこれは SPI が有効になったばかりのときか、あるいは定義されたサイズでトランザクションが始まったばかりのときの場合です。さらに、この場合、データのトランザクションが開始される、あまりに直前にスレーブソフトウェアが空の TxFIFO に最初のデータを書き込んだとき、最初の数ビットは破壊される可能性があります (データを TxFIFO に伝播するのに APB クロックの数サイクルがかかります)。

図 729. スレーブ検出中のアンダーラン状態に関するオプションの設定



注： UDR イベントをハードウェアで伝播するのにバス上で追加のトラフィックが必要になります。これには、イベント発生後（ハードウェアによるアンダーランのキャプチャ、およびソフトウェアによる UDR フラグのクリアの両方）、必ずいくつか追加の SPI クロック周期を要します。ソフトウェアによる UDR フラグのクリアがデータフレームトランザクションの終わり近くに適用された場合、または SCK ラインがフレーム間でアイドル状態の場合、次の追加のアンダーラン・パターンが、最初にスレーブによって、TxFIFO からの有効なデータが再度トランザクションされる前に送信されます。ユーザはアンダーランロジックを再開するセッション間で SPI 無効化/有効化操作により、これを防ぐことができます。これにより、次のセッションを有効なデータで開始できます。

モードフォールト (MODF)

モードフォールトは、マスタデバイスが内部 SS 信号（SS ハードウェアモードでは SS ピン、SS ソフトウェアモードでは SSI ビット）をローレベルにプルダウンしたときに発生します。これは、自動的に SPI インタフェースに次のような影響を与えます。

- MODF ビットがセットされ、MODFIE ビットがセットされている場合は SPI 割込みがトリガされます。
- SPE ビットは、MODF ビットがセットされるまで、強制的にゼロになります。これにより、SPI が無効になり、MODF 割込みリクエスト（有効な場合）を除くすべてのペリフェラル出力がブロックされます。
- MASTER ビットがクリアされ、デバイスは強制的にスレーブモードになります。

MODF は、SPI_IFCR レジスタの MODFC ビットに 1 を書き込むことによってクリアされます。

複数の MCU で構成されるシステムでスレーブ間の競合を避けるには、SPE ビットをセットすることによって SPI を再有効化する前に、SS ピンを非アクティブレベルにプルする必要があります。

安全のため、MODF ビットがセットされている間、ハードウェアは SPE ビットのセットを許可しません。スレーブデバイスでは、MODF ビットはセットできません。ただし、前回のマルチマスタ競合の結果としてセットする場合は例外です。

マルチマスタシステムにおいてマスタがバスを支配するときの正しいソフトウェア手順は次のとおりです。

- SSOE = 0 のとき、マスタモードに切り替えます（別のマスタがバスを占有していると、潜在的競合が起きる可能性があります。その場合、MODF が発生し、次のノードがマスタノードに切り替わることを防ぎます）。
- 別のマスタの SS 制御専用の GPIO ピンをアクティブレベルにします。
- データのトランザクションを実行します。
- 別のマスタの SS 制御専用の GPIO ピンを非アクティブレベルにします。
- スレーブモードに切り替えて戻します。

CRC エラー (CRCE)

このフラグを使用して、SPI_CFG1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。TSIZE によって定義された最後のデータが受信された後、シフトレジスタに受信された値が、レシーバの SPI_RXCRC の値と一致しなかった場合、SPI_SR レジスタの CRCE フラグがセットされます。CRCEIE ビットがセットされている場合、CRCE フラグは割込みをトリガします。CRCE ビットのクリアは、SPI_IFCR の CRCEC ビットに 1 を書き込むことによって行われます。

TI モードフレームフォーマットエラー (TIFRE)

SPI がスレーブモードで動作し、かつ TI モードプロトコルに準拠した設定となっている場合、通信の実行中に SS パルスが発生すると、TI モードフレームフォーマットエラーが検出されます。このエラー発生すると、SPI_SR レジスタの TIFRE フラグがセットされます。エラー発生時には SPI は無効にされず、この SS パルスは無視されます。SPI は次の SS パルスを待ってから新規の転送を開始します。このエラーの検出によりいくつかのデータフレームが失われるため、データは破壊される可能性があります。

TIFRE フラグは、SPI_IFCR の TIFREC ビットに 1 を書き込むことによってクリアされます。TIFREIE ビットがセットされている場合、SS エラー検出時に割込みが生成されます。データの一貫性がもはや保証されなくなるため、ソフトウェアによってマスタとスレーブ間で通信を再起動するべきです。

52.5.3 CRC 計算

2 つの別々の 33 ビットの、または 2 つの別々の 17 ビットの CRC 計算機が、送信データおよび受信データの信頼性をチェックするために実装されています。全機能がセットされたインスタンスで、SPI では CRC 多項式の長さが提供されており、その長さはデータの最大サイズが 32 ビットの場合は 5 ビットから 33 ビットまで、データサイズが 16 ビットに限られるペリフェラルインスタンスに対しては 9 ビットから 17 ビットまでです。機能が制限されているインスタンスでは、データサイズが 16 ビットに制限されている場合、CRC 多項式の長さは 9 または 17 のどちらかにセットできます。また、データサイズが 32 ビットに拡張されると、オプションで 33 にセットできます。多項式の長さは CRCPOLY レジスタに格納されている値の最上位ビットによって定義されます。その長さは DSIZE フィールドで定義されているデータフレーム長より大きくセットする必要があります。最大のデータサイズが適用されているときは、多項式ストリングのサイズを常にデータより大きく保ちながらその最上位ビットを定義するために、さらに CRC33_17 ビットをセットする必要があります。SPI_CFG1 の CRCSIZE フィールドでは、CRC 計算レジスタからいくつかの最上位ビットが CRC フレームとしてトランザクションされ、比較されるのを定義します。これはデータフレーム長とは独立に定義され

ますが、データフレームサイズと同じか、その倍数でなければなりません。ただし、サイズがインスタンスの最大データサイズを超えることはできません。

CRC 計算機能を最大限活用するには、多項式の長さ設定を CRC パターンサイズに一致させる必要があります。サイズが多項式の長さより大きくセットされている場合は、計算時の未使用ビットがトランザクションされ、CRC パターンの最後がすべてゼロになることが予想されます。

CRC の原理

CRC 計算は、SPI が有効 (SPE = 1) になる前に、SPI_CFG1 レジスタの CRCEN ビットをセットすることによって有効になります。CRC 値は、CRCPOLY レジスタと CRC33_17 ビットによって定義された CRC 多項式を使用して計算されます。SPI が有効になっているとき、CRC 多項式は変更できませんが、バス上にトラフィックがないときに限られます。

CRC 計算は、SPI_CR1 レジスタの CPHA ビットと CPOL ビットによって定義されたサンプリングクロックエッジで、ビットごとに行われます。計算された CRC 値は、SPI_CR2 レジスタによって定義されたデータブロックの最後のみ自動的にチェックされます。

受信データをもとに内部で計算された CRC 値とトランスミッタから受信した CRC 値の間に不一致が検出された場合、データ破壊エラーを示すために CRCE フラグがセットされます。CRC を処理する正しい手順は、SPI の設定および選択された転送管理によって変わります。

CRC 転送管理

通信が開始され、最後のデータフレームが SPI_DR レジスタに送信または受信されるまで正常に続きます。

転送の長さは TSIZE によって定義される必要があります。要求されたデータ数がトランザクション処理されると、TXCRC が送信され、ラインで受信されたデータが RXCRC 値と比較されます。

CRCSIZE の設定にかかわらず、TSIZE を全機能がセットされたインスタンスで 0xFFFF にセットしたり、CRC が有効な場合に機能が制限されたインスタンスで 0x3FF の値にセットしたりすることはできません。

送信では、CRC のトランザクション処理中、CRC 計算は停止し、TXCRC は CRCSIZE フィールドの値に等しい長さのフレームで送信されます。

受信では、必要な数のデータがトランザクションされると、RXCRC は同じく停止します。その後、RXCRC レジスタの内容と比較される情報が、CRCSIZE 値と等しい長さのフレームで受信されます。

CRC フレームの受信が完了すると、受信した CRC 値と SPI_RXCRC レジスタで計算された値を比較する自動チェックが行われます。ソフトウェアは、SPI_SR レジスタの CRCE フラグをチェックして、データ転送の内容が破壊されているか否かを判断する必要があります。ソフトウェアは、CRCEC に 1 を書き込んで CRCE フラグをクリアします。

ユーザは冗長な CRC 情報を一掃する面倒をみる必要はなく、それは自動的行われます。

SPI_TXCRC および SPI_RXCRC の値のリセット

CRC フェーズのあとで新しいデータがサンプリングされた時、SPI_TXCRC 値および SPI_RXCRC 値は自動的に初期化されます。これにより、データを中断することなく転送するために、DMA サーキュラモードが使用可能になります (いくつかのデータブロックは中間の CRC チェックフェーズによってカバーされます)。レシーバおよびトランスミッタ用の初期化パターンは、SPI_CR1 レジスタの TCRCINI および RCRCINI ビットの設定によってゼロまたはすべて 1 のどちらかに設定できます。

SPI が無効化されると、CRC の値はリセットされます。

52.6 SPI 低電力モード

表 555. 低電力モードが SPI に与える影響

モード	説明
SLEEP	影響はありません。SPI 割込みによって、デバイスは SLEEP モードを終了します。
STOP ⁽¹⁾	SPI レジスタの内容は保たれます。
STANDBY ⁽¹⁾	SPI のインスタンスはこのモードのパワーダウン時に機能せず、STANDBY モードの終了後に再初期化する必要があります。

1. インスタンスごとの STOP モードからのウェイクアップのサポートと、STANDBY モードを使用できるかどうかについては、[セクション 52.3 : SPI の実装](#)を参照してください。
インスタンスが STOP モードで機能しない場合は、STOP モードに移行する前に無効化する必要があります。

52.7 SPI 割込み

[表 556](#) に、有効な場合は割込みイベント (spi_it) を生成できる SPI イベントの概要を示します。そのうちのいくつかは、さらに低電力モードからのウェイクアップ機能を備えています。それらの多くは、特定の割込み有効制御ビットを使用することによって、独立に有効化および無効化することができます。イベントに関連するフラグは、特定の方法でクリアされます。イベントフラグについての詳細は、SPI レジスタの説明を参照してください。SPI が無効化されると、割込みサービスへの伝播を防ぐために、MODF 割込みリクエストを除く保留中の割込みリクエストがすべてブロックされます。

表 556. SPI のウェイクアップおよび割り込みリクエスト

割り込み ベクタ	割り込みイベント	イベント フラグ	イネーブル 制御 ビット	イベントのクリア方法	の終了 STOP および STANDBY モードの機能 ⁽¹⁾⁽²⁾
SPI	TxFIFO ロード準備済み (1 つのデータパケット - FIFO 閾値のスペースが利用可能)	TXP	TXPIE	TXP は、TxFIFO の空き容量が FTHLV より少なくなった時にハードウェアによってクリアされます。	可能
	データを RxFIFO に受信 (1 つのデータパケットが利用可能 - FIFO 閾値)	RXP	RXPIE	RXP は、RxFIFO に入っているサンプルが FTHLV より少なくなった時にハードウェアによってクリアされます。	可能
	TXP と RXP の両方がアクティブ	DXP	DXPIE	TXP または RXP がクリアされた時	可能
	送信の転送フル	TXTF	TXTFIE	TXTC に 1 を書き込む	不可
	アンダーラン	UDR	UDRIE	UDRC に 1 を書き込む	可能
	オーバーラン	OVR	OVRIE	OVRC に 1 を書き込む	可能
	CRC エラー	CRCE	CRCEIE	CRCEC に 1 を書き込む	可能
	TI フレームフォーマットエラー	TIFRE	TIFREIE	TIFREC に 1 を書き込む	不可
	モードフォールト	MODF	MODFIE	MODFC に 1 を書き込む	不可
	転送の終了 (転送の全シーケンス完了 - TSIZE 値に基づいて)	EOT	EOTIE	EOTC に 1 を書き込む	可能
	マスタモードサスペンド	SUSP		SUSPC に 1 を書き込む	可能
	TxFIFO 送信完了 (TxFIFO エンブティ)	TXC ⁽³⁾		TXC は、バス上で送信動作が開始された時にハードウェアによってクリアされます。	不可

- すべての割り込みイベントで、各インスタンスでシステムを SLEEP モードからウェイクアップできます。具体的な STOP および STANDBY モードを終了するためのインスタンスの機能について、詳しくは「動作モードに応じた機能」の表を参照してください。
- STANDBY モードを使用できるかどうかについては、[セクション 52.3 : SPI の実装](#)を参照してください。
- TXC フラグの動作は TSIZE 設定に依存します。TSIZE>0 の場合、フラグは EOTC のよるクリアを含め、完全に EOT に従います。

52.8 I²S の主な機能

- 全二重通信
- 単方向通信（トランスミッタまたはレシーバのみ）
- マスタまたはスレーブ動作
- 8 ビットのプログラム可能なリアプリスケール
- データ長は 16、24 または 32 ビット^(a)
- チャネル長は、マスタでは 16 または 32、スレーブでは任意の値
- プログラム可能なクロック極性
- 信頼性向上のためのエラーフラグ通知：アンダーラン、オーバーラン、およびフレームエラー
- 内蔵の RxFIFO および TxFIFO
- 以下の I²S プロトコルをサポート。
 - フィリップス I²S 規格
 - MSB 詰め規格（左詰め）
 - LSB 詰め規格（右詰め）
 - PCM 規格（ショートフレームおよびロングフレーム同期付き）
- データ順序はプログラム可能（LSb または MSb ファースト）
- 送受信の DMA 機能
- 外部オーディオコンポーネントを駆動するためのマスタクロックを出力可能。
 - $F_{MCK} = 256 \times F_{WS}$ （すべての I2S モード）
 - $F_{MCK} = 128 \times F_{WS}$ （すべての PCM モード）

注： F_{MCK} はマスタクロック周波数で、 F_{WS} はオーディオサンプリング周波数です。

52.9 I²S の機能詳細

52.9.1 I²S の概要

図 715 に示したブロック図は I2S モードにも適用されます。

SPI/I2S ブロックは I2SMOD ビットがセットされたとき、I2S/PCM モードで動作します。I2S パラメータ専用の設定のために専用のレジスタ（SPI_I2SCFGR）が用意されており、そこにはクロックジェネレータやシリアルリンクインタフェースが含まれています。

SPI/I2S がマスタモードに設定されているとき、I2S/PCM 機能は通信クロックを発生させるためにクロックジェネレータを使用します。このクロックジェネレータは、マスタクロック出力（MCK）のソースでもあります。

RxFIFO、TxFIFO、DMA、および割込み通知の部品といったリソースは SPI 機能と共用されています。低消費電力モード機能も I2S モードで利用可能です。[セクション 52.6 : SPI 低電力モード](#)および[セクション 52.10 : I²S 割込み](#)を参照してください。

a. 常に使用可能ではありません。24 および 32 ビットのデータ幅がサポートされているかどうかを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

52.9.2 SPI 機能とのピンの共用

I²S は SPI と、以下の 4 つの共通ピンを共用します。

- SDO : シリアルデータ出力 (MOSI ピンにマップされます) は、マスタでオーディオサンプルを送信し、スレーブでオーディオサンプルを受信します。 [セクション : 2385 ページのシリアルデータラインのスイッチング](#)を参照してください。
- SDI : シリアルデータ入力 (MISO ピンにマップされます) は、マスタでオーディオサンプルを受信し、スレーブでオーディオサンプルを送信します。 [セクション : 2385 ページのシリアルデータラインのスイッチング](#)を参照してください。
- WS : ワード選択 (SS ピンにマップされます) はフレーム同期用です。これはマスタモードでは出力として、スレーブモードでは入力として設定されます。
- CK : シリアルクロック (SCK ピンにマップされます) はシリアルビットクロックです。これはマスタモードでは出力として、スレーブモードでは入力として設定されます。

外部オーディオデバイスにマスタクロック出力が必要な場合、追加のピンを使用できます。

- MCK : マスタクロック (個別にマップされます) は、I2S がマスタモードに設定されるときに使用されます。

52.9.3 I2S/PCM モードで利用できるビットフィールド

I2S/PCM モードが選択されているとき (I2SMOD = '1')、いくつかのビットフィールドは無関係になり、I2S/PCM 機能の動作を保証するために特定の値に強制する必要があります。表 557 は、I2S/PCM モードで利用できるビットとフィールドの一覧を示し、どれを特定の値に強制する必要があるかを示します。

表 557. PCM/I2S モードで利用できるビットフィールド

レジスタ名	PCM/I2S モードで利用できるビットフィールド	他のビットフィールドでの制約
SPI/I2S 制御レジスタ 1 (SPI_CR1)	IOLOCK, CSUSP, CSTART, SPE	他のフィールドはリセット値にセット
SPI/I2S 制御レジスタ 2 (SPI_CR2)	-	リセット値にセット
SPI/I2S 設定レジスタ 1 (SPI_CFG1)	TXDMAEN, RXDMAEN, FTHLV	他のフィールドはリセット値にセット
SPI/I2S 設定レジスタ 2 (SPI_CFG2)	AFCNTR, LSBFRST, IOSWP	他のフィールドはリセット値にセット
SPI/I2S 割込み有効レジスタ (SPI_IER)	TIFREIE, OVRIE, UDRIE, TXPIE, RXPIE	
SPI/I2S ステータスレジスタ (SPI_SR)	SUSP, TIFRE, OVR, UDR, TXP, RXP	他のフラグは関係ない
SPI/I2S 割込み/ステータスフラグクリアレジスタ (SPI_IFCR)	SUSPC, TIFREC, OVRC, UDRC	他のフィールドはリセット値にセット
SPI/I2S 受信データレジスタ (SPI_RXDR)	レジスタ全体	-
SPI/I2S 多項式レジスタ (SPI_CRCPOLY)	-	リセット値にセット
SPI/I2S トランスミッタ CRC レジスタ (SPI_TXCRC)	-	
SPI/I2S レシーバ CRC レジスタ (SPI_RXCRC)	-	
SPI/I2S アンダーランデータレジスタ (SPI_UDRDR)	-	
SPI/I2S 設定レジスタ (SPI_I2SCFGR)	レジスタ全体	-

52.9.4 スレーブおよびマスタモード

SPI/I2S ブロックは、I²S および PCM プロトコルの両方に対してマスタとスレーブのモードをサポートしています。

マスタモードでは、CK、WS および MCK のすべての信号は出力にセットされます。

スレーブモードでは、CK および WS の両信号は入力にセットされます。MCK 信号はスレーブモードでは使用できません。

スレーブモードでの SPI/I2S ブロックの堅牢性を向上するために、ペリフェラルは各受信と送信を WS 信号に再同期させます。これは以下のことを意味します。

- フィリップス I²S 規格では、各データのシフトインまたはシフトアウトは、WS の各遷移の 1 ビットクロック後にトリガされます。
- I²S の MSB 詰め規格では、各データのシフトインまたはシフトアウトは、WS の遷移が検出されるとすぐにトリガされます。
- PCM の短い規格では、各データのシフトインまたはシフトアウトは、WS のアクティブエッジの 1 ビットクロック後にトリガされます。
- PCM の長い規格では、各データのシフトインまたはシフトアウトは、WS のアクティブエッジが検出されるとすぐにトリガされます。

注： この再同期メカニズムは I2S の LSB 詰め規格では使用できません。

SPI/I2S がスレーブモードに設定されているときは、カーネルクロックを供給する必要がないことに注意してください。

52.9.5 サポートされるオーディオプロトコル

I2S/PCM インタフェースは、SPI_I2SCFGR レジスタの I2SSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ標準をサポートします。

I²S プロトコルでは、オーディオデータは 2 つのチャネル（左チャネルと右チャネル）で時間多重化されています。WS 信号は、どちらのチャネルを左とし、どちらを右とする必要があるかを示すために使用されます。

I²S マスタモードでは、以下の 4 つのフレームフォーマットがサポートされています。

- 16 ビットチャネルにパックされた 16 ビットデータ
- 32 ビットチャネルにパックされた 16 ビットデータ
- 32 ビットチャネルにパックされた 24 ビットデータ^(a)
- 32 ビットチャネルにパックされた 32 ビットデータ^(a)

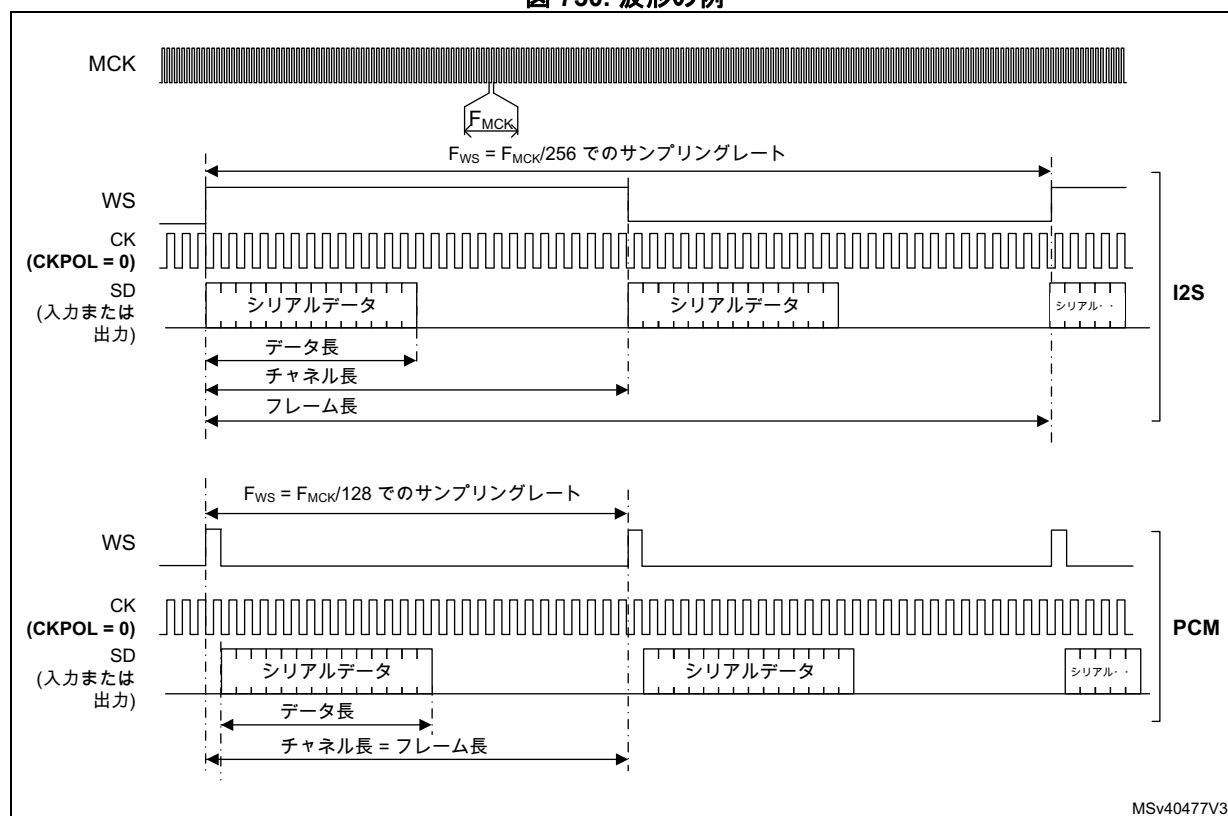
PCM マスタモードでは、以下の 3 つのフレームフォーマットがサポートされています。

- 16 ビットチャネルにパックされた 16 ビットデータ
- 32 ビットチャネルにパックされた 16 ビットデータ
- 32 ビットチャネルにパックされた 24 ビットデータ^(a)

以下の図に、本セクションで使用される、データ長、チャネル長、フレーム長の主な定義を示します。

a. 常に使用可能ではありません。24 および 32 ビットのデータ幅がサポートされているかどうかを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

図 730. 波形の例

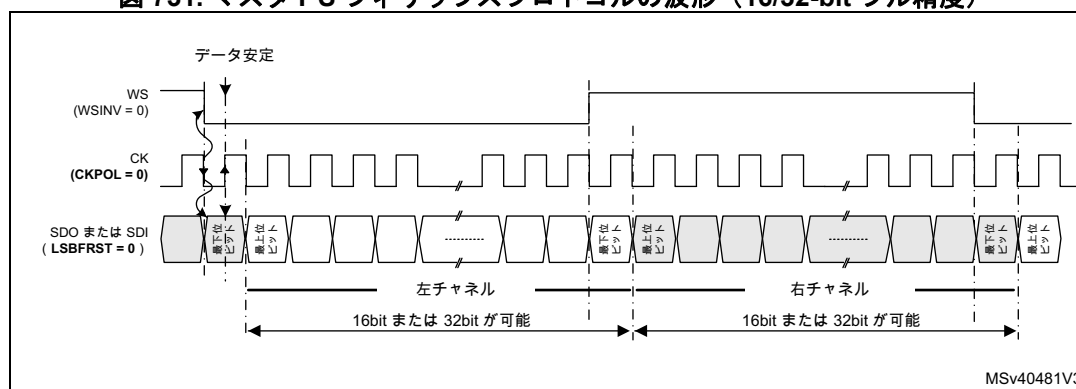


単純にするため、次の図では、SDI はシリアルデータ入力を表し、SDO はシリアルデータ出力を表します。シリアルデータラインの方向制御の詳細については、[セクション：シリアルデータラインのスワッピング](#)を参照してください。

フィリップス I²S 規格

I2SSTD を 0b00 にセットすることによってフィリップス I2S 規格を選択できます。この規格はマスタおよびスレーブのモードでサポートされています。

この規格では、WS 信号が反転してから 1 CK クロックサイクル後に最初のビット（フィリップス I²S 規格では MSb）が使用可能になります。WS の立下りエッジの遷移は、次の送信データが左チャンネルであることを示し、立上りエッジの遷移は、次の送信データが右チャンネルであることを示します。

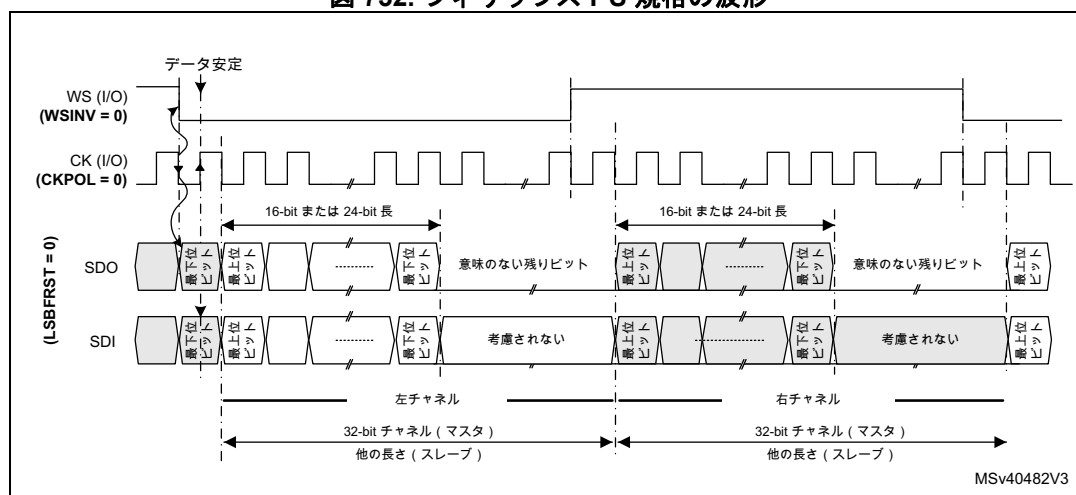
図 731. マスタ I²S フィリップスプロトコルの波形 (16/32-bit フル精度)

1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズについては、[セクション 52.3 : SPI の実装](#)を参照してください。

CKPOL は、I2S フィリップスプロトコルに合わせるためにクリアします。WS 信号の扱いに関する情報は [CK サンプリングエッジの選択](#)を参照してください。

[図 731](#) は、チャンネル長がデータ長に等しい場合に SPI/I2S によって生成される波形の例を示します。より正確には、CHLEN = 0 および DATLEN = 0b00 の場合、または CHLEN = 1 および DATLEN = 0b10 の場合です。

WS 信号の扱いに関する情報は [WS 反転の制御](#)を参照してください。

図 732. フィリップス I²S 規格の波形

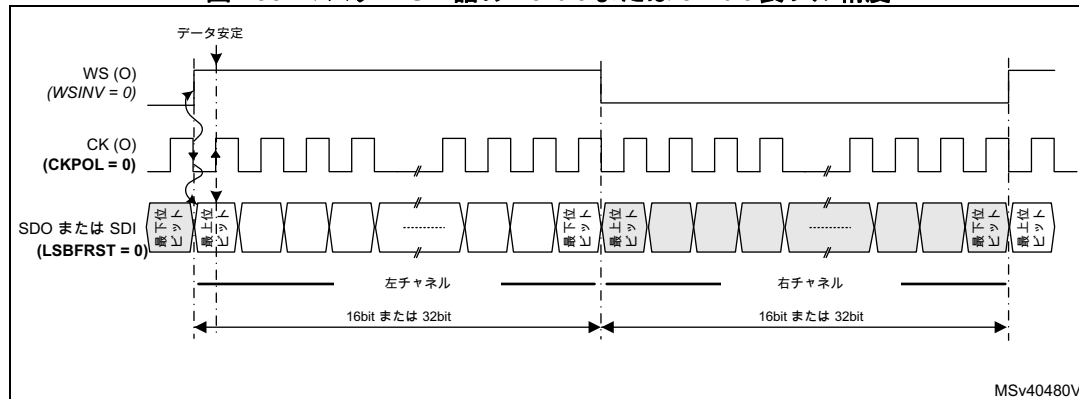
1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズについては、[セクション 52.3 : SPI の実装](#)を参照してください。

チャンネル長がデータ長より大きい場合、SPI/I2S が送信モードに設定されているとき、残りのビットは意味がありません。これはマスタモードとスレーブモードの両方に適用されます。

MSB 詰め規格

この規格では、WS 信号は最初のデータビットが提供される時に反転します。転送されるデータは、WS がハイの場合、左チャンネルを表し、WS がローの場合、右チャンネルを表します。

図 733. マスタ MSB 詰め 16-bit または 32-bit 長フル精度

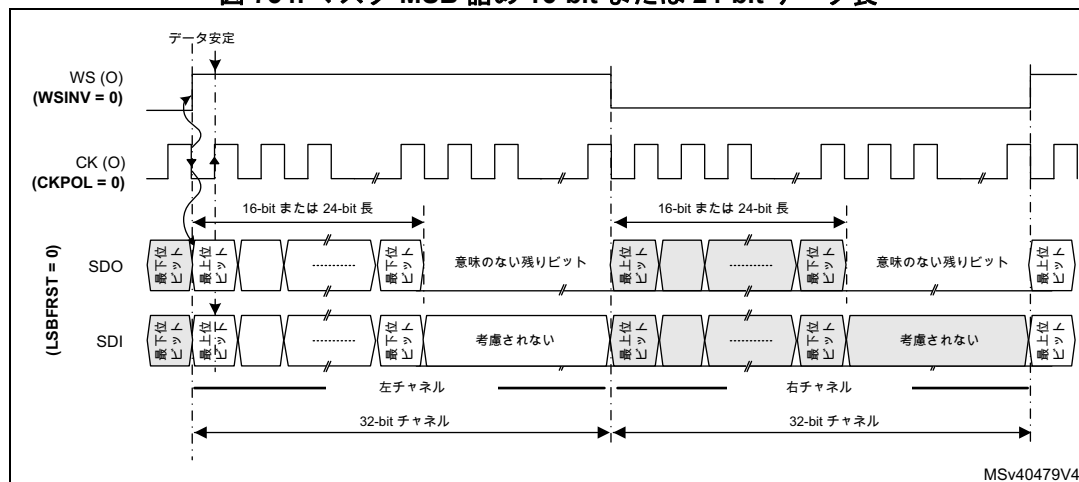


1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

CKPOL は、I2S MSB 詰めプロトコルに合わせるためにクリアします。WS 信号の扱いに関する情報は [CK サンプリングエッジの選択](#)を参照してください。

WS 信号の扱いに関する情報は [WS 反転の制御](#)を参照してください。

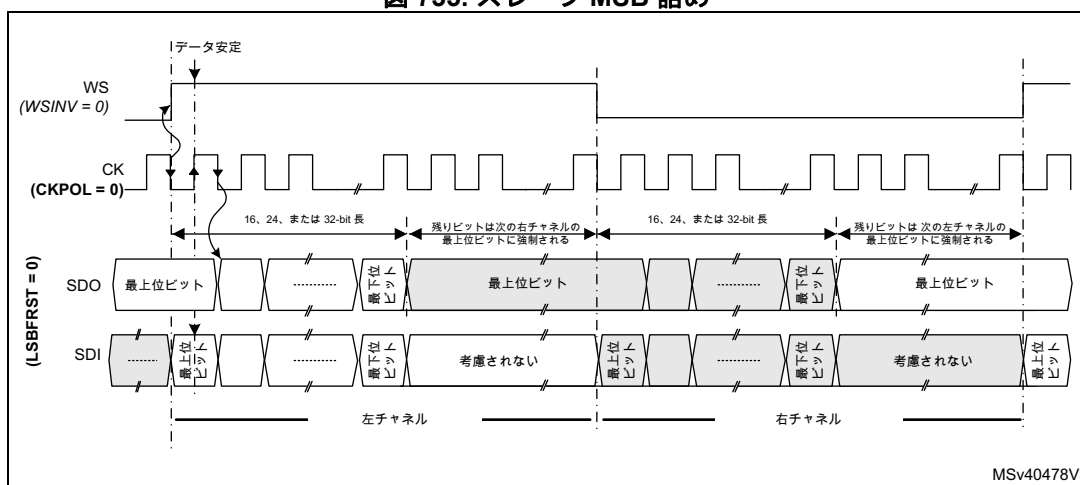
図 734. マスタ MSB 詰め 16-bit または 24-bit データ長



1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

チャンネル長がデータ長より大きい場合、SPI/I2S がマスタ送信モードに設定されているとき、残りのビットは意味がありません。スレーブ送信では、タイミングの問題を避けるために、残りのビットは生成される次のデータの最初のビット値に強制されます ([図 735](#) を参照)。

図 735. スレーブ MSB 詰め



1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

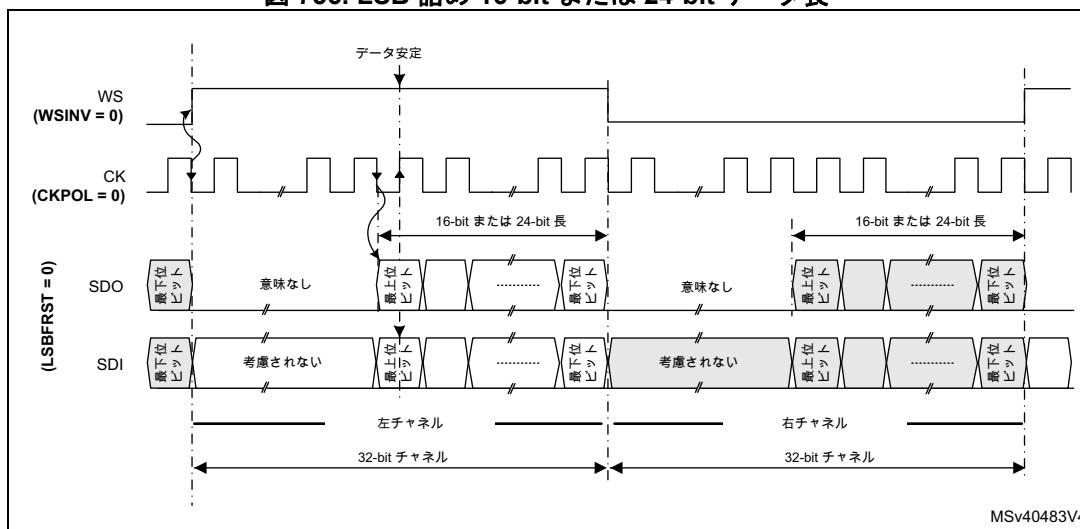
LSB 詰め規格

この規格は、マスタモードの MSB 詰め規格と似ています (16 ビットと 32 ビットのフル精度フレームフォーマットについて違いはありません)。LSB 詰め 16 または 32 ビットフル精度フォーマットでは、チャンネルとデータが同じ長さなので、MSB 詰めモード ([図 733](#) を参照) と同様の波形になります。

注：

LSB 詰めフォーマットでは、マスタおよびスレーブモードでは 16 および 32 ビットチャンネル長しかサポートされていません。これは、チャンネル長がトランスミッタおよびレシーバ側でわかっていないとデータを適切に転送することができないという事実のためです。

図 736. LSB 詰め 16-bit または 24-bit データ長



1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

CKPOL は、²S LSB 詰めプロトコルに合わせるためにクリアします。WS 信号の扱いに関する情報は [CK サンプリングエッジの選択](#) を参照してください。

WS 信号の扱いに関する情報は [WS 反転の制御](#) を参照してください。

PCM 規格

PCM 規格では、チャンネルサイド情報を使用する必要はありません。SPI_I2SCFGR レジスタの PCMSYNC ビットを使用して、2 つの PCM モード（ショートおよびロングフレーム）の使用および選択が可能です。

PCM ロングフレーム：

- WS 信号のアサーション時間は、マスタモードで CK の 13 サイクルに固定されます。
- 最初のデータビットは、WS 信号がアサートされるとすぐに受信または送信されます。

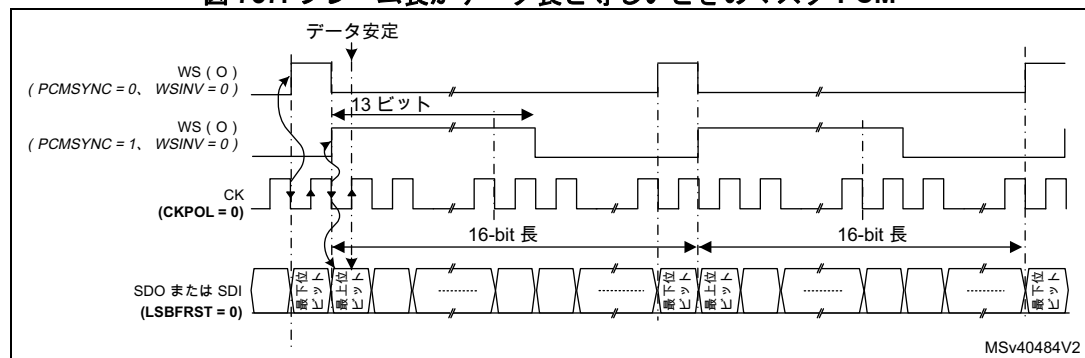
PCM ショートフレーム：

- WS 信号のアサーション時間は、マスタモードで CK の 1 サイクルに固定されます。
- 最初のデータビットは、CK の 1 サイクルが WS のアサーション後に受信または送信されます。

両方の PCM モード：

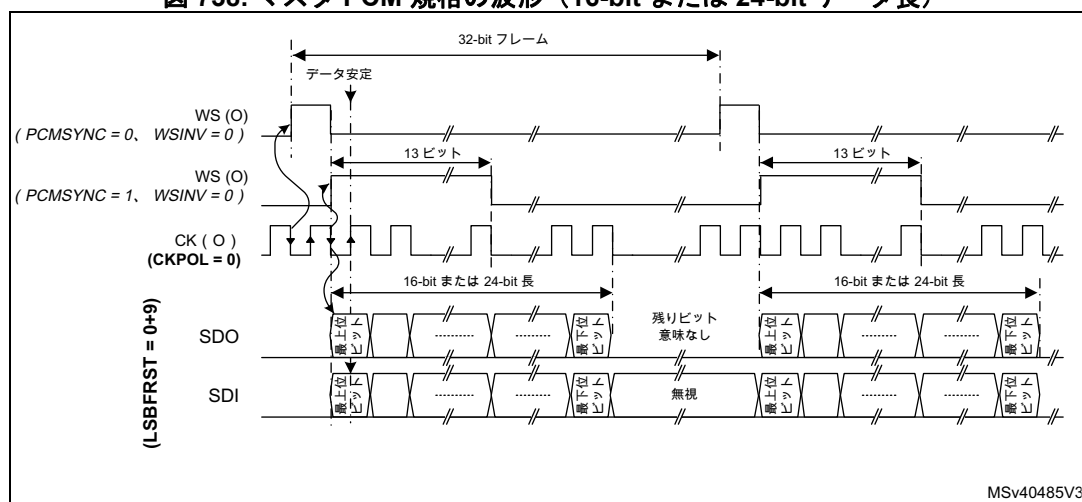
- 税所のデータビットは、LSBFIRST のビット値に応じて MSB または LSB です。
- CK サンプリングエッジは、CKPOL ビットによって選択できます。
- WS 信号は、WSINV ビットによって逆にすることができます。WS 信号の扱いに関する情報は [WS 反転の制御](#) を参照してください。

図 737. フレーム長がデータ長と等しいときのマスタ PCM



チャンネル長が 32 ビットにセットされているとき、16 または 24 ビットのデータサイズを使用することができます。

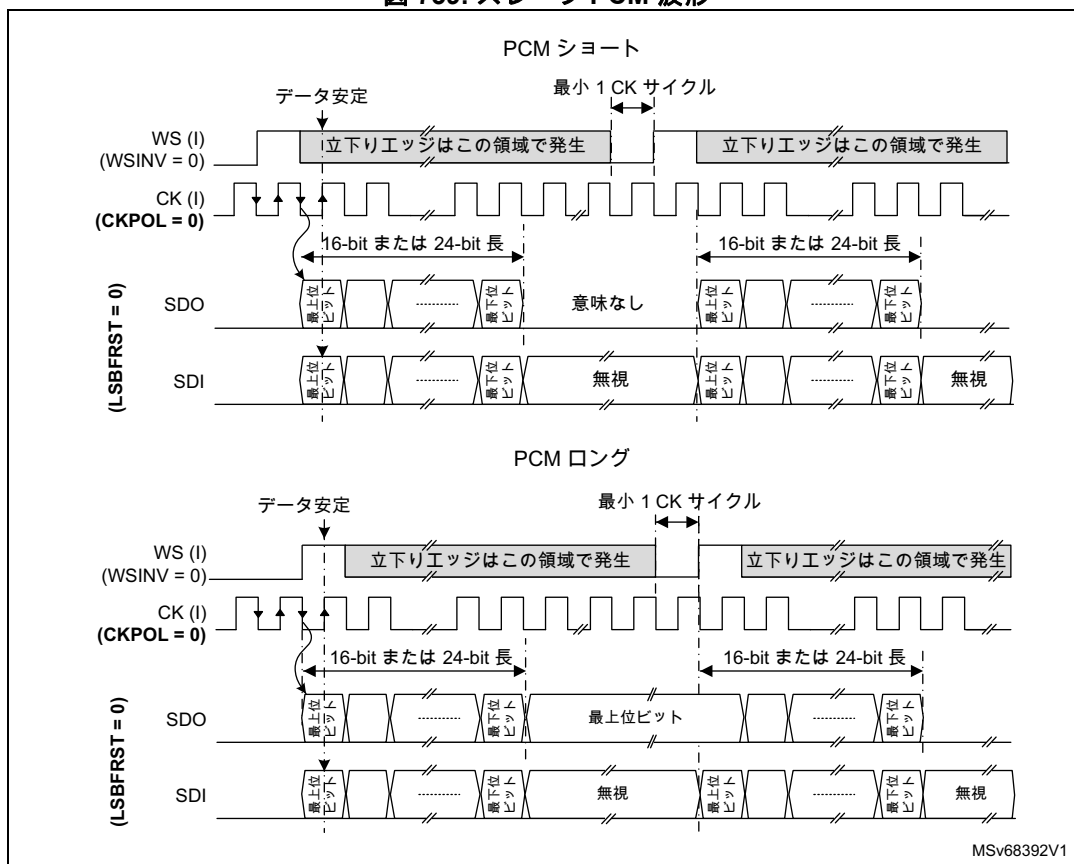
図 738. マスタ PCM 規格の波形 (16-bit または 24-bit データ長)



1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

スレーブモードで PCM プロトコルが使用された場合、フレーム長は 16 または 32 ビットとは異なる可能性があります。図 739 に示すように、スレーブモードでは、WS の立上リエッジによってフレームの開始が検出されるので、さまざまな WS のパルス幅が許されます。唯一の制約は、WS は少なくとも 1 CK サイクル前に非アクティブ状態に戻らなければならないということです。

図 739. スレーブ PCM 波形



1. 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3: SPI の実装](#)を参照してください。

注： チャネル長がデータ長より大きい場合、スレーブ PCM ロングでは、次に送信するデータが TXFIFO に含まれているときは、残りのビットの送信を、次に生成するデータの最初のビットの値に強制します。

注：スレーブモードでは、CHLEN を必ず適切にプログラムする必要があります（FIXCH 値にかかわらず）。たとえば、CHLEN がクリアされている場合（16 ビット長）、DATLEN が 0 以外であっても、データ転送は 16 ビットまで切捨てられます。この状況を避けるために、DATLEN = 1 または 2 のときに、CHLEN をセットする必要があります。

52.9.6 シリアルインタフェースの付加的な柔軟性

スレーブでの可変フレーム長

スレーブモードでは、チャネル長がデータ長より大きい限り、16 または 32 ビットと異なるチャネル長が許されます。これは、I2S LSB 詰めプロトコルを除いてすべてのプロトコルにあてはまります。

データ順序

すべてのデータフォーマットおよび通信規格に対して、**SPI/I2S 設定レジスタ 2 (SPI_CFG2)**にある **LSBFRST** ビットによってデータ順序 (MSb または Lsb ファースト) を選択することができます。

CK サンプリングエッジの選択

[SPI/I2S 設定レジスタ \(SPI_I2SCFGR\)](#)にある CKPOL ビットによって、すべてのプロトコルに対してスレーブおよびマスタモード用の CK のサンプリングエッジの極性を選択することができます。

- CKPOL = 0 のとき、シリアルデータ SDO および WS (マスタの時) は CK の立下りエッジで変化し、シリアルデータ SDI および WS (スレーブの時) は 立上りエッジで読み出されます。
- CKPOL = 1 のとき、シリアルデータ SDO および WS (マスタの時) は CK の立上りエッジで変化し、シリアルデータ SDI および WS (スレーブの時) は 立下りエッジで読み出されます。

WS 反転の制御

すべてのプロトコルに対して、WSINV をセットすることによって、マスタおよびスレーブモード用の WS 信号のデフォルトの極性を反転することができます。デフォルトでは、WS の極性は次のとおりです。

- フィリップス I²S 規格では、左チャンネルに対して WS は LOW で、右チャンネルに対して HIGH です。
- MSB/LSB 詰めモードでは、左チャンネルに対して WS は HIGH で、右チャンネルに対して LOW です。
- PCM モードでは、フレームの開始は WS の立上りエッジで指示されます。

WSINV がセットされているとき、WS の極性は反転され、したがって次のようになります。

- フィリップス I²S 規格では、左チャンネルに対して WS は HIGH で、右チャンネルに対して LOW です。
- MSB/LSB 詰めモードでは、左チャンネルに対して WS は LOW で、右チャンネルに対して HIGH です。
- PCM モードでは、フレームの開始は WS の立下りエッジで指示されます。

WSINV は [SPI/I2S 設定レジスタ \(SPI_I2SCFGR\)](#) に置かれています。

I/O の制御

[SPI/I2S 設定レジスタ 2 \(SPI_CFG2\)](#) の AFCNTR ビットによって、SPI/I2S ブロックは SPI/I2S を有効にする前に WS および CK 信号を非アクティブ状態に確定することができます。

これは、次の手順を使用することによって CKPOL および WSINV をプログラムすることによって行うことができます。

AFCNTR は最初にクリアされるものとします。

- I2SMOD = 1 にセットします (CK および WS の極性が CKPOL および WSINV 経由で制御されることをハードウェアに知らせるため)。
- CKPOL および WSINV ビットを希望の値にセットします。
- AFCNTR = 1 にセットします。
そうすると、SPI/I2S がまだ有効化されていなくても、CKPOL および WSINV の値に従って CK および WS I/O の非アクティブレベルがセットされます。
- その後、I2S/PCM のアクティブ化手順を実行します。

表 558 に、AFCNTR ビットがセットされ、SPI/I2S ブロックが有効化される前（すなわち、非アクティブレベル）の、WS および CK 信号のレベルを示します。WS のレベルは選択したプロトコルにも依存することに注意してください。

表 558. AFCNTR = 1 のとき、SPI/I2S が有効化される前の WS および CK のレベル

WSINV	I2SSTD		SPI/I2S が有効化される前の WS のレベル	CKPOL		SPI/I2S が有効化される前の CK のレベル
0	I ² S 規格 (00)	→	高	0	→	低
	その他	→	低		→	高
1	I ² S 規格 (00)	→	低	1	→	高
	その他	→	高		→	低

注： SPII2S がスレーブモードのとき、AFCNTR ビットをセットすることはできません。

シリアルデータラインのスワッピング

SDI および SDO の方向は、SPI/I2S 設定レジスタ 2 (SPI_CFG2) の IOSWP ビットと、スレーブ/マスターモードに依存します。表 559 に、この機能の詳細を示します。

表 559. シリアルデータラインのスワッピング

方向	IOSWP	マスターモード		スレーブモード	
		入力ライン	出力ライン	入力ライン	出力ライン
RX	0	SDI	-	SDO	-
	1	SDO	-	SDI	-
TX	0	-	SDO	-	SDI
	1	-	SDI	-	SDO
全二重	0	SDI	SDO	SDO	SDI
	1	SDO	SDI	SDI	SDO

52.9.7 起動シーケンス

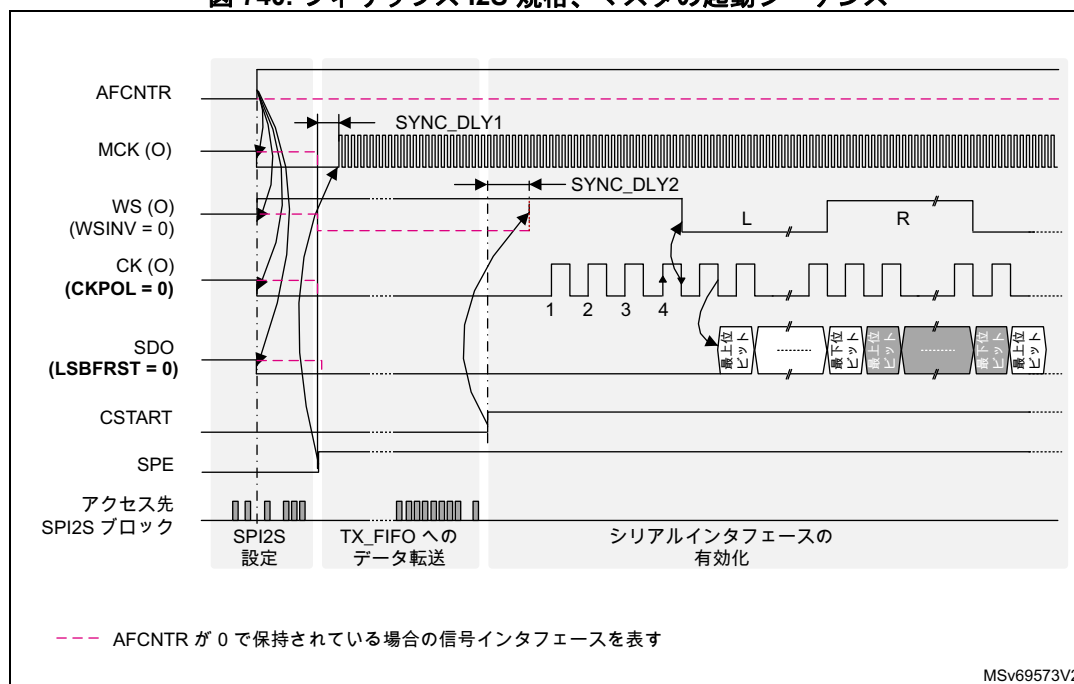
SPE ビットがクリアされているとき、SPI_RXDR と SPI_TXDR レジスタに読み書きすることはできませんが、他のレジスタへのアクセスは許可されます。

アプリケーションで SPI/I2S ブロックを使用したいときは、次の手順に従う必要があります。

1. SPE がクリアされていることを確認し、そうでなければ SPE に 0 を書き込みます。
2. 希望の設定に従ってすべての設定と制御レジスタをプログラムします。詳細なプログラミング例については、[セクション 52.9.16](#) を参照してください。
3. SPI_IFCR レジスタの USPC、TIFREC、OVRC、および UDRC ビットを設定してステータスフラグをすべてクリアします。フラグ SUSP がクリアされていない場合 (SUSPC ビット経由で)、CSTART 制御ビットには影響しません。
4. SPI/I2S ブロックをアクティブ化するために SPE ビットをセットします。このビットがセットされると、シリアルインタフェースはまだ無効ですが、DMA と割込みサービスは動作していて、たとえば、TxFIFO へのデータの転送は可能です。MCK の生成は、SPE が 1 になったときにも開始できます。
5. シリアルインタフェースをアクティブ化するために CSTART ビットをセットします。

図 740 に示すように、フィリップス I2S 規格のマスタ送信では、CSTART が 1 で TxFIFO が空でない場合、再同期遅延 (SYNC_DLY2) 後に、WS および CK 信号の生成が開始されます。外部のスレーブデバイスが確実に WS の遷移を適切に検出するために、WS の立下りエッジが発生する 4 回の立上りエッジ前に、ビットクロック CK がアクティブになることに注意してください。他の規格も同様に動作します。

図 740. フィリップス I2S 規格、マスタの起動シーケンス



- 図に示すとおり、MCK はビット SPE をセットするとすぐに有効化することができます。同期遅延 (SYNC_DLY1) 後に生成されます。詳細については、[セクション 52.9.9: クロックジェネレータ](#)の MCK の生成を参照してください。
- WS および CK 信号のレベルは、AFCNTR ビットがセットされるとすぐに、設定フェーズ中に SPI/I2S ブロックによって制御されることに注意してください。

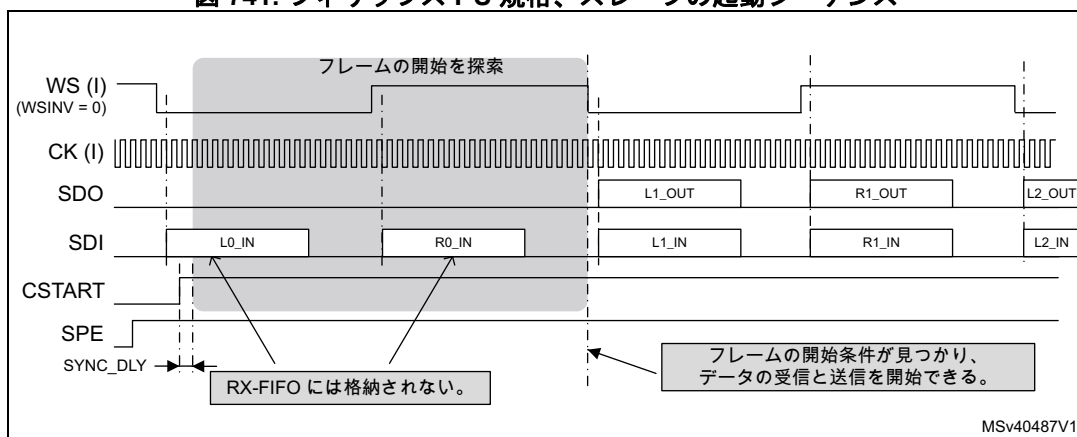
注： クロックドメインの再同期のために、CSTART ビットは CK クロックの約 3 周期 (SYNC_DLY2) 後にハードウェアによって考慮されます。

スレーブモードでは、CSTART がセットされると、以下のフレームの開始条件が満たされるときにデータ転送が開始されます。

- フィリップス I2S 規格では、フレームの開始条件は WS 信号の立下りエッジです。送信／受信は 1bit クロック後に開始されます。
WSINV = 1 の場合は、フレームの開始条件は 立上りエッジです。
- 他のプロトコルでは、フレームの開始条件は WS 信号の立上りエッジです。送信／受信は、MSB 詰めプロトコルでは WS の立上りエッジで開始されます。PCM プロトコルでは、送信／受信は 1bit クロック後に開始されます。
WSINV = 1 の場合は、フレームの開始条件は 立下りエッジです。

図 741 に、フィリップス I²S 規格、スレーブモードの起動シーケンスの例を示します。

図 741. フィリップス I²S 規格、スレーブの起動シーケンス



注：クロックドメインの再同期のために、CSTART ビットは CK クロックの 2 周期 (SYNC_DLY) 後にハードウェアによって考慮されます。

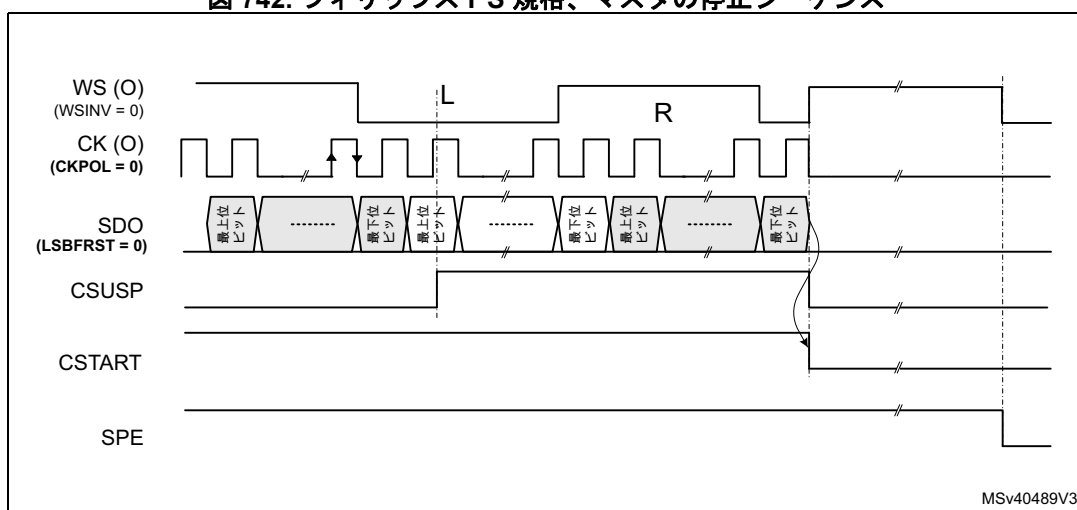
52.9.8 停止シーケンス

アプリケーションは SPE ビットをクリアすることによって I2S/PCM 転送を停止できます。その場合、現在のフレームの終了を待たずに通信は直ちに停止します。

マスターモードでは、現在のフレームの終了時点で I2S/PCM の転送を停止することもできます。その目的のために、CSUSP ビットをセットして、CSTART ビットが 0 になるまでポーリングする必要があります。CSTART ビットは、現在のステレオ (I2S モードが選択されていた場合) またはモノラルサンプルが完全にシフトイン/アウトされるとき、0 になります。その後、SPE ビットをクリアすることができます。

図 742 に、マスターモードの場合の停止シーケンスの例を示します。左サンプルの送信中に CSUSP ビットがセットされ、右サンプルの最後のビットが転送されるまで転送は続行されます。その後、CSTART および CSUSP は 0 に戻り、CK と WS 信号が非アクティブ状態に戻り、そして SPE ビットをクリアできます。

図 742. フィリップス I²S 規格、マスターの停止シーケンス



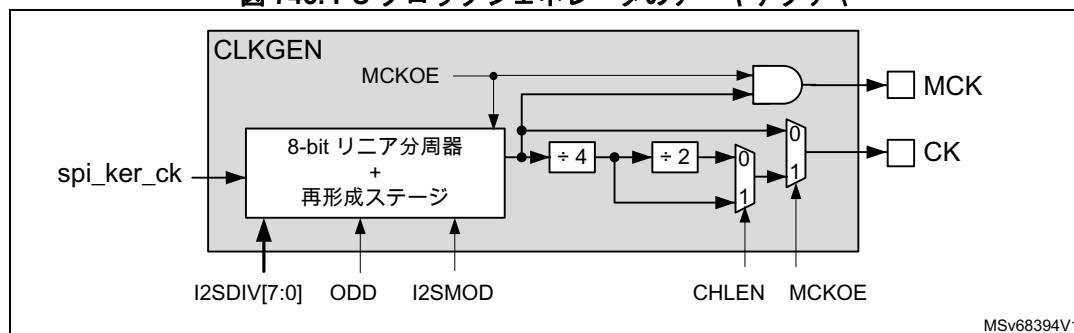
注： スレーブモードでは、停止シーケンスは SPE ビットによってのみ制御されます。

52.9.9 クロックジェネレータ

I2S または PCM をマスタモードに設定する場合、フレーム同期 (WS)、ビットクロック (CK)、およびマスタクロック (MCK) を希望の周波数で生成するためにクロックジェネレータをプログラムする必要があります。

I²S または PCM をスレーブモードで使用する場合は、クロックジェネレータを設定する必要はありません。

図 743. I²S クロックジェネレータのアーキテクチャ



MCK、CK、および WS に生成される周波数は主に I2SDIV、ODD、CHLEN、および MCKOE に依存します。MCKOE ビットはマスタクロックの生成の要否を示します。マスタクロックはフレーム同期よりも 256 または 128 倍高い周波数を持ちます。このマスタクロックは外部オーディオコーデックにリファレンスクロックを提供するのにしばしば必要となります。

注： マスタモードでは、バスクロック速度 (F_{pclk}) とビットクロック (F_{CK}) の比には特に制約はありません。バスクロック周波数は、データのスループットをサポートするために十分高くなければなりません。

マスタクロックが生成される場合 (MCKOE = 1)、フレーム同期の周波数は I2S モードでは次の式で与えられます。

$$F_{\text{WS}} = \frac{F_{\text{i2s_clk}}}{256 \times \{(2 \times \text{I2SDIV}) + \text{ODD}\}}$$

PCM モードでは次の式で与えられます。

$$F_{\text{WS}} = \frac{F_{\text{i2s_clk}}}{128 \times \{(2 \times \text{I2SDIV}) + \text{ODD}\}}$$

さらに、MCK (F_{MCK}) の周波数は次の式によって与えられます。

$$F_{\text{MCK}} = \frac{F_{\text{i2s_clk}}}{\{(2 \times \text{I2SDIV}) + \text{ODD}\}}$$

マスタクロックが無効の場合 (MCKOE = 0)、フレーム同期の周波数は I²S モードでは次の式で与えられます。

$$F_{WS} = \frac{F_{i2s_clk}}{32 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

PCM モードでは次の式で与えられます。

$$F_{WS} = \frac{F_{i2s_clk}}{16 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

ここで、 F_{WS} はフレーム同期の周波数、 F_{i2s_clk} は SPI/I²S ブロックに供給されるカーネルクロックの周波数です。

注： CHLEN および ODD は 0 または 1 です。

ODD が 0 のとき、I2SDIV は 0 から 255 までの任意の値をとることができますが、ODD = 1 のとき、I2SDIV = 1 の値は許されません。

I2SDIV = 0 のとき、 $\{(2 \times I2SDIV) + ODD\}$ は 1 に強制されます。

注： $\{(2 \times I2SDIV) + ODD\}$ が奇数のとき、MCK または CK 信号のデューティサイクルは 50% になりません。奇数の比を使用する場合は、セットアップおよびホールド時間のマージンに影響を与える可能性があるため注意が必要です。たとえば、 $\{(2 \times I2SDIV) + ODD\} = 5$ の場合、デューティサイクルは 40% になります。

表 560 に、I²S モードに対するクロックジェネレータのプログラミング例を示します。

MCK の生成

次の条件が満たされた場合、マスタクロック MCK が生成されます。

- I2SMOD は 1 でなければなりません。
- I2SCFG はマスタモードを選択する必要があります。
- MCKOE をセットする必要があります。
- SPE をセットする必要があります。

表 560. 通常の I²S 周波数での CLKGEN のプログラミング例

i2s_clk (MHz)	チャンネル長 (ビット)	I2SDIV	ODD	MCK	サンプリングレート: F _{ws} (kHz)
12.288	16	12	0	なし	16
12.288	32	6	0		16
12.288	16	6	0		32
12.288	32	3	0		32
49.152	16	16	0		48
49.152	32	8	0		48
49.152	16	8	0		96
49.152	32	4	0		96
49.152	16	4	0		192
49.152	32	2	0		192

表 560. 通常の I²S 周波数での CLKGEN のプログラミング例 (続き)

i2s_clk (MHz)	チャンネル長 (ビット)	I2SDIV	ODD	MCK	サンプリングレート : F _{ws} (kHz)
4.096	16 または 32	0	-	あり	16
24.576	16 または 32	3	0		32
49.152	16 または 32	3	0		48
12.288	16 または 32	0	-		96
49.152	16 または 32	2	0		192
61.44	16 または 32	2	1		
98.304	16 または 32	2	0		
196.608	16 または 32	2	0		

52.9.10 内部 FIFO

I²S インタフェースは RX および TX パス用の専用の FIFO を使用できます。送信サンプルは SPI_TXDR レジスタを経由して TxFIFO に書き込むことができます。Rx FIFO からの読出しは SPI_RXDR レジスタを経由して実行されます。

データの配置と順序

DATFMT ビットによって、SPI_RXDR および SPI_TXDR レジスタ内のデータ配置を選択することができます。

また、SPI_RXDR または SPI_TXDR に配置されるデータのフォーマットは、それらのレジスタが APB バス経由でアクセスされる方法にも依存することに注意してください。

図 744 は APB アクセスサイズ、DATFMT、および DATLEN の間で可能な設定を示します。

注： APB アクセスサイズが 32 ビットで、DATLEN = 0 の場合は注意が必要です。読出し動作では、Rx FIFO には少なくとも 2 つのデータが存在する必要があります。そうしないと、読み出しデータは不正になります。同様に書き込み動作では、TxFIFO には少なくとも 2 つの空き場所が存在する必要があります。そうしないと、データは失われる可能性があります。

図 744. データフォーマット

APB アクセス サイズ	DATLEN	SPI_RXDR, SPI_TXDR (DATFMT = 0)	SPI_RXDR, SPI_TXDR (DATFMT = 1)
16 ビット	0b00 (16 ビット)	15 0 有効なサンプル	15 0 有効なサンプル
32 ビット	0b00 (16 ビット)	31 16 15 0 有効なサンプル N+1 有効なサンプル N	31 16 15 0 有効なサンプル N+1 有効なサンプル N
32 ビット	0b01 (24 ビット)	31 24 23 0 ゼロ 有効なサンプル	31 8 7 0 有効なサンプル ゼロ
32 ビット	0b10 (32 ビット)	31 0 有効なサンプル	31 0 有効なサンプル

MSv40491V1

- I²S モードでは、サンプル N は左サンプルを表し、サンプル N+1 は右サンプルを表します。
- 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、セクション 52.3 : SPI の実装を参照してください。

プログラム可能な FIFO 閾値レベルに従って、割込みまたは DMA リクエストを生成することができます。FIFO 閾値は RXFIFO と TxFIFO で共通で、FTHLV によって調節できます。

I²S モードでは、左と右のオーディオサンプルは FIFO に交互に入ります。つまり、送信操作では、TxFIFO を左サンプルで埋めることから開始し、続いて右サンプルというように交互に充填する必要があります。受信モードでは、RxFIFO から読み出される最初のデータは左チャンネルを表すと考えられ、次のデータが右チャンネルとなります。

FIFO の読出し、書込みのポインタは、SPE ビットがクリアされると、リセットされることに注意してください。

詳細については、[セクション 52.9.11](#)および[セクション 52.9.15](#)を参照してください。

FIFO サイズの最適化

FIFO の基本要素はバイトです。これにより、FIFO の場所を最適化できます。たとえば、データサイズが 24bit に固定されているとき、各オーディオサンプルは 3つの FIFO 基本要素を占めます。

たとえば、16 基本要素の FIFO の深さは以下ようになります。

- DATLEN = 0 (16 ビット) の場合、8 サンプル
- DATLEN = 1 (24 ビット) の場合、5 サンプル^(a)
- DATLEN = 2 (32 ビット) の場合、4 サンプル^(a)

52.9.11 FIFO ステータスフラグ

アプリケーションが I2S インタフェースの状態を完全に監視できるように、2つのステータスフラグが用意されています。両方のフラグは割込みリクエストを生成できます。RXPIE ビットが有効化されている場合、受信割込みが生成され、TXPIE ビットが有効化されている場合、送信割込みが生成されます。これらのビットは SPI_IER レジスタに配置されています。

TxFIFO 閾値到達 (TXP)

このフラグがセットされると、TxFIFO には少なくとも FTHLV の空き領域があることが示され、それによって送信される新しい FTHLV データを SPI_TXDR に書き込むことができます。空き容量が FTHLV より少なくなると TXP フラグがリセットされます。I2S が無効にされている (SPE ビットがリセット) とき、TXP = 1 であることに注意してください。

RxFIFO 閾値到達 (RXP)

このフラグがセットされると、RxFIFO には少なくとも FTHLV の有効なデータがあることが示され、それによってユーザは SPI_RXDR を経由してそれらのデータを読み出すことができます。RxFIFO 内のデータが FTHLV より少なくなると、このフラグはリセットされます。

I2S モードでの割込み機能についての詳しい情報は[セクション 52.10](#)を参照してください。

a. 常に使用可能ではありません。24 および 32 ビットのデータ幅がサポートされているかどうかを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

52.9.12 アンダーラン状況の取り扱い

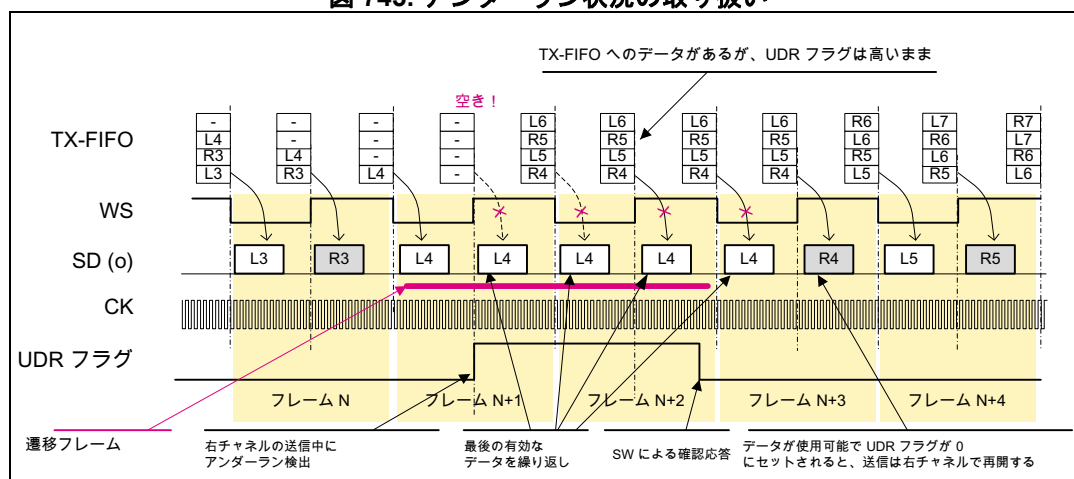
送信モードでは、TxFIFO がすでに空になっているときに新しいデータをシフトレジスタにロードする必要がある場合は、アンダーラン状況が検出されます。このような状況では、UDR フラグがセットされ、1 つ以上のオーディオフレームに予想外のデータが含まれます。

I²S モードには、誤配置状況（左右チャンネルの入れ替わり）を防止するためのハードウェアメカニズムがあります。アンダーラン状況が検出されると、実際のチャンネル長が CHLEN ビットで選択された長さと一致する場合、シフトレジスタにある最後の有効なデータが繰り返されます。そうでない場合、未定義のデータが繰り返されます。通常、ブロックがスレーブ TX モードでプログラムされていて、外部マスタオーディオデバイスが 16 または 32 ビット以外のチャンネル長を使用している場合、アンダーラン時に繰り返されるデータ値は未定義です。

次の図は、アンダーランの発生時に再開条件が満たされない限り、ペリフェラルが左と右のチャンネルの最後の有効なデータをリプレイすることを示しています。次の場合に送信が再開されます。

- TxFIFO に十分なデータがあるとき、および、
- UDR フラグがソフトウェアによってクリアされたとき、および、
- 再開の条件を満たしたとき：
 - 右のチャンネルデータを送信する必要があるときにアンダーランが発生した場合、右のチャンネルが送信を必要としたときに送信が再開されるか、
 - 左のチャンネルデータを送信する必要があるときにアンダーランが発生した場合、左のチャンネルが送信を必要としたときに送信が再開されます。

図 745. アンダーラン状況の取り扱い



ブロックを PCM モードの 1 つに設定した場合、送信は次のフレーム開始時に、TxFIFO に十分なデータがあり、UDR フラグがクリアされている場合に、再開されます。

SPI_IER レジスタの UDRIE ビットがセットされている場合、UDR フラグは割込みをトリガできます。UDR ビットは、SPI_IFCR レジスタの UDRC ビットに 1 を書き込むことによってクリアされます。

注：

アンダーラン状況は、マスタまたはスレーブモードで起きる可能性があります。マスタモードでは、アンダーランが起きると、WS、CK、および MCK 信号はゲートされません。

再同期のために、UDR フラグの変化は少なくとも CK クロックの 2 周期後にハードウェアによって考慮されます。

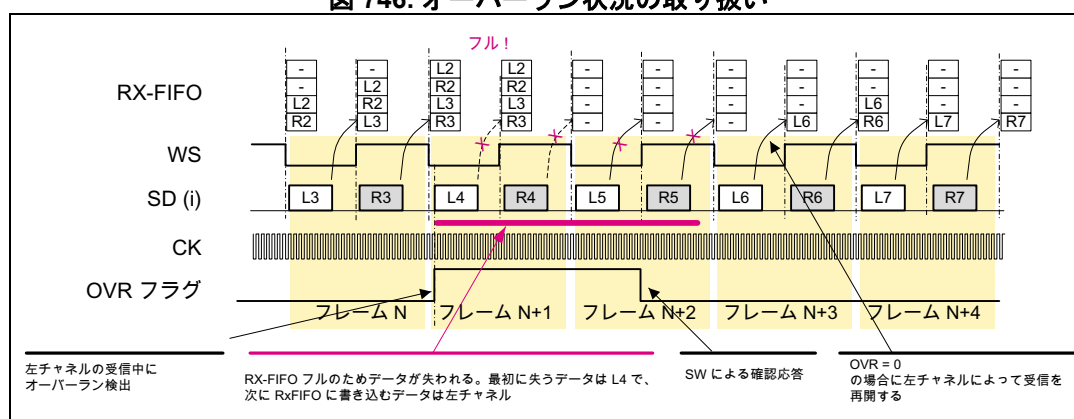
52.9.13 オーバーラン状況の取り扱い

受信モードで、フルの状態の RxFIFO にシフトレジスタが新しいデータの格納を求めたときに、オーバーラン状況が検出されます。このような状況では、OVR フラグがセットされ、受信データは失われます。

I²S モードでは、誤配置状況（左右チャンネルの入れ替わり）を防止するためのハードウェアメカニズムがあります。次の図に示すように、オーバーランが起きると、再開条件が満たされない限り、ペリフェラルはデータを RxFIFO に書き込むことを停止します。

受信は、RxFIFO に十分な空きができて、OVR フラグがクリアされたときに再開します。ブロックは、右のチャンネルデータを受信した時にオーバーランが発生した場合は次の右チャンネルを RxFIFO に書き込むことから開始し、左のチャンネルデータを受信した時にオーバーランが発生した場合は次の左チャンネルを書き込むことから開始します。

図 746. オーバーラン状況の取り扱い



ブロックを PCM モードに設定した場合、オーバーランエラー後、再開条件が満たされない限り、ブロックは RxFIFO へのデータの書き込みを停止します。RxFIFO に十分な空間があり、OVR フラグがクリアされている場合は、次の受信データが RxFIFO に書き込まれます。

SPI_IER レジスタの OVR1E ビットがセットされている場合は、割込みを生成できます。OVR ビットは、SPI_IFCR レジスタの OVR1C ビットに 1 を書き込むことによってクリアされます。

注： オーバーラン状況は、マスタまたはスレーブモードで起きる可能性があります。マスタモードでは、オーバーランが起きると、WS、CK、および MCK 信号はゲートされません。

52.9.14 フレームエラー検出

スレーブモードに設定されている場合、SPI/I2S ブロックは次の 2 種類のフレームエラーを検出します。

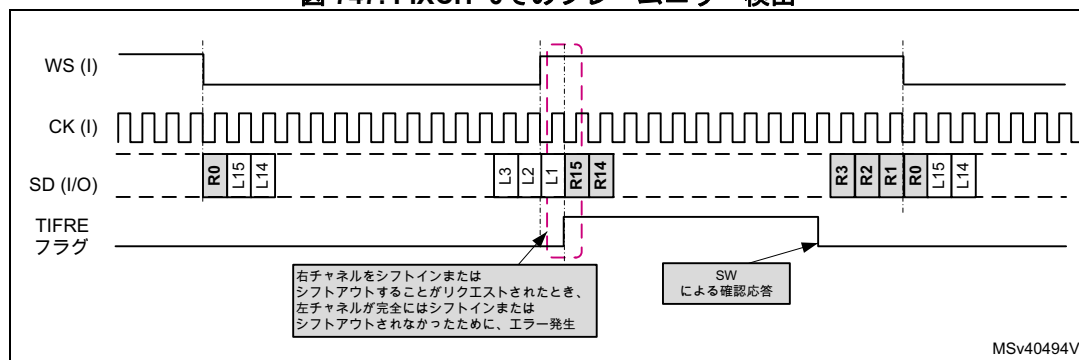
- 前のデータのシフトインまたはシフトアウトが完了していない時に受信したフレーム同期 (早期フレームエラー)。このモードは、FIXCH = 0 によって選択されます。
- 想定外の位置で発生するフレーム同期。このモードは、FIXCH = 1 によって選択されます。

スレーブモードでは、外部のマスタデバイスによって提供されるフレーム長が 32 または 64 ビットと異なっている場合、FIXCH をクリアする必要があります。SPI/I2S は各転送を WS で同期するので、ミスマライメントのリスクはありませんが、雑音の多い環境では、CK 信号にグリッチが起きた場合、サンプルは影響を受け、アプリケーションはこれに気が付きません。

外部のマスタデバイスによって提供されるフレーム長が 32 または 64 ビットに等しい場合、FIXCH をセットすることができ、それによって CHLEN を調節できます。SPI/I2S は各転送を WS で同期するので、同じくミスマライメントのリスクはありませんが、各チャンネル境界間のビットクロックの量が CHLEN と異なっていた場合、フレームエラーフラグ (TIFRE) がセットされます。

図 747 に、フレームエラー検出の例を示します。SPI/I2S ブロックはスレーブモードで、左チャンネル用のビットクロック周期の数がデータをシフトインまたはシフトアウトするのに十分ではありません。図は、WS 信号への整合を維持するために、実行中の転送が中断されて、次の転送が開始されたことを示しています。

図 747. FIXCH=0でのフレームエラー検出

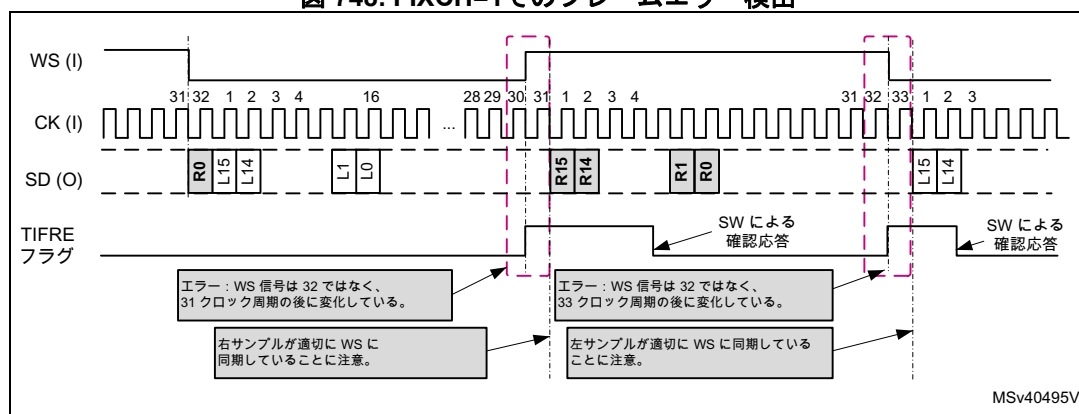


TIFREIE ビットをセットすれば、割込みを生成させることができます。フレームエラーフラグ (TIFRE) は、SPI_IFCR レジスタの TIFREC ビットに 1 を書き込むことによってクリアされます。

FIXCH ビットをセットすることにより、フレームエラーフラグの適用範囲を拡張することができます。このビットをセットすると、SPI/I2S はスレーブモードで固定チャネル長を想定します。つまり、想定されるチャネル長は CHLEN に従って 16 または 32 ビットの可能性があるということです。図 748 に示すように、このモードでは SPI/I2S ブロックは、WS 信号が予想時点で変化するかどうか (早すぎるか遅すぎるか) を検出することができます。

注： 図 747 および図 748 はスレーブ送信モードに対するメカニズムを示していますが、これはスレーブ受信およびスレーブ全二重に対してもあてはまります。

図 748. FIXCH=1でのフレームエラー検出



フレームエラー検出は、一般に WS や CK 信号の良好な受信を乱す雑音の多い環境が原因になる可能性があります。

注： オーバーランおよび早期フレームが同じフレーム内で起きた場合は、SPI/I2S は適切に回復することができません。この場合は、SPI/I2S を無効にして再度有効化する必要があります。

52.9.15 DMA インタフェース

I2S/PCM モードは SPI 機能と同じ DMA リクエストラインを共有します。送信パスと受信パスに対して別々の DMA チャンネルがあります。各 DMA チャンネルは、SPI_CFG1 レジスタの RXDMAEN と TXDMAEN ビットを介して有効にできます。

受信モードでは、DMA インタフェースは以下のように動作します。

1. ハードウェアが RxFIFO レベルを評価します。
2. RxFIFO に少なくとも FTHLV 個のサンプルが存在する場合、FTHLV DMA リクエストが生成されます。
 - FTHLV DMA リクエストが完了すると、ハードウェアは手順 1 に戻ります。
3. RxFIFO に FTHLV より少ないサンプルしかない場合、DMA リクエストは生成されず、ハードウェアは手順 1 に戻ります。

送信モードでは、DMA インタフェースは以下のように動作します。

1. ハードウェアが TxFIFO レベルを評価します。
2. TxFIFO に少なくとも FTHLV 個の空き場所が存在する場合、FTHLV DMA リクエストが生成されます。
 - FTHLV DMA リクエストが完了すると、ハードウェアは手順 1 に戻ります。
3. TxFIFO に FTHLV より少ない空き場所しかない場合、DMA リクエストは生成されず、ハードウェアは手順 1 に戻ります。

52.9.16 プログラミングの例

フィリップス I²S 規格のマスタ送信、全二重

この例では、マスタ全二重モードで、48 kHz のサンプリングレート、マスタクロックを使用しないで、フィリップス I²S 規格プロトコルをサポートするインタフェースのプログラム方法を示します。ここでは、SPI/I2S は回路のクロックコントローラから 61.44 mHz のカーネルクロック (i2s_clk) を受信していると想定します。上の例では、たとえば転送を開始する前に、I²C インタフェースを介して外部オーディオコーデックをプログラムする必要があると想定しました。さらに、この外部オーディオコーデックには、I2C コマンドを受け付けるために MCK が必要であると考えました。

手順

1. RCC ブロックを介して、バスインタフェースとカーネルクロックを有効にし、必要に応じてリセット信号をアサートし、リリースします。
2. 必要な I/O を選択するには、AFMUX をプログラムします。現行の例では、MCK、CK、WS、SDO、SDI が必要です。
3. MCK クロックを提供し、フレーム同期レートを正確に 48 kHz に設定するために、クロックジェネレータをプログラムします。I2SDIV を 2、ODD を 1、および MCKOE を 1 にセットします。
4. シリアルインタフェースプロトコルを次のようにプログラムします：CKPOL = 0、WSINV = 0、LSBFRST = 0、CHLEN = 1 (チャンネルあたり 32 ビット)、DATLEN = 1 (24 ビット)、I2SSTD = 0 (I2S フィリップス規格)、I2SCFG = 5 (マスタ全二重)、I2S/PCM モード用に I2SMOD = 1。
5. 希望する値を FTHLV に設定することによって FIFO 閾値を調節します。たとえば、2 つのオーディオサンプルの閾値が必要な場合は FTHLV = 1 とします。
6. アプリケーションが DMA を介したデータ転送を望んでいる場合は、ビット TXDMAEN および RXDMAEN をセットします。
7. SPI_IER レジスタに配置されたすべての割り込み有効化フィールドをクリアします。
8. SPI_IFCR レジスタの SUSPC、TIFREC、OVRRC、および UDRC を設定してステータスフラグをすべてクリアします。
9. SPE ビットをセットすると、MCK が生成されます。ここに示す例では、SPE = 1 にしても、すぐには TxFIFO に書き込まれません。送信するオーディオサンプルが用意できていなくても、アプリケーションは I²C インタフェースによって外部オーディオコーデックを設定できるようになります。以降は、[代替シーケンス](#)が提案されます。
10. アプリケーションがデータ転送の開始を求める場合：
 - データ転送が DMA を使用する場合は、
 - DMA ペリフェラルを 2 チャンネル、1 つは受信用、1 つは送信用にプログラムします。
 - メモリバッファを送信パス用の有効なオーディオサンプルで初期化します。
 - DMA チャンネルを有効にします。
 - 転送エラーを検出するために、必要に応じて UDRIE や OVRIE などの割り込みイベントを有効にします。
 - DMA チャンネルを有効にします。TXDMAEN がセットされたら、TxFIFO に書き込まれます。
 - データ転送が割り込みによって行われた場合：
 - 割り込みイベント UDRIE、OVRIE、TXPIE、および RXPIE を有効にします (1 を書き込みます)。割り込みリクエストがすぐにアクティブになり、割り込みハンドラは TxFIFO に書き込めるようになります。
11. 最後に、CSTART ビットをセットすることによって SPI/I2S シリアルインタフェースを有効にできます。CSTART ビットは SPI_CR1 レジスタにあります。

代替シーケンス

- 手順 1 ~ 8 は前のシーケンスと同じです。
- データ転送が DMA を使用する場合は、
 - DMA ペリフェラルを次のようにプログラムします：2 チャンネルで、1 つは受信用、1 つは送信用。
 - メモリバッファを送信パス用の有効なオーディオサンプルで初期化します。
 - 転送エラーを検出するために、必要に応じて UDRIE や OVRIE などの割り込みイベントを有効にします。
 - DMA チャンネルを有効にします。

- データ転送が割込みによって行われた場合：
 - 割込みイベント UDRIE、OVRIE、TXPIE、および RXPIE を有効にします (1 を書き込みます)。割込みリクエストがすぐにアクティブになり、割込みハンドラは TxFIFO に書き込めるようになります。
- SPE をセットします。このビットが 1 にセットされるとすぐに、以下の動作が行われます。
 - 割込み生成が有効になっている場合は、SPI/I2S は、割込みハンドラが TxFIFO を充填できるように割込みリクエストを生成します。
 - DMA 転送が有効な場合、SPI/I2S は、TxFIFO を充填するために DMA リクエストを生成します。
 - MCK が生成されます。これで、アプリケーションは I²C インタフェースを介して外部オーディオコーデックを設定できます。
- 最後に、CSTART ビットをセットすることによって SPI/I2S シリアルインタフェースを有効にできます。CSTART ビットは SPI_CR1 レジスタにあります。

停止手順 (マスタモード)

1. 実行中の転送を停止するために CSUSP ビットをセットします。
2. CSTART ビットの値をチェックして、0 になるまで待ちます。
3. SUSPC を設定して SUSP フラグをクリアします。
4. DMA ペリフェラル、バスクロック、等を停止します。
5. SPI/I2S ブロックを無効にするために SPE ビットをクリアします。

フィリップス I²S 規格のスレーブ受信

この例では、48 kHz のサンプリングレートで、フィリップス I²S 規格プロトコルのスレーブレシーバモードをサポートするインタフェースのプログラム方法を示します。スレーブモードでは、SPI/I2S ブロックは受信サンプルのサンプリングレートを制御できないことに注意してください。この例では、外部マスタデバイスが 24 ビットのチャンネル長の I²S フレーム構造を配信していると想定します。したがって、FIXCH がセットされているときのフレームエラー検出のために提供されている機能は使用できません。

手順

1. RCC ブロックを介して、バスインタフェースとカーネルクロックを有効にし、必要に応じてリセット信号をアサートし、リリースします。
2. 必要な I/O を選択するには、AFMUX をプログラムします。現在の例では CK、WS、SDI です。
3. シリアルインタフェースプロトコルを次のようにプログラムします：CKPOL = 0、WSINV = 0、LSBFRST = 0、FIXCH = 0 (チャンネル長が 16 および 32 ビットと異なっているため)、DATLEN = 0 (16 ビット)、I2SSTD = 0 (フィリップスプロトコル)、I2SCFG = 1 (スレーブ受信)、I2Sモード用に I2SMOD = 1。
4. 希望する値を FTHLV に設定することによって FIFO 閾値を調節します。たとえば、2つのオーディオサンプルの閾値が必要な場合は FTHLV = 1とします。
5. すべてのステータスフラグレジスタをクリアします。
6. OVRIE や TIFRE のような、割込みを生成するフラグを有効にします。
7. データ転送が DMA を使用する場合は、
 - DMA ペリフェラルを、1つの受信チャンネルにプログラムします。
 - DMA チャンネルを有効にします。
 - SPI/I2S ブロックで、RXDMAEN ビットをセットすることによって DMA を有効にします。

8. データ転送を割り込み経由で行う場合は、RXPIE ビットをセットして割り込みを有効にする必要があります。
9. SPE をセットします。
10. 最後に、シリアルインタフェースを有効にするために CSTART ビットをセットすることができます。外部マスタデバイスによる左データの送信が次に発生した時に、SPI/I2S は RxFIFO へのデータの格納を開始します。

停止手順 (スレーブモード)

1. SPI/I2S ブロックを無効にするために SPE ビットをクリアします。
2. DMA ペリフェラル、バスクロック、等を停止します。

52.10 I²S 割り込み

PCM/I2S モードでは、表 561 に記載されたイベントに従って割り込み (`spi_it`) またはウェイクアップイベント信号 (`spi_wkup`) を生成することができます。

割り込みイベントは個別に有効または無効にできます。

表 561. I2S 割り込みリクエスト

割り込みベクタ	割り込みイベント	イベントフラグ	イベント／割り込みのクリア方法	SLEEP モードの終了	STOP モードの終了	STANDBY モードの終了
SPI	TxFIFO 閾値到達	TXP	TxFIFO の空き領域が FTHLV より少ない場合	可能	はい	不可
	RxFIFO 閾値到達	RXP	RxFIFO のサンプルが FTHLV より少ない場合			
	オーバーランエラー	OVR	OVRC に 1 を書き込む			
	アンダーランエラー	UDR	UDRC に 1 を書き込む			
	フレームエラーフラグ	TIFRE	TIFREC に 1 を書き込む		不可	

52.11 SPI/I2S レジスタ

52.11.1 SPI/I2S 制御レジスタ 1 (SPI_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOLOCK
															rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRCINI	RCRCINI	CRC33_17	SSI	HDDIR	CSUSP	CSTART	MASRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE
rw	rw	rw	rw	rw	w	rs	rw								rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **IOLOCK** : 関連 I/O の AF 設定のロック

このビットは、SPI が無効 (SPE = 0) の場合にのみ、ソフトウェアで変更できます。MODF イベント発生時にハードウェアによってクリアされます。

0 : AF 設定はロックされません。

1 : AF 設定はロックされます。

このビットをセットすると、SPI_CFG2 レジスタの内容は変更できません。このビットは、SPI が有効 (SPE = 1) な場合に書き込み保護されます。

ビット 15 **TCRCINI** : トランスミッタ用 CRC 計算の初期化パターン制御

0 : すべて 0 のパターンを適用します。

1 : すべて 1 のパターンを適用します。

ビット 14 **RCRCINI** : レシーバ用 CRC 計算の初期化パターン制御

0 : すべてゼロのパターンを適用します。

1 : すべて 1 のパターンを適用します。

ビット 13 **CRC33_17** : 32 ビットの CRC 多項式設定

0 : フルサイズ (33 ビットまたは 17 ビット) の CRC 多項式は使用されません。

1 : フルサイズ (33 ビットまたは 17 ビット) の CRC 多項式が使用されます。

ビット 12 **SSI** : 内部 SS 信号入力レベル

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値はペリフェラル内部の SS 入力に強制的に設定され、SS ピンの I/O 値は無視されます。

ビット 11 **HDDIR** : 半二重モードでの Rx/Tx の方向

半二重の設定において、HDDIR ビットは Rx/Tx のデータ転送方向を確定します。このビットは、全二重または単方向の設定では無視されます。

0 : SPI はレシーバです。

1 : SPI はトランスミッタです。

ビット 10 CSUSP : マスタのサスペンドリクエスト

このビットは 0 として読み出されます。

マスタモードにおいて、このビットがソフトウェアによってセットされると、現在のフレームの終了時点で CSTART ビットがリセットされ、通信がサスペンドされます。フレームのトランザクションの終了を確認するために、ユーザは SUSP フラグをチェックする必要があります。

低電力モードに移行する前に、(このビットを使用するか、または TXDR を空に保持して) マスタモードの通信をサスペンドする必要があります。

ソフトウェアの中断後、SUSP フラグをクリアし、次のトランザクションが始まる前に SPI を無効化して再度有効化する必要があります。

ビット 9 CSTART : マスタ転送の開始

このビットは、SPI が有効な場合に、SPI または I2S/PCM 通信を開始するためだけに、ソフトウェアによってセットできます。SPI モードでは、転送の終了 (EOT) フラグがセットされたとき、またはトランザクションのサスペンド・リクエストが受理されたときに、ハードウェアによってクリアされます。I2S/PCM モードでも、[セクション 52.9.8 : 停止シーケンス](#)に記載されているように、ハードウェアによってクリアされます。

0 : マスタ転送はアイドル状態です。

1 : マスタ転送は動作中、または自動サスペンドによって一時的にサスペンドされています。

SPI モードでは、このビットはマスタモードでのみ考慮されます。送信が有効になっている場合、送信 FIFO でデータが利用可能になった場合のみ、通信が開始、または続行します。

ビット 8 MASRX : 受信モードでのマスタ自動サスペンド

このビットは、ソフトウェアによってセット／クリアされ、オーバーラン状態を避けるために、マスタレシーバモードで SPI の連続転送と自動管理を制御します。

0 : オーバーラン状態にかかわらず、SPI のフロー／クロック生成は続行します (データは失われます)。

1 : オーバーラン状態になる前に、RxFIFO がフル状態になった時、SPI フローは一時的にサスペンドされます。SPI 通信がサスペンドされたとき、SUSP フラグがセットされます。

ハードウェアによって自動的に SPI 通信がサスペンドされたとき、内部の同期遅延のために次のフレームの数ビットがすでにクロックアウトされている可能性があります。

このため、データのサイズが 8 ビット未満のとき、自動サスペンドの信頼性は低くなります。この場合、MIDI パラメータが非ゼロ値であるときに適用される、データフレーム間に挿入された遅延時間と組み合わせることによって安全なサスペンドが達成されます。データサイズとインターリーブされた SPI サイクルの合計により、少なくとも SPI クロックの 8 周期の長さの間隔が常に生成されます。SUSP ビットのソフトウェアをクリアした後で、通信はその後の制約なく後続ビットのトランザクションによって再開され、継続します。SUSP ビットをクリアする前に、ユーザは RxFIFO 空間を最大限解放する必要があります。これには、その後の中断を防ぐために、RXP フラグの指示に基づいて、RxFIFO で使用可能なすべてのデータパケットを読み出します。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 SPE : シリアルペリフェラル有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : シリアルペリフェラルは無効です。

1 : シリアルペリフェラルは有効です。

SPE = 1 の場合、SPI データ転送は有効化され、SPI_CFG1 および SPI_CFG2 の設定レジスタ、および SPI_CR1 レジスタの CRCPOLY、UDRDR、IOLOCK ビットは書き込み保護されます。それらは SPE = 0 のときのみ変更できます。

SPE = 0 の場合、任意の SPI 動作が停止され、無効化されると、MODF 割込みリクエストを除いて、割込みが有効化されたイベントのすべての保留中のリクエストがブロックされ (ただし保留により spi_plck クロックのリクエストが引き続き伝播される)、SS 出力がマスタで無効化され、RDY 信号はスレーブでノットレディステータスを保持し、内部ステートマシンはリセットされ、FIFO の内容はすべて一掃され、CRC 計算は初期化され、受信データレジスタはゼロを読み出します。

MODF エラーフラグがアクティブになっているとき、SPE はクリアされ、セットできません。

52.11.2 SPI/I2S 制御レジスタ 2 (SPI_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIZE[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TSIZE[15:0]** : 現在の転送でのデータ数

これらのビットがソフトウェアによって変更される場合、SPI を無効にする必要があります。

TSIZE にゼロ値が格納されているときに CSTART がセットされると、終わりのないトランザクションが開始されます。CRC が有効な場合、0x3FFF に対応した 0xFFFF の値に TSIZE をセットすることはできません。

注 : **TSIZE[15:10]** ビットは、制限された機能がセットされたインスタンスで予約され、リセット値で保持する必要があります。

52.11.3 SPI/I2S 設定レジスタ 1 (SPI_CFG1)

アドレスオフセット : 0x08

リセット値 : 0x0007 0007

このレジスタの内容は、SPI が有効になっているとき、書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BPASS	MBR[2:0]			Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	CRCSIZE[4:0]				
rW	rW	rW	rW						rW		rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDMA EN	RXDMA EN	Res.	Res.	Res.	Res.	UDRCFG	FTHLV[3:0]				DSIZE[4:0]				
rW	rW					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **BPASS** : マスタボーレートクロックジェネレータのプリスケアラのバイパス

0 : バイパスは無効です。

1 : バイパスは有効です。

ビット 30:28 **MBR[2:0]** : マスタボーレートのプリスケアラの設定

000 : SPI マスタクロック / 2
 001 : SPI マスタクロック / 4
 010 : SPI マスタクロック / 8
 011 : SPI マスタクロック / 16
 100 : SPI マスタクロック / 32
 101 : SPI マスタクロック / 64
 110 : SPI マスタクロック / 128
 111 : SPI マスタクロック / 256

注 : **MBR** 設定も **TI モード** で動作するスレーブで考慮されます([セクション 52.5.1: TI モード](#)を参照)。

ビット 27:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **CRCEN** : ハードウェア CRC 計算有効

0 : CRC 計算は無効です。
 1 : CRC 計算は有効です。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **CRCSIZE[4:0]** : トランザクションおよび比較される CRC フレームの長さ

CRC 結果がトランザクションまたは比較されるとき、多項式の計算から最上位の数ビットが考慮されます。多項式の長さは、この設定の影響を受けません。

00000 : 予約済み
 00001 : 予約済み
 00010 : 予約済み
 00011 : 4 ビット
 00100 : 5 ビット
 00101 : 6 ビット
 00110 : 7 ビット
 00111 : 8 ビット

.....

11101 : 30 ビット
 11110 : 31 ビット
 11111 : 32 ビット

値は、データサイズ (DSIZE[4:0]) と同じか、その倍数でセットする必要があります。最大サイズは、インスタンスでの DSIZE の最大値に対応します。

注 : **CRCSIZE** ビットフィールドの最上位ビットは、データサイズが 16 ビットに限定されているペリフェラルのインスタンスで予約済みです。

ビット 15 **TXDMAEN** : Tx DMA ストリーム有効

0 : Tx DMA は無効です。
 1 : Tx DMA は有効です。

ビット 14 **RxDMAEN** : Rx DMA ストリーム有効

0 : Rx-DMA は無効です。
 1 : Rx-DMA は有効です。

ビット 13:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **UDRCFG** : アンダーラン状態でのスレーブトランスミッタの動作

0 : スレーブは、ユーザによって SPI_UDRDR レジスタに定義された一定のパターンを送信します。
 1 : スレーブは、マスタから最後に受信したデータを繰り返します。スレーブが送信専用モード (COMM[1:0] = 01) で設定されている場合、すべてゼロのパターンが繰り返されます。
 詳細については、[図 729 : スレーブ検出中のアンダーラン状態に関するオプションの設定](#)を参照してください。

ビット 8:5 FTHLV[3:0] : FIFO 閾値レベル

1 つのデータパケットのデータフレーム数を定義します。パケットのサイズは FIFO スペースの 1/2 を超えてはいけません。

0000 : 1 データ
 0001 : 2 データ
 0010 : 3 データ
 0011 : 4 データ
 0100 : 5 データ
 0101 : 6 データ
 0110 : 7 データ
 0111 : 8 データ
 1000 : 9 データ
 1001 : 10 データ
 1010 : 11 データ
 1011 : 12 データ
 1100 : 13 データ
 1101 : 14 データ
 1110 : 15 データ
 1111 : 16 データ

以下のように、設定されたパケットサイズがデータレジスタのアクセス並行度に整合する場合、SPI インタフェースはより効率的になります。

- SPI データレジスタが 16 ビットレジスタとしてアクセスされ、DSIZE ≤ 8 ビットの場合、FTHLV=2、4、6 を選択することを推奨します。
- SPI データレジスタが 32 ビットレジスタとしてアクセスされ、DSIZE > 8 ビットの場合、FTHLV=2、4、6 を選択し、DSIZE ≤ 8 ビットの場合は FTHLV=4、8、12 を選択することを推奨します。

注： FTHLV[3:2] ビットは機能が制限されているインスタンスで予約済みです。

ビット 4:0 DSIZE[4:0] : 1つの SPI データフレームのビット数

00000 : 未使用
 00001 : 未使用
 00010 : 未使用
 00011 : 4 ビット
 00100 : 5 ビット
 00101 : 6 ビット
 00110 : 7 ビット
 00111 : 8 ビット

 11101 : 30 ビット
 11110 : 31 ビット
 11111 : 32 ビット

注： 最大データサイズは、一部のインスタンスでは 16 ビットまで制限可能です。機能が制限されているインスタンスでは、DSIZE[2:0] ビットは予約済みで、リセット状態で保持する必要があります。その後、DSIZE[4:3] ビットは、次のデータサイズの設定を制御します。

00xxx : 8 ビット
 01xxx : 16 ビット
 10xxx : 24 ビット
 11xxx : 32 ビット

52.11.4 SPI/I2S 設定レジスタ 2 (SPI_CFG2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

このレジスタの内容は、SPI が有効になっているか、または SPI_CR1 レジスタで IOLOCK ビットがセットされているとき、書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFCNTR	SSOM	SSOE	SSIOP	Res.	SSM	CPOL	CPHA	LSBFRST	マスタ	SP[2:0]			COMM[1:0]		Res.
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOSWP	RDIOP	RDIOM	Res.	Res.	Res.	Res.	Res.	MIDI[3:0]				MIDI[3:0]			
rw	rw	rw						rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **AFCNTR** : GPIO オルタネート機能制御

このビットは、SPE = 0 のときのみ考慮されます。

0 : ペリフェラルは無効になっているとき、GPIO の制御をしません。

1 : ペリフェラルは常に、関連するすべての GPIO を制御します。

特定の設定の理由（たとえば、CRC リセット、CPHA、または HDDIR 変更）のために SPI を一時的に無効にする必要があるときに、このビットをセットすると、オルタネート機能モードで設定された関連出力を、現在の SPI 設定に対応する状態に強制することによって、そのグリッチを防止できます。

注 : このビットは、PCM および I²S モードでも使用できます。**注 :** ブロックがスレープモードのとき、AFCNTR ビットをセットすることはできません。ビット 30 **SSOM** : マスタモードでの SS 出力管理

このビットは SSOE が有効なときにマスタモードで考慮されます。2 つの連続したデータ転送の間で SS 出力を設定することが可能になります。

0 : データ転送が完了するまで、SS はアクティブレベルに保持され、EOT フラグによって非アクティブになります。

1 : MIDI[3:0] > 1 のとき、SPI データフレームは SS の非アクティブパルスによってインターリーブされます。

ビット 29 **SSOE** : SS 出力有効

このビットは、マスタモードでのみ考慮されます。

0 : SS 出力は無効にされ、SPI はマルチマスタ設定で機能できます。

1 : SS 出力は有効です。SPI はマルチマスタ環境では機能できません。SSOM、MIDI、MSSI、SSIOP ビットの設定に従って転送を完了または SPI を無効化した後で、SS ピンを非アクティブレベルに強制します。

ビット 28 **SSIOP** : SS 入出力の極性

0 : SS 信号に関してローレベルがアクティブです。

1 : SS 信号に関してハイレベルがアクティブです。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **SSM** : SS 信号入力のソフトウェア管理

0 : SS 入力値は SS PAD によって決定されます。

1 : SS 入力値は SSI ビットによって決定されます。

マスタがハードウェアの SS 出力 (SSM = 0 および SSOE = 1) を使用する場合、SS 信号入力はマスタモードのフォールトエラーを防ぐために内部的に非アクティブ状態に強制されます。

ビット 25 **CPOL** : クロック極性

0 : アイドル時に SCK 信号は 0 となります。

1 : アイドル時に SCK 信号は 1 となります。

ビット 24 **CPHA** : クロックフェーズ

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2番目のクロック遷移が最初のデータキャプチャエッジです。

ビット 23 **LSBFRST** : データフレームフォーマット

0 : MSB が最初に送信されます。

1 : LSB が最初に送信されます。

注 : このビットは、PCM および I²S モードでも使用できます。

ビット 22 **MASTER** : SPI マスタ

0 : SPI スレーブ

1 : SPI マスタ

ビット 21:19 **SP[2:0]** : シリアルプロトコル

000 : SPI モトローラ

001 : SPI TI

その他。予約済み。使用できません。

ビット 18:17 **COMM[1:0]** : SPI 通信モード

00 : 全二重

01 : 単方向トランスミッタ

10 : 単方向レシーバ

11 : 半二重

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **IOSWP** : MISO および MOSI ピンのスワップ機能

0 : スワップなし。

1 : MOSI および MISO はスワップされます。

このビットがセットされた場合、MISO および MOSI ピンのオルタネートの機能は逆になります。

元の MISO ピンが MOSI になり、元の MOSI ピンが MISO になります。

注 : このビットは、SDO および SDI ピンをスワップするために、PCM および I²S モードでも使用できます。

ビット 14 **RDIOP** : RDY 信号の入出力の極性

RDIOPが0の場合、信号がハイレベルにおいてスレーブが通信できる状態であることを意味します。

RDIOPが1の場合、信号がローレベルにおいてスレーブが通信できる状態であることを意味します。

ビット 13 **RDIOM** : RDY 信号の入出力の管理

0 : RDY 信号は、永続的にアクティブとして内部的に固定されていると定義されます (RDIOP 設定は無効です)。

1 : RDY 信号は専用ピン (RDIOP 設定が有効) のオルタネート機能入力 (マスタケース) または出力 (スレーブケース) から得られます。

注 : SPI_CFG1 レジスタの DSIZE が 8 ビット未満で設定されている場合、RDIOM ビットはゼロで保持する必要があります。

ビット 12:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **MIDI[3:0]** : マスタのデータ間アイドル状態

マスタモードで 2 つの連続するデータフレーム間に挿入される最小時間遅延 (SPI クロックサイクルの周期で表される) を指定します。

0000 : 遅延なし

0001 : 1 クロックサイクル周期の遅延

.....

1111 : 15 クロックサイクル周期の遅延

注 : このビットは、TI モードではサポートされていません。

ビット 3:0 **MSSI[3:0]** : マスタの SS アイドル状態

SSOE が有効なとき、マスタモードで、セッションを開始する SS のアクティブエッジと、セッションの最初のデータ・フレームの初めの部分との間に追加で挿入される余分な遅延 (SPI クロックサイクル周期の数で表される) を指定します。

0000 : 余分な遅延なし

0001 : 1 クロックサイクル周期の遅延を付加

.....

1111 : 15 クロックサイクル周期の遅延を付加

注 : このビットは、TI モードではサポートされていません。

遅延を含めるには、SPI を無効にし、セッションを再度有効にする必要があります。

52.11.5 SPI/I2S 割込み有効レジスタ (SPI_IER)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DXPIE	TXPIE	RXPIE
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **MODFIE** : モードフォールト割込み有効

0 : MODF 割込み無効

1 : MODF 割込み有効

ビット 8 **TIFREIE** : TIFRE 割込み有効

0 : TIFRE 割込み無効

1 : TIFRE 割込み有効

ビット 7 **CRCEIE** : CRC エラー割込み有効

0 : CRC 割込み無効

1 : CRC 割込み有効

ビット 6 **OVRIE** : OVR 割込み有効

0 : OVR 割込み無効

1 : OVR 割込み有効

ビット 5 **UDRIE** : UDR 割込み有効

0 : UDR 割込み無効

1 : UDR 割込み有効

ビット 4 **TXTFIE** : TXTF 割込み有効

0 : TXTF 割込み無効

1 : TXTF 割込み有効

ビット 3 **EOTIE** : EOT、SUSP、および TXC 割込み有効

0 : EOT/SUSP/TXC 割込みは無効です。

1 : EOT/SUSP/TXC 割込みは有効です。

ビット 2 **DXPIE** : DXP 割込み有効

DXPIE は、ソフトウェアによってセットされ、TXTF フラグのセットイベントによってクリアされます。

0 : DXP 割込み無効

1 : DXP 割込み有効

ビット 1 **TXPIE** : TXP 割込み有効

TXPIE は、ソフトウェアによってセットされ、TXTF フラグのセットイベントによってクリアされます。

0 : TXP 割込み無効

1 : TXP 割込み有効

ビット 0 **RXPIE** : RXP 割込み有効

0 : RXP 割込み無効

1 : RXP 割込み有効

52.11.6 SPI/I2S ステータスレジスタ (SPI_SR)

アドレスオフセット : 0x14

リセット値 : 0x0000 1002

SPI を再度有効にしたときに、このレジスタのフラグはすべて、自動的にクリアされません。以下のビットの説明に示された、フラグクリアレジスタ経由の、クリアするための特定の専用アクセスが必要です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTSIZE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXWNE	RXPLVL[1:0]		TXC	SUSP	Res.	MODF	TIFRE	CRCE	OVR	UDR	TXTF	EOT	DXP	TXP	RXP
r	r	r	r	r		r	r	r	r	r	r	r	r	r	r

ビット 31:16 **CTSIZE[15:0]** : 現在の TSIZE セッションで残っているデータフレームの数

この値は、トラフィックがバスで進行中は完全には信頼できません。

注 : **CTSIZE[15:0]** ビットは機能が制限されているインスタンスでは使用できません。

ビット 15 **RXWNE** : RxFIFO ワードノットエンプティ

0 : RxFIFO スペースの 4 バイト未満がデータによって占有されます。

1 : RxFIFO スペースの 4 バイト以上がデータによって占有されます。

注 : このビット値は、DSIZE 設定に依存せず、残留データによる RxFIFO の占有に関する RXPLVL[1:0] の情報と一緒に保持されます。

ビット 14:13 **RXPLVL[1:0]** : RxFIFO パッキングレベル

RXWNE = 0 で、データサイズが最大 16 ビットの場合、この値は RxFIFO で続いている残りのデータフレーム数を示します。

00 : RxFIFO で使用可能な次のフレームはありません。

01 : 使用可能なフレームが 1 つあります。

10 : 使用可能なフレームが 2 つあります*。

11 : 使用可能なフレームが 3 つあります*。

注： (*) : データサイズが最大 8 ビットにセットされた場合に設定可能な値。

データサイズが 16 ビットより大きい場合、これらのビットは常に 00 として読み出されます。その場合、FIFO で受信した 1 つのデータフレームは、データサイズが 17 ~ 24 ビットの場合、RWNE ビットでも RXPLVL ビットでも検出することはできません。ユーザは、受信したデータ数を検出するため、TSIZE > 0 の場合の EOT イベントや FTHLV = 0 の場合の RXP イベントの監視など、その他の方法を適用する必要があります。

ビット 12 **TXC** : TxFIFO 送信完了

フラグの動作は TSIZE 設定に依存します。

TSIZE=0 のとき、TXC はハードウェアによってのみ変更され、TxFIFO が空になってバス上での動作がなくなるたびに発生します。

TSIZE ≠ 0 の場合、ソフトウェアのクリアを含む EOT フラグ値をコピーするだけの TXC を監視する理由は特にありません。EOTIE がセットされているとき、TXC は割込みを生成します。

このフラグは、SPI がリセットまたは無効化されるとセットされます。

0 : 現在のデータのトランザクションはまだ進行中です。データは TxFIFO で利用可能になっているか、または最後のフレームの送信が進行中です。

1 : 最後の TxFIFO フレーム送信完了

ビット 11 **SUSP** : サスペンド・ステータス

マスタモードにおいて、CSUSP リクエストが出され、現在のフレームが完了したとき、またはマスタ自動サスペンド受信モード (SPI_CR1 レジスタの MASRX ビットがセットされている) で RxFIFO がフル状態になったとき、直ちに SUSP がハードウェアによってセットされます。

EOTIE がセットされているとき、SUSP は割込みを生成します。

このビットは、SPI を無効にする前にクリアする必要があります。これを行うには、SPI_IFCR の SUSPC ビットに 1 だけを書き込みます。

0 : SPI はサスペンドされていません (マスタモードまたは他のモードがアクティブです)。

1 : マスタモードはサスペンドされています (現在のフレームは完了しています)。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **MODF** : モードフォールト

0 : モードフォールトはありません。

1 : モードフォールトを検出しました。

MODF をセットすると、SPI_CR1 レジスタの SPE および IOLOCK ビットがリセットされ、MODF がクリアされるまで、SPE の再設定はブロックされます。

このビットは、排他的に SPI_IFCR の MODFC ビットに 1 を書き込むことでクリアされます。

ビット 8 **TIFRE** : TI フレームフォーマットエラー

0 : TI フレームエラーはありません。

1 : TI フレームエラーを検出しました。

このビットは、排他的に SPI_IFCR の TIFREC ビットに 1 を書き込むことでクリアされます。

ビット 7 **CRCE** : CRC エラー

0 : CRC エラーはありません。

1 : CRC エラーが検出されました。

このビットは、オプションで SPI を再度有効にするか、SPI_IFCR の CRCEC ビットに 1 を書き込むことでクリアされます。

ビット 6 OVR : オーバーラン

0 : オーバーランはありません。

1 : オーバーランが検出されました。

このビットは、オプションで SPI を再度有効にするか、SPI_IFCR の OVRC ビットに 1 を書き込むことでクリアされます。

ビット 5 UDR : アンダーラン

0 : アンダーランはありません。

1 : アンダーランが検出されました。

このビットは、オプションで SPI を再度有効にするか、SPI_IFCR の UDRC ビットに 1 を書き込むことでクリアされます。

注 : SPI モードで、UDR フラグはスレーブモードにのみ適用されます。I2S/PCM モードで、(使用可能な場合は) このフラグはマスタおよびスレーブモードに適用されます。

ビット 4 TXTF : 送信の転送フル

0 : TxFIFO のアップロードは処理中または開始されていません。

1 : TxFIFO のアップロードは終了しています。

アプリケーションソフトウェアまたは DMA によって転送内のすべてのデータパケットが送信用に提出されると、つまり TSIZE 数のデータが TxFIFO にプッシュされた時、すぐにハードウェアによって TXTF がセットされます。

このビットは、排他的にソフトウェアによって SPI_IFCR の TXTFC ビットに 1 を書き込むことでクリアされます。

TXTFIE ビットがセットされている場合、TXTF フラグは割り込みをトリガします。

TXTF をセットすると TXPIE および DXPIE マスクがクリアされるので、アプリケーションソフトウェアは TXP および DXP 割り込みをいつ無効にするかの計算の負担を免れます。

ビット 3 EOT : 転送終了

全転送が完了すると、つまり SPI が再有効化されたときや、SPI で TSIZE 数のデータが送信および/または受信されたときに、すぐにハードウェアによって EOT がセットされます。EOT は、オプションで SPI を再度有効にするか、SPI_IFCR の EOTC ビットに 1 を書き込むことでクリアされます。

EOTIE ビットがセットされている場合、EOT フラグは割り込みをトリガします。

TXTF フラグがセットされるまで DXP フラグが使用され、DXPIE がクリアされている場合、RxFIFO に入っている最後のパケットを 1 回でダウンロードするために EOT を使用することができます。

0 : 転送は処理中または開始されていません。

1 : 転送が完了しました。

マスタでは、EOT イベントによって、データのトランザクションが終了し、オプションで SS 出力が処理されます。CRC が適用されると、EOT イベントは CRC フレームのトランザクションまで拡張されます。

内部ステートマシンを適切に再開するには、設定を変更しない場合でも、次のトランザクションが始まる前に SPI を無効にして再度有効化することを強く推奨します。

ビット 2 DXP : 二重パケット

0 : TxFIFO はフルで、および/または RxFIFO は空です。

1 : TxFIFO には書き込むスペースがあり、RxFIFO には少なくとも 1 つの読み出すパケットがあります。

DXP フラグは、TXP と RXP の両フラグがセットされたとき、SPI モードにかかわらずセットされます。

ビット 1 TXP : 送信パケットスペース利用可能

0 : TxFIFO に次のデータパケットを提供するための十分な空きスペースがありません。

1 : TxFIFO に 1 つ以上のデータパケットを提供するための十分な空きスペースがあります。

TXP フラグはハードウェアによってのみ変更できます。その値は FIFO の物理サイズとその閾値 (FTHLV[3:0])、データフレームサイズ (SPI モードの DSIZE[4:0] および I2S/PCM モードの対応する DATLEN[1:0])、および実際の通信フローに依存します。データパケットが SPI_TXDR に連続書き込みを行うことで格納される場合、完了したデータパケットが TxFIFO に格納されたら、TXP フラグを再度確認する必要があります。TXP は、SPI のリセットまたは無効化により、SPI TxFIFO がアクセスできなくなってもセットされます。

ビット 0 **RXP** : 受信パケット使用可能

0 : RxFIFO は空であるか、または不完全なデータパケットが受信されています。

1 : RxFIFO には少なくとも 1 つのデータパケットが含まれています。

このフラグは、ハードウェアによって変更されます。SPI が有効な場合、RxFIFO で現在使用可能な全体データ数を監視します。RXP の値は FIFO の閾値 (FTHLV[3:0])、データフレームサイズ (SPI モードの DSIZE[4:0] および I2S/PCM モードの DATLEN[1:0])、および実際の通信フローに依存します。データパケットが SPI_RXDR から連続読出しを行うことで読み出される場合、完了したデータパケットが RxFIFO から読み出されたら、RXP フラグを再度確認する必要があります。

52.11.7 SPI/I2S 割込み/ステータスフラグクリアレジスタ (SPI_IFCR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SUSPC	Res.	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res.	Res.	Res.
				w		w	w	w	w	w	w	w			

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SUSPC** : サスペンドフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの SUSP フラグがクリアされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **MODFC** : モードフォールトフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの MODF フラグがクリアされます。

ビット 8 **TIFREC** : TI フレームフォーマットエラーフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの TIFRE フラグがクリアされます。

ビット 7 **CRCEC** : CRC エラーフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの CRCE フラグがクリアされます。

ビット 6 **OVRC** : オーバーランフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの OVR フラグがクリアされます。

ビット 5 **UDRC** : アンダーランフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの UDR フラグがクリアされます。

ビット 4 **TXTFC** : 送信転送フルフラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの TXTF フラグがクリアされます。

ビット 3 **EOTC** : 転送の終了フラグクリア

このビットに 1 を書き込むと、SPI_SR レジスタの EOT フラグがクリアされます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

52.11.8 SPI/I2S 送信データレジスタ (SPI_TXDR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXDR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **TXDR[31:0]** : 送信データレジスタ

このデータレジスタは、TxFIFO とのインタフェースとして使用できます。これに書き込むと TxFIFO へアクセスします。

注 : SPI モードでは、データは常に右詰めです。I²S モードでのデータの配置は DATLEN および DATFMT 設定に依存します。未使用のビットは、レジスタへの書き込み時に無視され、レジスタの読出し時にゼロとして読み出されます。

注 : DR はバイト単位に (8 ビットアクセス) アクセス可能です。この場合、1 回のアクセスで 1 データバイトのみが書き込まれます。

ハーフワード単位 (16 ビットアクセス) の場合、1 回のアクセスで 2 データバイトまたは 1 ハーフワードデータが書き込まれます。

ワード単位 (32 ビットアクセス) の場合、1 回のアクセスで 4 データバイト、または 2 ハーフワードデータ、またはワードデータが書き込まれます。

設定したデータサイズより小さいレジスタの書き込みアクセスは禁止されます。

52.11.9 SPI/I2S 受信データレジスタ (SPI_RXDR)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXDR[31:0]** : 受信データレジスタ

このレジスタは、RxFIFO とのインタフェースとして使用できます。これが読み出されるとき、RxFIFO がアクセスされます。

注 : SPI モードでは、データは常に右詰めです。I²S モードでのデータの配置は DATLEN および DATFMT 設定に依存します。未使用のビットは、レジスタの読出し時にはゼロとして読み出されます。このレジスタへの書き込みは無視されます。

注 : DR はバイト単位 (8 ビットアクセス) でアクセス可能です。この場合、1 回のアクセスで 1 データバイトのみが読み出されます。

ハーフワード単位 (16 ビットアクセス) の場合、1 回のアクセスで 2 データバイトまたは 1 ハーフワードデータが読み出されます。

ワード単位 (32 ビットアクセス) の場合、1 回のアクセスで 4 データバイト、または 2 ハーフワードデータ、またはワードデータが読み出されます。

設定したデータサイズより小さいレジスタの読出しアクセスは禁止されます。

52.11.10 SPI/I2S 多項式レジスタ (SPI_CRCPOLY)

アドレスオフセット : 0x40

リセット値 : 0x0000 0107

このレジスタの内容は、SPI が有効になっているとき、書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRCPOLY[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 CRCPOLY[31:0] : CRC 多項式レジスタ

このレジスタは、CRC 計算用の多項式を格納します。

デフォルトの 9 ビット多項式の設定 0x107 は DSIZE がデフォルトの 8 ビット設定に対応しています。これは、ストリングの最上位ビットが常に隠されている固定長の多項式ストリングを備えた他のいくつかの ST 製品で使用されている設定 0x07 と互換性があります。

多項式の長さはこのレジスタに格納されている値の最上位ビットによって与えられます。その長さは DSIZE より大きく設定する必要があります。DSIZE が最大の 32 ビットまたは 16 ビットのサイズに設定され、CRC が有効になっているとき、多項式の長さをデータサイズより大きく保つために、CRCPOLY レジスタとともにさらに CRC33_17 ビットをセットする必要があります。

注： CRCPOLY[31:16] ビットは、16 ビットに制限されたデータサイズのインスタンスで予約済みです。これらのアドレスに 32 ビットアクセスが適用される場合、何も制約はありません。予約済みのビット 31~16 は常にゼロが読み出され、それらへの書き込みは無視されます。

52.11.11 SPI/I2S トランスミッタ CRC レジスタ (SPI_TXCRC)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXCRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 TXCRC[31:0] : トランスミッタ用 CRC レジスタ

CRC 計算が有効なとき、TXCRC[31:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。CRC 計算は、SPI_CR1 の CRCEN ビットがセットされたとき、またはデータブロックが完全にトランザクションされたときに初期化されます。CRC は、SPI_CRCPOLY レジスタにプログラムされた多項式を使用して連続的に計算されます。

計算で考慮されるビット数は、SPI_CRCPOLY レジスタと、SPI_CFG1 レジスタの CRCSIZE ビットの設定に依存します。

注： 通信が進行中にこのレジスタを読み出すと、誤った値が返されることがあります。

注： このビットフィールドは I2S モードでは使用しません。

注： TXCRC[31:16] ビットは、16 ビットに制限されたデータサイズのインスタンスで予約済みです。これらのアドレスに 32 ビットアクセスが適用される場合、何も制約はありません。予約済みのビット 31~16 は常にゼロが読み出され、それらへの書き込みは無視されます。

注： このレジスタの内容がソフトウェアによって読み出された場合、CRCSIZE ビットフィールドの設定は考慮されません。この場合、未使用ビットにマスクは適用されません。

52.11.12 SPI/I2S レシーバ CRC レジスタ (SPI_RXCRC)

アドレスオフセット : 0x48

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXCRC[31:0]** : レシーバ用 CRC レジスタ

CRC 計算が有効なとき、RXCRC[31:0] ビットには、その後に受信したバイトから算出された CRC 値が格納されています。CRC 計算は、SPI_CR1 の CRCEN ビットがセットされたとき、またはデータブロックが完全にトランザクションされたときに初期化されます。CRC は、SPI_CRCPOLY レジスタにプログラムされた多項式を使用して連続的に計算されます。

計算で考慮されるビット数は、SPI_CRCPOLY レジスタと、SPI_CFG1 レジスタの CRCSIZE ビットの設定に依存します。

注： 通信が進行中にこのレジスタを読み出すと、誤った値が返されることがあります。

このビットフィールドは I2S モードでは使用しません。

RXCRC[31:16] ビットは、16 ビットに制限されたデータサイズのペリフェラルインスタンスで予約済みです。これらのアドレスに 32 ビットアクセスが適用される場合、何も制約はありません。予約済みのビット 31~16 は常にゼロが読み出され、それらへの書き込みは無視されます。

注： このレジスタの内容がソフトウェアによって読み出された場合、CRCSIZE ビットフィールドの設定は考慮されません。この場合、未使用ビットにマスクは適用されません。

52.11.13 SPI/I2S アンダーランデータレジスタ (SPI_UDRDR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

このレジスタの内容は、SPI が有効になっているとき、書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UDRDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDRDR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **UDRDR[31:0]** : スレーブのアンダーラン状態でのデータ

このレジスタは、スレーブモードで、アンダーラン状態でのみ考慮されます。考慮されるビット数は、SPI_CFG1 レジスタの DSIZE ビットの設定に依存します。アンダーラン状態の処理は SPI_CFG1 レジスタの UDRCFG ビットの設定に依存します。

注： UDRDR[31:16] ビットは、16 ビットに制限されたデータサイズのペリフェラルインスタンスで予約済みです。これらのアドレスに 32 ビットアクセスが適用される場合、何も制約はありません。予約済みのビット 31~16 は常にゼロが読み出され、それらへの書き込みは無視されます。

52.11.14 SPI/I2S 設定レジスタ (SPI_I2SCFGR)

アドレスオフセット : 0x50

リセット値 : 0x0000 0000

このレジスタは、I²S が無効 (SPE=0) のときに設定する必要があります。このレジスタの内容は、0 で保持する必要のある I2SMOD ビットを除いて、SPI モードでは考慮されません。

このレジスタは、I²S モードをサポートしないインスタンスで予約済みです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV [7:0]							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DATFMT	WSINV	FIXCH	CKPOL	CHLEN	DATLEN[1:0]		PCMSYNC	Res.	I2SSTD [1:0]		I2SCFG [2:0]		I2SMOD	
	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **MCKOE** : マスタクロック出力イネーブル

0 : マスタクロック出力は無効です。

1 : マスタクロック出力は有効です。

ビット 24 **ODD** : プリスケアラの奇数分周比

0 : 真の分周値 = I2SDIV * 2

1 : 真の分周値 = (I2SDIV * 2) + 1

詳細については、[セクション 52.9.9 : クロックジェネレータ](#)を参照してください。

ビット 23:16 **I2SDIV[7:0]** : I²S リニアプリスケアラ

ODD が 1 のとき、I2SDIV は 1 の値を除いて任意の値をとることができます。

詳細については、[セクション 52.9.9 : クロックジェネレータ](#)を参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **DATFMT** : データフォーマット

0 : SPI_RXDR または SPI_TXDR 内のデータは右詰めです。

1 : SPI_RXDR または SPI_TXDR 内のデータは左詰めです。

ビット 13 **WSINV** : ワード選択の反転

このビットは、WS 信号のデフォルト極性を反転するために使用します。

0 : フィリップス I2S 規格では、左チャンネル転送は WS の立下りエッジの 1 CK サイクル後に開始され、右チャンネルは WS の立上りエッジの 1 CK サイクル後に開始されます。

MSB または LSB 詰めモードでは、WS が HIGH のとき、左チャンネルが転送され、WS が LOW のとき、右チャンネルが転送されます。

PCM のショートモードは PCM のロングモードで WS の立上りエッジで開始しますが、そのデータ転送は WS の立下りエッジから開始します。

1 : フィリップス I2S 規格では、左チャンネル転送は WS の立上りエッジの 1 CK サイクル後に開始され、右チャンネルは WS の立下りエッジの 1 CK サイクル後に開始されます。

MSB または LSB 詰めモードでは、WS が LOW のとき、左チャンネルが転送され、WS が HIGH のとき、右チャンネルが転送されます。

PCM のショートモードは PCM のロングモードで WS の立下りエッジで開始しますが、そのデータ転送は WS の立上りエッジから開始します。

ビット 12 **FIXCH** : スレーブでの固定チャンネル長

0 : スレーブモードでのチャンネル長は 16 または 32 ビットと異なります (CHLEN は考慮されません)。

1 : スレーブモードでのチャンネル長は 16 または 32 ビットと推測されます (CHLEN による)。

ビット 11 **CKPOL** : シリアルオーディオクロックの極性

0 : SPI/I2S によって生成される信号 (すなわち SDO および WS) は CK の立下りエッジで変化し、SPI/I2S によって受信される信号 (すなわち SDI および WS) は CK の立上りエッジで読み出されます。

1 : SPI/I2S によって生成される信号 (すなわち SDO および WS) は CK の立上りエッジで変化し、SPI/I2S によって受信される信号 (すなわち SDI および WS) は CK の立下りエッジで読み出されます。

ビット 10 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

0 : 16 ビット幅

1 : 32 ビット幅

ビット 9:8 **DATLEN[1:0]** : 転送されるデータ長

00 : 16 ビットデータ長

01 : 24 ビットデータ長

10 : 32 ビットデータ長

11 : 設定禁止

注 : 24 および 32 ビットのデータ幅は常にサポートされません (DATLEN = 01 または 10)。サポートされるデータサイズを確認するには、[セクション 52.3 : SPI の実装](#)を参照してください。

ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期

1 : ロングフレーム同期

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **I2SSTD[1:0]** : I²S 規格選択

00 : フィリップス I²S 規格

01 : MSB 詰め規格 (左詰め)

10 : LSB 詰め規格 (右詰め)

11 : PCM 規格

I²S 規格に関する詳細は、[セクション 52.9.5 : サポートされるオーディオプロトコル](#)を参照してください。

ビット 3:1 **I2SCFG[2:0]** : I2S 設定モード

000 : スレーブ - 送信

001 : スレーブ - 受信

010 : マスタ - 送信

011 : マスタ - 受信

100 : スレーブ - 全二重

101 : マスタ - 全二重

その他、未使用

ビット 0 **I2SMOD** : I2S モード選択

0 : SPI モードが選択されます。

1 : I2S/PCM モードが選択されます。

52.11.15 SPI/I2S レジスタマップ

表 562. SPI レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	SPI_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOLOCK	TCRCINI	RCRCINI	CRC33_17	SSI	HDDIR	CSUSP	CSTART	MASRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE	
	リセット値																0	0	0	0	0	0	0	0	0								0	
0x04	SPI_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSIZE[15:0]																	
	リセット値																	(1)	(1)	(1)	(1)	(1)	(1)	0	0	0	0	0	0	0	0	0	0	
0x08	SPI_CFG1	BPASS	MBR[2:0]			Res.	Res.	Res.	Res.	Res.	CRCE	Res.	CRCSIZE[4:0]				TXDMAEN	RCDMAEN	Res.	Res.	Res.	Res.	UDRCFG	FTHLV[3:0]			DSIZE[4:0]							
	リセット値	0	0	0	0						0		1	1	1	1	1	0	0				0	0 ⁽¹⁾	0 ⁽¹⁾	0	0	0	0	0	0 ⁽¹⁾	1 ⁽¹⁾	1 ⁽¹⁾	
0x0C	SPI_CFG2	AFNTR	SSOM	SSOE	SSIOP	SSM	CPOL	CPHA	LSBFIRST	マスタ		SP[2:0]		COMM	[1:0]	Res.	IOSWP	RDIOP	RDIOM	Res.	Res.	Res.	Res.	Res.	MIDI[3:0]			MIDI[3:0]						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SPI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DXPIE	TXPIE	RXPIE	
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	
0x14	SPI_SR	CTSIZE[15:0] ⁽¹⁾															RXWNE	RXPVL	[1:0]	TXC	SUSP	Res.	MODF	TIFRE	CRCE	OVR	UDR	TXTF	EOT	DPXP	TXP	RXP		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0		0	0	0	0	0	0	0	0	0	1	0	
0x18	SPI_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSPC	SUSP	Res.	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res.	Res.	Res.		
	リセット値																			0	0		0	0	0	0	0	0	0					
0x1C	予約済みです。																																	
0x20	SPI_TXDR	TXDR[31:16]															TXDR[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x24-0x2C	予約済みです。																																	
0x30	SPI_RXDR	RXDR[31:16]															RXDR[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x34 - 0x3C	予約済みです。																																	
0x40	SPI_CRCPOLY	CRCPOLY[31:16] ⁽²⁾															CRCPOLY[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	
0x44	SPI_TXCRC	TXCRC[31:16] ⁽²⁾															TXCRC[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x48	SPI_RXCRC	RXCRC[31:16] ⁽²⁾															RXCRC[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x4C	SPI_UDRDR	UDRDR[31:16] ⁽²⁾															UDRDR[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50	SPI_I2SCFGR	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV [7:0]								Res.	Res.	WSINV	FIXCH	CKPOL	CHLEN	DATLEN[1:0]		PCMSSYNC	Res.	I2SSTD [1:0]		I2SCFG [2:0]		I2SMOD			
	リセット値						0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	

- ビットフィールドは、機能が制限されているインスタンスのために予約済みで、リセット値で保持する必要があります。詳細については、[セクション 52.11 : SPI/I2S レジスタ](#)の具体的なレジスタの説明を参照してください。
- ビット 31 ~ 16 は、データサイズが 16 ビットに限定されているペリフェラルのインスタンスで予約済みです。これらのアドレスに 32 ビットアクセスが適用される場合、何も制約はありません。予約済みのビット 31 ~ 16 は常にゼロが読み出され、それらへの書込みは無視されます。

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

53 シリアルオーディオインタフェース (SAI)

53.1 概要

この SAI インタフェース (シリアルオーディオインタフェース) は、柔軟性があり幅広い構成が可能のため、多様なオーディオプロトコルに対応できます。多くのステレオまたはモノラルオーディオアプリケーションが対象になります。例えば、I2S 規格、LSB/MSB 詰め、PCM/DSP、TDM、AC'97 の各プロトコルに対応することができます。オーディオブロックがトランスミッタとして設定された場合、SPDIF 出力が提供されます。

このレベルの柔軟性と再設定機能を実現するため、SAI は独立した 2 つのオーディオサブブロックを内蔵しています。各ブロックには専用のクロックジェネレータと I/O ラインコントローラが備わっています。

この SAI は、マスタまたはスレーブの設定で動作します。このオーディオサブブロックはレシーバにもトランスミッタにもなり、他の一方に対して同期または非同期で動作します。

SAI を他の SAI と接続することで、互いに同期して動作することができます。

53.2 SAI の主な機能

- 個別に FIFO を内蔵し、トランスミッタまたはレシーバとなることができる 2 つの独立したオーディオサブブロック。
- 各オーディオサブブロックに 8 ワード FIFO を内蔵。
- オーディオサブブロック間での同期モードまたは非同期モードが可能。
- 複数の SAI 間の同期が可能。
- 両オーディオサブブロックに対して独立してマスタ設定またはスレーブ設定が可能。
- 両オーディオサブブロックをマスタモードに設定した場合に、個別のオーディオサンプリング周波数に対応する各オーディオブロック用クロックジェネレータ。
- 設定可能なデータサイズ : 8、10、16、20、24、32 ビット
- オーディオプロトコル : I2S、LSB/MSB 詰め、PCM/DSP、TDM、AC'97
- 最大 4 台のマイクロフォンのペアをサポートする PDM インタフェース。
- 必要に応じて、SPDIF 出力を利用可能。
- サイズ設定が可能な最大 16 個のスロットが利用可能。
- フレームごとのビット数が設定可能。
- フレーム同期アクティブレベルが設定可能 (オフセット、ビット長、レベル)。
- スロット内の先頭アクティブビット位置が設定可能。
- LSB が先にまたは MSB が先にデータ転送が可能。
- ミュートモード。
- ステレオ/モノラルオーディオフレーム機能。
- 通信クロックストローブエッジが設定可能 (SCK)。

- 対応する割込み付きのエラーフラグ（それぞれイネーブル付き）。
 - オーバーラン検出およびアンダーラン検出、
 - スレーブモードにおける予測フレーム同期信号検出、
 - スレーブモードにおけるレイトフレーム同期信号検出、
 - 受信時 AC'97 モードに対するコーデックノットレディ
- 割込みソース（有効時）：
 - エラー
 - FIFO リクエスト
- 2 チャンネル DMA インタフェース

53.3 SAI の実装

表 563. STM32H563/H573および STM32H562 SAI の機能 ⁽¹⁾

SAI の機能	SAI1	SAI2
I2S、LSB/MSB 詰め、PCM/DSP、TDM、AC'97	X	X
FIFO サイズ	8 ワード	8 ワード
SPDIF	X	X
PDM	X ⁽²⁾	-

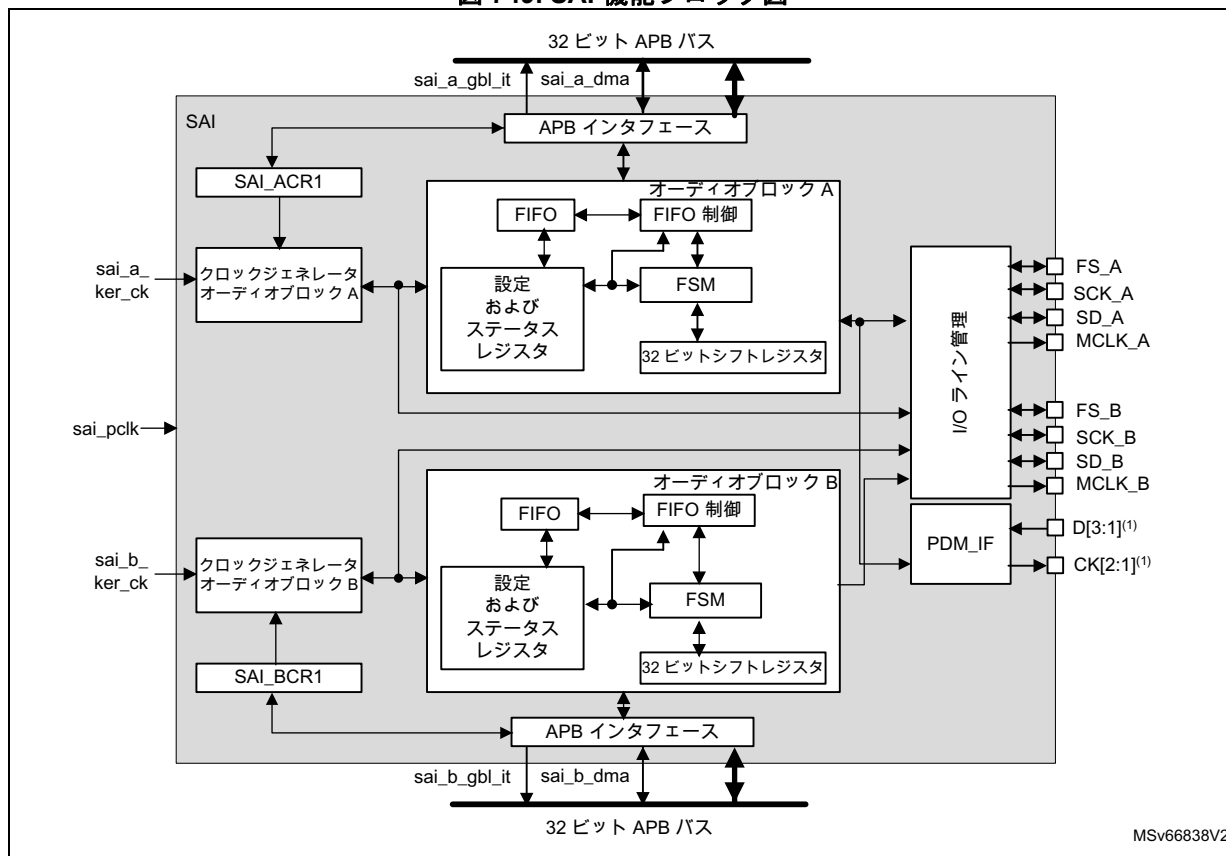
1. “X” はサポートされています。“-” はサポートされていません。
2. D[3:1] および CK[2:1] の信号のみ使用可能です。

53.4 SAI の機能説明

53.4.1 SAI ブロック図

図 749 に SAI ブロック図を、表 564 および表 565 に SAI 内部および外部信号のリストを示します。

図 749. SAI 機能ブロック図



1. これらの信号は、すべての SAI インタフェースで利用できるわけではありません。詳細については、[セクション 53.3: SAI の実装](#)を参照してください。

この SAI は大部分が 2つのオーディオサブブロックから構成されており、それぞれに固有のクロックジェネレータを装備しています。各オーディオブロックは、それぞれの機能ステートマシンによって制御される 32 ビットシフトレジスタを内蔵しています。データは、専用 FIFO に保存されたり、そこから読み出されたりします。FIFO は CPU からアクセスすることもできれば、通信時に CPU を開放するため DMA からアクセスすることもできます。各オーディオブロックは独立しています。互いに同期することもできます。

I/O ラインコントローラは、SAI 内の指定オーディオブロックに対する 4 本の専用ピン (SD、SCK、FS、MCLK) を管理します。この 2 つのサブブロックを同期するように指定すると、いくつかのピンを共用していくつかのピンを開放し、汎用 I/O として使用することができます。アプリケーション、デコーダ要件、オーディオブロックがマスタとして設定されているか否かによって、MCLK ピンを出力にしたり、それ以外にしたりすることができます。

1 つの SAI が別の SAI と同期して動作するように設定されている場合、より多くの I/O を開放することができます (SD_x ピンを除く)。

この機能ステートマシンは、広範囲なオーディオプロトコルを処理するように設定することができます。所望のプロトコルを設定するため、いくつかのレジスタがあります（オーディオフレーム波形ジェネレータ）。

オーディオサブブロックは、マスタモードまたはスレーブモードでトランスミッタまたはレシーバになることができます。マスタモードでは、SCK_x ビットクロックとフレーム同期信号が SAI で生成されますが、スレーブモードでは、これらの信号を別の外部マスタまたは内部マスタから受け取ります。FS 信号の方向がマスタモードまたはスレーブモードの定義に直接結びつかない特別なケースがあります。AC'97 プロトコルでは、SAI（リンクコントローラ）が SCK クロックを入力するように設定されている場合でも（つまりスレーブモードでも）、FS 信号は SAI の出力です。

注：読みやすくするために、このセクションでは SAI_A または SAI_B を SAI_x と表記しています。「x」とは SAI A サブブロックまたは SAI B サブブロックのことです。

53.4.2 SAI ピンおよび内部信号

表 564. SAI 内部入力／出力信号

内部信号名	信号タイプ	説明
sai_a_gbl_it/ sai_b_gbl_it	出力	オーディオブロック A および B グローバル割込み
sai_a_dma、sai_b_dma	入力／出力	オーディオブロック A および B DMA 確認応答およびリクエスト
sai_sync_out_sck、 sai_sync_out_fs	出力	他の SAI ブロックと交換される内部クロックおよびフレーム同期出力信号
sai_sync_in_sck、 sai_sync_in_fs	入力	他の SAI ブロックと交換される内部クロックおよびフレーム同期入力信号
sai_a_ker_ck/ sai_b_ker_ck	入力	オーディオブロック A/B カーネルクロック
sai_pclk	入力	APB クロック

表 565. SAI の入出力ピン

名前	信号タイプ	コメント
SAI_SCK_A/B	入力／出力	オーディオブロック A/B ビットクロック
SAI_MCLK_A/B	出力	オーディオブロック A/B マスタクロック
SAI_SD_A/B	入力／出力	ブロック A/B のデータライン
SAI_FS_A/B	入力／出力	オーディオブロック A/B のフレーム同期ライン
SAI_CK[2:1]	出力	PDM ビットストリームクロック ⁽¹⁾
SAI_D[3:1]	入力	PDM ビットストリームデータ ⁽¹⁾

1. これらの信号は、すべての SAI インタフェースで利用できるわけではありません。詳細については、[セクション 53.3：SAI の実装](#)を参照してください。

53.4.3 主要な SAI モード

SAI の各オーディオサブブロックは、選択したオーディオブロックの SAI_xCR1 レジスタの MODE ビットを使ってマスタまたはスレーブに設定することができます。

マスタモード

マスタモードでは、SAI は接続された外部デバイスへタイミング信号を送り出します。

- ビットクロックとフレーム同期はそれぞれ SCK_x ピンと FS_x ピンに出力されます。
- 必要であれば、SAI は MCLK_x ピンにマスタクロックを生成することもできます。

SCK_x、FS_x、および MCLK_x のいずれも出力として設定されます。

スレーブモード

SAI は、外部デバイスからのタイミング信号の受信を待ちます。

- SAI サブブロックが非同期モードに設定されている場合、SCK_x ピンおよび FS_x ピンは入力として設定されます。
- SAI サブブロックが他の SAI インタフェースまたは2番目のオーディオサブブロックと同期して動作するよう設定されている場合、対応する SCK_x および FS_x ピンは汎用 I/O として使用できます。

MCLK_x ピンはスレーブモードでは使用されず、別の機能に割り当てることができます。

スレーブデバイスを有効にした後でマスタを有効にすることをお勧めします。

SAI モードの設定と有効化

各オーディオサブブロックは、該当するオーディオブロックにある SAI_xCR1 レジスタの MODE ビットを使って個別にトランスミッタまたはレシーバに指定することができます。そのため、SAI_SD_x ピンは出力または入力としてそれぞれ設定されます。

同じ SAI にある 2 個のマスタオーディオブロックは、2 つの異なる MCLK および SCK クロック周波数で設定することができます。この場合、オーディオブロックは非同期モードに設定する必要があります。

SAI 内の各オーディオブロックは、SAI_xCR1 レジスタの SAIEN ビットにより有効化されます。このビットがアクティブになると直ちに、トランスミッタまたはレシーバはスレーブモードでクロックライン、データライン、同期ラインの動作に反応するようになります。

マスタ TX モードでは、オーディオブロックを有効化すると直ちに、FIFO にデータがなくとも、外部スレーブに対するビットクロックを生成しますが、FS 信号の生成は、FIFO 内のデータの存在が条件となります。FIFO が先頭の送信データを受け取った後、このデータが外部スレーブに出力されます。FIFO に送信データがない場合、オーディオフレームにおいては 0 値が送信されて、アンダーランフラグが生成されます。

スレーブモードでは、オーディオブロックが有効化されて、フレームの開始が検出されたときに、オーディオフレームが開始されます。

スレーブ TX モードでは、オーディオブロックを有効化した後の最初のフレームでアンダーランイベントが生成されることはありません。理由は、次の動作シーケンスが必要となるためです。

1. ソフトウェアまたは DMA により SAI_xDR へ書込みを行います。
2. FIFO 閾値 (FLH) フラグが 0b000 (FIFO エンプティ) でなくなるまで待ちます。
3. スレーブ送信ミッタモードでオーディオブロックを有効化します。

53.4.4 SAI 同期モード

オーディオサブブロックレベルと SAI レベルの 2 種類の同期レベルがあります。

内部同期

オーディオサブブロックは、同じ SAI 内の 2 番目のオーディオサブブロックと同期して動作できるよう設定できます。この場合、ビットクロック信号とフレーム同期信号が共用され、通信に使用する外部ピン数を少なくすることができます。同期モードに設定されたオーディオブロックでは、その SCK_x、FS_x、および MCLK_x ピンは GPIO として開放状態に戻されています。非同期モードに設定されたオーディオブロックのほうが、FS_x、SCK_x、MCLK_x の I/O ピンに適したブロックとなります（オーディオブロックをマスタと考えた場合）。

一般に、同期モードのオーディオブロックを使って、SAI を全二重モードに設定することができます。2 個のオーディオブロックの片方をマスタとして設定し、他方をスレーブとして設定することができます。あるいは、両方をスレーブとして設定し、1 つは非同期ブロック (SAI_xCR1 の対応する SYNCEN[1:0] ビットを 00 にセット) に、もう一方を同期ブロック (SAI_xCR1 の対応する SYNCEN[1:0] ビットを 01 にセット) にすることもできます。

注： 内部再同期ステージのため、PCLK APB 周波数はビットレートクロック周波数の 2 倍以上である必要があります。

外部同期

オーディオサブブロックは他の SAI と同期して動作するよう設定することもできます。この方法は以下の通りです。

1. SAI が、他の SAI の同期元として設定されている場合は、その相手の SAI に FS および SCK 信号を提供するオーディオサブブロックを定義する必要があります。これは SYNCOUT[1:0] ビットを設定することによって行います。
2. 同期信号を受信する SAI は、SYNCIN[1:0] ビットに適正な値をセットすることによって同期を提供する SAI を選択する必要があります。2 個の SAI オーディオサブブロックのそれぞれに対して、ユーザはそのサブブロックが他方の SAI と同期して動作するかどうかを SYNCEN ビットで指定します。

注： SYNCIN[1:0] および SYNCOUT[1:0] ビットは SAI_GCR レジスタ内にあり、SYNCEN ビットは SAI_xCR1 レジスタ内にあります。

SAI の 2 個のオーディオサブブロックがどちらも他の SAI と同期して動作する必要がある場合は、次の設定のいずれかを選択することができます。

- 他の SAI ブロックと同期させる各オーディオブロックを SYNCEN[1:0] ビットにより設定します。
- 他の SAI と同期させる 1 個のオーディオブロックを SYNCEN[1:0] ビットにより設定します。このとき、もう一方のオーディオブロックは 2 番目の SAI オーディオブロックと同期させるように、SYNCEN[1:0] ビットによって設定されます。

下表は、使用される SAI ブロックに応じた適切な同期信号の選び方を示します。たとえば、SAI2 は SAI2 SYNCIN を 0 に設定することで、SAI1 からの同期を選択できます。SAI1 に SAI2 からの同期を選択させたい場合は、SAI1 SYNCIN を 1 にセットする必要があります。「予約済み」と記載されている位置は使用できません。

表 566. 外部同期の選択

ブロックの例	SYNCIN = 1	SYNCIN = 0
SAI1	SAI2 同期	予約済みです。
SAI2	予約済みです。	SAI1 同期

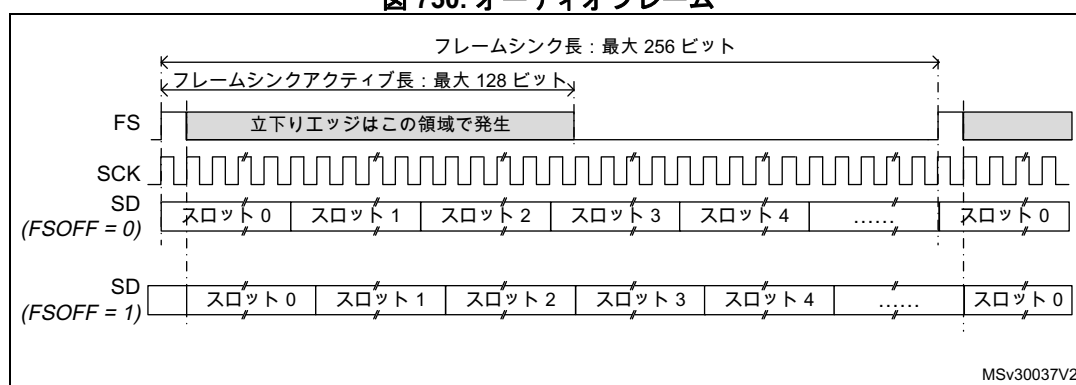
53.4.5 オーディオデータサイズ

オーディオフレームは、SAI_xCR1 レジスタの DS[2:0] ビットを設定することにより、様々なデータサイズを対象とすることができます。データサイズとしては、8、10、16、20、24、32 ビットが可能です。転送時、SAI_xCR1 レジスタの LSBFIRST ビットの設定によってデータの MSB を先頭にして送信することもできれば LSB を先頭にすることもできます。

53.4.6 フレーム同期

FS 信号は、オーディオフレームにおいてフレーム同期信号として機能します（フレーム開始）。この信号の波形は、このフレーム同期動作に関する独自の仕様を持つ様々なオーディオプロトコルに対応するため、全面的に設定可能となっています。この再設定は SAI_xFRCR レジスタを使って実行されます。図 750 にこの柔軟性について図示します。

図 750. オーディオフレーム



AC'97 モードまたは SPDIF モード（SAI_xCR1 レジスタの PRTCFG[1:0] ビット = 10 または PRTCFG[1:0] ビット = 01）では、フレーム同期波形を強制的に AC'97 プロトコルに適合させます。SAI_xFRCR レジスタ値は無視されます。

各オーディオブロックは独立であるため、それぞれに固有の設定が必要となります。

フレーム長

- マスタモード

オーディオフレーム長は、SAI_xFRCR レジスタの FRL[7:0] ビットを設定して、最大 256 ビットクロックに設定することができます。

フレーム長がフレームに対して指定されたスロット数より大きい場合、SAI_xCR2 レジスタの TRIS ビットの状態に応じて、残りの送信ビットを 0 に拡張するか、SD ラインをハイインピーダンス（HI-Z）状態にします（FS 信号の機能を参照）。受信モードでは、残りのビットは無視されます。

NODIV ビットがクリアされている場合、(FRL + 1) は 8 から 256 のうちの 2 の累乗の数に調整する必要があります。これは、オーディオフレームがビットクロックサイクル当たり整数個の MCLK パルスを含むようにするためです。

NODIV ビットがセットされている場合、(FRL + 1) フィールドは 8 から 256 のうちの任意の値をとることができます。セクション 53.4.8 : SAI クロックジェネレータを参照してください。

- スレーブモード

オーディオフレーム長は主に、外部マスタから送信されるオーディオフレーム当たりのビットクロック数をスレーブに対して指定するために使用されます。また、処理中のオーディオフレームでマスタからの想定されるフレーム同期信号または遅延したフレーム同期信号の発生を検出す

するためにも使用されます。この場合、エラーが生成されます。詳細については、[セクション 53.4.14 : エラーフラグ](#)を参照してください。

スレーブモードでは、SAI_xFRCR レジスタの FRL[7:0] ビット設定に制約はありません。

フレーム内のビット数は $FRL[7:0] + 1$ に等しくなります。

オーディオフレーム内で転送する最小ビット数は 8 です。

フレーム同期極性

SAI_xFRCR レジスタの FSPOL ビットにより、フレームが開始される FS ピンのアクティブ極性が設定されます。フレーム開始は、エッジに反応します。

スレーブモードにおいて、オーディオブロックは有効なフレームを待って、送信または受信を開始します。フレーム開始はこの信号に同期します。これは、通信中のフレーム開始が検出されず、予測のフレーム開始と一致しない場合にのみ有効です ([セクション 53.4.14 : エラーフラグ](#)を参照)。

マスタモードでは、SAI_xCR1 レジスタの SAIEN ビットがクリアされるまで、オーディオフレームが完了するごとにフレーム同期が毎回送信されます。前のオーディオフレームの終了時に FIFO にデータがない場合、アンダーラン状態は[セクション 53.4.14 : エラーフラグ](#)の説明のように扱われますが、オーディオ通信フローは中断されません。

フレーム同期アクティブレベル長

SAI_xFRCR レジスタの FSALL[6:0] ビットにより、フレーム同期信号のアクティブレベル長の設定が可能になります。この長さは、1 から 128 のビットクロックに設定することができます。

例えば、I2S、LSB 詰めまたは MSB 詰めモードでは、アクティブ長をフレーム長の 1/2 に、PCM/DSP、TDM では 1 ビット幅に、それぞれ設定することができます。

フレーム同期オフセット

アプリケーションで対象とするオーディオプロトコルに応じて、オーディオフレームの最終ビットまたは先頭ビットを送信したときフレーム同期信号をアサートすることができます (例えば、それぞれ I2S 規格プロトコルおよび MSB 詰めプロトコルの場合)。SAI_xFRCR レジスタの FSOFF ビットにより、2 つの設定のいずれかを選択可能です。

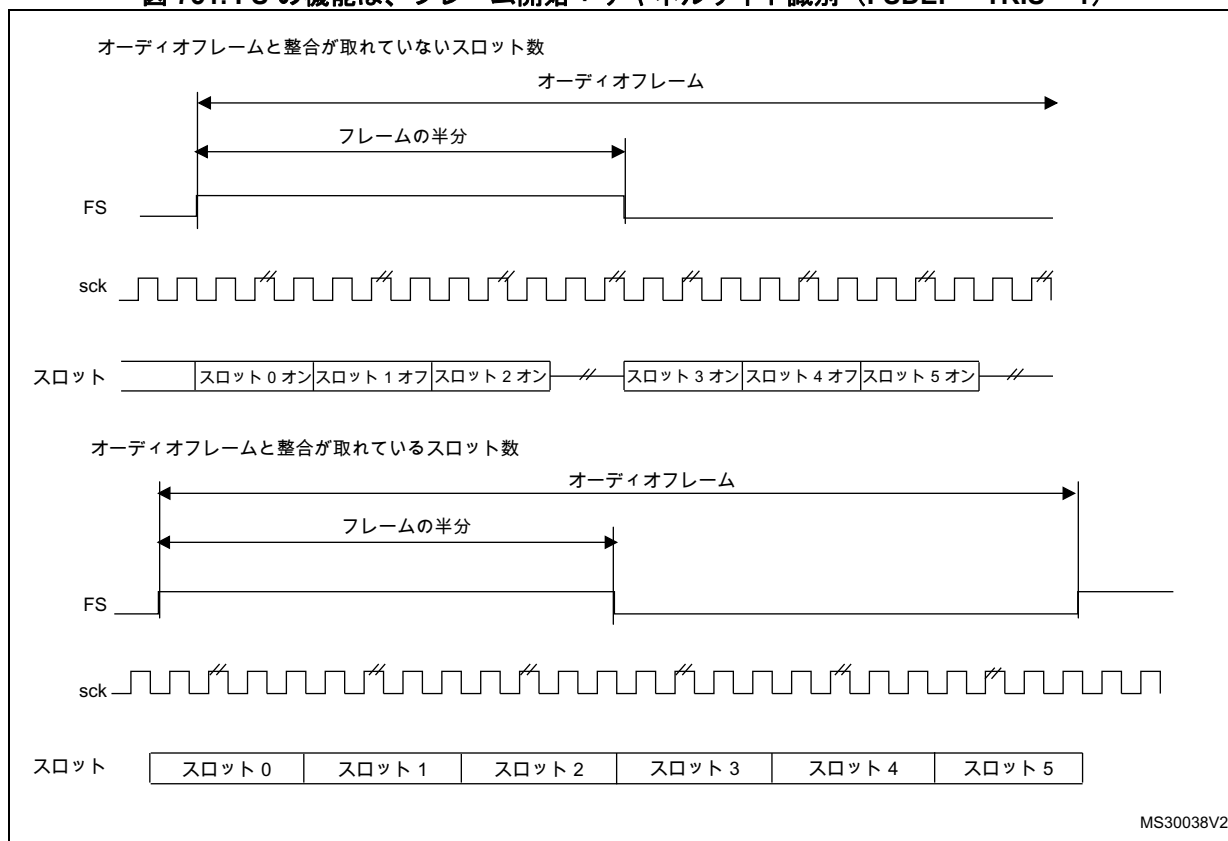
FS 信号の機能

FS 信号は、FS の機能によって意味が異なります。SAI_xFRCR レジスタの FSDEF ビットにより意味を選択します。

- 0 : 例えば PCM/DSP、TDM、AC'97 オーディオプロトコルのようなフレーム開始
- 1 : 例えば I2S、MSB 詰めまたは LSB 詰めプロトコルのような、オーディオフレーム内のチャネルサイド識別およびフレーム開始

FS 信号をフレーム開始およびフレーム内チャネルサイド識別と見なす場合、指定スロット数は半分を左チャネル用、もう半分を右チャネル用と見なす必要があります。ハーフオーディオフレームのビットクロック数がチャネルサイド専用スロット数より大きい場合、TRIS = 0 であれば、SAI_xCR2 レジスタの残りのビットクロックに対して 0 が送信されます。そうしないと、TRIS = 1 の場合、SD ラインがハイインピーダンスに開放されます。受信モード時、残りのビットクロックサイクルはチャネルサイドが変化するまで無視されます。

図 751. FS の機能は、フレーム開始 + チャネルサイド識別 (FSDEF = TRIS = 1)

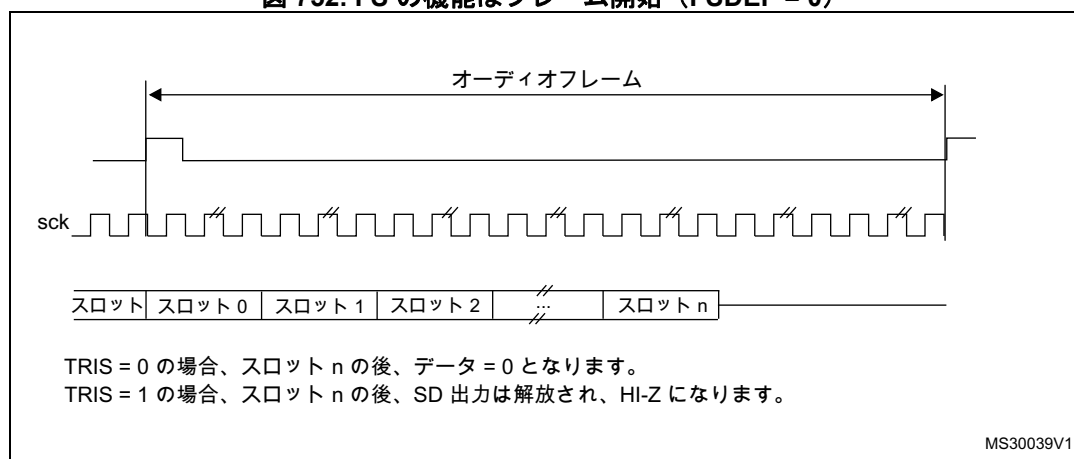


1. フレーム長は偶数である必要があります。

SAI_xFRCR の FSDEF ビットがクリアされたままの場合、FS 信号はフレーム開始に相当し、さらに SAI_xSLOTR の NBSLOT[3:0] ビットで指定されたスロット数に SAI_xSLOTR の SLOTSZ[1:0] ビットで設定されたスロット当たりのビット数を乗算した値が、フレームサイズ (SAI_xFRCR レジスタの FRL[7:0] ビット) より小さい場合、

- SAI_xCR2 レジスタで TRIS = 0 のとき、送信の場合はフレームの終わりまで、最終スロット後の残りのビットは強制的に 0 に設定されます。
- TRIS = 1 のとき、これら残りのビットの転送中、ラインはハイインピーダンス (HI-Z) になります。受信モードでは、これらのビットは破棄されます。

図 752. FS の機能はフレーム開始 (FSDEF = 0)



送信モードでオーディオブロックが SD ラインの SPDIF 出力を取得するように設定されている場合は、FS 信号は使用されません。対応する FS の I/O は開放され、他の目的のために空けられます。

53.4.7 スロットの設定

スロットは、オーディオフレームの基本要素です。オーディオフレーム内のスロット数は $NBSLOT[3:0] + 1$ に等しくなります。

オーディオフレーム当たりの最大スロット数は 16 に固定されています。

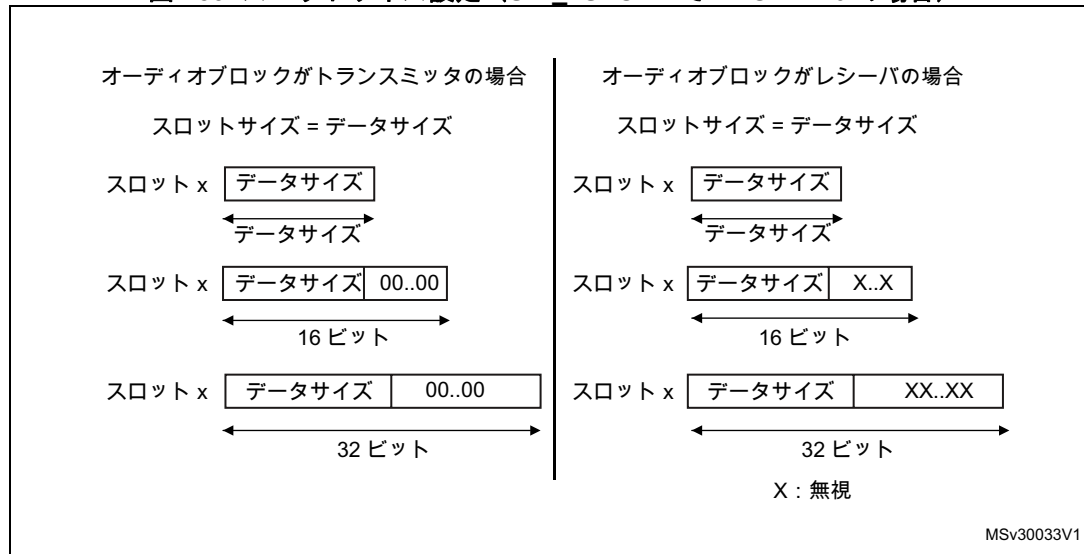
AC'97 プロトコルまたは SPDIF (PRTCFG[1:0] ビット = 10 または PRTCFG[1:0] ビット = 01 の場合) に対しては、スロット数は、そのプロトコル仕様を対象とするよう自動的に設定され、 $NBSLOT[3:0]$ の値は無視されます。

各スロットは、SAI_xSLOTR レジスタの SLOTEN[15:0] ビットにより有効/無効を指定することができます。

送信モードでは、無効なスロットが転送されると、SD データラインは TRIS ビットの設定に応じて強制的に 0 に設定されるか、または開放されてハイインピーダンス (HI-Z) 状態になります ([非アクティブスロットでの出力データライン管理](#)を参照)。受信モードでは、このスロットの終わりからの受信した値は無視されます。結果として、FIFO アクセスがないため、この非アクティブなスロット状態にリンクした FIFO からの読出しリクエストや FIFO への書込みリクエストは起こりません。

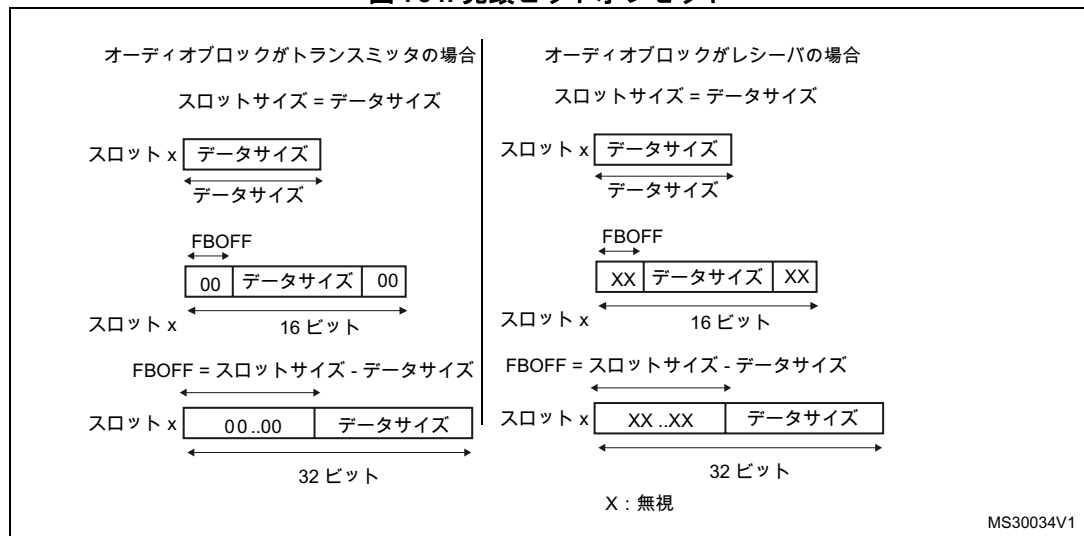
また、スロットのサイズも [図 753](#) に示すように設定可能です。スロットのサイズは、SAI_xSLOTR レジスタの SLOTSZ[1:0] ビットを設定することで選択できます。サイズは、オーディオフレームの各スロットに同じように適用されます。

図 753. スロットサイズ設定 (SAI_xSLOTR で FBOFF = 0 の場合)



スロット内で転送する先頭データビットの位置を選択することができます。このオフセットは、SAI_xSLOTR レジスタの FBOFF[4:0] ビットによって設定されます。送信モードでは、スロットの開始からこのオフセット位置に到達するまでの間 0 値が挿入されます。受信時、オフセットフェーズ内のビットは無視されます。この機能は LSB 詰めプロトコルを対象としています (オフセットが、スロットサイズ - データサイズに等しい場合)。

図 754. 先頭ビットオフセット



SAI の誤動作を避けるためには、次の条件を満足する必要があります。

$FBOFF \leq (SLOTSZ - DS)$ 、

$DS \leq SLOTSZ$ 、

$NBSLOT \times SLOTSZ \leq FRL$ (フレーム長)、

SAI_xFRCR レジスタの FSDEF ビットがセットされている場合でも、スロット数は偶数でなければなりません。

AC'97 および SPDIF プロトコル (PRTCFG[1:0] ビット = 10 または PRTCFG[1:0] ビット = 01) では、スロットサイズは、[セクション 53.4.11 : AC'97 リンクコントローラ](#)で示されるように自動的に設定されます。

53.4.8 SAI クロックジェネレータ

各オーディオブロックには専用のクロックジェネレータが備わっています。クロックジェネレータは、sai_x_ker_ck からマスタクロック (MCLK_x) およびビットクロック (SCK_x) 信号を作成します。sai_x_ker_ck クロックは、製品のクロックコントローラ (RCC) によって供給されます。

マスタクロック (MCLK_x) の生成

クロックジェネレータは、オーディオブロックがマスタまたはスレーブとして定義されている場合にマスタクロック (MCLK_x) を供給します。マスタクロックは、該当ブロックの SAIEN ビットが 0 にセットされていても、MCKEN ビットを 1 にセットするとすぐに、生成されます。この機能は、オーディオストリームをアクティブにする前に MCLK_x を生成可能にするため、MCLK_x クロックが外部オーディオデバイスのシステムクロックとして使用される場合に役立ちます。

オーディオサンプルを転送する前に MCLK_x 出力にマスタクロックを生成するには、ユーザアプリケーションは以下のシーケンスに従う必要があります。

1. SAIEN = 0 であることを確認します。
2. MCKDIV[5:0] 分周器を必要な値にプログラムします。
3. MCKEN ビットを 1 にセットします。
4. 後に、アプリケーションは SAI の他の部分を構成し、SAIEN ビットを 1 に設定してオーディオサンプルの転送を開始できます。

MCLK_x 出力に生成されたクロックの乱れを避けるため、次の操作は推奨されません。

- MCKEN = 1 のときに、MCKDIV を変更すること
- SAIEN = 1 の場合に、MCKEN を 0 にセットすること

SAI では、MCLK_x が (SAIEN = 0 の状態で) MCKEN ビットによってオンおよびオフに切り替えられたときに、MCLK_x 出力にスプールがないことを保証します。

[表 567](#) に、MCLK_x の有効化条件を示します。

表 567. MCLK_x の有効化条件

MCKEN	NODIV	ブロック x の SAIEN	MCLK_x
0	X	0	無効
1			有効
0	1	1	無効
1			有効
X	0		有効

注： MCKEN が 1 にセットされている場合、MCLK_x は AC'97 モードで生成することもできます。

ビットクロック (SCK_x) の生成

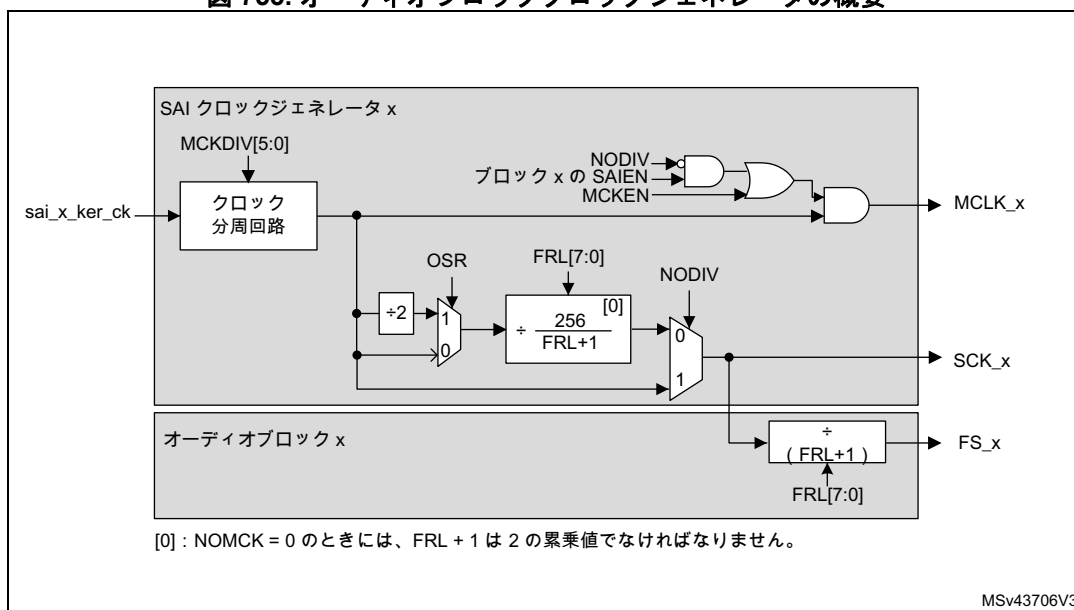
クロックジェネレータは、オーディオブロックがマスタとして定義されている場合にビットクロック (SCK_x) を供給します。フレーム同期 (FS_x) は、クロックジェネレータから供給される信号からも得られます。

スレーブモードでは、NODIV および OSR フィールドの値は無視され、SCK_x クロックは生成されません。

SCK_x のビットクロックストロブエッジは、CKSTR フィールドで設定することができ、これはマスタモードとスレーブモードの両方で機能します。

図 755 に、オーディオブロッククロックジェネレータのアーキテクチャを示します。

図 755. オーディオブロッククロックジェネレータの概要



マスタクロック (MCLK_x) とフレーム同期 (FS_x) 周波数の比率を 256 または 512 に強制するために、NODIV ビットを使用する必要があります。

- NODIV を 0 にセットした場合、フレーム同期とマスタクロックとの周波数比は、OSR 値に従って 512 または 256 に固定されますが、フレーム長は 2 の累乗値でなければなりません。詳細については、以下を参照してください。
- NODIV を 1 にセットした場合、ビットクロック (SCK_x) の周波数は、アプリケーションによって、MCKDIV を介して調整できます。さらに、フレーム長が 8 以上 (すなわち、FRL[7:0] > 6) であるかぎり、フレーム長の値に何ら制約はありません。フレーム同期周波数は MCKDIV およびフレーム長 (FRL[7:0]) に依存します。この場合、MCLK_x の周波数は SCK_x に等しくなります。

NODIV、MCKEN、SAIEN、OVR、CKSTR、および MCKDIV[5:0] ビットは SAI_xCR1 レジスタに属しますが、FRL[7:0] ビットは SAI_xFRCR レジスタに属します。

NODIV = 0 のときのクロックジェネレータのプログラミング

この場合、MCLK_x 周波数は次のようになります。

- $F_{MCLK_x} = 256 \times F_{FS_x}$ (OSR = 0 の場合)
- $F_{MCLK_x} = 512 \times F_{FS_x}$ (OSR = 1 の場合)

MCKDIV が 0 ではない場合、MCLK_x 周波数は下の計算式によって与えられます。

$$F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

フレーム同期周波数は次のようになります。

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV \times (OSR + 1) \times 256}$$

ビットクロック周波数 (SCK_x) は次の計算式で与えられます。

$$F_{SCK_x} = \frac{F_{sai_x_ker_ck} \times (FRL + 1)}{MCKDIV \times (OSR + 1) \times 256}$$

注 : NODIV が 0 に等しい場合、(FRL+1) は 2 の累乗値でなければなりません。加えて、(FRL+1) は 8 ~256 でなければなりません。(FRL+1) はオーディオフレーム内のビットクロック数を表します。

MCKDIV 分周比が奇数である場合、MCLK のデューティサイクルは 50% ではありません。MCKDIV が奇数、OSR が 0 に等しく、かつ (FRL+1) = 2^8 であると、ビットクロック信号 (SCK_x) についてもデューティサイクルが 50% とは異なることがあります。

MCKDIV を偶数が大きな値 (10 を超える値) にプログラミングすることを推奨します。

MCKDIV = 0 であると、MCKDIV = 1 のときと同じ結果となることに注意してください。

NODIV = 1 のときのクロックジェネレータのプログラミング

MCKDIV が 0 ではない場合、ビットクロック (SCK_x) 周波数は下の計算式によって与えられます。

$$F_{SCK_x} = F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

フレーム同期 (FS_x) 周波数は次の計算式によって与えられます。

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{(FRL + 1) \times MCKDIV}$$

注 : NODIV が 1 にセットされている場合、(FRL+1) は 8 から 256 のうちの任意の値をとることができます。

MCKDIV = 0 であると、MCKDIV = 1 のときと同じ結果となることに注意してください。

クロックジェネレータのプログラミング例

表 568 に、48、96、192 kHz に対するプログラミング例を示します。

表 568. クロックジェネレータのプログラミング例

入力 sai_x_ker_ck クロック周波数	MCLK	F_{MCLK}/F_{FS}	FRL ⁽¹⁾	OSR	NODIV	MCKEN	MCKDIV[5:0]	オーディオ サンプリング 周波数 (F_{FS})
98.304 MHz	Y	512	2^{N-1}	1	0	1	0 または 1	192 kHz
		512	2^{N-1}	1	0	1	2	96 kHz
		512	2^{N-1}	1	0	1	4	48 kHz
		256	2^{N-1}	0	0	1	2	192 kHz
		256	2^{N-1}	0	0	1	4	96 kHz
		256	2^{N-1}	0	0	1	8	48 kHz
	N	-	63	-	1	0	8	192 kHz
		-	63	-	1	0	16	96 kHz
		-	63	-	1	0	32	48 kHz

1. N は 3～8 の整数値です。

53.4.9 内部 FIFO

SAI 内の各オーディオブロックには固有の FIFO を搭載しています。そのブロックがトランスミッタに指定されるかレシーバに指定されるかによって、FIFO にそれぞれ書き込んだり読み出したりすることができます。したがって、SAI_xSR レジスタの FREQ ビットに対応した FIFO リクエストは 1 つしかありません。

SAI_xIM レジスタの FREQIE ビットが有効化されている場合、割込みが生成されます。これは次の条件によって変わります：

- FIFO 閾値設定 (SAI_xCR2 の FLVL ビット)
- 通信方向 (トランスミッタまたはレシーバ) [送信モードでの割込み生成](#) および [受信モードでの割込み生成](#) を参照してください。

送信モードでの割込み生成

送信モードでの割込みの生成は、次のように FIFO の設定によって変わります：

- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO エンプティ (FTH[2:0] = 0b000) に設定されている場合、SAI_xDR レジスタに有効なデータがない (SAI_xSR の FLVL[2:0] ビット < 001b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO がエンプティ (空) でなくなった場合 (SAI_xSR の FLVL[2:0] ビット ≠ 0b000)、すなわち FIFO に 1 個以上のデータが格納されている状態になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/4 (FTH[2:0] = 001b) に設定されている場合、データが FIFO の 1/4 より少ない (SAI_xSR の FLVL[2:0] ビット < 0b010) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 1/4 以上にデータがある状態 (SAI_xSR の FLVL[2:0] ビット ≥ 0b010) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。

- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/2 (FTH[2:0] = 0b010) に設定されている場合、データが FIFO の 1/2 より少ない (SAI_xSR の FLVL[2:0] ビット < 011b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 1/2 以上にデータがある状態 (SAI_xSR の FLVL[2:0] ビット ≥ 011b) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 3/4 (FTH[2:0] = 011b) に設定されている場合、データが FIFO の 3/4 より少ない (SAI_xSR の FLVL[2:0] ビット < 0b100) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 3/4 以上にデータがある状態 (SAI_xSR の FLVL[2:0] ビット ≥ 0b100) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフル (FTH[2:0] = 0b100) に設定されている場合、FIFO がフルでない (SAI_xSR の FLVL[2:0] ビット < 101b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO データがフル (SAI_xSR の FLVL[2:0] ビット = 101b) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。

受信モードでの割込み生成

受信モードでの割込みの発生は、次のように FIFO の設定によって変わります：

- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO エンプティ (FTH[2:0] = 0b000) に設定されている場合、SAI_xDR レジスタに 1 個以上のデータがある (SAI_xSR の FLVL[2:0] ビット ≥ 001b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO がエンプティになる (SAI_xSR の FLVL[2:0] ビット = 0b000)、すなわち FIFO にデータがなくなると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/4 (FTH[2:0] = 001b) に設定されている場合、FIFO データロケーションの 1/4 以上が使用可能 (SAI_xSR の FLVL[2:0] ビット ≥ 0b010) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 1/4 未満 (SAI_xSR の FLVL[2:0] ビット < 0b010) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/2 (FTH[2:0] = 0b010) に設定されている場合、FIFO データロケーションの 1/2 以上が使用可能 (SAI_xSR の FLVL[2:0] ビット ≥ 011b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 1/2 未満 (SAI_xSR の FLVL[2:0] ビット < 011b) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 3/4 (FTH[2:0] = 011b) に設定されている場合、FIFO データロケーションの 3/4 以上が使用可能 (SAI_xSR の FLVL[2:0] ビット ≥ 0b100) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 3/4 未満 (SAI_xSR の FLVL[2:0] ビット < 0b100) になると、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI_xCR2 レジスタの FIFO 閾値ビットが FIFO がフル (FTH[2:0] = 0b100) に設定されている場合、FIFO がフル (SAI_xSR の FLVL[2:0] ビット = 101b) という条件で、割込みが生成されます (SAI_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO データがフルでなくなると (SAI_xSR の FLVL[2:0] ビット < 101b)、この割込み (SAI_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。

SAI_xCR1 レジスタの DMAEN ビットがセットされている場合、SAI は、割込み生成と同様に DMA を使用することができます。FREQ ビットアサーションメカニズムは、上記 FREQIE について説明した割込み生成メカニズムと同じです。

各 FIFO は 8 ワードの FIFO です。FIFO に対する書込み操作や読出し操作は、そのアクセスサイズによらず、1 回につき FIFO の 1 ワード分を対象とします。FIFO の 1 ワードには 1 オーディオスロットが格納されます。FIFO ポインタは、SAI_xDR レジスタに対する各アクセスの後に 1 ワードずつインクリメントされます。

SAI_xDR に書き込む際、データは右詰めである必要があります。

受信データは SAI_xDR で右詰めとなっています。

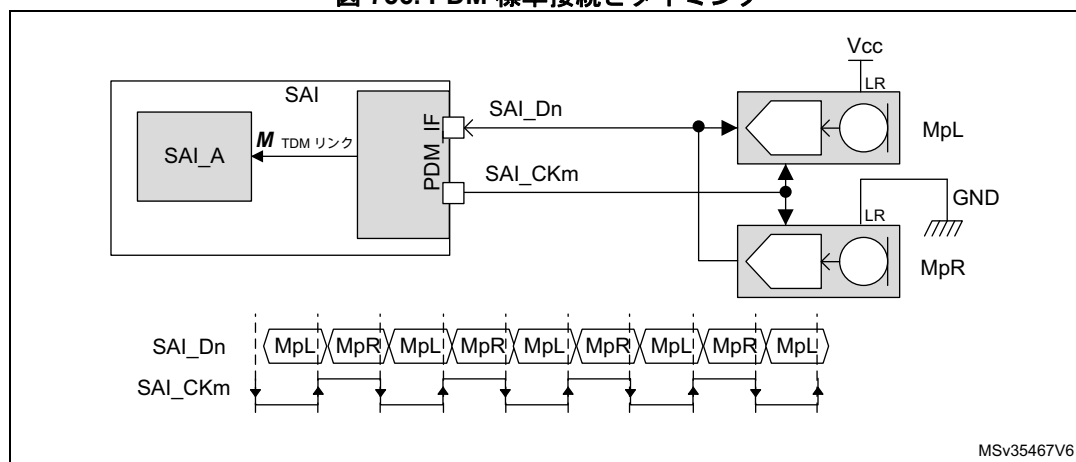
SAI_xCR2 レジスタの FFLUSH ビットをセットして SAI を無効化すると、FIFO ポインタを再初期化することができます。SAI が有効なときに FFLUSH をセットすると、FIFO 内にあるデータは自動的に失われます。

53.4.10 PDM インタフェース

PDM (パルス密度変調) インタフェースは、デジタルマイクロフォン対応のために提供されています。最大 4 組のデジタルマイクロフォンのペアに並列接続可能です。製品の実装に応じて、対応可能なマイクロフォンの数は少なくなります (セクション 53.3 : SAI の実装を参照)。

図 756 に、PDM インタフェースを介したデジタルマイクロフォンのペアの標準的な接続を示します。どちらのマイクロフォンも同じビットストリームクロックとデータのラインを共有しています。設定ピン (LR) によって、片方のマイクロフォンが SAI_CK[m] の立上りエッジで有効なデータを供給すると同時に、もう一方が SAI_CK[m] の立下りエッジで有効なデータを供給します (m はクロックライン数)。

図 756. PDM 標準接続とタイミング



1. n はデータライン数、p はマイクロフォンのペア数を表します。

PDM 機能は、TDM マスタモードで設定された SAI_A サブクロックとともに使用されるためのものです。SAI_B サブクロックとは使用できません。PDM インタフェースは、SAI_A の TDM インタフェースから供給されるタイミング信号を使用して、それを適合させてビットストリームクロック (SAI_CK[m]) を生成します。

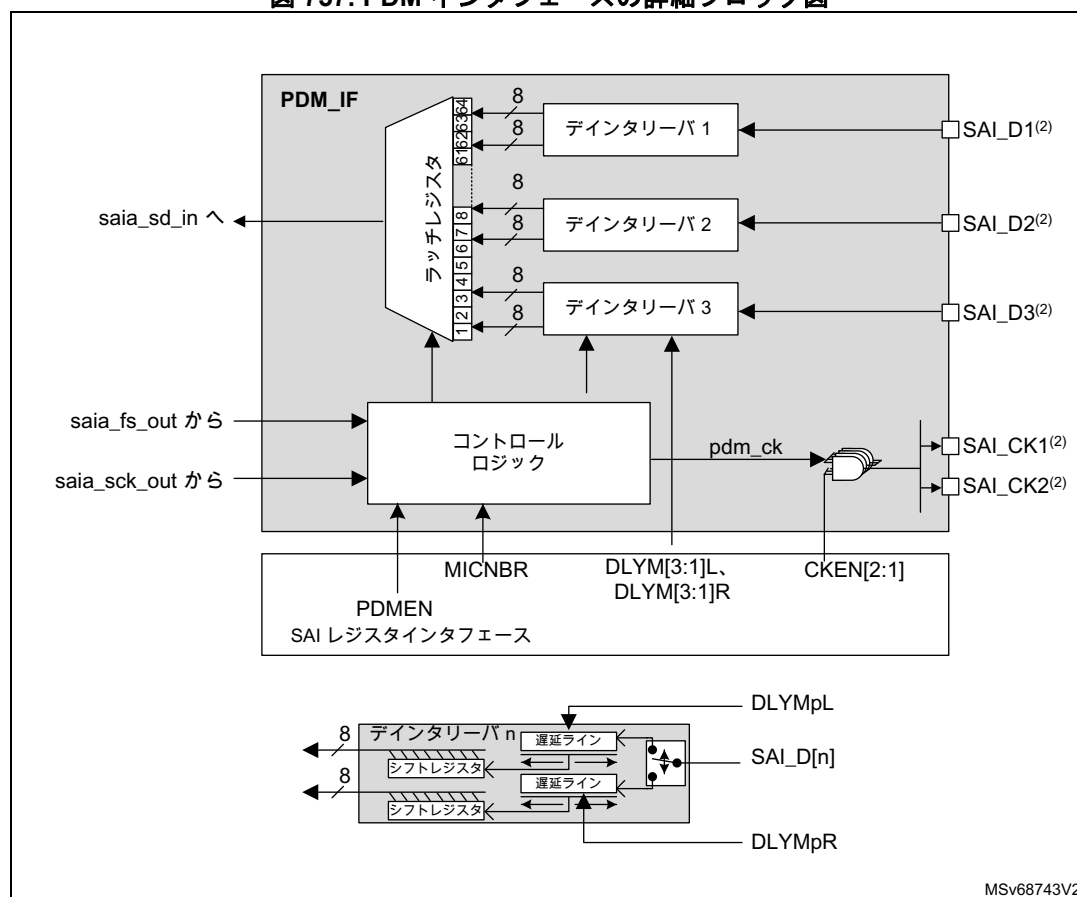
PDM へのデータ処理シーケンスは、次のとおりです。

1. PDM インタフェースは、SAI_A の TDM インタフェースから受信したビットクロックからビットストリームクロックを構築します。
2. マイクロフォンから受信したビットストリームデータ (SAI_D[n]) はインタリーブを解かれ、ビットストリームクロック精度で各マイクrofホンの遅延を微調整するために、7 ビット遅延ラインを通ります。
3. シフトレジスタによって、それぞれのシリアルビットストリームがバイト列に変換されます。
4. 最後の操作では、TDM インタフェースのシリアルデータラインを通じて、結果として得られたバイト列が SAI_A にシフトアウトされます。

以下の図 757 に、PDM インタフェースのブロック図をデインタリーブの詳細図とともに示します。

注： PDM インタフェースには、ビットストリームから PCM オーディオサンプルを構築するために必要なデシメーションフィルタは組み込まれていません。この操作を行うかどうかは、アプリケーションソフトウェア次第です。

図 757. PDM インタフェースの詳細ブロック図



1. n はデータライン数、 p はマイクrofホンのペア数を表します。
2. これらの信号は、すべての SAI インスタンスで使用できるわけではありません。詳細については、[セクション 53.3: SAI の実装](#)を参照してください。

PDM インタフェースは、SAI_PDMCR レジスタの PDMEN ビットを通じて有効にできます。ただし、SAI_A ブロックを有効にする前に PDM インタフェースを有効にする必要があります。

メモリのフットプリントを削減するため、アプリケーションが必要とするマイクrofホンの本数を選択できます。この選択は、MICNBR[1:0] ビットで行うことができます。マイクrofホンの本数は、2

本、4 本、6 本、8 本から選択可能です。たとえば、アプリケーションがマイクロホンに 3 本使用するのであれば、4 本を選択する必要があります。

PDM インタフェースの有効化

PDM インタフェースを有効にするには、以下のシーケンスに従います。

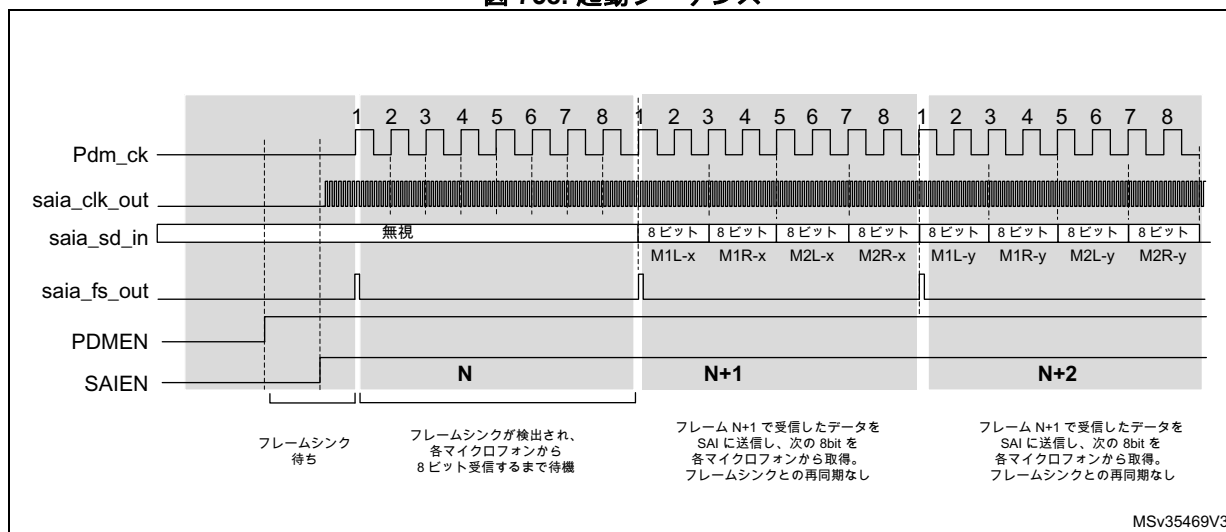
1. SAI_A を TDM マスタモードに設定します (表 569 参照)。
2. 次のように PDM インタフェースを設定します。
 - a) MICNBR にデジタルマイクロホンの本数を設定します。
 - b) CKEN の対応するビットを 1 にセットして、アプリケーションで必要なビットストリームクロックを有効にします。
3. PDMEN ビットで PDM インタフェースを有効にします。
4. SAI_A を有効にします。

注： PDM インタフェースと SAI_A が有効になると、SAI_ADR で受信する最初の 2 つの TDMA フレームは無効ですので、破棄する必要があります。

起動シーケンス

図 758 に起動シーケンスを示します。PDM インタフェースが有効になると、マイクロホンサンプルの取得を始める前に、フレーム同期イベントを待ちます。SAI_CK クロック 8 周期後、各マイクロホンから入力されるデータバイトが利用可能となり、TDM インタフェースを通じて SAI に転送されます。

図 758. 起動シーケンス



SAI_ADR データフォーマット

マイクロホンから SAI_ADR レジスタに入力されるデータの配列は、次のパラメータに依存しています。

- マイクロホンの本数
- 選択されたスロット幅
- LSBFIRST ビット

スロット幅によって、SAI_ADR に入力可能な各ワードの有効ビット数が設定されます。

32 ビットのスロット幅が選択された場合には、SAI_ADR に入力可能な各データには 32 ビットの有効なビットが含まれます。これによって、メモリに格納されるワード数が減ります。ただし、対応するには、ソフトウェアが各マイクロフォンのデータのインタリーブを解くための何らかの操作を行う必要があります。

その一方で、8 ビットのスロット幅が選択された場合には、SAI_ADR に入力可能な各データには 8 ビットの有効ビットが含まれます。これによって、メモリに格納されるワード数が増えます。ただし、各ワードにはマイクロフォン 1 本からの情報が含まれているため、これには余計な処理を避けるという利点があります。

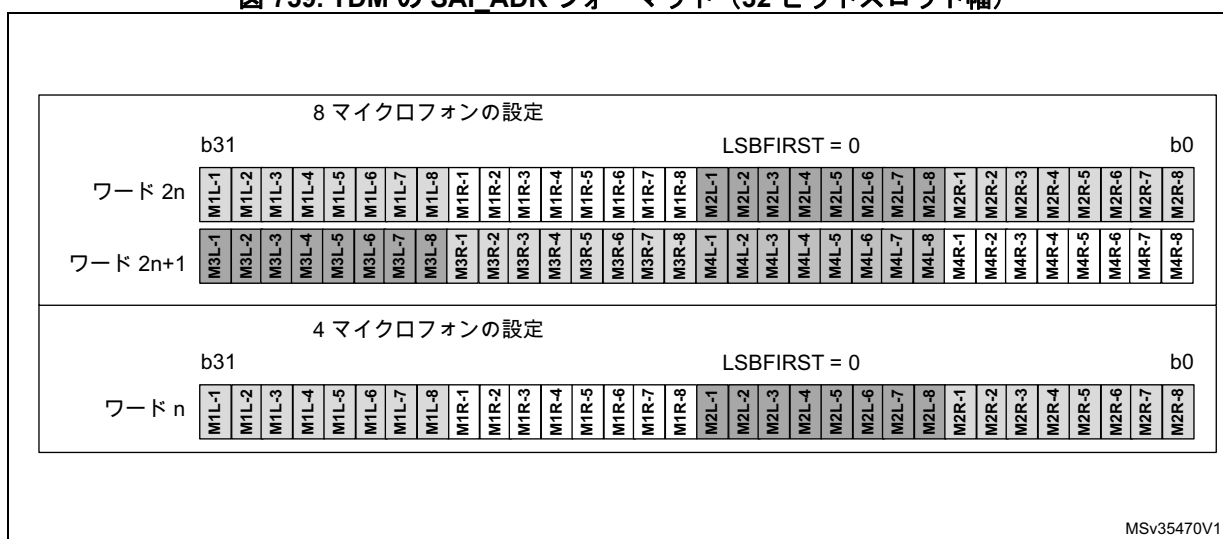
SAI_ADR データフォーマット例

- 32 ビットスロット幅 (DS = 0b111 かつ SLOTSZ = 0) [図 759](#) を参照してください。

マイクロフォン 8 本の設定では、SAI_ADR レジスタから読み込まれた連続する 2 ワードに、各マイクロフォンからの 1 データバイトが含まれています。

マイクロフォン 4 本の設定では、SAI_ADR レジスタから読み込まれた各ワードに、各マイクロフォンからの 1 データバイトが含まれています。

図 759. TDM の SAI_ADR フォーマット (32 ビットスロット幅)

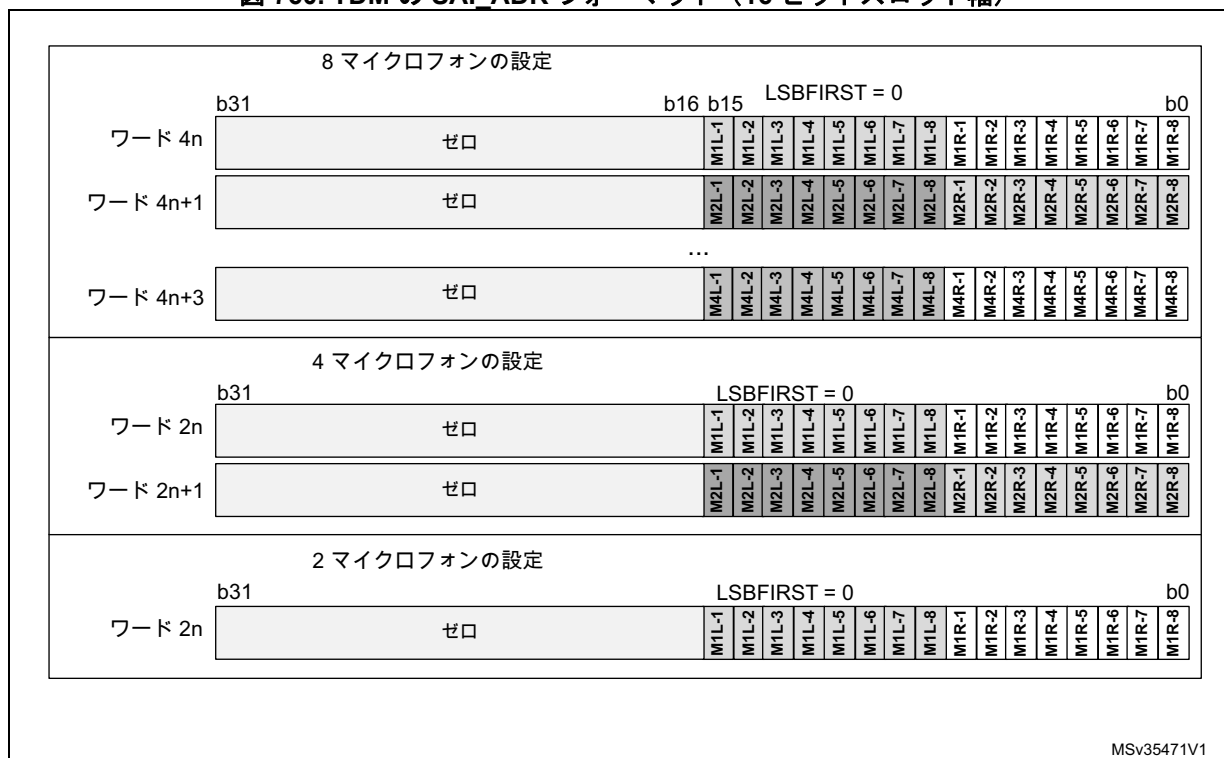


- 16 ビットスロット幅 (DS = 0b100 かつ SLOTSZ = 0) [図 760](#) を参照してください。

マイクロフォン 8 本の設定では、SAI_ADR レジスタから読み込まれた連続する 4 ワードに、各マイクロフォンからの 1 データバイトが含まれています。SAI_ADR の 16 ビットデータは右詰めされていることに注意してください。

マイクロフォン 4 本または 2 本の設定では、SAI の動作はマイクロフォン 8 本の設定とほぼ同じです。マイクロフォン 4 本からバイトを取得するには 16 ビットワードが最大で 2 ワード必要であり、マイクロフォン 2 本からであれば 1 ワード必要です。

図 760. TDM の SAI_ADR フォーマット (16 ビットスロット幅)

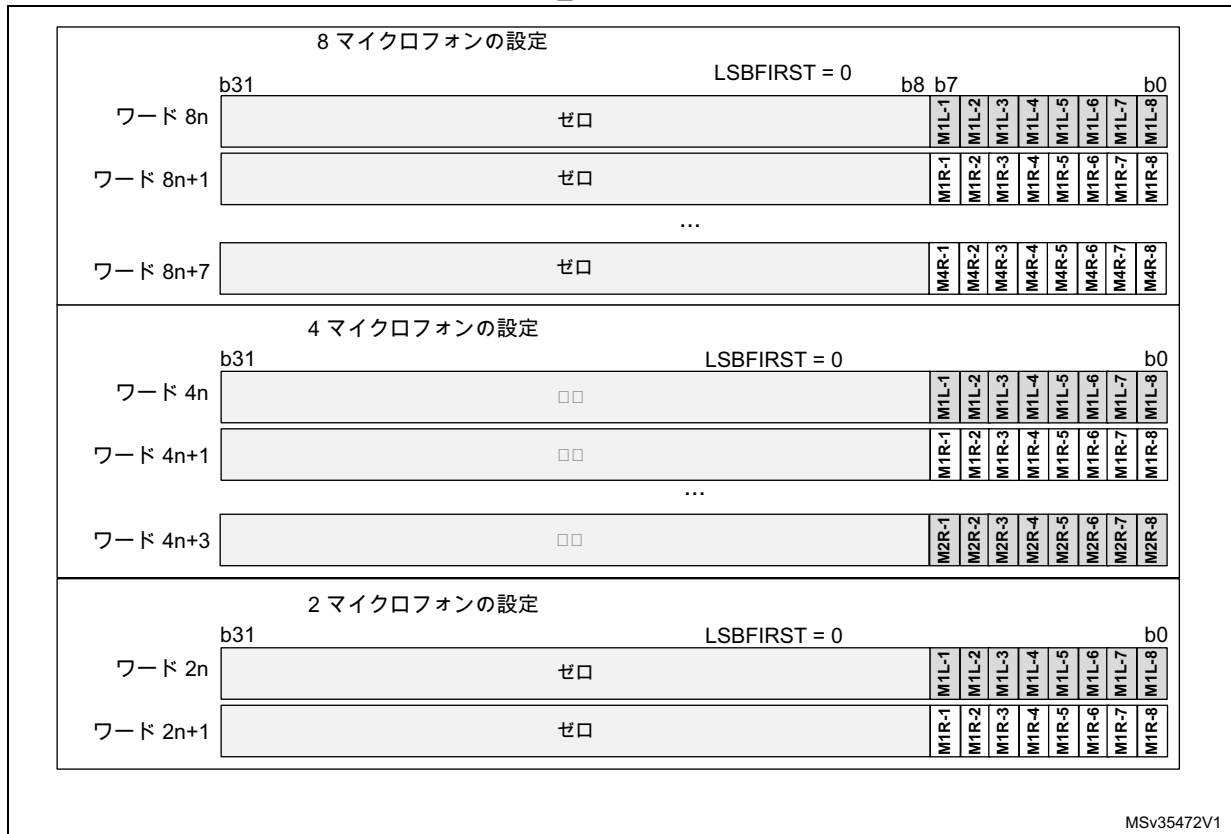


- 8 ビットスロット幅を使用 (DS = 0b010 かつ SLOTSZ = 0) 図 761 を参照してください。

マイクロフォン 8 本の設定では、SAI_ADR レジスタから読み込まれた連続する 8 ワードに、各マイクロフォンからのデータが 1 バイト含まれています。SAI_ADR の 8 ビットデータは右詰めされていることに注意してください。

マイクロフォン 4 本または 2 本の設定では、SAI の動作はマイクロフォン 8 本の設定とほぼ同じです。マイクロフォン 4 本からバイトを取得するには 8 ビットワードが最大で 4 ワード必要であり、マイクロフォン 2 本からであれば 2 ワード必要です。

図 761. TDM の SAI_ADR フォーマット (8 ビットスロット幅)



PDM インタフェースの TDM 設定

SAI_A TDM インタフェースは、マイクロフォンのサンプルを取得するために、内部で PDM インタフェースに接続されています。ユーザアプリケーションは、PDM インタフェースを [表 569](#) に示すように設定して、PDM インタフェースとの良好な接続を保证する必要があります。

表 569. TDM 設定

ビットフィールド	値	コメント
MODE	0b01	モードは MASTER レシーバである必要があります。
PRTCFCG	0b00	TDM 用フリープロトコル
DS	X	フレーム長とスロット数 (FRL と NBSLOT) に基づいて、必要なデータフォーマットに応じて調整します。 表 570 を参照してください。
LSBFIRST	X	このパラメータは必要なデータフォーマットに応じて使用します。
CKSTR	0	信号の遷移は、SCK_A ビットクロックの立ち上りエッジで発生します。信号はビットクロックの立下りでは安定しています。
MONO	0	ステレオモード
FRL	X	マイクロフォンの本数 (MICNBR) に応じて調整します。 表 570 を参照してください。
FSALL	0	ハルス幅は 1 ビットクロックサイクルです。
FSDEF	0	FS 信号がフレームの開始です。

表 569. TDM 設定 (続き)

ビット フィールド	値	コメント
FSPOL	1	FS はアクティブハイです。
FSOFF	0	FS はスロット 0 の先頭ビットでアサートされます。
FBOFF	0	スロットにはオフセットはありません。
SLOTSZ	0	スロットサイズ = データサイズ
NBSLOT	X	スロットサイズとフレーム長 (FRL と DS) に基づいて、必要なデータフォーマットに応じて調整します。表 570 を参照してください。
SLOTEN	X	NBSLOT に応じて設定します。
NODIV	1	マスタクロック MCLK の生成は不要です。
MCKDIV	X	sai_a_ker_ck 入力に供給される周波数に依存します。 このパラメータは、適切なビットストリームクロック周波数を生成するために調整する必要があります。表 570 を参照してください。

ビットストリームクロックレートの調整

SAI TDM インタフェースを適切にプログラムするには、ユーザアプリケーションは表 569 に記載された設定を考慮し、次の手順に従う必要があります。

1. 次の式を使用して、PDM ビットストリームクロックに必要な周波数に応じてビットクロック周波数 (F_{SCK_A}) を調整します。

$$F_{SCK_A} = F_{PDM_CK} \times (MICNBR + 1) \times 2$$

MICNBR は、0、1、2、または 3 の値を取ります (0 = マイクフォン 2 本、[セクション 53.6.18](#) 参照)。

2. 次の式を使用して、フレーム長 (FRL) を設定します。

$$FRL = (16 \times (MICNBR + 1)) - 1$$

3. スロットサイズ (DS) を (FRL+1) の倍数に設定します。

表 570. TDM フレーム設定の例⁽¹⁾⁽²⁾

マイクロフォン サンプリング レート	マイクロフォン 本数	必要な SAI_CK _n 周波数 ⁽³⁾	ビット クロック (SCK_A)	フレーム シンク (FS_A)	FRL	DS	NBSLOT	コメント
48 kHz	最大 8	3.072 MHz	24.576 MHz	384 kHz	63	0b111	1	フレームあたり 32 ビット 2 スロット
		3.072 MHz	24.576 MHz	384 kHz	63	0b100	3	フレームあたり 16 ビット 4 スロット
		3.072 MHz	24.576 MHz	384 kHz	63	0b010	7	フレームあたり 8 ビット 8 スロット
	最大 6	3.072 MHz	18.432 MHz	384 kHz	47	0b110	1	フレームあたり 24 ビット 2 スロット
		3.072 MHz	18.432 MHz	384 kHz	47	0b100	2	フレームあたり 16 ビット 3 スロット
		3.072 MHz	18.432 MHz	384 kHz	47	0b010	5	フレームあたり 8 ビット 6 スロット
	最大 4	3.072 MHz	12.288 MHz	384 kHz	31	0b111	0	フレームあたり 32 ビット 1 スロット
		3.072 MHz	12.288 MHz	384 kHz	31	0b100	1	フレームあたり 16 ビット 2 スロット
		3.072 MHz	12.288 MHz	384 kHz	31	0b010	3	フレームあたり 8 ビット 4 スロット
	最大 2	3.072 MHz	6.144 MHz	384 kHz	15	0b100	0	フレームあたり 16 ビット 1 スロット
		3.072 MHz	6.144 MHz	384 kHz	15	0b010	1	フレームあたり 8 ビット 2 スロット
16 kHz	最大 8	1.024 MHz	8.192 MHz	128 kHz	63	0b111	1	フレームあたり 32 ビット 2 スロット
		1.024 MHz	8.192 MHz	128 kHz	63	0b100	3	フレームあたり 16 ビット 4 スロット
		1.024 MHz	8.192 MHz	128 kHz	63	0b010	7	フレームあたり 8 ビット 8 スロット
	最大 6	1.024 MHz	6.144 MHz	128 kHz	47	0b110	1	フレームあたり 24 ビット 2 スロット
		1.024 MHz	6.144 MHz	128 kHz	47	0b010	5	フレームあたり 8 ビット 6 スロット
		1.024 MHz	4.096 MHz	128 kHz	31	0b111	0	フレームあたり 32 ビット 1 スロット
	最大 4	1.024 MHz	4.096 MHz	128 kHz	31	0b100	1	フレームあたり 16 ビット 2 スロット
		1.024 MHz	4.096 MHz	128 kHz	31	0b010	3	フレームあたり 8 ビット 4 スロット
		1.024 MHz	2.048 MHz	128 kHz	15	0b100	0	フレームあたり 16 ビット 1 スロット
	最大 2	1.024 MHz	2.048 MHz	128 kHz	15	0b010	1	フレームあたり 8 ビット 2 スロット

1. TDM 設定の詳細については、表 569 : TDM 設定を参照してください。SAI に供給される sai_a_ker_ck クロック周波数は、SCK_A 周波数の倍数である必要があり、MCKDIV はそれに応じてプログラムされる必要があります。
2. 上記の sai_a_ker_ck 周波数は、あくまで一例です。「デバイスに生成できるかどうか確認するためのリセットおよびクロックコントローラ (RCC)」のセクションを参照してください。
3. 上表は、デシメーション比が 64 である場合に許容される設定です。

ディレイラインの調整

PDM インタフェースが有効になると、アプリケーションは、SAI_PDMDLY レジスタを通じて各マイクロフォン入力のディレイセルを実行中に調整できます。

新たなディレイ値は、TDM フレームの 2 周期後に有効になります。

53.4.11 AC'97 リンクコントローラ

SAI は AC'97 リンクコントローラとしても機能することができます。このプロトコルの仕様は次のとおりです：

- スロット番号とスロットサイズは固定です。
- フレーム同期信号は完全に定義されており波形は固定です。

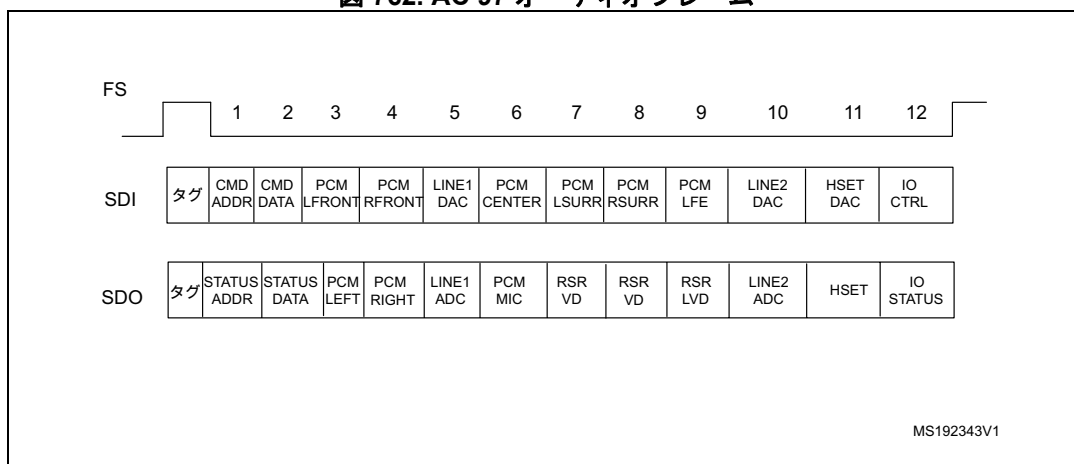
このプロトコルを選択するには、SAI_xCR1 レジスタの PRTCFG[1:0] ビットを 10 に設定します。AC'97 モードを選択した場合、使用可能なデータサイズは 16 ビットまたは 20 ビットのみであり、その他の場合、SAI の動作は保証されません。

- したがって、NBSLOT[3:0] ビットと SLOTSZ[1:0] ビットは無視されます。
- スロット数は 13 スロットに固定されています。最初のスロットは 16 ビット幅で、他はすべて 20 ビット幅（データスロット）です。
- SAI_xSLOTR レジスタの FBOFF[4:0] ビットは無視されます。
- SAI_xFRCCR レジスタは無視されます。
- MCLK は使用されません。

非同期指定されたブロックからの FS 信号は、自動的に出力として設定されます。これは、マスタ/スレーブの設定にかかわらず AC'97 コントローラリンクが FS 信号を駆動するためです。

図 762 に、AC'97 オーディオフレーム構造を示します。

図 762. AC'97 オーディオフレーム



注： AC'97 プロトコルでは、タグのビット 2 は予約済みです（常に 0）。したがって、SAI FIFO に書き込まれる値によらず、タグのビット 2 は 0 レベルに固定されます。

タグ表示についての詳細は、AC'97 プロトコル 規格を参照してください。

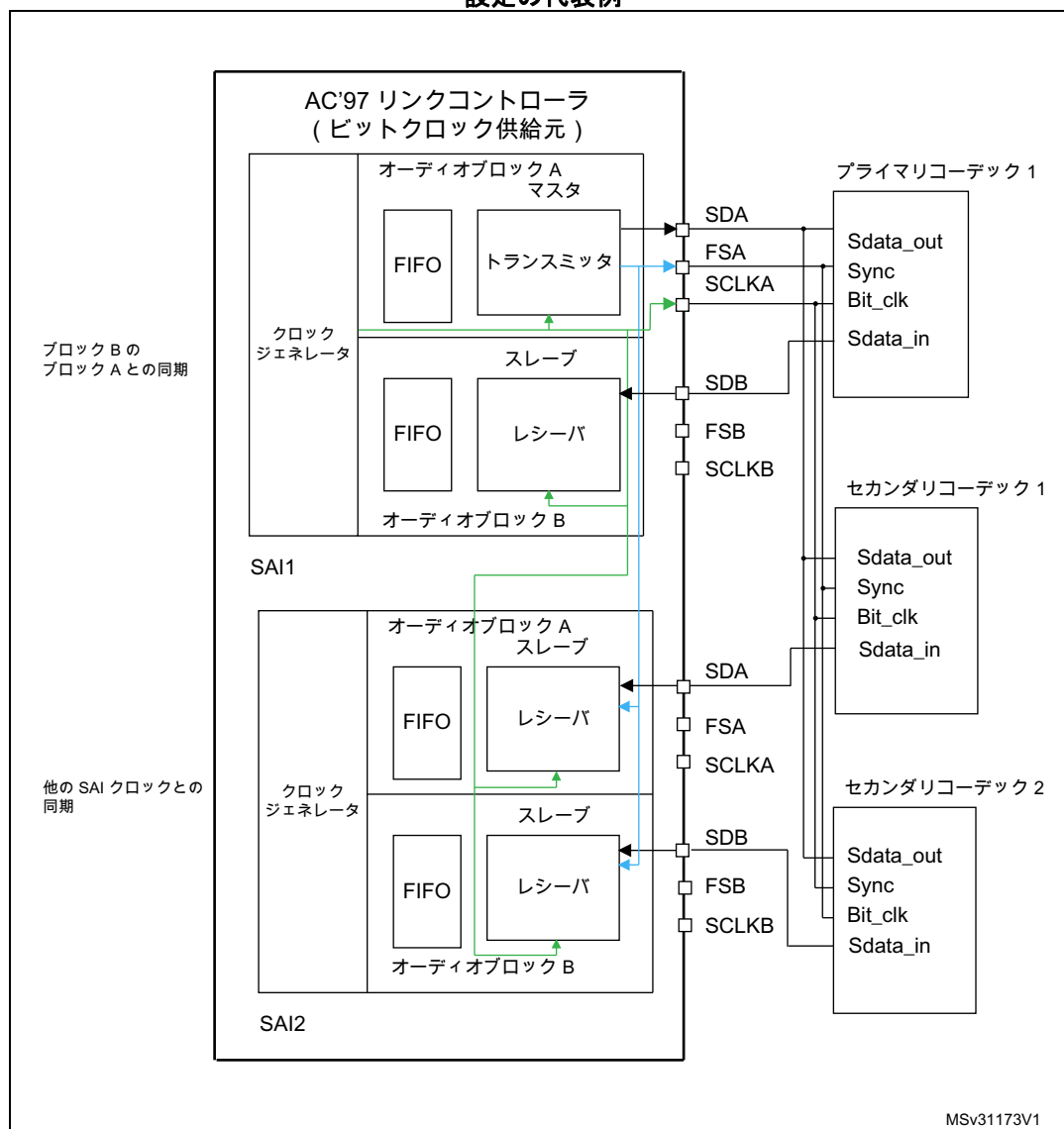
1 つの SAI を AC'97 の 2 点間通信に的を絞って使用することができます。

2 つの SAI を使用することで（デバイスに 2 つの内蔵 SAI を搭載の場合）、図 763 に示す 3 個の外部 AC'97 デコーダが有効化されます。

SAI1 では、オーディオブロック A を非同期マスタトランスミッタに指定する必要がありますが、オーディオブロック B はスレーブレシーバに指定され、オーディオブロック A と内部的に同期します。

SAI2 では、オーディオブロック A と B がどちらもスレーブ受信モードで外部 SAI1 と同期するように設定されます。

図 763. 2 つ以上の内蔵SAI を搭載したデバイス (3 個の外部 AC'97 デコーダ) の AC'97 設定の代表例



受信モードでは、スロット 0 のコーデックレディビットがローにデコードされた場合、AC'97 リンクコントローラとして機能する SAI は、FIFO リクエストを必要としないため FIFO 内のデータストレージも不要です。SAI_xIM レジスタの CNRDYIE ビットが有効化された場合、SAI_xSR レジスタの CNRDY フラグがセットされ、割込みが生成されます。このフラグは、AC'97 プロトコル専用です。

AC'97 モードでのクロックジェネレータのプログラミング

AC'97 モードでは、フレーム長は 256 ビットに固定され、その周波数は 48 kHz に設定する必要があります。[セクション 53.4.8: SAI クロックジェネレータ](#)に示す式には、適切なフレームレート (F_{FS_x}) を生成するために、 $FRL = 255$ という条件を適用する必要があります。

53.4.12 SPDIF 出力

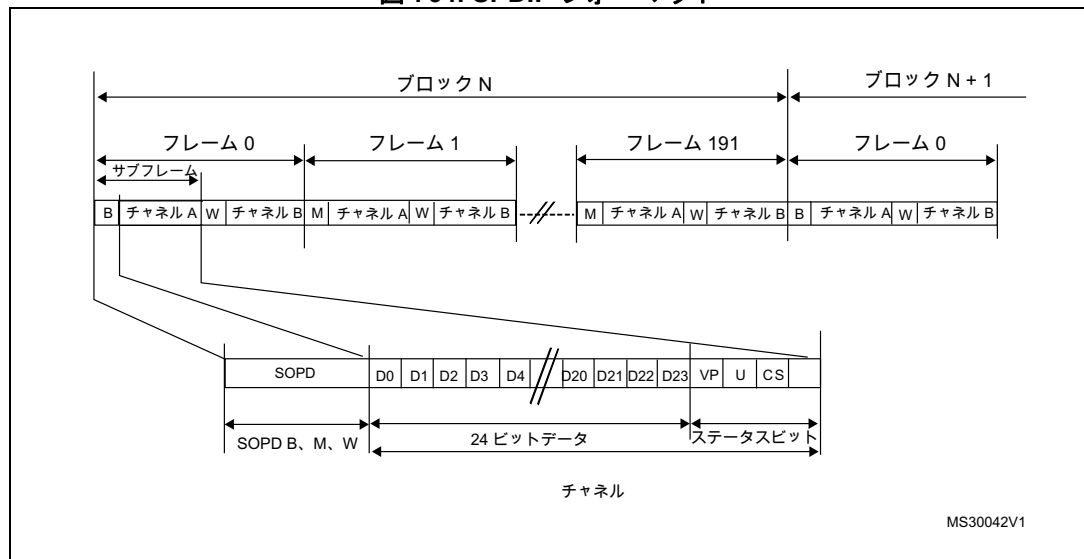
SPDIF インタフェースは送信モードでのみ使用できます。このインタフェースはオーディオ IEC60958 をサポートします。

SPDIF モードを選択するには、SAI_xCR1 レジスタの PRTCFG[1:0] ビットに 01 をセットします。

SPDIF プロトコルを使用する場合

- SD データラインのみが有効です。
- I/O ピンの FS、SCK、MCLK は空けてあります。
- MODE[1] ビットは、SAI のクロックジェネレータを有効にしたり SD ラインでのデータレートを管理するために強制的に 0 にして、マスタモードを選択します。
- データサイズは強制的に 24 ビットに設定されます。SAI_xCR1 レジスタの DS[2:0] ビットにセットされた値は無視されます。
- クロックジェネレータのシンボルレートを、ビットクロックがシンボルレートの 2 倍でなければならないことを考慮して、定義する必要があります。データはマンチェスタープロトコルでコード化されます。
- SAI_xFRCR レジスタおよび SAI_xSLOTR レジスタは無視されます。SAI は、SPDIF プロトコル要件（[図 764](#) を参照）に適合するように内部的に設定されます。

図 764. SPDIF フォーマット



SPDIF ブロックには 192 個のフレームが含まれています。各フレームは、2 個の 32 ビットサブフレーム（一般に左チャンネル用と右チャンネル用に 1 つずつ）で構成されています。各サブフレームは SOPD パターン（4 ビット）で構成されています。このパターンを使用して、サブフレームがブロックの先頭である（つまり、チャンネル A を識別している）、ブロックのどこかでチャンネル A を識別している、またはチャンネル B を参照しているのいずれかを指定します（[表 571](#) を参照）。チャンネル情報の次の 28 ビットは 24 ビットのデータと 4 ビットのステータス情報から成ります。

表 571. SOPD パターン

SOPD	プリアンプルのコード化		説明
	最終ビットが 0	最終ビットが 1	
B	11101000	00010111	ブロックの始めにあるチャンネル A データ
W	11100100	00011011	ブロックのどこかにあるチャンネル B データ
M	11100010	00011101	チャンネル A データ

SAI_xDR に格納されているデータは次のような内容でなければなりません。

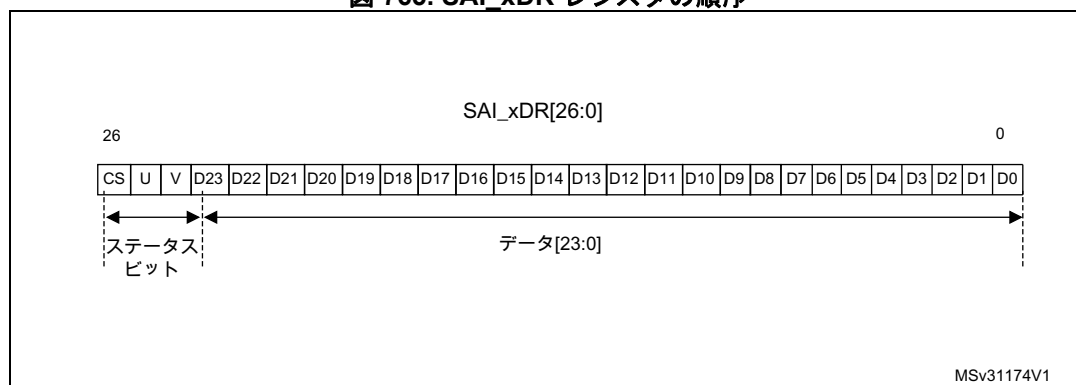
- SAI_xDR[26:24] は、チャンネルのステータスビット、ユーザビット、有効性ビットを含みます。
- SAI_xDR[23:0] は、検討されたチャンネルに関するデータ 24 ビットを含みます。

データサイズが 20 ビットの場合、データは SAI_xDR[23:4] に配置する必要があります。

データサイズが 16 ビットの場合、データは SAI_xDR[23:8] に配置する必要があります。

SAI_xDR[23] は常に MSB を表します。

図 765. SAI_xDR レジスタの順序



注：必ず LSB（最下位ビット）から転送されます。

SAI はまず最初にブロック内の各サブフレームの適切なプリアンプル部を送信します。次に、SAI_xDR が SD ラインに送信されます（マンチェスターコード化）。SAI は、表 572 の記載通りに計算されたパリティビットを転送することで、サブフレームを終了させます。

表 572. パリティビットの計算

SAI_xDR[26:0]	転送されるパリティビット P の値
奇数個の "0 "	0
奇数個の "1 "	1

SAI は送信モードでしか動作できないため、アンダーランは SPDIF モードの SAI_xSR レジスタで有効な唯一のエラーフラグです。そのため、アンダーラン割込みまたはアンダーランステータスビットによって検出されたアンダーランエラーから復帰するためには、次のシーケンスを実行する必要があります。

- DMA が使用されている場合は、DMA ストリームを無効にします（DMA ペリフェラルを使用）。
- SAI を無効にし、SAI_xCR1 レジスタの SAIEN ビットをポーリングすることでペリフェラルが物理的に無効化されていることを確認します。

3. SAI_xCLRFR レジスタの COVRUNDR フラグをクリアします。
4. FIFO を一掃するために、SAI_xCR2 の FFLUSH ビットをセットします。
ソフトウェアで、新しいブロックの始めに今後格納するデータ（プリアンブル部 B 用のデータ）のアドレスを指し示す必要があります。DMA が使用されている場合、DMA ソースベースアドレスのポインタは適宜更新する必要があります。
5. DMA を使用して、新しいソースベースアドレスに従ってデータ転送を管理する場合は、DMA ストリーム（DMA ペリフェラル）を再び有効にします。
6. SAI_xCR1 レジスタの SAIEN ビットをセットして、再び SAI を有効にします。

SPDIF ジェネレータモードでのクロックジェネレータのプログラミング

SPDIF ジェネレータモードの場合、SAI はシンボルレートの 2 倍の速度のビットクロックを供給します。下表では、オーディオサンプリングレートに対するシンボルレートの一般例を示します。

表 573. オーディオサンプリング周波数とシンボルレート

オーディオサンプリング周波数 (F _S)	シンボルレート
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

より一般的には、オーディオサンプリング周波数 (F_S) とビットクロックレート (F_{SCK_X}) の関係は、次の式で求められます。

$$F_S = \frac{F_{SCK_X}}{128}$$

ビットクロックレートは次のように求められます。

$$F_{SCK_X} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

注： 上式は、SAI_ACR1 レジスタで NODIV が 1 にセットされている場合にのみ有効です。

53.4.13 特有の機能

SAI インタフェースには、選択したオーディオプロトコルに応じて役立つ特定の機能が組み込まれています。これらの機能へは、SAI_xCR2 レジスタの特定ビットを使ってアクセスすることができます。

ミュートモード

ミュートモードは、オーディオサブブロックがトランスミッタまたはレシーバであるとき使用することができます。

送信モード時のオーディオサブブロック

送信モード時、ミュートモードはいつでも選択可能です。ミュートモードは、オーディオフレーム全体に対して有効です。処理中のフレーム内で SAI_xCR2 レジスタの MUTE ビットがセットされると、ミュートモードが有効になります。

ミュートモードビットは、フレームの終わりでのみ認識されます。この時点でミュートモードビットがセットされた場合は、ミュートモードは新しいオーディオフレームの先頭から次に来るフレームの

終了までフレーム全体に対して有効になります。その後、このビットを認識して、次のフレームもミュートフレームか否かを決定します。

SAI_xSLOTR レジスタの NBSLOT[3:0] ビットに設定されたスロット数が 2 以下の場合、ミュートモード中に送信される値が 0 なのか、または各スロットの最後の値なのかを指定することができます。この選択は、SAI_xCR2 レジスタの MUTEVAL ビットを使って行います。

SAI_xSLOTR レジスタの NBSLOT[3:0] ビットに設定されたスロット数が 2 より大きい場合、各スロットの各ビットで 0 値が送信されるため、SAI_xCR2 の MUTEVAL ビットは無意味となります。

ミュートモード時は、FIFO ポインタのインクリメントは続きます。つまり、FIFO にデータは供給され続けますが、ミュートモードの間は破棄されます。

受信モード時のオーディオサブブロック

受信モードでは、指定されかつ有効なオーディオフレームのすべてのスロットが、指定された連続した数のオーディオフレームに対して 0 を受信したとき、外部 トランスミッタから送信されたミュートモードを検出することが可能です (SAI_xCR2 レジスタの MUTECONT[5:0] ビット)。

ミュートフレーム数が検出されると、SAI_xSR レジスタの MUTEDET フラグがセットされて、SAI_xCR2 の MUTEDETIE ビットがセットされている場合は割込みを生成することができます。

オーディオサブブロックが無効化されたとき、または有効なスロットがオーディオフレーム内で 1 つ以上のデータを受信したとき、ミュートフレームカウンタがクリアされます。カウンタが MUTECONT[5:0] ビットに指定された値に到達すると、割込みが 1 回だけ生成されます。次に、カウンタがクリアされると割込みイベントが再初期化されます。

注： ミュートモードは SPDIF オーディオブロックには使用できません。

モノラル/ステレオモード

送信モードでは、スロット数が 2 と想定し (SAI_xSLOTR で NBSLOT[3:0] = 0001)、メモリ内でデータを前処理することなくモノラルモードを指定できます。このような場合、スロット 0 のデータがデータスロット 1 へ複製されるため、FIFO へのアクセスタイムは半減されます。

モノラルモードを有効にするには、

1. SAI_xCR1 レジスタの MONO ビットに 1 をセットします。
2. SAI_xSLOTR レジスタの NBSLOT ビットに 1、SLOTEN ビットに 3 をセットします。

受信モードでは、MONO ビットをセットすることができますが、送信モードの場合と同様にスロット数が 2 のときだけ意味があります。このビットをセットした場合、スロット 0 のデータだけが FIFO に格納されます。この場合、前のスロットと同じと見なされるため、スロット 1 に属するデータは無視されます。受信時のデータの流れが左右でデータが異なる真のステレオオーディオフローである場合、MONO ビットは意味を成しません。出力ステレオファイルから等価なモノラルファイルへの変換は、ソフトウェアで行われます。

コンパANDINGモード

通信アプリケーションでは、データコンパANDINGアルゴリズムにより送信データまたは受信データの処理が必要となることがあります。

[図 766](#) に示すように、SAI_xCR2 レジスタの COMP[1:0] ビットに応じて (フリープロトコルモードを選択した場合のみ使用)、SD シリアル出力ラインに送信する前に処理 (圧縮) するか否か、あるいは SD シリアル入力ラインから受信した後にデータを処理 (伸長) するか否かをソフトウェアで選択することができます。サポートされている 2 つのコンパANDINGモードは μ -Law と A-Law (対数) で、これらは CCITT G.711 勧告の一部となっています。

米国と日本で採用されているコンパANDING規格は μ -Law であり、14 ビットのダイナミックレンジが確保できます (SAI_xCR2 レジスタで COMP[1:0] = 10)。

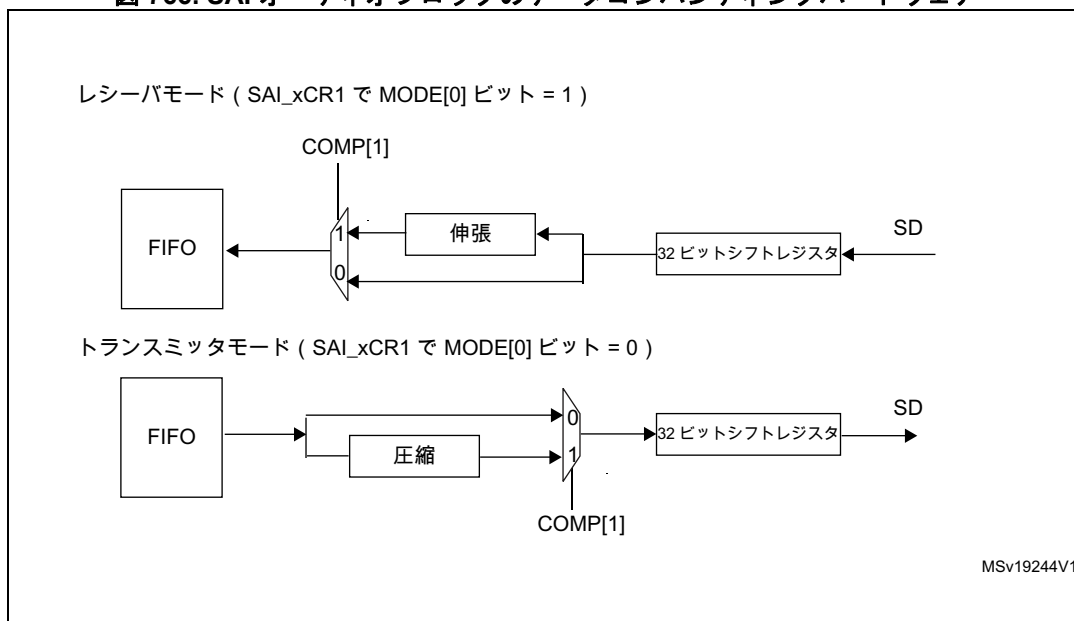
ヨーロッパのコンパニング規格は A-Law であり、13 ビットのダイナミックレンジが確保できます (SAI_xCR2 レジスタで COMP[1:0] = 11)。

コンパニング規格 (μ-Law または A-Law) は、SAI_xCR2 レジスタの CPL ビットの設定に応じて、1 の補数表現または 2 の補数表現に基づいて計算することができます。

μ-Law および A-Law 規格では、データを MSB 詰めの 8 ビットコードにコード化します。コンパンドされたデータは常に 8 ビット幅です。このため、SAI オーディオブロックが有効化されていて (SAI_xCR1 レジスタで SAIEN ビット = 1)、さらに、COMP[1:0] ビットによりこれら 2 つのコンパニングモードのいずれかを選択している場合、SAI_xCR1 レジスタの DS[2:0] ビットは強制的に 010 に固定されます。

コンパニングが不要な場合は、COMP[1:0] ビットをクリアしたままにする必要があります。

図 766. SAI オーディオブロックのデータコンパニングハードウェア



1. AC'97 または SPDIF を選択した場合は適用されません。

伸張モードまたは圧縮モードは SAI_xCR2 により自動的に選択されます。

- SAI オーディオブロックがトランスミッタに設定され、かつ SAI_xCR2 レジスタの COMP[1] ビットがセットされている場合、圧縮モードが適用されます。
- SAI オーディオブロックがレシーバに指定された場合、伸長アルゴリズムが適用されます。

非アクティブスロットでの出力データライン管理

送信モードでは、非アクティブスロットがデータラインに送信されたときに出力となっている SD ラインの動作を選択することができます (TRIS ビットを使用)。

- 非アクティブスロットが送信されるとき、SAI が SD 出力ラインを 0 に固定することもできれば、
- 送信データの最終ビットの終わりでラインを HI-Z 状態にして、このノードに接続された他のトランスミッタへラインを開放することもできます。

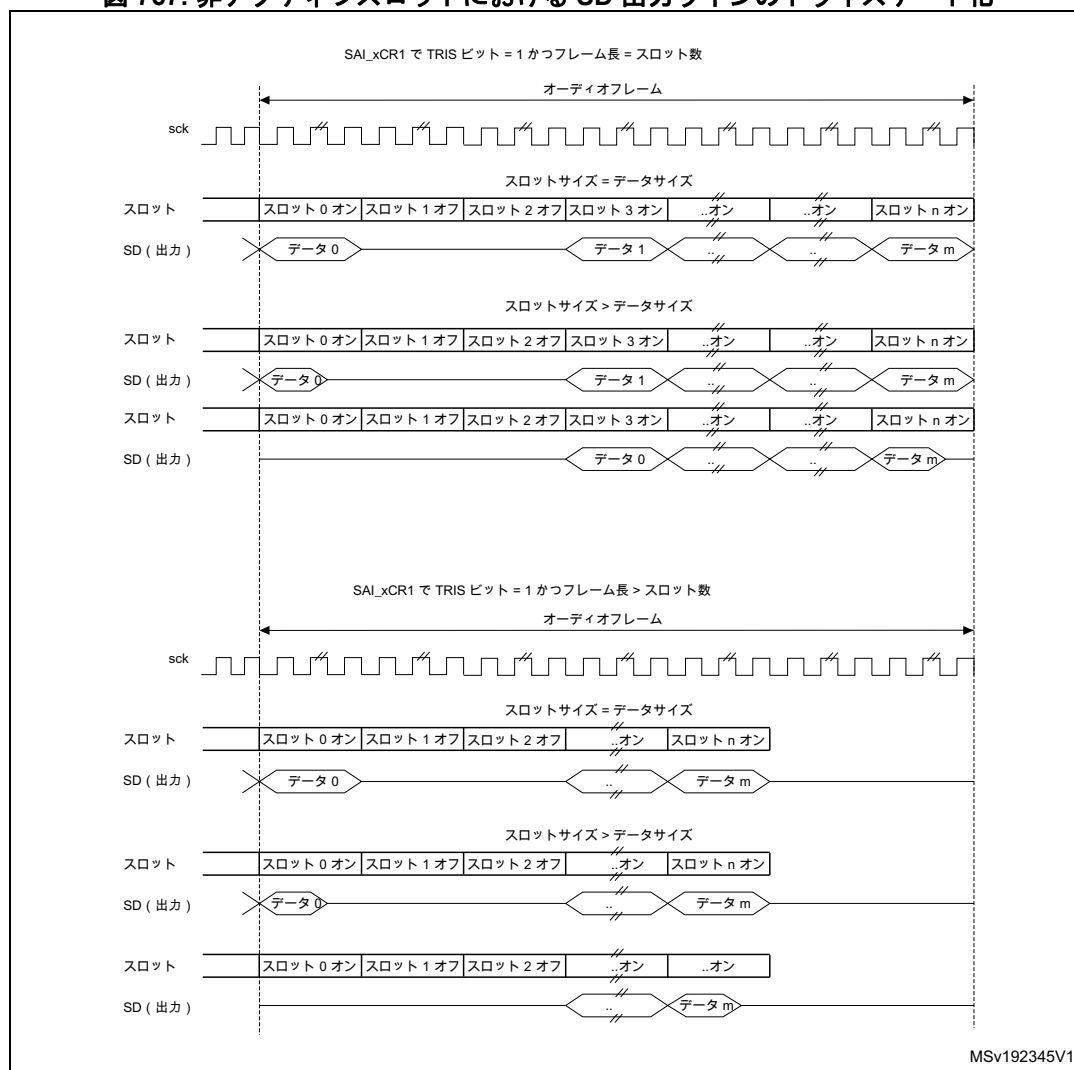
2 つのトランスミッタが同じ SD 出力ピンを同時に駆動しないように注意することが重要です。同時駆動した場合、短絡の原因となります。送信間の間隔を確保するため、データが 32 ビットより短い場合は、SAI_xSLOTR レジスタの SLOTSZ[1:0] ビット = 10 に設定することにより、データを 32 ビットに拡張することができます。すると、次のスロットが非アクティブに指定された場合、アクティブ

スロットの LSB の終わりで（データを 32 ビットに拡張するため 0 を詰めるフェーズで）SD 出力ピンがトライステートになります。

さらに、スロット数とスロットサイズの積がフレーム長より小さい場合、0 を詰めてオーディオフレームを完成させるときに SD 出力ラインがトライステートになります。

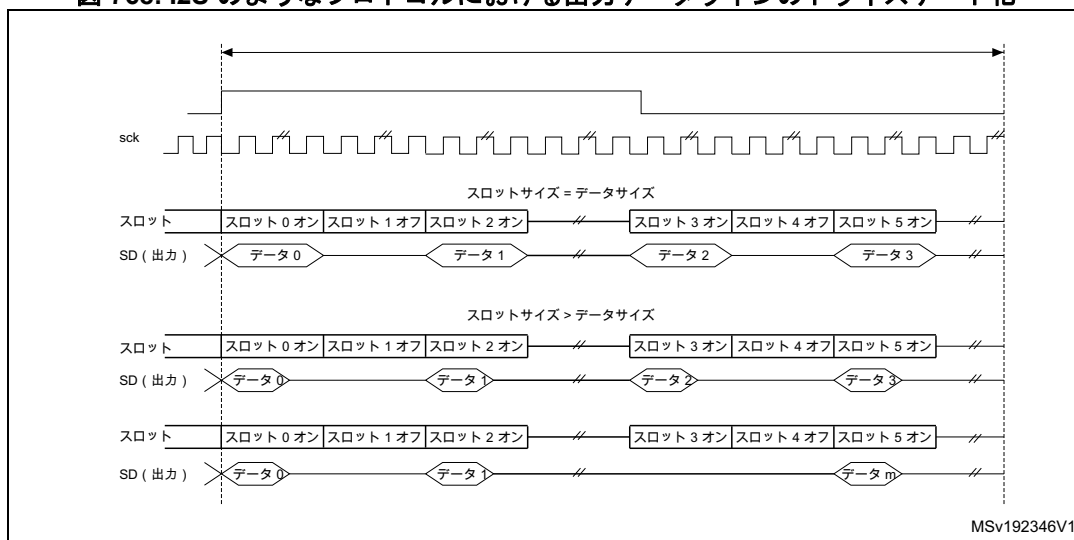
これらの動作を図 767 に示します。

図 767. 非アクティブスロットにおける SD 出力ラインのトライステート化



選択したオーディオプロトコルが FS 信号をフレーム開始とチャネルサイド識別として使用する場合 (SAI_xFRCR レジスタでビット FSDEF = 1)、図 768 に従ってトライステートモードが管理されます (この図では、SAI_xCR1 レジスタでビット TRIS = 1、FSDEF = 1、ハーフフレーム長 > スロット数 / 2、NBSLOT = 6)。

図 768. I2S のようなプロトコルにおける出力データラインのトライステート化



MSv192346V1

SAI_xCR2 レジスタの TRIS ビットがクリアされている場合、図 767 と 図 768 に示す SD 出力ライン上のすべてのハイインピーダンス状態は 0 値での駆動として置き換えます。

53.4.14 エラーフラグ

SAI は次のエラーフラグを実装しています。

- FIFO オーバーラン/アンダーラン
- 予測フレーム同期検出
- レイトフレーム同期検出
- コーデックノットレディ (AC'97 専用)
- マスタモードにおける不正クロック設定

FIFO オーバーラン/アンダーラン (OVRUDR)

FIFO オーバーラン/アンダーランビットは、SAI_xSR レジスタで OVRUDR と呼ばれます。

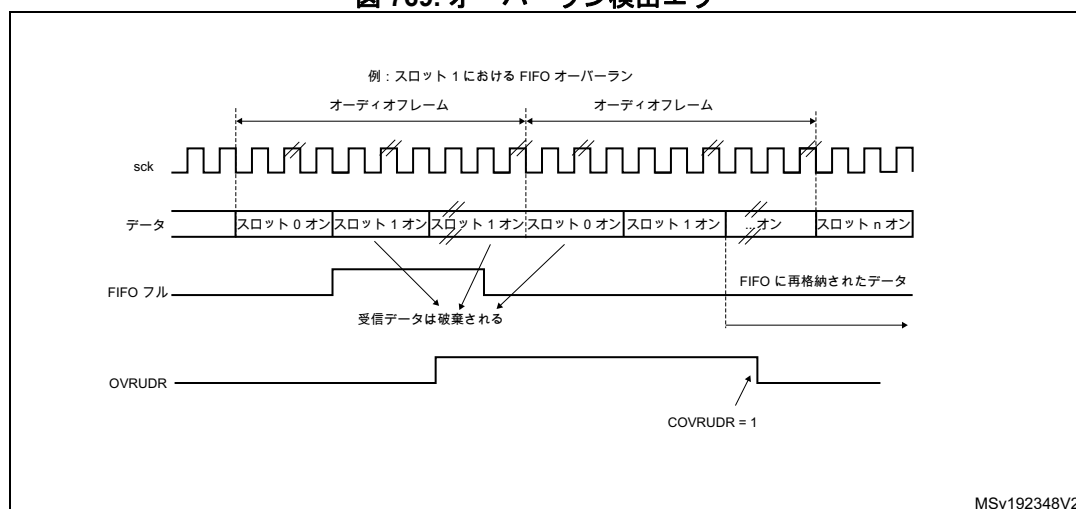
オーディオブロックはレシーバまたはトランスミッタのいずれかであり、SAI 内の各オーディオブロックに固有の SAI_xSR レジスタを搭載しているため、オーバーランエラーとアンダーランエラーは、同じビットを使用しています。

オーバーラン

オーディオブロックがレシーバに設定されていて、FIFO がフルで受信データを保存できないときにオーディオフレームでデータが受信された場合に、オーバーラン状態が発生します。この場合、受信データが失われ、SAI_xSR レジスタの OVRUDR フラグがセットされ、SAI_xIM レジスタの OVRUDRIE ビットがセットされている場合は割込みが生成されます。オーバーランが発生したスロット番号が、内部に保存されます。FIFO が新しいデータを格納できるようになるまで、データは FIFO に保存されません。FIFO に 1 データ分以上の空きがある場合、SAI オーディオブロックレシーバは、オーバーラン状態が検出されたとき内部に保存されたスロット番号からの新しいデータ（新しいオーディオフレームの）を保存します。これにより、転送先メモリでのデータスロットの配置の狂いを回避します（図 769 を参照）。

SAI_xCLRFR レジスタの COVRUDR ビットがセットされると、OVRUDR フラグがクリアされます。

図 769. オーバーラン検出エラー



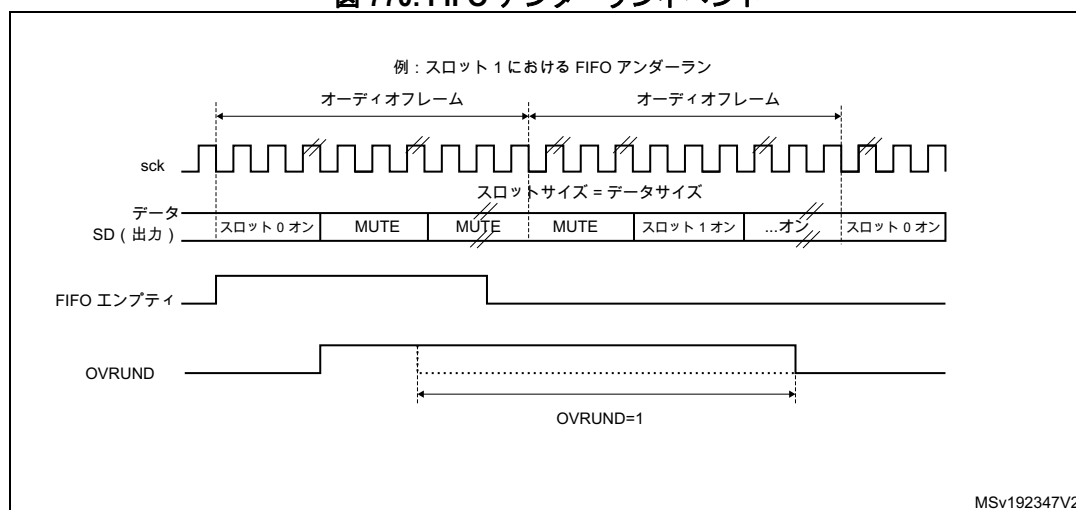
アンダーラン

SAI 内のオーディオブロックがトランスミッタで、かつデータの送信が必要なとき FIFO がエンプティ（空）の場合に、アンダーランが発生します。アンダーランが検出されると、イベントが発生したスロット番号が保存され、FIFO がアンダーランが検出されたスロットに対応するデータを送信する準備ができるまで MUTE 値 (00) が送信されます（図 770 を参照）。これにより、メモリポインタとオーディオフレーム内のスロットの同期はずれを回避します。

アンダーランイベントにより SAI_xSR レジスタの OVRUDR フラグがセットされ、SAI_xIM レジスタの OVRUDRIE ビットがセットされている場合、割込みが生成されます。このフラグをクリアするには、SAI_xCLRFR レジスタの COVRUDR ビットをセットします。

アンダーランイベントは、オーディオサブブロックがマスタまたはスレーブとして設定されている場合に発生します。

図 770. FIFO アンダーランイベント



予測フレーム同期検出 (AFSDET)

AFSDET フラグはスレーブモードでのみ使用されます。マスタモードでは決してアサートされません。フレーム長、フレーム極性、フレームオフセットが設定されていて既知であるため、フレーム同期 (FS) が予測より早く検出されることを示します。

予測フレームの検出により、SAI_xSR レジスタの AFSDET フラグがセットされます。

この検出は、現在のオーディオフレームに影響を与えません。現在のオーディオフレームは予測 FS に影響されません。つまり、現在のオーディオフレームを乱すことなく、FS 信号に関する「寄生」イベントフラグが立つということです。

SAI_xIM レジスタの AFSDETIE ビットがセットされている場合は、割込みが生成されます。AFSDET フラグをクリアするには、SAI_xCLRFR レジスタの CAFSDET ビットをセットする必要があります。

予測フレーム検出エラーの後にマスタと再同期させるには、次の 4 ステップが必要です。

1. SAI_xCR1 レジスタの SAIEN ビットをリセットして、SAI ブロックを無効化します。SAI が無効であることを確認するには、SAIEN ビットを読み返し、それが 0 にセットされていることを確認します。
2. SAI_xCR2 レジスタの FFLUS ビットを使って FIFO を一掃します。
3. SAI ペリフェラルを再度有効化します (SAIEN ビットを 1 に設定)。
4. SAI ブロックは FS の開始を待ってマスタとの同期を再開します。

注： AFSDET フラグは AC'97 モードではアサートされません。これは SAI オーディオブロックがリンクコントローラとして機能し、スレーブに指定された場合でも FS 信号を生成するためです。SPDIF モードでは FS 信号は使用されないため、このフラグは意味を持ちません。

レイトフレーム同期検出

SAI オーディオブロックがスレーブとして動作する場合のみ、SAI_xSR レジスタの LFSDET フラグをセットすることができます。フレーム長、フレーム極性、フレームオフセット設定は、SAI_xFRCR レジスタから知ることができます。

外部マスタが FS 信号を予想タイミングで送信しない (信号の発生が遅すぎる) 場合、LFSDET フラグがセットされ、SAI_xIM レジスタの LFSDETIE ビットがセットされている場合は割込みが生成されます。

SAI_xCLRFR レジスタの CLFSDET ビットがセットされると、LFSDET フラグがクリアされます。

レイトフレーム同期検出フラグは、対応するエラーが検出されるとセットされます。SAI はマスタと再同期させる必要があります (シーケンスは、[予測フレーム同期検出 \(AFSDET\)](#) を参照)。

ノイズの多い環境では、オーディオブロックステートマシンが SCK クロック上のグリッチを間違っ
て検出し、SAI データが間違っ
たフレーム位置にシフトされることがあります。このようなイベントは SAI によって検出され、レイトフレーム同期検出エラーとして報告することができます。

外部マスタが連続モードでオーディオデータフレーム転送を管理していない場合、破壊は起こりませんが、ほとんどの場合、アプリケーションでこのような使い方はできません。この場合、LFSDET フラグがセットされます。

注： LFSDET フラグは AC'97 モードではアサートされません。これは SAI オーディオブロックがリンクコントローラとして機能し、スレーブと指定された場合でも FS 信号を生成するためです。SPDIF モードでは FS 信号はプロトコルで使用されないため、このフラグは意味を持ちません。

コーデックノットレディ (CNRDY AC'97)

SAI オーディオブロックが AC'97 モードで動作するように設定されている場合にのみ SAI_xSR レジスタの CNRDY フラグが関係します (SAI_xCR1 レジスタの PRTCFG[1:0] ビット = 10)。SAI_xIM レジスタの CNRDYIE ビットがセットされている場合、CNRDY フラグがセットされたときに割込みが生成されます。

AC'97 オーディオフレームのタグ 0 (スロット 0) の受信中にコーデックが通信の準備ができていないとき、CNRDY フラグがアサートされます。この場合、コーデックが動作可能であることをタグ 0 が示すまではコーデックは動作可能状態にはないため、FIFO に自動的に保存されるデータはありません。コーデックが動作可能状態になると、SAI_xSLOTR レジスタで指定されたすべてのアクティブスロットがキャプチャされます。

CNRDY フラグをクリアするには、SAI_xCLRFR レジスタの CCNRDY ビットをセットする必要があります。

マスタモードでの不正なクロック設定 (NODIV = 0 の場合)

オーディオブロックがマスタとして動作し (MODE[1] = 0)、NODIV ビットが 0 に等しい場合、次の条件が満たされて SAI が有効になるとすぐに、WCKCFG フラグがセットされます。

- (FRL + 1) は 2 の累乗値ではない、かつ
- (FRL + 1) は 8~256 の範囲外である。

MODE、NODIV、SAIEN ビットは SAI_xCR1 レジスタに、FRL ビットは SAI_xFRCR レジスタに属します。

WCKCFGIE ビットがセットされている場合、SAI_xSR レジスタで WCKCFG フラグがセットされると割込みが生成されます。このフラグをクリアするには、SAI_xCLRFR レジスタの CWCKCFG ビットをセットします。

WCKCFG ビットがセットされると、オーディオブロックが自動的に無効化され、ハードウェアによって SAIEN ビットがクリアされます。

53.4.15 SAI の無効化

SAI_xCR1 レジスタの SAIEN ビットをクリアすることによって、SAI オーディオブロックをいつでも無効化することができます。すでに開始しているすべてのフレームは、SAI が動作を停止する前に自動的に完了します。現在のオーディオフレーム転送の最後に SAI が完全にオフするまで、SAIEN ビットはハイを保持します。

他の SAI と同期している SAI 内にオーディオブロックがある場合、マスタになっている SAI を先に無効化する必要があります。

53.4.16 SAI DMA インタフェース

CPU を開放し、バスの帯域幅を最適化するため、各 SAI オーディオブロックには SAI_xDR レジスタ (内部 FIFO へのアクセス用) に対して読み書きを行うための独立した DMA インタフェースを搭載しています。オーディオサブブロック毎に 1 つの DMA チャンネルがあり、基本的な DMA リクエスト/確認応答プロトコルに対応しています。

DMA 転送を行うようにオーディオサブブロックを設定するには、SAI_xCR1 レジスタの DMAEN ビットをセットします。DMA リクエストは、FIFO 閾値レベルに応じて FIFO コントローラによって直接管理されます (詳細は、[セクション 53.4.9: 内部 FIFO](#) を参照してください)。DMA 転送方向は、SAI オーディオサブブロック設定と次の様に関連付けられています。

- オーディオブロックがトランスミッタとして動作する場合、オーディオブロックの FIFO コントローラが DMA リクエストを出力して SAI_xDR レジスタに書き込まれたデータを FIFO にロードします。
- オーディオブロックがレシーバとして動作する場合、DMA リクエストは SAI_xDR レジスタからの読出し動作に関わります。

SAI インタフェースを DMA モードに設定するには、次のシーケンスに従ってください。

- SAI と FIFO 閾値レベルを設定して、DMA リクエストを開始するタイミングを指定します。
- SAI DMA チャンネルを設定します。
- DMA を有効にします。
- SAI インタフェースを有効にします。

注： SAI ブロックを設定する前に、SAI DMA チャンネルを無効化する必要があります。

53.5 SAI 割込み

SAI は表 574 に示す 7 つの割込みソースに対応しています。

表 574. SAI 割込みソース

項目 (割込みの 略称)	割込み ソース	割込み グループ	オーディオブロック モード	割込みイネーブル	割込みクリア
SAI	FREQ	FREQ	マスタ/スレーブ 受信または送信モード	SAI_xIM レジスタ の FREQIE	下記に応じてクリアします。 – FIFO 閾値設定 (SAI_xCR2 の FLVL ビット) – 通信方向 (トランスミッタ/レシーバ) 詳細については、 セクション 53.4.9 : 内部 FIFO を参照してください。
	OVRUDR	ERROR	マスタ/スレーブ 受信または送信モード	SAI_xIM レジスタ の OVRUDRIE	SAI_xCLRFR レジスタの COVRUDR = 1
	AFSDDET	ERROR	スレーブ (AC'97 モードおよび SPDIF モードでは使用 しません。)	SAI_xIM レジスタ の AFSDETIE	SAI_xCLRFR レジスタで CAFSDDET = 1
	LFSDDET	ERROR	スレーブ (AC'97 モードおよび SPDIF モードでは使用 しません。)	SAI_xIM レジスタ の LFSDETIE	SAI_xCLRFR レジスタで CLFSDDET = 1
	CNRDY	ERROR	スレーブ (AC'97 モードの場合の み)	SAI_xIM レジスタ の CNRDYIE	SAI_xCLRFR レジスタで CCNRDY = 1
	MUTEDET	MUTE	マスタ/スレーブ 受信モードのみ	SAI_xIM レジスタ の MUTEDETIE	SAI_xCLRFR レジスタで CMUTEDET = 1
	WCKCFG	ERROR	マスタ (SAI_xCR1 レジ スタで NODIV = 0 の場 合)	SAI_xIM レジスタ の WCKCFGIE	SAI_xCLRFR レジスタで CWCKCFG = 1

割込みを有効にするには、次のシーケンスに従ってください。

1. SAI 割込みを無効化します。
2. SAI を設定します。
3. SAI 割込みソースを設定します。
4. SAI を有効化します。

53.6 SAI レジスタ

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。

53.6.1 SAI グローバル設定レジスタ (SAI_GCR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
										rw	rw			rw	rw

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **SYNCOUT[1:0]** : 同期出力

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : 同期信号は出力されません。オーディオブロックが SPDIF として設定されている場合、SYNCOUT[1:0] は「同期信号出力なし」に設定する必要があります。

01 : ブロック A は他の SAI 用に追加された同期に使用します。

10 : ブロック B は他の SAI 用に追加された同期に使用します。

11 : 予約済み。両方のオーディオブロック（A および B）を無効化して、これらのビットを設定する必要があります。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **SYNCIN[1:0]** : 同期入力

これらのビットは、ソフトウェアによってセット／クリアされます。

このフィールドの設定方法については、表 566 : 外部同期の選択を参照してください。

両方のオーディオブロック（A および B）を無効化して、これらのビットを設定する必要があります。

2 つのオーディオブロックのうち 1 つが外部 SAI で同期モードで動作するように設定されている場合 (SAI_ACR1 または SAI_BCR1 レジスタで SYNCEN[1:0] = 10)、これらのビットは意味を持ちます。

53.6.2 SAI 設定レジスタ 1 (SAI_ACR1)

アドレスオフセット : 0x004

リセット値 : 0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCK EN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDR V	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFCG[1:0]		MODE[1:0]	
		rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **MCKEN** : マスタクロック生成イネーブル

0 : マスタクロックは生成されません。

1 : マスタクロックが SAIEN ビットから独立して生成されます。

ビット 26 **OSR** : マスタクロックのオーバーサンプリング比

このビットは、NODIV ビットが 0 にセットされているときにのみ意味を持ちます。

0 : マスタクロック周波数 = $F_{FS} \times 256$

1 : マスタクロック周波数 = $F_{FS} \times 512$

ビット 25:20 **MCKDIV[5:0]** : マスタクロック分周回路

これらのビットは、ソフトウェアによってセット／クリアされます。

000000 : 分周比 1 のカーネルクロック入力 (sai_x_ker_ck)。

その他の場合、マスタクロック周波数は[セクション 53.4.8: SAI クロックジェネレータ](#)で指定された式で計算されます。

オーディオブロックがスレーブの場合、これらのビットに意味はありません。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

ビット 19 **NODIV** : 分周器なし

このビットは、ソフトウェアによってセット／クリアされます。

0 : マスタクロックジェネレータとフレーム同期の比率が 256 または 512 に固定されます。

1 : マスタクロックジェネレータとフレーム同期の比率が FRL[7:0] に依存します。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DMAEN** : DMA 有効化

このビットは、ソフトウェアによってセット／クリアされます。

0 : DMA は無効です。

1 : DMA は有効です。

注 : オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの DMA リクエストを回避するために、MODE[1:0] ビットは DMAEN をセットする前に設定する必要があります。

ビット 16 **SAIEN** : オーディオブロック有効化

このビットはソフトウェアでセットされます。

オーディオブロックをスイッチオフするには、アプリケーションソフトウェアでこのビットを 0 に設定し、このビットが 0 に読み返されるまでポーリングする必要があります。これは、ブロックを完全に無効にするということです。このビットを 1 にセットする前に、それが 0 にセットされていることを確認します。そうしないと、有効化コマンドは考慮されません。

このビットは、SAI オーディオブロックの状態を制御できるようにします。オーディオフレーム転送の途中で無効化した場合、処理中の転送は完了し、このオーディオフレーム転送の転送後にセル全体が完全に無効化されます。

0 : SAI オーディオブロックは無効です。

1 : SAI オーディオブロックは有効です。

注 : **SAI ブロック (A または B) をマスタモードに設定した場合、SAIEN ビットをセットする前に SAI の入力にクロックが存在する必要があります。**

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **OUTDRIV** : 出力駆動

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAIEN がセットされたときオーディオブロック出力が駆動されます。

1 : このビットをセットした直後にオーディオブロック出力が駆動されます。

注 : **このビットは、オーディオブロックを有効化する前で、かつオーディオブロックを設定した後にセットする必要があります。**

ビット 12 **MONO** : モノラルモード

このビットは、ソフトウェアによってセット/クリアされます。このビットはスロット数が 2 のときにのみ意味を持ちます。モノラルモードを選択した場合、オーディオブロックがトランスミッタとして動作する場合にスロット 0 のデータがスロット 1 に複製されます。受信モードでは、スロット 1 は破棄され、スロット 0 から受け取ったデータのみが保存されます。詳細については、[セクション : モノラル/ステレオモード](#)を参照してください。

0 : ステレオモード

1 : モノラルモード

ビット 11:10 **SYNCEN[1:0]** : 同期有効化

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオサブブロックを無効化して、これらのビットを設定する必要があります。

00 : オーディオサブブロックは非同期モードです。

01 : オーディオサブブロックは他の内部オーディオサブブロックと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

10 : オーディオサブブロックは他の SAI 内蔵ペリフェラルと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

11 : 予約済みです。

注 : **オーディオサブブロックは、SPDIF モードが有効な場合、非同期として設定する必要があります。**

ビット 9 **CKSTR** : クロックストローブエッジ

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。このビットは、SPDIF オーディオプロトコルでは意味を持ちません。

0 : SCK 立上りエッジでの SAI の変化によって信号が生成され、SCK 立下りエッジでの SAI のサンプリングによって信号が受信されます。

1 : SCK 立下りエッジでの SAI の変化によって信号が生成され、SCK 立上りエッジでの SAI のサンプリングによって信号が受信されます。

ビット 8 **LSBFIRST** : LSB ファースト

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。AC'97 ではデータは常に MSB が先で転送されるため、このビットは AC'97 オーディオプロトコルでは意味を持ちません。SPDIF ではデータは常に LSB 先頭で転送されるため、このビットは SPDIF オーディオプロトコルでは意味を持ちません。

0 : データは MSB が先に転送されます。

1 : データは LSB から先に転送されます。

ビット 7:5 DS[2:0] : データサイズ

これらのビットは、ソフトウェアによってセット／クリアされます。SPDIF プロトコルを選択した場合 (PRTCFG[1:0] ビット)、フレームおよびデータサイズは固定されているため、これらのビットは無視されます。コンパニディングモードを選択した場合 (COMP[1:0] ビット)、データサイズはアルゴリズムによって 8 ビットに固定されているため、DS[1:0] は無視されます。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

000 : 予約済みです。

001 : 予約済みです。

010 : 8 ビット

011 : 10 ビット

100 : 16 ビット

101 : 20 ビット

110 : 24 ビット

111 : 32 ビット

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3:2 PRTCFG[1:0] : プロトコル設定

これらのビットは、ソフトウェアによってセット／クリアされます。オーディオブロックを無効化して、これらのビットを設定する必要があります。

00 : フリープロトコル。フリープロトコルを選択すると、オーディオブロックの強力な設定機能を使用できるようになり、大部分の設定レジスタビットとフレーム設定レジスタを設定して、特定のオーディオプロトコル（たとえば、I2S、LSB/MSB 詰め、TDM、PCM/DSP など）を処理することができます。

01 : SPDIF プロトコル

10 : AC'97 プロトコル

11 : 予約済みです。

ビット 1:0 MODE[1:0] : SAIx オーディオブロックモード

これらのビットは、ソフトウェアによってセット／クリアされます。SAIx オーディオブロックを無効化した場合、これらのビットを設定する必要があります。

00 : マスタトランスミッタ

01 : マスタレシーバ

10 : スレーブトランスミッタ

11 : スレーブレシーバ

注 : オーディオブロックを SPDIF モードに設定した場合、マスタ送信モードが強制されます (MODE[1:0] = 00)。

53.6.3 SAI 設定レジスタ 1 (SAI_BCR1)

アドレスオフセット : 0x024

リセット値 : 0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCK EN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDR V	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFCG[1:0]		MODE[1:0]	
		rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **MCKEN** : マスタクロック生成イネーブル

0 : マスタクロックは生成されません。

1 : マスタクロックが SAIEN ビットから独立して生成されます。

ビット 26 **OSR** : マスタクロックのオーバーサンプリング比

このビットは、NODIV ビットが 0 にセットされているときにのみ意味を持ちます。

0 : マスタクロック周波数 = $F_{FS} \times 256$

1 : マスタクロック周波数 = $F_{FS} \times 512$

ビット 25:20 **MCKDIV[5:0]** : マスタクロック分周回路

これらのビットは、ソフトウェアによってセット／クリアされます。

000000 : 分周比 1 のカーネルクロック入力 (sai_x_ker_ck)。

その他の場合、マスタクロック周波数は[セクション 53.4.8: SAI クロックジェネレータ](#)で指定された式で計算されます。

オーディオブロックがスレーブの場合、これらのビットに意味はありません。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

ビット 19 **NODIV** : 分周器なし

このビットは、ソフトウェアによってセット／クリアされます。

0 : マスタクロックジェネレータとフレーム同期の比率が 256 または 512 に固定されます。

1 : マスタクロックジェネレータとフレーム同期の比率が FRL[7:0] に依存します。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DMAEN** : DMA 有効化

このビットは、ソフトウェアによってセット／クリアされます。

0 : DMA は無効です。

1 : DMA は有効です。

注 : オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの DMA リクエストを回避するために、MODE[1:0] ビットは DMAEN をセットする前に設定する必要があります。

ビット 16 SAIEN : オーディオブロック有効化

このビットはソフトウェアでセットされます。

オーディオブロックをスイッチオフするには、アプリケーションソフトウェアでこのビットを 0 に設定し、このビットが 0 に読み返されるまでポーリングする必要があります。これは、ブロックを完全に無効にするということです。このビットを 1 にセットする前に、それが 0 にセットされていることを確認します。そうしないと、有効化コマンドは考慮されません。

このビットは、SAI オーディオブロックの状態を制御できるようにします。オーディオフレーム転送の途中で無効化した場合、処理中の転送は完了し、このオーディオフレーム転送の転送後にセル全体が完全に無効化されます。

0 : SAI オーディオブロックは無効です。

1 : SAI オーディオブロックは有効です。

注 : SAI ブロック (A または B) をマスタモードに設定した場合、SAIEN ビットをセットする前に SAI の入力にクロックが存在する必要があります。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 OUTDRIV : 出力駆動

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAIEN がセットされたときオーディオブロック出力が駆動されます。

1 : このビットをセットした直後にオーディオブロック出力が駆動されます。

注 : このビットは、オーディオブロックを有効化する前で、かつオーディオブロックを設定した後にセットする必要があります。

ビット 12 MONO : モノラルモード

このビットは、ソフトウェアによってセット/クリアされます。このビットはスロット数が 2 のときにのみ意味を持ちます。モノラルモードを選択した場合、オーディオブロックがトランスミッタとして動作する場合にスロット 0 のデータがスロット 1 に複製されます。受信モードでは、スロット 1 は破棄され、スロット 0 から受け取ったデータのみが保存されます。詳細については、[セクション : モノラル/ステレオモード](#)を参照してください。

0 : ステレオモード

1 : モノラルモード

ビット 11:10 SYNCEN[1:0] : 同期有効化

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオサブブロックを無効化して、これらのビットを設定する必要があります。

00 : オーディオサブブロックは非同期モードです。

01 : オーディオサブブロックは他の内部オーディオサブブロックと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

10 : オーディオサブブロックは他の SAI 内蔵ペリフェラルと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

11 : 予約済みです。

注 : オーディオサブブロックは、SPDIF モードが有効な場合、非同期として設定する必要があります。

ビット 9 CKSTR : クロックストローブエッジ

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。このビットは、SPDIF オーディオプロトコルでは意味を持ちません。

0 : SCK 立上りエッジでの SAI の変化によって信号が生成され、SCK 立下りエッジでの SAI のサンプリングによって信号が受信されます。

1 : SCK 立下りエッジでの SAI の変化によって信号が生成され、SCK 立上りエッジでの SAI のサンプリングによって信号が受信されます。

ビット 8 LSBFIRST : LSB ファースト

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。AC'97 ではデータは常に MSB が先で転送されるため、このビットは AC'97 オーディオプロトコルでは意味を持ちません。SPDIF ではデータは常に LSB 先頭で転送されるため、このビットは SPDIF オーディオプロトコルでは意味を持ちません。

0 : データは MSB が先に転送されます。

1 : データは LSB から先に転送されます。

ビット 7:5 **DS[2:0]** : データサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。SPDIF プロトコルを選択した場合 (PRTCFG[1:0] ビット)、フレームおよびデータサイズは固定されているため、これらのビットは無視されます。コンパニングモードを選択した場合 (COMP[1:0] ビット)、データサイズはアルゴリズムによって 8 ビットに固定されているため、DS[1:0] は無視されます。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

000 : 予約済みです。

001 : 予約済みです。

010 : 8 ビット

011 : 10 ビット

100 : 16 ビット

101 : 20 ビット

110 : 24 ビット

111 : 32 ビット

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3:2 **PRTCFG[1:0]** : プロトコル設定

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、これらのビットを設定する必要があります。

00 : フリープロトコル。フリープロトコルを選択すると、オーディオブロックの強力な設定機能を使用できるようになり、大部分の設定レジスタビットとフレーム設定レジスタを設定して、特定のオーディオプロトコル (たとえば、I2S、LSB/MSB 詰め、TDM、PCM/DSP など) を処理することができます。

01 : SPDIF プロトコル

10 : AC'97 プロトコル

11 : 予約済みです。

ビット 1:0 **MODE[1:0]** : SAIx オーディオブロックモード

これらのビットは、ソフトウェアによってセット/クリアされます。SAIx オーディオブロックを無効化した場合、これらのビットを設定する必要があります。

00 : マスタトランスミッタ

01 : マスタレシーバ

10 : スレーブトランスミッタ

11 : スレーブレシーバ

注 : オーディオブロックを SPDIF モードに設定した場合、マスタ送信モードが強制されます (MODE[1:0] = 00)。マスタ送信モードでは、オーディオブロックはただちに FS およびクロックの生成を開始します。

53.6.4 SAI 設定レジスタ 2 (SAI_ACR2)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0]		CPL	MUTECONT[5:0]					MUTEV AL	MUTE	TRIS	F FLUSH	FTH[2:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	W	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:14 **COMP[1:0]** : コンパANDINGモード。

これらのビットは、ソフトウェアによってセット/クリアされます。μ-Law と A-Law (対数) は CCITT G.711 勧告の一部であり、使用される補数のタイプは **CPL** ビットによって決まります。

データ伸張かデータ圧縮かは、MODE[0] ビットの状態で決まります。

オーディオブロックがトランスミッタとして設定された場合、データ圧縮が適用されます。

オーディオブロックがレシーバとして設定された場合、自動的にデータ伸張が適用されます。

詳細については、[セクション : コンパANDINGモード](#)を参照してください。

00 : コンパANDINGアルゴリズムなし

01 : 予約済み。

10 : μ-Law アルゴリズム

11 : A-Law アルゴリズム

注 : コンパANDINGモードは、フリープロトコルモードが選択された場合にのみ適用されます。

ビット 13 **CPL** : 補数ビット。

このビットは、ソフトウェアによってセット/クリアされます。

このビットにより、コンパANDINGモードに使用される補数のタイプを指定します。

0 : 1 の補数表現。

1 : 2 の補数表現。

注 : コンパANDINGモードが μ-Law アルゴリズムまたは A-Law アルゴリズムの場合にのみ、このビットは有効です。

ビット 12:7 **MUTECONT[5:0]** : ミュートカウンタ。

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは受信モードでのみ使用されます。

これらのビットに設定された値は、受信で検出された連続ミュートフレーム数と比較されます。ミュートフレーム数がこの値と一致すると、MUTEDET フラグがセットされ、MUTEDETIE ビットがセットされている場合、割込みが生成されます。

詳細については、[セクション : ミュートモード](#)を参照してください。

ビット 6 MUTEVAL : ミュート値。

このビットは、ソフトウェアによってセット／クリアされます。オーディオブロックを有効化 (SAIEN) する前に書き込む必要があります。オーディオブロックがトランスミッタとして動作し、スロット数が 2 以下で、かつ MUTE ビットがセットされている場合にのみ、このビットは意味を持ちます。

これより多くのスロットが指定された場合、ミュートモードでの送信時に送信されるビット値は、この MUTEVAL ビットの値とは無関係に 0 になります。

スロット数が 2 以下で MUTEVAL = 1 の場合、送信される各スロットのミュート値は、前のフレームで送信したものと同じになります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモード時にビット値 0 を送信します。

1 : ミュートモード時に最後の値を送信します。

注 : このビットは意味を持たないので、SPDIF オーディオブロックには使用できません。

ビット 5 MUTE : ミュート。

このビットは、ソフトウェアによってセット／クリアされます。このビットは、オーディオブロックがトランスミッタとして動作する場合にのみ意味を持ちます。スロット数が 2 以下の場合、ミュート値は MUTEVAL 値に関連付けられますが、スロット数が 2 より大きい場合、ミュート値は 0 になります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモードではありません。

1 : ミュートモードは有効です。

注 : このビットは意味を持たないので、SPDIF オーディオブロックには使用できません。

ビット 4 TRIS : データラインのトライステート管理。

このビットは、ソフトウェアによってセット／クリアされます。このビットは、オーディオブロックがトランスミッタとして設定された場合にのみ意味を持ちます。オーディオブロックが SPDIF モードに設定されている場合、このビットは使用されません。このビットは SAI を無効にして設定する必要があります。

詳細については、[セクション : 非アクティブスロットでの出力データライン管理](#)を参照してください。

0 : スロットが非アクティブのときでも、SD 出力ラインは SAI によって駆動されます。

1 : 次のスロットが非アクティブの場合、最終アクティブスロットの最終データビットの終わりで SD 出力ラインを開放 (Hi-Z) します。

ビット 3 FFLUSH : FIFO を一掃します。

このビットはソフトウェアでセットされます。常に 0 が読み出されます。このビットは SAI が無効なときに設定する必要があります。

0 : FIFO を一掃しません。

1 : FIFO を一掃します。このビットを 1 に設定すると、FIFO の一掃がトリガされます。すべての内部 FIFO ポインタ (読み出しと書き込み) がクリアされます。この場合、FIFO に残っているデータが失われます (送受信データはそれ以上失われません)。FIFO を一掃する前に、SAI DMA ストリーム／割込みを無効化する必要があります。

ビット 2:0 FTH[2:0] : FIFO 閾値。

このビットは、ソフトウェアによってセット／クリアされます。

000 : FIFO エンプティ

001 : FIFO の ¼

010 : FIFO の ½

011 : FIFO の ¾

100 : FIFO フル

101 : 予約済みです。

110 : 予約済みです。

111 : 予約済みです。

53.6.5 SAI 設定レジスタ 2 (SAI_BCR2)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0]		CPL	MUTECONT[5:0]					MUTEV AL	MUTE	TRIS	F FLUSH	FTH[2:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	W	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:14 **COMP[1:0]** : コンパANDINGモード。

これらのビットは、ソフトウェアによってセット/クリアされます。μ-Law と A-Law (対数) は CCITT G.711 勧告の一部であり、使用される補数のタイプは **CPL** ビットによって決まります。

データ伸張かデータ圧縮かは、MODE[0] ビットの状態で決まります。

オーディオブロックがトランスミッタとして設定された場合、データ圧縮が適用されます。

オーディオブロックがレシーバとして設定された場合、自動的にデータ伸張が適用されます。

詳細については、[セクション : コンパANDINGモード](#)を参照してください。

00 : コンパANDINGアルゴリズムなし

01 : 予約済み。

10 : μ-Law アルゴリズム

11 : A-Law アルゴリズム

注 : コンパANDINGモードは、フリープロトコルモードが選択された場合にのみ適用されます。

ビット 13 **CPL** : 補数ビット。

このビットは、ソフトウェアによってセット/クリアされます。

このビットにより、コンパANDINGモードに使用される補数のタイプを指定します。

0 : 1 の補数表現。

1 : 2 の補数表現。

注 : コンパANDINGモードが μ-Law アルゴリズムまたは A-Law アルゴリズムの場合にのみ、このビットは有効です。

ビット 12:7 **MUTECONT[5:0]** : ミュートカウンタ。

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは受信モードでのみ使用されます。

これらのビットに設定された値は、受信で検出された連続ミュートフレーム数と比較されます。ミュートフレーム数がこの値と一致すると、MUTEDET フラグがセットされ、MUTEDETIE ビットがセットされている場合、割込みが生成されます。

詳細については、[セクション : ミュートモード](#)を参照してください。

ビット 6 MUTEVAL : ミュート値。

このビットは、ソフトウェアによってセット／クリアされます。オーディオブロックを有効化 (SAIEN) する前に書き込む必要があります。オーディオブロックがトランスミッタとして動作し、スロット数が 2 以下で、かつ MUTE ビットがセットされている場合にのみ、このビットは意味を持ちます。

これより多くのスロットが指定された場合、ミュートモードでの送信時に送信されるビット値は、この MUTEVAL ビットの値とは無関係に 0 になります。

スロット数が 2 以下で MUTEVAL = 1 の場合、送信される各スロットのミュート値は、前のフレームで送信したものと同じになります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモード時にビット値 0 を送信します。

1 : ミュートモード時に最後の値を送信します。

注 : このビットは意味を持たないので、SPDIF オーディオブロックには使用できません。

ビット 5 MUTE : ミュート。

このビットは、ソフトウェアによってセット／クリアされます。このビットは、オーディオブロックがトランスミッタとして動作する場合にのみ意味を持ちます。スロット数が 2 以下の場合、ミュート値は MUTEVAL 値に関連付けられますが、スロット数が 2 より大きい場合、ミュート値は 0 になります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモードではありません。

1 : ミュートモードは有効です。

注 : このビットは意味を持たないので、SPDIF オーディオブロックには使用できません。

ビット 4 TRIS : データラインのトライステート管理。

このビットは、ソフトウェアによってセット／クリアされます。このビットは、オーディオブロックがトランスミッタとして設定された場合にのみ意味を持ちます。オーディオブロックが SPDIF モードに設定されている場合、このビットは使用されません。このビットは SAI を無効にして設定する必要があります。

詳細については、[セクション : 非アクティブスロットでの出力データライン管理](#)を参照してください。

0 : スロットが非アクティブのときでも、SD 出力ラインは SAI によって駆動されます。

1 : 次のスロットが非アクティブの場合、最終アクティブスロットの最終データビットの終わりで SD 出力ラインを開放 (Hi-Z) します。

ビット 3 FFLUSH : FIFO を一掃します。

このビットはソフトウェアでセットされます。常に 0 が読み出されます。このビットは SAI が無効なときに設定する必要があります。

0 : FIFO を一掃しません。

1 : FIFO を一掃します。このビットを 1 に設定すると、FIFO の一掃がトリガされます。すべての内部 FIFO ポインタ (読み出しと書き込み) がクリアされます。この場合、FIFO に残っているデータが失われます (送受信データはそれ以上失われません)。FIFO を一掃する前に、SAI DMA ストリーム／割込みを無効化する必要があります。

ビット 2:0 FTH[2:0] : FIFO 閾値。

このビットは、ソフトウェアによってセット／クリアされます。

000 : FIFO エンプティ

001 : FIFO の ¼

010 : FIFO の ½

011 : FIFO の ¾

100 : FIFO フル

101 : 予約済みです。

110 : 予約済みです。

111 : 予約済みです。

53.6.6 SAI フレーム設定レジスタ (SAI_AFRCCR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0007

注 : このレジスタは、AC'97および SPDIF オーディオプロトコルでは意味を持ちません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rW	rW	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FSALL[6:0]							FRL[7:0]							
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **FSOFF** : フレーム同期オフセット。

このビットは、ソフトウェアによってセット/クリアされます。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はスロット 0 の先頭ビットで開始されます。

1 : FS はスロット 0 の先頭ビットの 1 ビット前でアサートされます。

ビット 17 **FSPOL** : フレーム同期極性。

このビットは、ソフトウェアによってセット/クリアされます。このビットを使用して、FS 信号上のフレーム開始のレベルを設定します。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。

オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はアクティブラロー (立下りエッジ) になります

1 : FS はアクティブハイ (立上りエッジ) になります

ビット 16 **FSDEF** : フレーム同期の定義。

このビットは、ソフトウェアによってセット/クリアされます。

0 : FS 信号がフレーム開始信号になります

1 : FS 信号がフレーム開始信号 + チャネルサイド識別子になります

このビットがセットされている場合、SAI_xSLOTR レジスタで指定されるスロット数は偶数である必要があります。これは、このスロット数の半分を左チャンネルに、残りのスロットを右チャンネルに、それぞれ割り当てることを意味します (たとえば、このビットは、I2S プロトコルまたは MSB/LSB 詰めプロトコルなどに対して設定する必要があります)。

AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:8 **FSALL[6:0]** : フレーム同期アクティブレベル長。

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、オーディオフレームにおける FS 信号のアクティブレベルの長さを、オーディオフレームにおける FS 信号のアクティブレベルの長さを、ビットクロック数 (SCK) + 1 (FSALL[6:0] + 1) で、指定します。

これらのビットは意味を持たないので、AC'97 または SPDIF オーディオブロックの設定には使用されません。オーディオブロックを無効化して、これらのビットを設定する必要があります。

ビット 7:0 **FRL[7:0]** : フレーム長。

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、オーディオフレーム長を SCK クロックサイクル数で定義します。フレーム内のビット数は $FRL[7:0] + 1$ に等しくなります。

オーディオフレームで転送する最小ビット数は 8 でなければならず、その他の場合、オーディオブロックの動作は予測できません。これは、データサイズが 8 ビットで、SAI_xSLOTR レジスタの NBSLOT[4:0] でスロット 0 が 1 つだけ指定されている場合の例です (NBSLOT[3:0] = 0000)。

マスタモードで、マスタクロック (MCLK_x ピンで使用可能) が使用されている場合、フレーム長は、8 から 256 のうちの 2 の累乗の数に調整する必要があります。マスタクロックが使用されていない場合 (NODIV = 1)、フレーム長を 8 から 256 のうちの任意の値に設定することをお勧めします。

これらのビットは意味を持たないので、AC'97 または SPDIF オーディオブロックの設定には使用されません。オーディオブロックを無効化して、これらのビットを設定する必要があります。

53.6.7 SAI フレーム設定レジスタ (SAI_BFRCR)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0007

注 : このレジスタは、AC'97 および SPDIF オーディオプロトコルでは意味を持ちません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													r/w	r/w	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FSALL[6:0]							FRL[7:0]							
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **FSOFF** : フレーム同期オフセット。

このビットは、ソフトウェアによってセット／クリアされます。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はスロット 0 の先頭ビットで開始されます。

1 : FS はスロット 0 の先頭ビットの 1 ビット前でアサートされます。

ビット 17 **FSPOL** : フレーム同期極性。

このビットは、ソフトウェアによってセット／クリアされます。このビットを使用して、FS 信号上のフレーム開始のレベルを設定します。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。

オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はアクティブラー (立下りエッジ) になります

1 : FS はアクティブハイ (立上りエッジ) になります

ビット 16 **FSDEF** : フレーム同期の定義。

このビットは、ソフトウェアによってセット／クリアされます。

0 : FS 信号がフレーム開始信号になります

1 : FS 信号がフレーム開始信号 + チャネルサイド識別子になります

このビットがセットされている場合、SAI_xSLOTR レジスタで指定されるスロット数は偶数である必要があります。これは、このスロット数の半分を左チャンネルに、残りのスロットを右チャンネルに、それぞれ割り当てることを意味します (たとえば、このビットは、I2S プロトコルまたは MSB/LSB 詰めプロトコルなどに対して設定する必要があります)。

AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:8 **FSALL[6:0]** : フレーム同期アクティブレベル長。

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、オーディオフレームにおける FS 信号のアクティブレベルの長さを、オーディオフレームにおける FS 信号のアクティブレベルの長さを、ビットクロック数 (SCK) + 1 (FSALL[6:0] + 1) で、指定します。

これらのビットは意味を持たないので、AC'97 または SPDIF オーディオブロックの設定には使用されません。オーディオブロックを無効化して、これらのビットを設定する必要があります。

ビット 7:0 **FRL[7:0]** : フレーム長。

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、オーディオフレーム長を SCK クロックサイクル数で定義します。フレーム内のビット数は FRL[7:0] + 1 に等しくなります。

オーディオフレームで転送する最小ビット数は 8 でなければならず、その他の場合、オーディオブロックの動作は予測できません。これは、データサイズが 8 ビットで、SAI_xSLOTR レジスタの NBSLOT[4:0] でスロット 0 が 1 つだけ指定されている場合の例です (NBSLOT[3:0] = 0000)。

マスタモードで、マスタクロック (MCLK_x ピンで使用可能) が使用されている場合、フレーム長は、8 から 256 のうちの 2 の累乗の数に調整する必要があります。マスタクロックが使用されていない場合 (NODIV = 1)、フレーム長を 8 から 256 のうちの任意の値に設定することをお勧めします。

これらのビットは意味を持たないので、AC'97 または SPDIF オーディオブロックの設定には使用されません。

53.6.8 SAI スロットレジスタ (SAI_ASLOTR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

注 : このレジスタは、AC'97 および SPDIF オーディオプロトコルでは意味を持ちません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOTEN[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW

ビット 31:16 **SLOTEN[15:0]** : スロット有効化。

これらのビットは、ソフトウェアによってセット／クリアされます。

各 SLOTEN ビットは 0 から 15 までのスロット位置に対応します (最大 16 スロット)。

0 : 非アクティブスロット。

1 : アクティブスロット。

オーディオブロックを無効化して、スロットを有効にする必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **NBSLOT[3:0]** : オーディオフレーム内スロット数。

これらのビットは、ソフトウェアによってセット／クリアされます。

このビットフィールドに設定された値は、オーディオフレーム内のスロット数 + 1 を表します (非アクティブスロット数を含む)。最大スロット数は 16 です。

SAI_xFRCR レジスタの FSDEF ビットがセットされている場合、スロット数は偶数である必要があります。

オーディオブロックを無効化して、スロット数を設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 7:6 **SLOTSZ[1:0]** : スロットサイズ

このビットは、ソフトウェアによってセット／クリアされます。
スロットサイズは、データサイズ以上である必要があります。この条件が満たされない場合、SAI の動作は不確定です。

SD ラインの駆動方法については、[非アクティブスロットでの出力データライン管理](#)を参照してください。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

00 : スロットサイズをデータサイズ (SAI_xCR1 レジスタの DS[3:0] で指定) に一致させます。

01 : 16 ビット

10 : 32 ビット

11 : 予約済みです。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **FBOFF[4:0]** : 先頭ビットオフセット

これらのビットは、ソフトウェアによってセット／クリアされます。

このビットフィールドに設定された値は、スロット内の転送データの先頭ビット位置を定義します。これはオフセット値となっています。送信モードでは、データフィールド範囲外のビットは強制的に 0 に設定されます。受信モードでは、余分な受信ビットは無視されます。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

53.6.9 SAI スロットレジスタ (SAI_BSLOTR)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

注 : このレジスタは、AC'97 および SPDIF オーディオプロトコルでは意味を持ちません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOTEN[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

ビット 31:16 **SLOTEN[15:0]** : スロット有効化。

これらのビットは、ソフトウェアによってセット／クリアされます。

各 SLOTEN ビットは 0 から 15 までのスロット位置に対応します (最大 16 スロット)。

0 : 非アクティブスロット。

1 : アクティブスロット。

オーディオブロックを無効化して、スロットを有効にする必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **NBSLOT[3:0]** : オーディオフレーム内スロット数。

これらのビットは、ソフトウェアによってセット／クリアされます。

このビットフィールドに設定された値は、オーディオフレーム内のスロット数 + 1 を表します (非アクティブスロット数を含む)。最大スロット数は 16 です。

SAI_xFRCR レジスタの FSDEF ビットがセットされている場合、スロット数は偶数である必要があります。

オーディオブロックを無効化して、スロット数を設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 7:6 **SLOTSZ[1:0]** : スロットサイズ

このビットは、ソフトウェアによってセット／クリアされます。

スロットサイズは、データサイズ以上である必要があります。この条件が満たされない場合、SAI の動作は不確定です。

SD ラインの駆動方法については、[非アクティブスロットでの出力データライン管理](#)を参照してください。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

00 : スロットサイズをデータサイズ (SAI_xCR1 レジスタの DS[3:0] で指定) に一致させます。

01 : 16 ビット

10 : 32 ビット

11 : 予約済みです。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **FBOFF[4:0]** : 先頭ビットオフセット

これらのビットは、ソフトウェアによってセット／クリアされます。

このビットフィールドに設定された値は、スロット内の転送データの先頭ビット位置を定義します。これはオフセット値となっています。送信モードでは、データフィールド範囲外のビットは強制的に 0 に設定されます。受信モードでは、余分な受信ビットは無視されます。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

これらのビットは、AC'97 または SPDIF モードでは無視されます。

53.6.10 SAI 割込みマスキレジスタ (SAI_AIM)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETI E	AFSDETI E	CNRDYI E	FREQI E	WCKCFGI E	MUTEDETI E	OVRUDRI E
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **LFSDETI** : レイトフレーム同期検出割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの LFSDET ビットがセットされている場合は割込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 5 **AFSDETIE** : 予測フレーム同期検出割込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの AFSDET ビットがセットされている場合は割込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 4 **CNRDYIE** : コーデックノットレディ割込み有効化 (AC'97)

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

割込みを有効化すると、オーディオブロックは、AC'97 フレームのスロット 0 (tag0) でこのラインに接続されているコーデックが動作可能状態 (レディ) であるか否かを検出します。動作可能状態にない場合は、SAI_xSR レジスタの CNRDY フラグがセットされ、割込みが生成されます。

このビットは、PRTCFG[1:0] ビットを介して AC'97 モードが選択され、かつオーディオブロックがレシーバとして動作している場合のみ意味を持ちます。

ビット 3 **FREQIE** : FIFO リクエスト割込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの FREQ ビットがセットされている場合は割込みが生成されます。オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの寄生割込みを回避するために、MODE ビットは FREQIE をセットする前に設定する必要があります。

ビット 2 **WCKCFGIE** : 不正クロック設定割込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットは、オーディオブロックがマスタとして設定されている場合 (MODE[1] = 0 かつ NODIV = 0) にのみ考慮されます。

SAI_xSR レジスタの WCKCFG フラグがセットされている場合は割込みが生成されます。

注 : このビットはフリープロトコルモードでのみ使用され、他のモードでは意味を持ちません。

ビット 1 **MUTEDETIE** : ミュート検出割込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの MUTEDET ビットがセットされている場合は割込みが生成されます。

オーディオブロックが受信モードに設定されている場合にのみ、このビットは意味を持ちます。

ビット 0 **OVRUDRIE** : オーバーラン/アンダーラン割込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの OVRUDR ビットがセットされている場合は割込みが生成されます。

53.6.11 SAI 割込みマスクレジスタ (SAI_BIM)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETI E	AFSDETI E	CNRDYI E	FREQI E	WCKCFGI E	MUTEDETI E	OVRUDRI E
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **LFSDETI** : レイトフレーム同期検出割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの LFSDET ビットがセットされている場合は割込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 5 **AFSDETI** : 予測フレーム同期検出割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの AFSDET ビットがセットされている場合は割込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 4 **CNRDYIE** : コーデックノットレディ割込み有効化 (AC'97)

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

割込みを有効化すると、オーディオブロックは、AC'97 フレームのスロット 0 (tag0) でこのラインに接続されているコーデックが動作可能状態 (レディ) であるか否かを検出します。動作可能状態にない場合は、SAI_xSR レジスタの CNRDY フラグがセットされ、割込みが生成されます。

このビットは、PRTCFG[1:0] ビットを介して AC'97 モードが選択され、かつオーディオブロックがレシーバとして動作している場合のみ意味を持ちます。

ビット 3 **FREQIE** : FIFO リクエスト割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの FREQ ビットがセットされている場合は割込みが生成されます。オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの寄生割込みを回避するために、MODE ビットは FREQIE をセットする前に設定する必要があります。

ビット 2 **WCKCFGIE** : 不正クロック設定割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットは、オーディオブロックがマスタとして設定されている場合 (MODE[1] = 0 かつ NODIV = 0) にのみ考慮されます。

SAI_xSR レジスタの WCKCFG フラグがセットされている場合は割込みが生成されます。

注 : このビットはフリープロトコルモードでのみ使用され、他のモードでは意味を持ちません。

ビット 1 **MUTEDETIE** : ミュート検出割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの MUTEDET ビットがセットされている場合は割込みが生成されます。

オーディオブロックが受信モードに設定されている場合にのみ、このビットは意味を持ちます。

ビット 0 **OVRRUDRIE** : オーバーラン/アンダーラン割込み有効化。

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割込みは無効です。

1 : 割込みは有効です。

このビットをセットすると、SAI_xSR レジスタの OVRRUDR ビットがセットされている場合は割込みが生成されます。

53.6.12 SAI ステータスレジスタ (SAI_ASR)

アドレスオフセット : 0x018

リセット値 : 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRRUDR
									r	r	r	r	r	r	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **FLVL[2:0]** : FIFO レベル閾値。

このビットは読み出し専用です。FIFO レベル閾値フラグはハードウェアにより管理され、その設定は SAI ブロックの設定 (送信モードまたは受信モード) によって決まります。

000 : FIFO エンプティ (送信モードおよび受信モード)

001 : FIFO < ¼ ただしエンプティではない (送信モード)、FIFO < ¼ ただしエンプティではない (受信モード)

010 : ¼ < FIFO ≤ ½ (送信モード)、¼ ≤ FIFO < ½ (受信モード)

011 : ½ < FIFO ≤ ¾ (送信モード)、½ ≤ FIFO < ¾ (受信モード)

100 : ¾ < FIFO ただしフルではない (送信モード)、¾ ≤ FIFO ただしフルではない (受信モード)

101 : FIFO フル (送信モードおよび受信モード)

その他 : 予約済みです。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 LFSDET : レイトフレーム同期検出。

このビットは読み出し専用です。

0 : エラーなし。

1 : フレーム同期信号が正しいタイミングで現れない。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このビットは、AC'97 または SPDIF モードでは使用されません。

SAI_xIM レジスタの LFSDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CLFSDET ビットをセットすると、このフラグはクリアされます。

ビット 5 AFSDET : 予測フレーム同期検出。

このビットは読み出し専用です。

0 : エラーなし。

1 : 予想より前にフレーム同期信号が検出されました。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このビットは AC'97 または SPDIF モードでは使用されません。

SAI_xIM レジスタの AFSDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CAFSDET ビットをセットすると、このフラグはクリアされます。

ビット 4 CNRDY : コーデックノットレディ。

このビットは読み出し専用です。

0 : 外部 AC'97 コーデックが動作可能状態です。

1 : 外部 AC'97 コーデックが動作可能状態ではありません。

SAI_xCR1 レジスタで AC'97 オーディオプロトコルが選択され、かつ受信モードに設定された場合にのみ、このビットが使用されます。

SAI_xIM レジスタの CNRDYIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CCNRDY ビットをセットすると、このフラグはクリアされます。

ビット 3 FREQ : FIFO リクエスト。

このビットは読み出し専用です。

0 : FIFO リクエストはありません。

1 : SAI_xDR に対する読み出しまたは書き込みの FIFO リクエストがあります。

このリクエストはオーディオブロック設定によって決まります。

- ブロックが送信モードに設定されている場合、FIFO リクエストは SAI_xDR 内の書き込みリクエスト動作に関わります。
- ブロックが受信モードに設定されている場合、FIFO リクエストは SAI_xDR 内の読み出しリクエスト動作に関わります。

SAI_xIM レジスタの FREQIE ビットがセットされている場合、このフラグによって割込みを生成することができます。

ビット 2 **WCKCFG** : 不正クロック設定フラグ。

このビットは読み出し専用です。

0 : クロック設定は正常です。

1 : クロック設定が [セクション 53.4.6 : フレーム同期](#) で規定するフレーム長仕様に関する規則に準拠していません (SAI_xFRCR レジスタの FRL[7:0] ビットの設定)。

このビットは、オーディオブロックがマスタモードで動作する場合 (MODE[1] = 0 かつ NODIV = 0) にのみ使用されます。

SAI_xIM レジスタの WCKCFGIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CWCKCFG ビットをセットすると、このフラグはクリアされます。

ビット 1 **MUTEDET** : ミュート検出。

このビットは読み出し専用です。

0 : SD 入カラインでミュートは検出されません

1 : 指定数の連続オーディオフレームに対して SD 入カラインでミュート値が検出されました (0 値)

連続するオーディオフレームに対してオーディオフレームの各スロットで連続する 0 値が受信された場合、このフラグがセットされます (SAI_xCR2 レジスタの MUTECONT ビットをセット)。

SAI_xIM レジスタの MUTEDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CMUTEDET ビットをセットすると、このフラグはクリアされます。

ビット 0 **OVRUDR** : オーバーラン/アンダーラン。

このビットは読み出し専用です。

0 : オーバーラン/アンダーランエラーはありません。

1 : オーバーラン/アンダーランエラーが検出されました。

オーバーランおよびアンダーラン状態は、オーディオブロックがそれぞれレシーバとトランスミッタとして設定されている場合にのみ発生します。

SAI_xIM レジスタの OVRUDRIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの COVRUDR ビットをセットすると、このフラグはクリアされます。

53.6.13 SAI ステータスレジスタ (SAI_BSR)

アドレスオフセット : 0x038

リセット値 : 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **FLVL[2:0]** : FIFO レベル閾値。

このビットは読出し専用です。FIFO レベル閾値フラグはハードウェアにより管理され、その設定は SAI ブロックの設定 (送信モードまたは受信モード) によって決まります。

000 : FIFO エンプティ (送信モードおよび受信モード)

001 : $FIFO < \frac{1}{4}$ ただしエンプティではない (送信モード)、 $FIFO < \frac{1}{4}$ ただしエンプティではない (受信モード)

010 : $\frac{1}{4} < FIFO \leq \frac{1}{2}$ (送信モード)、 $\frac{1}{4} \leq FIFO < \frac{1}{2}$ (受信モード)

011 : $\frac{1}{2} < FIFO \leq \frac{3}{4}$ (送信モード)、 $\frac{1}{2} \leq FIFO < \frac{3}{4}$ (受信モード)

100 : $\frac{3}{4} < FIFO$ ただしフルではない (送信モード)、 $\frac{3}{4} \leq FIFO$ ただしフルではない (受信モード)

101 : FIFO フル (送信モードおよび受信モード)

その他 : 予約済みです。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **LFSDET** : レイトフレーム同期検出。

このビットは読出し専用です。

0 : エラーなし。

1 : フレーム同期信号が正しいタイミングで現れない。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このビットは、AC'97 または SPDIF モードでは使用されません。

SAI_xIM レジスタの LFSDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CLFSDET ビットをセットすると、このフラグはクリアされます。

ビット 5 **AFSDET** : 予測フレーム同期検出。

このビットは読出し専用です。

0 : エラーなし。

1 : 予想より前にフレーム同期信号が検出されました。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このビットは AC'97 または SPDIF モードでは使用されません。

SAI_xIM レジスタの AFSDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CAFSDET ビットをセットすると、このフラグはクリアされます。

ビット 4 **CNRDY** : コーデックノットレディ。

このビットは読出し専用です。

0 : 外部 AC'97 コーデックが動作可能状態です。

1 : 外部 AC'97 コーデックが動作可能状態ではありません。

SAI_xCR1 レジスタで AC'97 オーディオプロトコルが選択され、かつ受信モードに設定された場合にのみ、このビットが使用されます。

SAI_xIM レジスタの CNRDYIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CCNRDY ビットをセットすると、このフラグはクリアされます。

ビット 3 **FREQ** : FIFO リクエスト。

このビットは読出し専用です。

0 : FIFO リクエストはありません。

1 : SAI_xDR に対する読出しまたは書込みの FIFO リクエストがあります。

このリクエストはオーディオブロック設定によって決まります。

– ブロックが送信モードに設定されている場合、FIFO リクエストは SAI_xDR 内の書込みリクエスト動作に関わります。

– ブロックが受信モードに設定されている場合、FIFO リクエストは SAI_xDR 内の読出しリクエスト動作に関わります。

SAI_xIM レジスタの FREQIE ビットがセットされている場合、このフラグによって割込みを生成することができます。

ビット 2 **WCKCFG** : 不正クロック設定フラグ。

このビットは読み出し専用です。

0 : クロック設定は正常です。

1 : クロック設定が [セクション 53.4.6 : フレーム同期](#) で規定するフレーム長仕様に関する規則に準拠していません (SAI_xFRCR レジスタの FRL[7:0] ビットの設定)。

このビットは、オーディオブロックがマスタモードで動作する場合 (MODE[1] = 0 かつ NODIV = 0) にのみ使用されます。

SAI_xIM レジスタの WCKCFGIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CWCKCFG ビットをセットすると、このフラグはクリアされます。

ビット 1 **MUTEDET** : ミュート検出。

このビットは読み出し専用です。

0 : SD 入カラインでミュートは検出されません

1 : 指定数の連続オーディオフレームに対して SD 入カラインでミュート値が検出されました (0 値)

連続するオーディオフレームに対してオーディオフレームの各スロットで連続する 0 値が受信された場合、このフラグがセットされます (SAI_xCR2 レジスタの MUTECONT ビットをセット)。

SAI_xIM レジスタの MUTEDETIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの CMUTEDET ビットをセットすると、このフラグはクリアされます。

ビット 0 **OVRUDR** : オーバーラン/アンダーラン。

このビットは読み出し専用です。

0 : オーバーラン/アンダーランエラーはありません。

1 : オーバーラン/アンダーランエラーが検出されました。

オーバーランおよびアンダーラン状態は、オーディオブロックがそれぞれレシーバとトランスミッタとして設定されている場合にのみ発生します。

SAI_xIM レジスタの OVRUDRIE ビットがセットされている場合は割込みが生成されます。

ソフトウェアによって SAI_xCLRFR レジスタの COVRUDR ビットをセットすると、このフラグはクリアされます。

53.6.14 SAI クリアフラグレジスタ (SAI_ACLRFR)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTED ET	COVRUDR
									w	w	w		w	w	w

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CLFSDET** : レイトフレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの LFSDET フラグがクリアされます。

このビットは、AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 5 **CAFSDET** : 予測フレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの AFSDET フラグがクリアされます。

このビットは AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 4 **CCNRDY** : コーデックノットレディフラグのクリア

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの CNRDY フラグがクリアされます。

SAI_xCR1 レジスタで AC'97 オーディオプロトコルが選択された場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CWCKCFG** : 不正クロック設定フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの WCKCFG フラグがクリアされます。

オーディオブロックがマスタとして設定され (MODE[1] = 0)、かつ SAI_xCR1 レジスタで NODIV = 0 の場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 1 **CMUTEDET** : ミュート検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの MUTEDET フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

ビット 0 **COVRUDR** : オーバーラン/アンダーランクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの OVRUDR フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

53.6.15 SAI クリアフラグレジスタ (SAI_BCLRFR)

アドレスオフセット : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTED ET	COVRUDR
									w	w	w		w	w	w

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CLFSDET** : レイトフレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの LFSDET フラグがクリアされます。

このビットは、AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 5 **CAFSDET** : 予測フレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの AFSDET フラグがクリアされます。

このビットは AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 4 **CNDRDY** : コーデックノットレディフラグのクリア

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの CNRDY フラグがクリアされます。

SAI_xCR1 レジスタで AC'97 オーディオプロトコルが選択された場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CWCKCFG** : 不正クロック設定フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの WCKCFG フラグがクリアされます。

オーディオブロックがマスタとして設定され (MODE[1] = 0)、かつ SAI_xCR1 レジスタで NODIV = 0 の場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 1 **CMUTEDET** : ミュート検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの MUTEDET フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

ビット 0 **COVRUDR** : オーバーラン/アンダーランクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI_xSR レジスタの OVRUDR フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

53.6.16 SAI データレジスタ (SAI_ADR)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **DATA[31:0]** : データ

FIFO がフルでない場合、このレジスタへ書き込むと、FIFO への書き込みが行われます。

FIFO がエンプティでない場合、このレジスタを読み出すと、FIFO からの読出しが行われます。

53.6.17 SAI データレジスタ (SAI_BDR)

アドレスオフセット : 0x040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DATA[31:0]** : データ

FIFO がフルでない場合、このレジスタへ書き込むと、FIFO への書込みが行われます。

FIFO がエンプティでない場合、このレジスタを読み出すと、FIFO からの読出しが行われます。

53.6.18 SAI PDM 制御レジスタ (SAI_PDMCR)

アドレスオフセット : 0x0044

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
						rW	rW			rW	rW				rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CKEN2** : ビットストリームクロック 2 番のクロックイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : SAI_CK2 クロック無効

1 : SAI_CK2 クロック有効

注 : このビットを **PDMEN = 1** のときに設定するのは推奨しません。

SAI_CK2 は、すべての SAI インスタンスで使用できるわけではありません。詳細については、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 8 **CKEN1** : ビットストリームクロック 1 番のクロックイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : SAI_CK1 クロック無効

1 : SAI_CK1 クロック有効

注 : このビットを **PDMEN = 1** のときに設定するのは推奨しません。

SAI_CK1 は、すべての SAI インスタンスで使用できるわけではありません。詳細については、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **MICNBR[1:0]** : マイクロフォンの本数

このビットは、ソフトウェアによってセット／クリアされます。

00 : マイクロフォン 2 本の設定

01 : マイクロフォン 4 本の設定

10 : マイクロフォン 6 本の設定

11 : マイクロフォン 8 本の設定

注 : このフィールドを **PDMEN = 1** のときに設定するのは推奨しません。*

データライン一式は、すべての SAI インスタンスで使用できるわけではありません。詳細については、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **PDMEN** : PDM 有効化

このビットは、ソフトウェアによってセット／クリアされます。このビットを使って、PDM インタフェースブロックの状態を制御できます。

PDM インタフェースを有効にする前に SAI がすでに TDM マスタモードで動作していることを確認してください。

0 : PDM インタフェースが無効です。

1 : PDM インタフェースが有効です。

53.6.19 SAI PDM ディレイレジスタ (SAI_PDMDLY)

アドレスオフセット : 0x0048

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYM4R [2:0]			Res.	DLYM4L [2:0]			Res.	DLYM3R [2:0]			Res.	DLYM3L [2:0]		
	r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYM2R [2:0]			Res.	DLYM2L [2:0]			Res.	DLYM1R [2:0]			Res.	DLYM1L [2:0]		
	r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w	r/w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **DLYM4R[2:0]** : ペア No.4 の 2 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : 1 T_{SAI_CK} 周期のディレイ

010 : 2 T_{SAI_CK} 周期のディレイ

.....

111 : 7 T_{SAI_CK} 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D4 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **DLYM4L[2:0]** : ペア No.4 の 1 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D4 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **DLYM3R[2:0]** : ペア No.3 の 2 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D3 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **DLYM3L[2:0]** : ペア No.3 の 1 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D3 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **DLYM2R[2:0]** : ペア No.2 の 2 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D2 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **DLYM2L[2:0]** : ペア No.2 の 1 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D2 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **DLYM1R[2:0]** : ペア No.1 の 2 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D1 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **DLYM1L[2:0]** : ペア No.1 の 1 番目のマイクロフォンのディレイライン

このビットは、ソフトウェアによってセット／クリアされます。

000 : 遅延なし

001 : $1 T_{SAI_CK}$ 周期のディレイ

010 : $2 T_{SAI_CK}$ 周期のディレイ

.....

111 : $7 T_{SAI_CK}$ 周期のディレイ

このフィールドは動作中に変更することができます。

注 : このフィールドは、D1 ラインが使用可能な場合にのみ使用できます。使用可能かどうか確認するには、[セクション 53.3 : SAI の実装](#)を参照してください。

53.6.20 SAI レジスタマップ

表 575. SAI レジスタのマップとリセット値

オフセット	レジスタ名 リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x0000	SAI_GCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
	リセット値																											0	0			0	0		
0x0004 または 0x0024	SAI_xCR1	Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]							NODIV	Res.	DMAEN	SAIEN	Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]		Res.	PRTCFG[1:0]		MODE[1:0]			
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
0x0008 または 0x0028	SAI_xCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP[1:0]		CPL	MUTEEN[5:0]					MUTE VAL	MUTE	TRIS	FFLUS	FTH[2:0]					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x000C または 0x002C	SAI_xFRCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF	Res.	FSALL[6:0]					FRL[7:0]											
	リセット値														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
0x0010 または 0x0030	SAI_xSLOTR	SLOTEN[15:0]															Res.	Res.	Res.	Res.	NBSLOT[3:0]			SLOTSZ[1:0]		Res.	FBOFF[4:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0		
0x0014 または 0x0034	SAI_xIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE		
	リセット値																	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0		
0x0018 または 0x0038	SAI_xSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR		
	リセット値															0	0	0									0	0	0	0	1	0	0		
0x001C または 0x003C	SAI_xCLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CNRDY	Res.	WCKCFG	CMUTEDET	COVRUDR		
	リセット値																	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0		0	0	0		
0x0020 または 0x0040	SAI_xDR	DATA[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0044	SAI_PDMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	Res.	PDMEN	
	リセット値																	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0							

表 575. SAI レジスタのマップとリセット値 (続き)

オフセット	レジスタ名 リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0048	SAI_PDMDLY	Res.	DLYM4R [2:0]			Res.	DLYM4L [2:0]			Res.	DLYM3R [2:0]			Res.	DLYM3L [2:0]			Res.	DLYM2R [2:0]			Res.	DLYM2L [2:0]			Res.	DLYM1R [2:0]			Res.	DLYM1L [2:0]		
	リセット値		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0

レジスタ境界アドレスについては、[セクション 2.3](#) を参照してください。

54 FD コントローラエリアネットワーク (FDCAN)

54.1 概要

コントローラエリアネットワーク (CAN) のサブシステム (図 771 を参照) は、1 つの CAN モジュール、共有メッセージ RAM、および設定ブロックで構成されます。これらの部分の各ベースアドレスについてはメモリマップを参照してください。

モジュール (FDCAN) は、ISO 11898-1: 2015 (CAN プロトコル仕様バージョン 2.0 パート A、B) および CAN FD プロトコル仕様バージョン 1.0 に準拠しています。

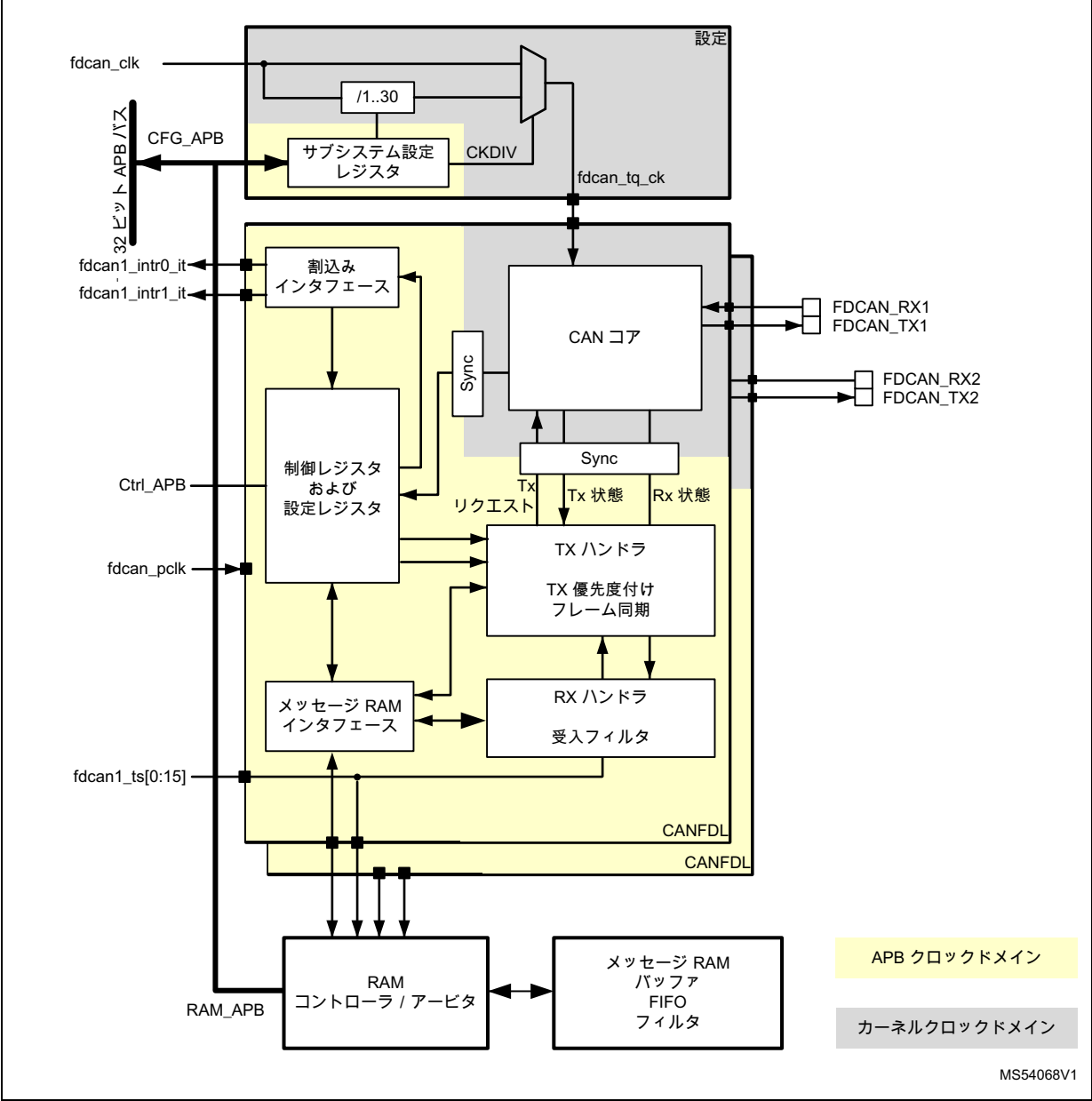
FDCAN インスタンスあたり 0.8 KB のメッセージ RAM には、フィルタ、受信 FIFO、送信イベント FIFO、および送信 FIFO が実装されます。

CAN サブシステム I/O 信号およびピンの詳細は、それぞれ表 576 および図 771 に示します。

表 576. CAN サブシステム I/O 信号

名前	タイプ	説明
fdcan_ck	デジタル入力	CAN サブシステムカーネルクロック入力
fdcan_pclk		CAN サブシステム APB インタフェースクロック入力
fdcan_intr0_it	デジタル出力	FDCAN interrupt0
fdcan_intr1_it		FDCAN interrupt1
fdcan_ts[0:15]	-	外部タイムスタンプベクトル
FDCAN_RX	デジタル入力	FDCAN 受信ピン
FDCAN_TX	デジタル出力	FDCAN 送信ピン
APB インタフェース	デジタル入出力	設定、制御、および RAM アクセス用の複数の psel を持つシングル APP

図 771. CAN サブシステム

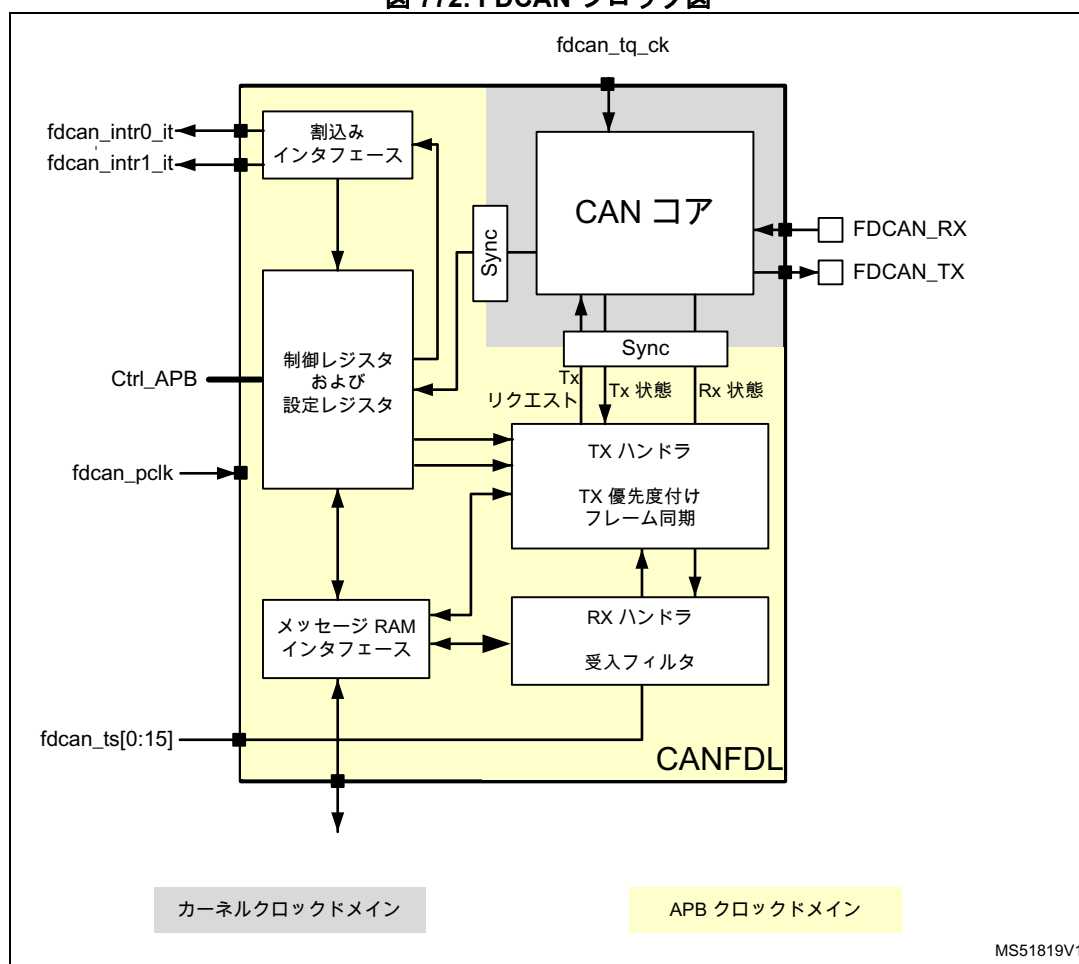


54.2 FDCAN の主な機能

- CAN プロトコルバージョン 2.0 パート A、B および ISO 11898-1: 2015, -4
- 最大 64 データバイトをサポートする CAN FD
- CAN エラーログ記録
- AUTOSAR および J1939 サポート
- 改良された受入フィルタ
- それぞれ 3 つのペイロードを持つ 2 つの受信 FIFO (ペイロード当たり最大 64 バイト)
- 高優先メッセージの受信で個別の信号
- 3 つのペイロードの設定可能な送信 FIFO (ペイロード当たり最大 64 バイト)
- 送信イベント FIFO
- プログラム可能なループバックテストモード
- マスク可能なモジュール割込み
- 2 つのクロックドメイン : APB バスインタフェースおよび CAN コアカーネルクロック
- パワーダウンサポート

54.3 FDCAN の機能詳細

図 772. FDCAN ブロック図



2つの割込みライン

FDCAN ペリフェラルには、`fdcan_intr0_it` および `fdcan_intr1_it` の2つの割込みラインがあります。

FDCAN_ILE レジスタで EINT0 および EINT1 ビットをプログラムすることで、割込みラインを個別に有効化または無効化できます。

CAN コア

CAN コアには、プロトコルコントローラおよび受信／送信シフトレジスタがあります。コアはすべての ISO 11898-1 を扱います: 2015 プロトコル機能进行处理し、11 ビットおよび 29 ビットの識別子両方をサポートします。

Sync

同期ブロックは、APB クロックドメインと CAN カネルクロックドメインとの信号を同期します。

Tx ハンドラ

メッセージ RAM から CAN コアへのメッセージ転送を制御します。送信に対して、最大 3 個の Tx バッファを使用できます。Tx バッファは、Tx FIFO または Tx キューとして使用できます。Tx イベント FIFO には、対応するメッセージ ID とともに Tx タイムスタンプを格納できます。送信キャンセルもサポートしています。

Rx ハンドラ

CAN コアから外部メッセージ RAM への受信したメッセージの転送を制御します。Rx ハンドラは、受入フィルタを通過したすべてのメッセージの記録用に、2 つの受信 FIFO をサポートします。Rx タイムスタンプが、各メッセージとともに格納されます。11 ビットの ID には最大 28 個のフィルタ、29 ビットの ID には最大 8 個のフィルタを定義できます。

APB インタフェース

設定レジスタ、コントローラの設定、および RAM アクセス用の APB バスに FDCAN を接続します。

メッセージ RAM インタフェース

FDCAN には RAM コントローラ/アービタ経由で外部の 1 KB メッセージ RAM が接続されています。

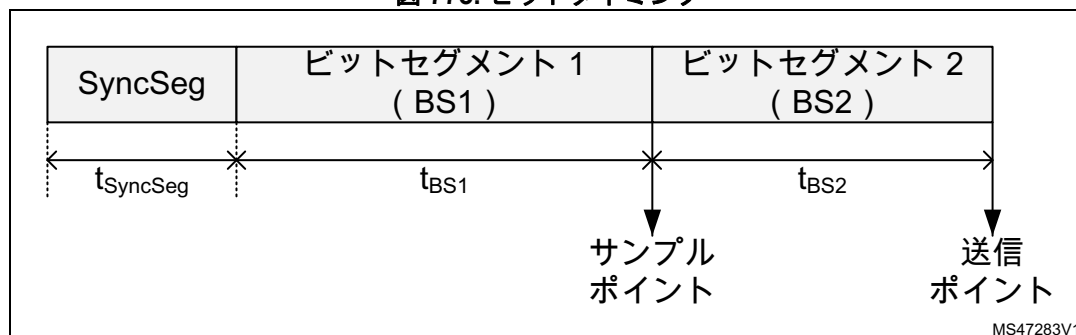
54.3.1 ビットタイミング

ビットタイミング回路は、シリアルバスラインを監視し、スタートビットエッジでの同期と後続エッジでの再同期によって、サンプリングとサンプリングポイントの調整を行います。

図 773 に示すように、この動作は、ビット時間を次のように 3 つのセグメントに分割するとわかりやすく説明できます。

- 同期セグメント (SYNC_SEG) : この 1 タイムクオンタム ($1 \times tq$) に固定された長さの時間セグメント内にビット変化が生じることが想定されます。
- ビットセグメント 1 (BS1) : サンプルポイントの位置を定義します。これには、CAN 規格の PROP_SEG と PHASE_SEG1 が含まれます。セグメントの時間は 1~16 タイムクオンタムの間でプログラミング可能ですが、ネットワークのさまざまなノード間の周波数の差に起因する正の位相ドリフトを補正するために、自動的に延長されることがあります。
- ビットセグメント 2 (BS2) : 送信ポイントの位置を定義します。これは、CAN 規格の PHASE_SEG2 に相当します。セグメントの時間は 1~8 タイムクオンタムの間でプログラミング可能ですが、負の位相ドリフトを補正するために、自動的に短縮されることもあります。

図 773. ビットタイミング



ボーレートはビット時間の逆数（ボーレート = $1 / \text{ビット時間}$ ）で、3つのコンポーネントの和です。
 図 773 は、ビット時間 = $t_{\text{SyncSeg}} + t_{\text{BS1}} + t_{\text{BS2}}$ を示します。ここで、

- 公称ビット時間の場合
 - $t_q = (\text{FDCAN_NBTP.NBRP}[8:0] + 1) * t_{\text{fdcan_tq_clk}}$
 - $t_{\text{SyncSeg}} = 1 t_q$
 - $t_{\text{BS1}} = t_q * (\text{FDCAN_NBTP.NTSEG1}[7:0] + 1)$
 - $t_{\text{BS2}} = t_q * (\text{FDCAN_NBTP.NTSEG2}[6:0] + 1)$
- データビット時間の場合
 - $t_q = (\text{FDCAN_DBTP.DBRP}[4:0] + 1) * t_{\text{fdcan_tq_clk}}$
 - $t_{\text{SyncSeg}} = 1 t_q$
 - $t_{\text{BS1}} = t_q * (\text{FDCAN_DBTP.DTSEG1}[4:0] + 1)$
 - $t_{\text{BS2}} = t_q * (\text{FDCAN_DBTP.DTSEG2}[3:0] + 1)$

（再）同期ジャンプ幅（SJW）は、ビットセグメントの延長／短縮量の上限を定め、1～4 タイムクォンタムの中でプログラミング可能です。

有効なエッジは、コントローラ自体はリセシブビットを送信しないという条件で、あるビット時間において、ドミナントからリセシブへのバスレベルの最初の遷移と定義されます。

有効なエッジが SYNC_SEG ではなく BS1 で検出された場合、サンプルポイントが遅くなるように、BS1 は最大で SJW 延長されます。

逆に、有効なエッジが SYNC_SEG ではなく BS2 で検出された場合、送信ポイントが早くなるように、BS2 は最大で SJW 短縮されます。

プログラミングエラーに対する安全対策として、ビットタイミングレジスタは、デバイスが STANDBY モードの場合にのみ設定可能です。レジスタ FDCAN_DBTP および FDCAN_NBTP（それぞれデータビット時間および公称ビット時間専用）は、CCCR.CCE と CCCR.INIT がセットされている場合にのみアクセス可能です。

注： CAN ビットタイミングと再同期メカニズムの詳細については、ISO 11898-1 規格を参照してください。

54.3.2 動作モード

設定

IP バージョン、ハードウェア、および入力クロック分周回路を設定します。クロック分周回路を 0 にセットすると、主な入力クロックはそのまま使用されます。

ソフトウェアの初期化

ソフトウェアの初期化は、FDCAN_CCCR レジスタで INIT ビットをセットするか、ソフトウェアやハードウェアリセット、Bus_Off への移行で開始されます。FDCAN_CCCR レジスタで INIT ビットをセットしている間、CAN バスとのメッセージ転送は停止し、CAN バス出力 FDCAN_TX のステータスはリセシブ（ハイ）になります。EML（エラー管理ロジック）のカウンタは変更されません。FDCAN_CCCR の INIT ビットのセットで、設定レジスタは変更されません。FDCAN_CCCR の INIT ビットのクリアで、ソフトウェア初期化は終了します。その後、ビットストリームプロセッサ（BSP）は 11 個の連続したリセシブビットの発生を待つことで（Bus_Idle）、CAN バスでのデータ転送と同期してから、バスアクティビティに移行してメッセージ転送を開始できます。

FDCAN 設定レジスタへのアクセスは、FDCAN_CCCR レジスタの INIT ビットと FDCAN_CCCR レジスタの CCE ビットの両方がセットされているときのみ有効です。

FDCAN_CCCR レジスタの CCE ビットは、FDCAN_CCCR の INIT ビットがセットされている間のみセット／クリアできます。FDCAN_CCCR レジスタの CCE ビットは、FDCAN_CCCR の INIT ビットがクリアされているとき、自動的にクリアされます。

FDCAN_CCCR レジスタの CCE ビットがセットされている場合、次のレジスタはリセットされます。

- FDCAN_HPMS - 高優先メッセージステータス
- FDCAN_RXF0S - Rx FIFO 0 ステータス
- FDCAN_RXF1S - Rx FIFO 1 ステータス
- FDCAN_TXFQS - Tx FIFO／キューステータス
- FDCAN_TXBRP - Tx バッファリクエストペンディング
- FDCAN_TXBTO - Tx バッファ送信発生
- FDCAN_TXBCF - Tx バッファキャンセル終了
- FDCAN_TXEFS - Tx イベント FIFO ステータス

FDCAN_TOCV レジスタのタイムアウトカウンタ値 TOC ビットは、FDCAN_CCCR の CCE ビットがセットされているとき、FDCAN_TOCC レジスタの TOP ビットによって設定された値にプリセットされます。

さらに、Tx ハンドラおよび Rx ハンドラのステートマシンは、FDCAN_CCCR の CCE ビットがセットされている間はアイドル状態で保持されます。

FDCAN_CCCR レジスタの CCE ビットがクリアされている場合のみ、次のレジスタは書き込み可能です。

- TXBAR - Tx バッファ追加リクエスト
- TXBCR - Tx バッファキャンセルリクエスト

FDCAN_CCCR の TEST ビットおよび FDCAN_CCCR の MON ビットは、CCCR の INIT ビットと CCCR レジスタの CCE ビットの両方がセットされているときのみ、ソフトウェアによってセットできます。両ビットは常時リセットできます。FDCAN_CCCR の DAR ビットは、FDCAN_CCCR の INIT ビットと FDCAN_CCCR レジスタの CCE ビットの両方がセットされているときのみセット／クリアできます。

通常動作

ハードウェアリセット後の FDCAN のデフォルト動作モードは、イベント駆動 CAN 通信です。TT 動作モードはサポートされていません。

FDCAN が初期化され、FDCAN_CCCR レジスタの INIT ビットがクリアされると、FDCAN は CAN バスと同期し、通信できるようになります。

受入フィルタを通過した後、メッセージ ID と DLC を含む受信メッセージは、Rx FIFO 0 や Rx FIFO 1 に格納されます。

送信されるメッセージの場合、Tx FIFO や Tx キューを初期化または更新できます。リモートフレーム受信時の自動送信はサポートされません。

CAN FD 動作

FDCAN プロトコルは 2 種類あります。

1. CAN フレームのデータフィールドが 8 バイトより長くなるロングフレームモード (LFM)
2. CAN フレームの制御フィールド、データフィールド、および CRC フィールドがフレームの開始と終了時より高いビットレートで送信される高速フレームモード (FFM)

高速フレームモードは、ロングフレームモードと組み合わせて使用できます。

11 ビットの識別子の CAN フレームで以前に予約済みのビットと、29 ビットの識別子の CAN フレームで一番最初に予約済みのビットは、FDF ビットとしてデコードされます。FDF リセッспは、CAN FD フレームを示し、FDF ドミナントはクラシック CAN フレームを示します。

CAN FD フレームでは、FDF に続く 2 つのビットの res および BRS によって、この CAN FD フレーム内のビットレートを切り替えるかどうかが決まります。CAN FD ビットレートの切り替えは、res ドミナントと BRS リセッспによって示されます。res リセッспのコーディングは、プロトコルを今後拡張するために予約済みとなります。FDCAN が FDF リセッспおよび res リセッспでフレームを受信する場合、ビット PSR.PXE をセットすることでプロトコル例外イベントの信号を送信します。プロトコル例外処理が有効である場合 (CCCR.PXHD = 0)、動作状態が次のサンプルポイントで Receiver (PSR.ACT = 10) から Integrating (PSR.ACT = 00) に変わります。プロトコル例外処理が無効である場合 (CCCR.PXHD = 1)、FDCAN はリセッспの res ビットをフォームエラーとして扱い、エラーフレームで応答します。

CAN FD 動作は、CCCR.FDOE をプログラムすることで有効になります。CCCR.FDOE = 1 の場合、CAN FD フレームの送受信は有効です。クラシック CAN フレームの送受信はいつでも可能です。CAN FD フレームまたはクラシック CAN フレームが送信されるかどうかは、それぞれの Tx バッファ要素のビット FDF 経由で設定できます。CCCR.FDOE = 0 では、受信フレームはクラシック CAN フレームとして解釈され、CAN FD フレーム受信時にエラーフレームを送信します。CAN FD 動作が無効である場合、Tx バッファ要素のビット FDF がセットされている場合でも CAN FD フレームは送信されません。CCCR.FDOE および CCCR.BRSE は、CCCR.INIT および CCCR.CCE が両方セットされている間のみ変更できます。

CCCR.FDOE = 0 では、ビット FDF および BRS の設定は無視され、フレームはクラシック CAN フォーマットで送信されます。CCCR.FDOE = 1 かつ CCCR.BRSE = 0 では、Tx バッファ要素のビット FDF だけが評価されます。CCCR.FDOE = 1 かつ CCCR.BRSE = 1 では、ビットレート切り替えでの CAN FD フレームの送信は有効です。ビット FDF および BRS がセットされたすべての Tx バッファ要素は、ビットレート切り替えして CAN FD フォーマットで送信されます。

CAN 動作中のモード変更は、次の条件下でのみ推奨されます。

- CAN FD データフェーズでの失敗率は、CAN FD アービトレーションフェーズより大幅に高くなります。この場合、送信用の CAN FD ビットレート切り替えオプションを無効にします。
- システム起動中、すべてのノードが CAN FD フォーマットで通信できることを検証するまでクラシック CAN メッセージを送信します。これが当てはまる場合、すべてのノードは CAN FD 動作に切り替わります。
- CAN パーシャルネットワークのウェイクアップメッセージは、クラシック CAN フォーマットで送信する必要があります。
- すべてのノードが CAN FD に対応していない場合のプログラミングの終了。非 CAN FD ノードは、プログラミングが完了するまでサイレントモードで保持されます。その後、すべてのノードがクラシック CAN 通信に切り替わります。

FDCAN フォーマットでは、DLC のコーディングは標準の CAN フォーマットの 1 つとは異なります。DLC コード 0~8 は標準 CAN と同じコーディングで、コード 9~15 (標準 CAN ではすべて 8 バイトのデータフィールドのコーディング) は表 577 に従ってコーディングされます。

表 577. FDCAN での DLC コーディング

DLC	9	10	11	12	13	14	15
データバイト数	12	16	20	24	32	48	64

CAN FD 高速フレームでは、BRS (ビットレート切り替え) ビットがリセッспである場合、このビットの後のフレーム内でビットタイミングが切り替わります。FDCAN アービトレーションフェーズで BRS ビットまでは、ビットタイミングおよびブリスケーラレジスタ BTP によって定義されたと

おりに、標準の CAN ビットタイミングが使用されます。次の FDCAN データフェーズでは、高速ビットタイミングおよびプリスケアラレジスタ FBTP によって定義されたとおりに、高速 CAN ビットタイミングが使用されます。CRC デリミタまたはエラー検出時にどちらが先に発生しても、ビットタイミングが高速タイミングから再度切り替わります。

CAN FD データフェーズで設定可能な最大ビットレートは、FDCAN カーネルクロック周波数に依存します。たとえば、20 MHz の FDCAN カーネルクロック周波数で 4 タイムクオンタム (tq) の設定可能な最小ビット時間とした場合、データフェーズのビットレートは 5 Mbit/s となります。

両方のデータフレームフォーマット (CAN FD ロングフレームおよび CAN FD 高速フレーム) で、ビット ESI (エラーステータスインジケータ) の値は送信開始時のトランスミッタのエラー状態によって決まります。トランスミッタがエラーパッシブである場合、ESI はリセツプで送信され、その他の場合はドミナントで送信されます。CAN FD リモートフレームでは、ESI ビットはトランスミッタのエラー状態にかかわらず、常にドミナントで送信されます。CAN FD リモートフレームのデータ長コードは 0 で送信されます。

FDCAN Tx バッファが DLC > 8 での FDCAN 送信に設定されている場合、最初の 8 バイトは Tx バッファで設定されたとおり送信され、データフィールドの残りの部分は 0xCC でパディングされます。FDCAN が DLC > 8 で FDCAN フレームを受信する場合、フレームの最初の 8 バイトは適合する Rx FIFO に格納されます。残りのバイトは破棄されます。

トランシーバ遅延補正

FDCAN 送信のデータフェーズの間、1 つのノードだけが送信して、その他すべてはレシーバになります。バスラインの長さに影響はありません。ピン FDCAN_TX 経由で送信する場合、プロトコルコントローラはピン FDCAN_RX 経由でローカル CAN トランシーバから送信されたデータを受信します。受信されるデータは、CAN トランシーバループ遅延により遅れます。この遅延が TSEG1 (サンプリングポイント前の時間セグメント) より大きくなる場合、ビットエラーが検出されます。トランシーバ遅延補正がない場合、FDCAN フレームのデータフェーズのビットレートはトランシーバループ遅延によって制限されます。

FDCAN は遅延補正メカニズムを実装して、CAN トランシーバループ遅延を補正することで、特定の CAN トランシーバの遅延にかかわらず、FDCAN データフェーズ中により高いビットレートでの送信を可能にします。

ノード送信のデータフェーズ中にビットエラーをチェックするために、遅延した送信データはセカンダリサンプリングポイント (SSP) で受信したデータと比較されます。ビットエラーが検出される場合、トランスミッタは次のレギュラサンプリングポイントでこのビットエラーに反応します。アービトレーションフェーズ中、遅延補正は常に無効です。

トランスミッタ遅延補正は、データビット時間がトランスミッタ遅延より短い場合に設定が可能になります。詳細は、新しい ISO11898-1 に記載されています。ビット DBTP.TDC をセットすることで有効になります。

受信したビットは、SSP で送信したビットと比較されます。SSP の位置は、トランシーバ経由で FDCAN の送信出力ピン FDCAN_TX から受信入力ピン FDCAN_RX までに測定された遅延と、TDCR.TDCO によって設定されたトランスミッタ遅延補正オフセットの合計として定義されます。トランスミッタ遅延補正オフセットは、受信したビット内の SSP の位置を調整するために使用します (例: データフェーズのビット時間の半分)。セカンダリサンプリングポイントの位置は、次の mtq (fdcan_tq_ck クロックの 1 周期である最小タイムクオンタム) の整数に切り捨てられます。

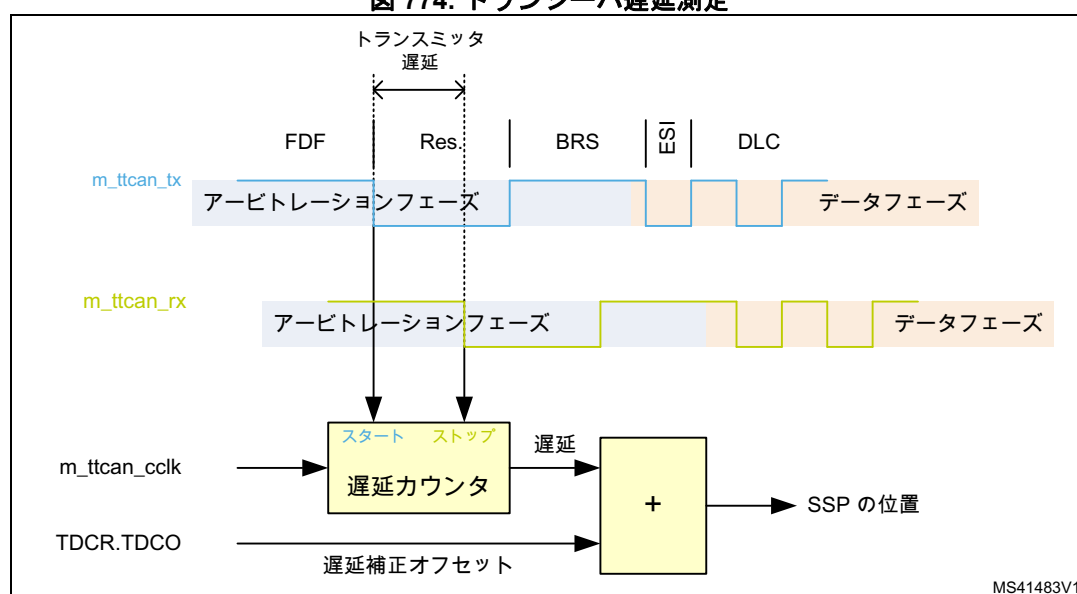
PSR.TDCV は、実際のトランスミッタ遅延補正値を示します。PSR.TDCV は、CCCR.INIT がセットされているとクリアされ、DBTP.TDC がセットされている間 FD フレームの各送信で更新されます。

FDCAN に実装されたトランスミッタ遅延補正では、次の境界条件を考慮する必要があります。

- FDCAN_Tx から FDCAN_Rx までの測定された遅延と、設定されたトランスミッタ遅延補正オフセット TDCR.TDCO の合計は、データフェーズで 6 ビット時間未満でなければなりません。
- FDCAN_TX から FDCAN_RX までの測定された遅延と、設定されたトランスミッタ遅延補正オフセット TDCR.TDCO の合計は、127 mtq 以下でなければなりません。合計がこの値を超える場合、最大値 (127 mtq) はトランスミッタ遅延補正に使用されます。
- データフェーズは、CRC デリミタのサンプルポイントで終了し、SSP で受信したビットのチェックを停止します。

トランスミッタ遅延補正が、DBTP.TDC = 1 をプログラムすることで有効になった場合、ビット FDF からビット res の立下りエッジで送信した各 CAN FD フレーム内で測定が開始されます。このエッジがトランスミッタの受信入力ピン FDCAN_TX で見られると測定は停止します。この測定の解析時間は、1 mtq です。

図 774. トランシーバ遅延測定



MS41483V1

受信した FDF ビット内のドミナントグリッチによって、受信した res ビットの立下りエッジ前に遅延補正測定を終了する（早い SSP の位置となる）ことを避けるために、TDCR.TDCF をプログラムすることでトランスミッタ遅延補正フィルタウィンドウの使用が有効になります。これで SSP の位置の最小値を定義します。より早い SSP の位置につながる FDCAN_RX のドミナントエッジは、トランスミッタ遅延測定では無視されます。SSP の位置が少なくとも TDCR.TDCF で、FDCAN_RX がローになると、測定は停止します。

制限動作モード

制限動作モードでは、ノードはデータとリモートフレームを受信でき、有効なフレームに確認応答を付与できますが、データフレーム、リモートフレーム、アクティブエラーフレーム、またはオーバーロードフレームは送信しません。エラー状態やオーバーロード状態になった場合、ドミナントビットを送らずに、バスアイドル状態の発生を待って CAN 通信と再同期します。エラーカウンタ (ECR.REC、ECR.TEC) は、エラーログ記録 (ECR.CEL) がアクティブになっている間は停止します。ソフトウェアは、ビット CCCR.ASM をセットすることで FDCAN を制限動作モードにセットできます。このビットは、CCCR.CCE および CCCR.INIT の両方が 1 にセットされているときのみソフトウェアによってセットできます。このビットはいつでもソフトウェアによってクリアできます。

Tx ハンドラがメッセージ RAM から時間内にデータを読み出せなかったとき、自動的に制限動作モードに移行します。制限動作モードを終了するには、ソフトウェアで CCCR.ASM をリセットする必要があります。

制限動作モードは、異なる CAN ビットレートに適応できるアプリケーションで使用できます。この場合、アプリケーションは異なるビットレートをテストし、有効なフレームを受信すると制限動作モードを終了します。

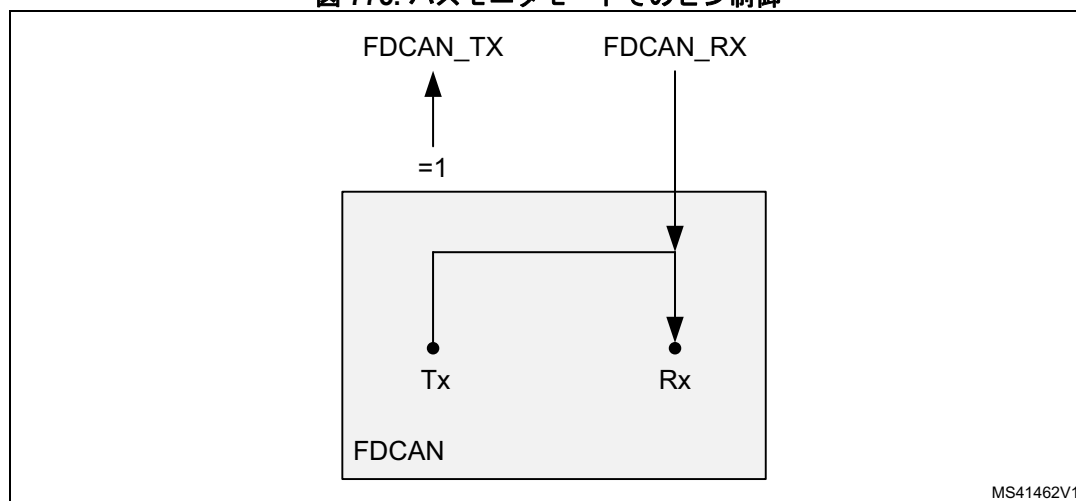
注： 制限動作モードは、ループバックモード（内部または外部）と組み合わせないでください。

バスモニタモード

FDCAN は、CCCR.MON ビットをセットすることでバスモニタモードにセットされます。バスモニタモード（詳細は ISO11898-1、10.12 Bus monitoring を参照してください）では、FDCAN は有効なデータフレームとリモートフレームを受信できますが、送信は開始できません。このモードでは、CAN バスでリセッシブビットのみを送信します。FDCAN がドミナントビット（ACK ビット、オーバーロードフラグ、アクティブエラーフラグ）を送信する必要がある場合、CAN バスがリセッシブ状態にとどまることがあっても FDCAN がこのドミナントビットを監視できるように、このビットは内部でルート変更されます。バスモニタモードでは、TXBRP レジスタがリセット状態のまま保持されます。

バスモニタモードを使用すると、ドミナントビットの送信による影響を受けずに、CAN バス上のトラフィックを解析できます。図 775 では、バスモニタモードでの FDCAN への FDCAN_TX および FDCAN_RX 信号の接続を示します。

図 775. バスモニタモードでのピン制御



MS41462V1

自動再送信モード無効 (DAR)

CAN 仕様 (ISO11898-1、6.3.3 Recovery Management) に従い、FDCAN はアービトレーションを喪失したフレームや送信中にエラーで中断されたフレームを自動的に再送信する方法を提供します。デフォルトでは自動再送信は有効化されています。

DAR モードでのフレーム送信

DAR モードでは、すべての送信が CAN バスで開始した後で自動的にキャンセルされます。Tx バッファ Tx リクエストペンディングビット TXBRP.TRPx は、転送が成功した後、転送がまだ始まっていない時点でのキャンセル時、ロストアービトレーションによるアボート時、もしくはフレーム転送中にエラーが発生した時にリセットされます。

- 送信成功
 - 対応する Tx バッファ送信発生ビット TXBTO[TOx] をセット
 - 対応する Tx バッファキャンセル終了ビット TXBCF[CFx] はセットされない
- キャンセル時に送信成功
 - 対応する Tx バッファ送信発生ビット TXBTO[TOx] をセット
 - 対応する Tx バッファキャンセル終了ビット TXBCF[CFx] をセット
- アービトラージョン喪失またはフレーム送信中断
 - 対応する Tx バッファ送信発生ビット TXBTO[TOx] はセットされない
 - 対応する Tx バッファキャンセル終了ビット TXBCF[CFx] をセット

フレーム送信が成功した場合、かつ Tx イベントのストレージが有効になっている場合、Tx イベント FIFO 要素はイベントタイプ ET = 10（キャンセル時送信）で書き込まれます。

パワーダウン (SLEEP モード)

FDCAN は、CC 制御レジスタ CCCR[CSR] 経由のクロック停止リクエスト入力によって制御されるパワーダウンモードにセットできます。クロック停止リクエストがアクティブである限り、ビット CCCR[CSR] は 1 で読み出されます。

すべてのペンディングされた送信リクエストが完了すると、FDCAN はバスアイドル状態が検出されるまで待ちます。その後、FDCAN は CCCR[INIT] を 1 にセットして、追加の CAN 転送を防ぎます。FDCAN は、CCCR[CSA] を 1 にセットすることでパワーダウンの準備が整ったことを確認します。この状態では、クロックがオフに切り替わる前に、追加のレジスタアクセスが可能です。CCCR[INIT] への書込みアクセスは効果がありません。これで、モジュールクロック入力はオフに切り替えられます。

パワーダウンモードを終了するには、アプリケーションが CC 制御レジスタフラグ CCCR.CSR をリセットする前にモジュールクロックをオンにしなければなりません。FDCAN は、CCCR[CSA] をリセットすることでこれを確認します。その後、アプリケーションは、ビット CCCR[INIT] をリセットすることで CAN 通信を再開できます。

テストモード

FDCAN テストレジスタ (FDCAN_TEST) への書込みアクセスを有効にするには、ビット CCCR.TEST を 1 にセットして、テストモードおよび機能の設定を有効にしなければなりません。

TEST.TX をプログラムすることで、CAN 送信ピン FDCAN_TX で 4 つの出力機能が使用できます。デフォルト機能（シリアルデータ出力）に加えて、CAN サンプルポイント信号を駆動して FDCAN ビットタイミングを監視し、一定のドミナントまたはリセッティング値を駆動できます。ピン FDCAN_RX で実際の値は、TEST.RX から読み出せます。両方の機能を使用して、CAN バス物理層をチェックできます。

CAN カーネルクロックと APB クロックドメインとの同期メカニズムによって、新しい設定が FDCAN_TX 出力ピンで確認できるようになるまで、TEST.TX への書込み間で APB クロック周期数周期分の遅延が発生する場合があります。これは、TEST.RX 経由で FDCAN_RX 入力ピンを読み出すときにも適用されます。

注： テストモードは、製造テストまたは自己診断機能にのみ使用する必要があります。FDCAN_TX ピンのソフトウェア制御は、すべての CAN プロトコル機能に干渉します。アプリケーションにテストモードを使用することは推奨されません。

外部ループバックモード

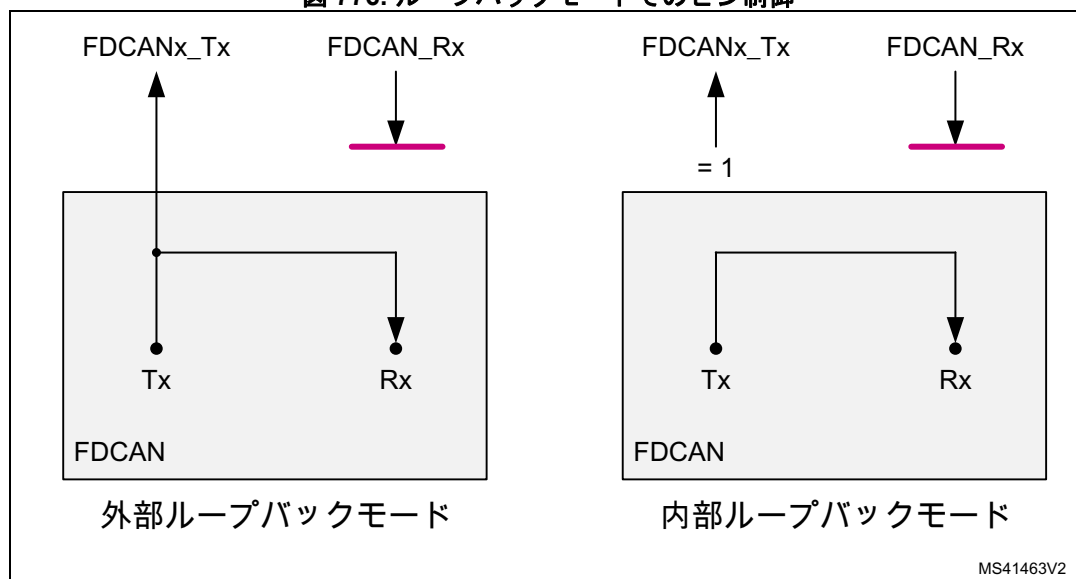
FDCAN を外部ループバックモードに設定するには、TEST.LBCK を 1 にセットします。ループバックモードでは、FDCAN はそれ自体が送信したメッセージを受信メッセージとして扱い、それらを Rx FIFO に格納します（受入フィルタを通過した場合）。図 776 では、外部ループバックモードでの FDCAN への送受信信号 FDCAN_TX および FDCAN_RX の接続を示します。

このモードは、ハードウェアの自己診断機能のために用意されています。外部通信から独立させるために、ループバックモードでは、FDCAN は ACK エラーを無視します（データ/リモートフレームの ACK スロットではリセシブビットがサンプリングされます）。このモードでは、FDCAN は送信出力から受信入力へ内部フィードバックを行います。FDCAN は、FDCAN_RX 入力ピンの実際の値を無視します。送信メッセージは FDCAN_TX 送信ピンで監視できます。

内部ループバックモード

TEST.LBCK と CCCR.MON を 1 にプログラムすることによって、内部ループバックモードに移行します。このモードは「ホットセルフテスト」に使用できます。つまり、FDCAN_TX および FDCAN_RX ピンに接続されている実行中の CAN システムに影響を与えずに、FDCAN をテストできます。このモードでは、FDCAN_RX ピンは FDCAN から切断され、FDCAN_TX ピンはリセシブに保持されます。図 776 では、内部ループバックモードの場合の FDCAN への FDCAN_TX および FDCAN_RX ピンの接続を示します。

図 776. ループバックモードでのピン制御



タイムスタンプの生成

タイムスタンプの生成について、FDCAN は 16 ビットラップアラウンドカウンタを提供します。プリスケアラ TSCC.TCP を設定して、CAN ビット時間の倍数（1～16）でカウンタにクロック供給できます。16)。カウンタは、TSCV[TCV] で読み出せます。レジスタ TSCV への書き込みアクセスによって、カウンタは 0 にリセットされます。タイムスタンプカウンタが 0 に戻ると、割込みフラグ IR[TSW] がセットされます。

フレーム受信/送信の開始時、カウンタ値がキャプチャされ、Rx FIFO (RXTS[15:0]) または Tx イベント FIFO (TXTS[15:0]) 要素のタイムスタンプセクションに格納されます。

ビット TSCC.TSS をプログラムすることで、16 ビットタイムスタンプを使用できます。

デバッグモードの動作

デバッグモードでは、読出し機能のセット／リセットはデバッガレジスタアクセス中に自動的に無効になり、通常の MCU 動作中に有効になります。

タイムアウトカウンタ

Rx FIFO 0、Rx FIFO 1、および Tx イベント FIFO のタイムアウト条件の信号を送信するために、FDCAN は 16-bit タイムアウトカウンタを提供します。ダウンカウンタとして動作し、TSCC[TCP] で制御される同じプリスケールをタイムスタンプカウンタとして使用します。タイムアウトカウンタは、レジスタ TOCC 経由で設定されます。実際のカウンタ値は、TOCV[TOC] から読み出せます。タイムアウトカウンタは CCCR[INIT] = 0 の間だけ開始できます。FDCAN が Bus_Off 状態に移行したときなど CCCR[INIT] = 1 の場合は停止します。

動作モードは TOCC[TOS] で選択されます。連続モードで動作時、CCCR[INIT] がリセットされるとカウンタが開始します。TOCV への書込みによって、カウンタは TOCC[TOP] で設定された値にプリセットされ、カウントダウンを続行します。

タイムアウトカウンタが FIFO のうち 1 つで制御されている場合、空の FIFO は TOCC[TOP] で設定された値にカウンタをプリセットします。カウントダウンは最初の FIFO 要素が格納されたときに開始します。TOCV への書込みは無効です。

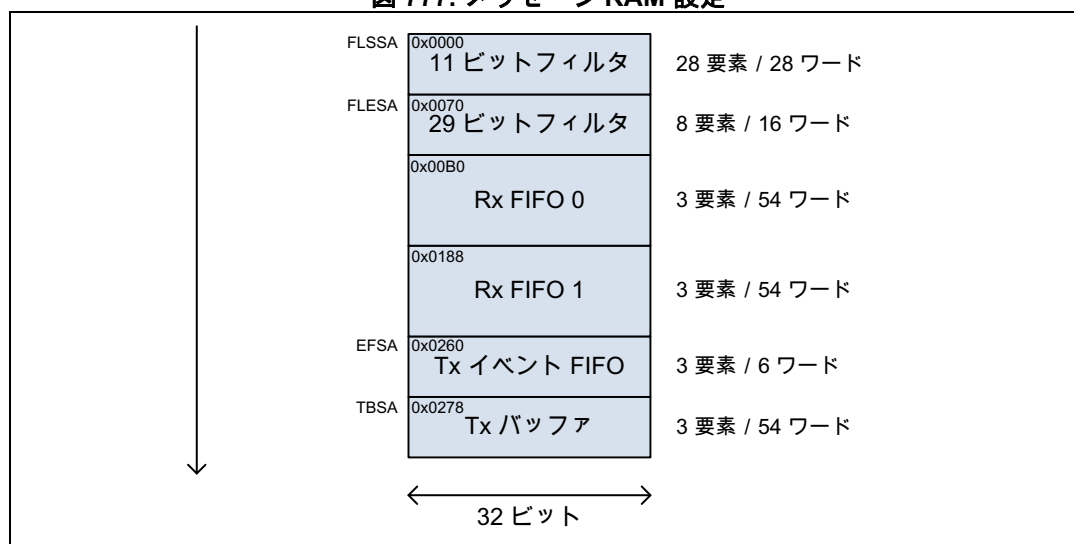
カウンタが 0 に達すると、割込みフラグ IR[TOO] がセットされます。連続モードでは、カウンタは TOCC[TOP] ですぐに再開されます。

注： タイムアウトカウンタのクロック信号は、CAN コアサンプルポイント信号から生成されます。そのため、タイムアウトカウンタがデクリメントされる時間内のポイントは、CAN コアの同期／再同期メカニズムによって異なる場合があります。FDCAN のポーレートスイッチ機能が使用されている場合、タイムアウトカウンタはアービトレーションおよびデータフィールドで別にクロック供給されます。

54.3.3 メッセージ RAM

メッセージ RAM は 32 ビット幅で、これに最大 212 ワードを割り当てるために FDCAN モジュールが設定されます。図 777 に示された各セクションを設定する必要はありません。

図 777. メッセージ RAM 設定



FDCAN がメッセージ RAM のアドレスを指定する場合、シングルバイトではなく 32 ビットワード（整列）を指定します。RAM アドレスは 32 ビットワードで、ビット 15 ~ 2 だけが評価され、2 つの最下位ビットは無視されます。

インスタンスが複数ある場合、FDCANn の RAM 開始アドレスは FDCANn-1 の終了アドレス + 4 で計算され、FDCANn の終了アドレスは FDCANn 開始アドレス + 0x0350 - 4 で計算されます。

たとえば、次の 2 つのインスタンスの場合：

- FDCAN1 :
 - 開始アドレス 0x0000
 - 終了アドレス 0x034C（図 777 を参照）
- FDCAN2 :
 - 開始アドレス = 0x034C（FDCAN1 終了アドレス）+ 4 = 0x0350
 - 終了アドレス = 0x0350（FDCAN2 開始アドレス）+ 0x0350 - 4 = 0x069C

Rx の処理

Rx ハンドラは、受入フィルタ、2 つの Rx FIFO のうちの 1 つへの受信メッセージの転送、および Rx FIFO の Put と Get のインデックスを制御します。

受入フィルタ

FDCAN は、2 セットの受入フィルタを設定でき、1 つは標準識別子用で、もう 1 つは拡張識別子用です。これらのフィルタは、Rx FIFO 0 または Rx FIFO 1 に割り当てられます。受入フィルタでは、フィルタの各リストが要素 #0 から最初に一致する要素まで実行されます。受入フィルタは最初に一致する要素で停止します。次のフィルタ要素はこのメッセージでは評価されません。

主な機能：

- 各フィルタ要素は次のように設定できます。
 - 範囲フィルタ（～から～まで）
 - 1 つまたは 2 つの専用 ID のフィルタ
 - クラシックビットマスクフィルタ
- 各フィルタ要素は、受入フィルタまたは拒否フィルタで設定できます。
- 各フィルタ要素は個別に有効化／無効化できます。
- フィルタは連続でチェックされ、最初の一致フィルタ要素で実行を停止します。

関連する設定レジスタ：

- グローバルフィルタ設定 (RXGFC)
- 拡張 ID ANDマスク (XIDAM)

フィルタ要素の設定 (SFEC/EFEC) に応じて、一致すると次の操作のいずれかがトリガされます。

- FIFO 0 または FIFO 1 に受信フレームを格納
- 受信フレームを拒否
- 高優先メッセージ割込みフラグ IR[HPM] のセット
- 高優先メッセージ割込みフラグ IR[HPM] をセットし、FIFO 0 または FIFO 1 に受信フレームを格納

受入フィルタは、完了識別子を受信した後に開始されます。受入フィルタが完了した後、一致する Rx FIFO が見つかった場合、メッセージハンドラが 32 ビットの単位で受信したメッセージデータを一致する Rx FIFO に書き込み始めます。CAN プロトコルコントローラがエラー条件 (CRC エラーなど) を検出した場合、このメッセージは次の影響を与えつつ破棄されます。

- **Rx FIFO**

一致する Rx FIFO の プット インデックスは更新されませんが、関連する FIFO 要素 (一部) が受信したデータで上書きされます。エラータイプについては、PSR.LEC および PSR.DLEC を参照してください。一致する Rx FIFO が上書きモードで動作している場合、[Rx FIFO 上書きモード](#)に記載された境界条件を考慮する必要があります。

注： 許可したメッセージが 2 つの Rx FIFO のうちの 1 つに書き込まれると、変更されていない受信 ID は、使用されるフィルタとは別に格納されます。受入フィルタプロセスの結果は、設定されたフィルタ要素のシーケンスに強く依存します。

範囲フィルタ

フィルタが SF1ID/SF2ID および EF1ID/EF2ID で定義された範囲のメッセージ ID を持つすべての受信フレームに一致します。

範囲フィルタを拡張フレームと使用する場合、2 つのことが可能です。

- EFT = 00 : 範囲フィルタが適用される前に、受信フレームのメッセージ ID は拡張 ID AND マスク (XIDAM) と AND でつながれます。
- EFT = 11 : 拡張 ID AND マスク (XIDAM) は、範囲フィルタに使用されません。

専用 ID のフィルタ

フィルタ要素を設定して、1 つまたは 2 つの固有 ID のフィルタでフィルタできます。1 つの固有メッセージ ID でフィルタするには、フィルタを SF1ID = SF2ID および EF1ID = EF2ID で設定する必要があります。

クラシックビットマスクフィルタ

クラシックビットマスクフィルタは、受信したメッセージ ID のシングルビットをマスクして、メッセージ ID のグループをフィルタすることを目的としています。クラシックビットマスクフィルタで、SF1ID/EF1ID はメッセージ ID フィルタとして使用され、SF2ID/EF2ID はフィルタマスクとして使用されます。

フィルタマスクでの 0 ビットは、設定された ID フィルタの対応するビット位置をマスクします。たとえば、そのビット位置で受信したメッセージ ID は受入フィルタの影響を受けません。対応するマスクビットが 1 の場合、受信したメッセージ ID のそれらのビットだけが受入フィルタの影響を受けます。

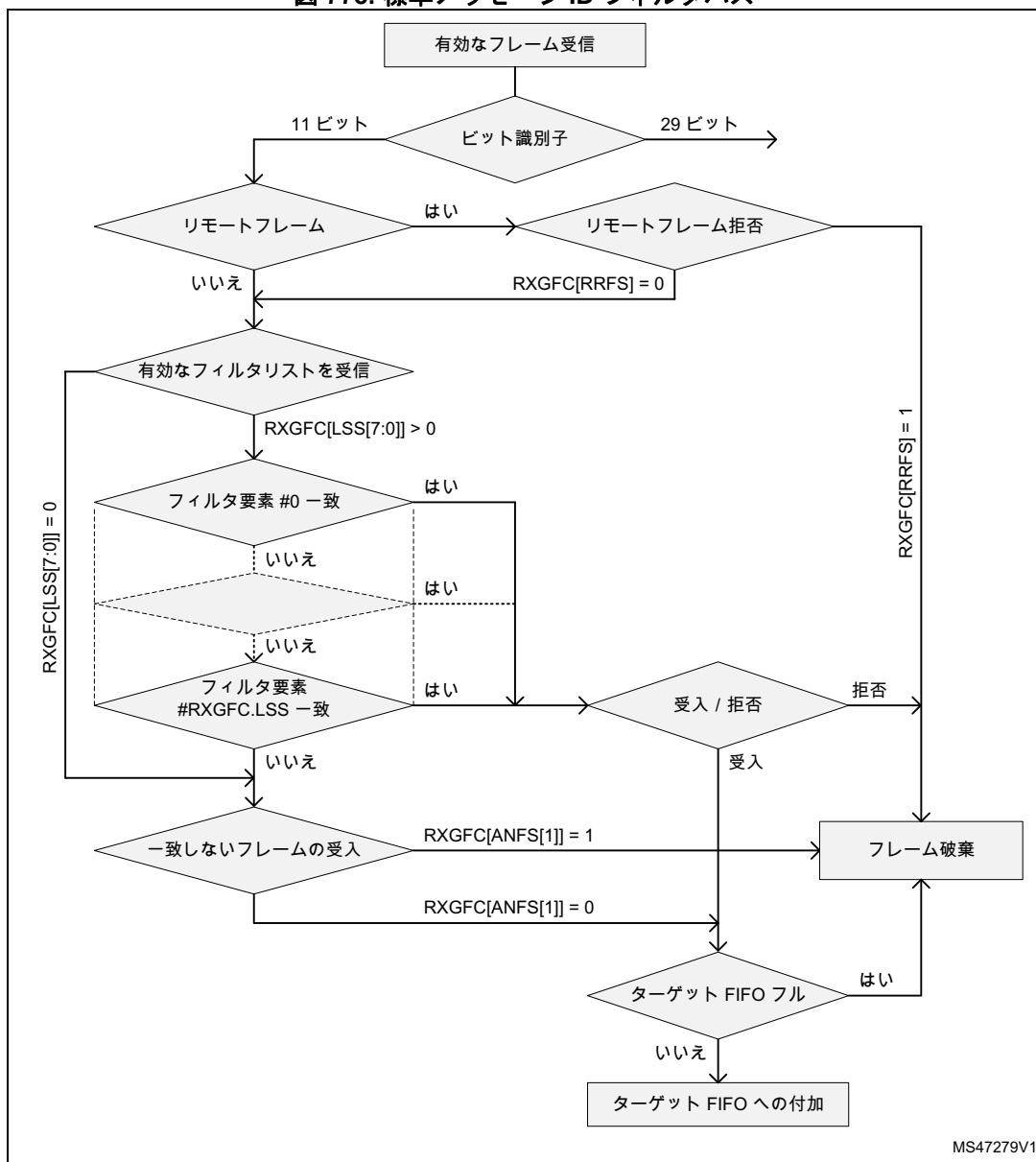
すべてのマスクビットが 1 の場合、受信したメッセージ ID とメッセージ ID フィルタが同一である場合のみ一致が発生します。すべてのマスクビットが 0 の場合、すべてのメッセージ ID が一致します。

標準メッセージ ID フィルタ

[図 778](#)では、標準メッセージ ID (11-bit 識別子) フィルタのフローを示します。標準メッセージ ID フィルタ要素は、[セクション 54.3.8](#)を参照してください。

グローバルフィルタ設定 (RXGFC) メッセージ ID で制御されている場合、受信フレームのリモート送信リクエストビット (RTR) と識別子拡張ビット (IDE) は設定されたフィルタ要素のリストと比較されます。

図 778. 標準メッセージ ID フィルタパス

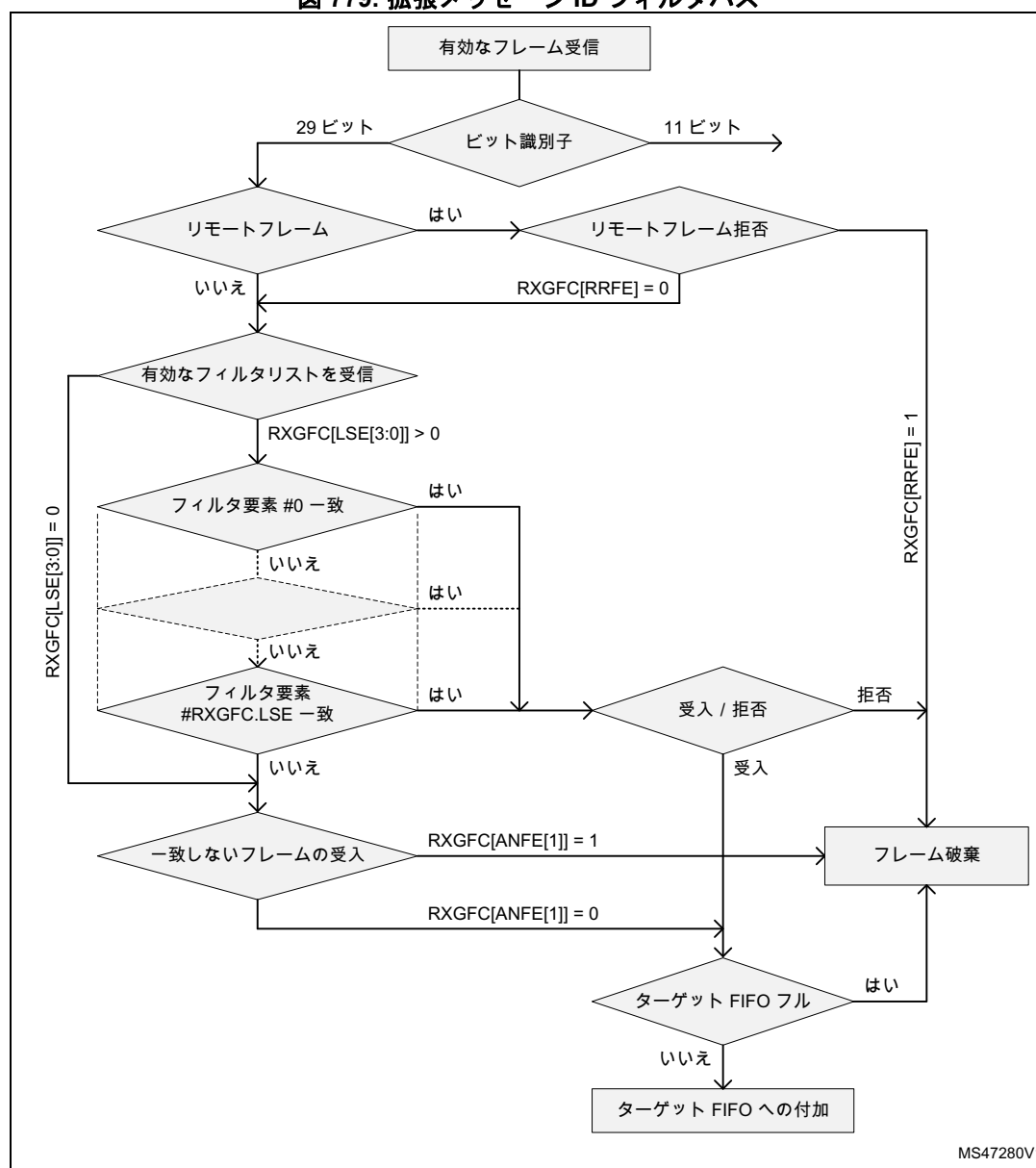


拡張メッセージ ID フィルタ

図 779では、拡張メッセージ ID (29-bit 識別子) フィルタのフローを示します。拡張メッセージ ID フィルタ要素は、[セクション 54.3.9](#) を参照してください。

グローバルフィルタ設定 RXGFC および拡張 ID フィルタ設定 RXGFC メッセージ ID で制御されている場合、受信フレームのリモート送信リクエストビット (RTR) と識別子拡張ビット (IDE) が設定されたフィルタ要素のリストと比較されます。

図 779. 拡張メッセージ ID フィルタパス



MS47280V1

フィルタリストが実行される前に、拡張 ID AND マスク (XIDAM) は受信識別子と AND でつながれます。

Rx FIFO

Rx FIFO 0 および Rx FIFO 1 は、それぞれ最大 3 個の要素を保持できます。

受入フィルタを通過した受信メッセージは、一致フィルタ要素で設定されたとおりに Rx FIFO に転送されます。Rx FIFO 0 および Rx FIFO 1 で使用できるフィルタメカニズムの詳細については、[受入フィルタ](#)を参照してください。Rx FIFO 要素は、[セクション 54.3.5](#)を参照してください。

Rx FIFO フル条件が IR[RFnF] によって信号送信されると、最低 1 つのメッセージが読み出され、Rx FIFO Get インデックスがインクリメントされるまで、対応する Rx FIFO に追加のメッセージは書き込まれません。対応する Rx FIFO がフルになっているときにメッセージを受信した場合、このメッセージは破棄され、割込みフラグ IR[RFnL] がセットされます。

Rx FIFO から読み出すときに、Rx FIFO Get インデックス $RxFnS[FnGI] + FIFO$ 要素サイズを対応する Rx FIFO 開始アドレス [FnSA] に追加する必要があります。

Rx FIFO ブロックモード

Rx FIFO ブロックモードは、RXGFC.FnOM = 0 で設定されます。これは、Rx FIFO のデフォルトの動作モードです。

Rx FIFO フル条件 ($RxFnS.FnPI = RxFnS.FnGI$) に達すると、最低 1 つのメッセージが読み出され、Rx FIFO Get インデックスがインクリメントされるまで、対応する Rx FIFO に追加のメッセージは書き込まれません。Rx FIFO フル条件は、 $RxFnS.FnF = 1$ で信号送信されます。さらに、割込みフラグ IR.RFnF がセットされます。

対応する Rx FIFO がフルになっているときにメッセージを受信した場合、このメッセージは破棄され、メッセージ喪失条件が $RxFnS.RFnL = 1$ で信号送信されます。さらに、割込みフラグ IR.RFnL がセットされます。

Rx FIFO 上書きモード

Rx FIFO 上書きモードは、RXGFC.FnOM = 1 で設定されます。

Rx FIFO フル条件 ($RxFnS.FnPI = RxFnS.FnGI$) が $RxFnS.FnF = 1$ によって信号送信されると、FIFO に許可された次のメッセージは一番古い FIFO メッセージを上書きします。Put および Get インデックスは両方 1 インクリメントされます。

Rx FIFO が上書きモードで動作し、Rx FIFO フル条件の信号を送信したとき、Rx FIFO 要素の読出しは最低 Get インデックス + 1 で開始する必要があります。これは、CPU がメッセージ RAM (Get インデックス) から読み出しつつ、受信メッセージがメッセージ RAM (Put インデックス) に書き込まれる状況になり得るためです。この場合、一貫していないデータがそれぞれの Rx FIFO 要素から読み出される場合があります。Rx FIFO から読出し時に 取得インデックスにオフセットを加えることで、この問題を回避しています。このオフセットは、CPU が Rx FIFO にアクセスする速度に依存します。

Rx FIFO から読み出した後、最後の要素読出しの数を Rx FIFO 確認応答インデックス $RxFnA.FnA$ に書き込む必要があります。これにより、取得インデックスがその要素数にインクリメントされます。プット インデックスがこの Rx FIFO 要素にインクリメントされなかった場合、Rx FIFO フル条件はリセットされます ($RxFnS.FnF = 0$)。

Tx の処理

Tx ハンドラは、Tx FIFO および Tx キューの送信リクエストを処理します。CAN コア、Put および Get インデックス、Tx イベント FIFO への送信メッセージの転送を制御します。最大 3 個の Tx バッファをメッセージ送信にセットアップできます。CAN メッセージデータフィールドは 64 バイトに設定され、Tx FIFO は Tx 要素の格納に 18 個の 32 ビットワードを配置します。

表 578. フレーム送信の可能な設定

CCCR		Tx バッファ要素		フレーム送信
BRSE	FDOE	FDF	BRS	
無視	0	無視	無視	クラシック CAN
0	1	0	無視	クラシック CAN
0	1	1	無視	ビットレート切り替えなしの FD
1	1	0	無視	クラシック CAN
1	1	1	0	ビットレート切り替えなしの FD
1	1	1	1	ビットレート切り替えありの FD

注: AUTOSAR は最低 3つの Tx キューバッファと送信キャンセルのサポートが必須です。

Tx ハンドラは、Tx バッファリクエストペンディングレジスタ TXBRP が更新されたときや、送信が開始されたときに、Tx スキャンを開始して、最優先のペンディング中の Tx リクエスト（一番小さいメッセージ ID の Tx バッファ）をチェックします。

送信一時停止

送信一時停止機能は、CAN メッセージ識別子が（完全に）固有の値に指定され、容易に変更できない CAN システムで使用することを目的とします。これらのメッセージ識別子は、他の定義されたメッセージより高い CAN アービトレーション優先度である場合がありますが、個別のアプリケーションでは関連するアービトレーション優先度が逆でなければなりません。これにより、他のメッセージが低い CAN アービトレーション優先度であるため、ある ECU が一気に CAN メッセージを送信して別の ECU CAN メッセージを遅延させる場合があります。

たとえば、CAN ECU-1 が機能を有効にして、アプリケーションソフトウェアによって 4 件のメッセージを送信するようリクエストされた場合、最初のメッセージ送信が成功した後、次のリクエストされたメッセージの開始が許可される前に 2 CAN ビット時間のバスアイドルを待ちます。ペンディング中のメッセージを保持する他の ECU がある場合、それらのメッセージはアイドル時間で開始され、ECU-1 の次のメッセージとアービトレーションする必要はありません。メッセージを受信した後、受信メッセージが CAN バスを開放すると、ECU-1 は次の送信開始が許可されます。

この機能は、CCCR レジスタの TXP ビットで制御されます。このビットがセットされている場合、FDCAN がメッセージの送信に成功するたびに、次の送信開始前の 2 CAN ビット時間、一時停止します。これにより、ネットワーク内の他の CAN ノードのメッセージが低い優先度識別子を持つものであってもメッセージを送信できます。デフォルトでは無効です (CCCR.TXP = 0)。

この機能は、1 つのノードからのバースト送信を緩和し、アプリケーションプログラムが誤って大量の送信をリクエストする「バブリングイディオット」シナリオから保護します。

Tx FIFO

Tx FIFO 動作は、TXBC[TFQM] を 0 にプログラムすることによって設定されます。Tx FIFO に格納されたメッセージは、Get インデックス TXFQS[TFGI] によって参照されるメッセージから送信されます。各送信後、Get インデックスは Tx FIFO がエンプティになるまで周期的にインクリメントされます。Tx FIFO は、メッセージが Tx FIFO に書き込まれた順番で、異なる Tx バッファからの同じメッセージ ID を持つメッセージを送信できます。FDCAN は、Tx FIFO フリーレベル TXFQS[TFFL] を Get および Put インデックス間の差として計算します。これは、使用可能な（空き）Tx FIFO 要素の数を示します。

新しい送信メッセージは、Put インデックス TXFQS[TFQPI] によって参照される Tx バッファから始まる Tx FIFO に書き込まれる必要があります。追加リクエストは、次の空き Tx FIFO 要素に Put イ

ンデックスをインクリメントします。Put インデックスが Get インデックスに達すると、Tx FIFO フル (TXFQS[TFQF]=1) の信号が送信されます。この場合、次のメッセージが送信され、Get インデックスがインクリメントされるまで、Tx FIFO に追加のメッセージを書き込まないでください。

1 つのメッセージが Tx FIFO に追加される場合、Tx FIFO Put インデックスによって参照される Tx バッファに関連した TXBAR ビットに 1 を書き込むことで送信がリクエストされます。

複数 (n) 件のメッセージが Tx FIFO に追加される場合、Put インデックスから始まる n 個の連続した Tx バッファに書き込まれます。送信は TXBAR 経由でリクエストされます。Put インデックスは周期的に n ずつインクリメントされます。リクエストされる Tx バッファの数は、Tx FIFO フリーレベルで示された空きの Tx バッファの数を超えてはなりません。

Get インデックスによって参照される Tx バッファに対する送信リクエストがキャンセルされた場合、Get インデックスは送信リクエストをペンディングして次の Tx バッファにインクリメントされ、Tx FIFO フリーレベルが再計算されます。送信キャンセルが他のあらゆる Tx バッファに適用される場合、Get インデックスと FIFO フリーレベルに変化はありません。

Tx FIFO 要素はメッセージ RAM で 18 の 32 ビットワードを割り当てます。そのため、次に使用可能な (空き) Tx FIFO バッファの開始アドレスは、Put インデックス TXFQS[TFQPI] (0 ~ 2) を Tx バッファ開始アドレス TBSA に 4 回追加することで計算されます。

Tx キュー

Tx キュー動作は、TXBC[TFQM] を 1 にプログラムすることによって設定されます。Tx キューに格納されたメッセージは、一番小さいメッセージ ID (最優先) のメッセージから送信されます。

標準メッセージ ID と拡張メッセージ ID が混在している場合、標準メッセージ ID は拡張メッセージ ID のビット [28:18] と比較されます。

複数のキューバッファが同じメッセージ ID で設定される場合、一番小さいバッファ数のキューバッファが先に送信されます。

新しいメッセージは、Put インデックス TXFQS[TFQPI] によって参照される Tx バッファに書き込まれる必要があります。追加リクエストは、次の空き Tx バッファに Put インデックスをインクリメントします。Tx キューがフル (TXFQS[TFQF]=1) の場合、Put インデックスは有効ではなく、リクエストされるメッセージのうち最低 1 件が送信されるか、ペンディング中の送信リクエストがキャンセルされるまで、追加のメッセージを Tx キューに書き込まないでください。

アプリケーションは Put インデックスの代わりにレジスタ TXBRP を使用し、送信リクエストをペンディングせずに Tx バッファにメッセージを配置する場合があります。

Tx キューバッファはメッセージ RAM で 18 の 32 ビットワードを割り当てます。そのため、次に使用可能な (空き) Tx キューバッファの開始アドレスは、Tx キュー Put インデックス TXFQS[TFQPI] (0 ~ 2) を Tx バッファ開始アドレス TBSA に 4 回追加することで計算されます。

送信キャンセル

FDCAN は送信キャンセルをサポートします。Tx キューバッファからリクエストされた送信をキャンセルするには、ホストはレジスタ TXBCR の対応するビット位置 (= Tx バッファの数) に 1 を書き込む必要があります。送信キャンセルは Tx FIFO 動作を対象としません。

成功したキャンセルは、レジスタ TXBCF の対応するビットを 1 にセットすることで信号送信されます。

Tx バッファからの送信がすでに進行中で、送信キャンセルがリクエストされた場合、対応する TXBRP ビットは送信が進行している限りセットされたままとなります。送信に成功した場合、対応する TXBTO および TXBCF ビットがセットされます。送信に成功しなかった場合、繰り返されず、対応する TXBCF ビットのみセットされます。

注： ペンディング中の送信が開始直前にキャンセルされた場合、別のメッセージがこのノードでペンディングされている場合でも送信が開始されない短いタイムウィンドウが続きます。これにより、別のノードはそのノードの 2 番目の優先度のメッセージより低い優先度のメッセージを送信できます。

Tx イベントの処理

Tx イベントの処理をサポートするために、FDCAN は Tx イベント FIFO を実装しました。FDCAN が CAN バスでメッセージを送信すると、メッセージ ID およびタイムスタンプが Tx イベント FIFO 要素に格納されます。Tx イベントを Tx イベント FIFO 要素にリンクするために、送信された Tx バッファのメッセージマーカーが Tx イベント FIFO 要素にコピーされます。

Tx イベント FIFO は、最大 3 個の要素に設定できます。Tx イベント FIFO 要素は、[Tx FIFO](#) を参照してください。

Tx イベント FIFO の目的は、Tx バッファが送信すべきメッセージのみを保持しつつ、送信ステータスは Tx イベント FIFO に別に格納されるなど、送信ステータス情報の処理を送信メッセージ処理から切り離すことです。これには、特に動的に管理される送信キューを操作する際に、Tx バッファが送信成功直後に新しいメッセージに使用できるという利点があります。Tx バッファを上書きする前に、その Tx バッファからの送信ステータス情報を保存する必要はありません。

Tx イベント FIFO フル条件が IR[TEFF] によって信号送信されると、最低 1 つの要素が読み出され、Tx イベント FIFO Get インデックスがインクリメントされるまで、Tx イベント FIFO に追加の要素は書き込まれません。Tx イベント FIFO がフルになっているときに Tx イベントが発生した場合、このイベントは破棄され、割込みフラグ IR[TEFL] がセットされます。

Tx イベント FIFO から読み出すときに、2 倍の Tx イベント FIFO Get インデックス TXEFS[EFGI] を Tx イベント FIFO 開始アドレス EFSA に追加する必要があります。

54.3.4 FIFO 確認応答の処理

Rx FIFO 0、Rx FIFO 1、および Tx イベント FIFO の Get インデックスは、対応する FIFO 確認応答インデックスへの書き込みによって制御されます。[セクション 54.4.23](#) および [セクション 54.4.25](#) を参照してください。FIFO 確認応答インデックスへの書き込みは FIFO Get インデックスを FIFO 確認応答インデックス +1 にセットし、FIFO フィルレベルを更新します。これには、2 つの使用例があります。

1. 要素が 1 つだけ FIFO (Get インデックスによって指定されているもの) から読み出された場合、この Get インデックス値は FIFO 確認応答インデックスに書き込まれます。
2. 一連の要素が FIFO から読み出された場合、読出しシーケンスの最後に 1 回だけ FIFO 確認応答インデックス (値: 最後の要素読出しのインデックス) を書き込んで、FIFO Get インデックスを更新するだけで十分です。

CPU が FDCAN メッセージ RAM に自由にアクセスできるため、任意の (Get インデックスが考慮されない) 順番で FIFO 要素を読み出す際に特別な注意が必要です。これは、2 つの Rx FIFO のうち 1 つから高優先メッセージを読み出すときに役立つ場合があります。この場合、Get インデックスが誤った位置にセットされ、FIFO フィルレベルも変わるため、FIFO 確認応答インデックスへの書き込みは行わないでください。この場合、古いほうの FIFO 要素の一部は失われます。

注： アプリケーションは、有効な値が FIFO 確認応答インデックスに書き込まれていることを確認する必要があります。FDCAN は誤った値をチェックしません。

54.3.5 FDCAN Rx FIFO 要素

2 つの Rx FIFO がメッセージ RAM に設定されています。各 Rx FIFO セクションを設定して、最大 3 個の受信メッセージを格納できます。Rx FIFO 要素の構造は、表 579 に示され、表 580 に説明が示されています。

表 579. Rx FIFO 要素

ビット	31				24	23				16	15		8	7		0
R0	ESI	XTD	RTR	ID[28:0]												
R1	ANMF	FIDX[6:0]				Res.	FDF	BRS	DLC[3:0]	RXTS[15:0]						
R2	DB3 [7:0]				DB2 [7:0]					DB1 [7:0]			D[7:0]			
R3	DB7 [7:0]				DB6 [7:0]					DB5 [7:0]			DB4 [7:0]			
⋮	⋮				⋮					⋮						
Rn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]			DBm-3[7:0]			

CAN FD メッセージの格納について設定された要素サイズは、64 バイトのデータフィールドに設定されます。

表 580. Rx FIFO 要素の説明

フィールド	説明
R0 ビット 31 ESI	エラー状態インジケータ – 0 : 送信中のノードはエラーアクティブです。 – 1 : 送信中のノードはエラーパッシブです。
R0 ビット 30 XTD	拡張識別子 受信フレームに標準識別子または拡張識別子があるかどうかをホストに信号送信します。 – 0 : 11 ビット標準識別子です。 – 1 : 29 ビット拡張識別子です。
R0 ビット 29 RTR	リモート送信リクエスト 受信フレームがデータフレームかリモートフレームかどうかをホストに信号送信します。 – 0 : 受信フレームはデータフレームです。 – 1 : 受信フレームはリモートフレームです。
R0 ビット 28:0 ID[28:0]	識別子 ビット XTD に応じた標準識別子または拡張識別子です。標準識別子は、ID[28:18] に保存されます。
R1 ビット 31 ANMF	許可される一致しないフレーム 一致しないフレームの許可は、RXGFC[ANFS] および RXGFC[ANFE] 経由で有効にできます。 – 0 : 受信フレームがフィルタインデックスに一致します。 – 1 : 受信フレームがどの Rx フィルタ要素にも一致しません。
R1 ビット 30:24 FIDX[6:0]	フィルタインデックス 0-27 = 一致している Rx 受入フィルタ要素のインデックス (ANMF = 1 の場合は無効)。 範囲は、0~RXGFC[LSS] - 1 または RXGFC[LSE] - 1 です。
R1 ビット 21 FDF	FD フォーマット – 0 : 標準フレームフォーマットです。 – 1 : FDCAN フレームフォーマット (新しい DLC コーディングおよび CRC)

表 580. Rx FIFO 要素の説明 (続き)

フィールド	説明
R1 ビット 20 BRS	ビットレート切り替え – 0 : ビットレート切り替えなしで受信したフレームです。 – 1 : ビットレート切り替えありで受信したフレームです。
R1 ビット 19:16 DLC[3:0]	データ長コード – 0~8 : クラシック CAN + CAN FD : 受信フレームに 0 ~ 8 のデータバイトがあります。 – 9~15 : クラシック CAN : 受信フレームに 8 データバイトがあります。 – 9~15 : CAN FD : 受信フレームに 12/16/20/24/32/48/64 のデータバイトがあります。
R1 ビット 15:0 RXTS[15:0]	Rx タイムスタンプ フレーム受信の開始時にキャプチャされるタイムスタンプカウンタ値です。分解能はタイムスタンプカウンタプリスケラ TSCC[TCP] の設定に依存します。
R2 ビット 31:24 DB3 [7:0]	データバイト 3
R2 ビット 23:16 DB2 [7:0]	データバイト 2
R2 ビット 15:8 DB1 [7:0]	データバイト 1
R2 ビット 7:0 D[7:0]	データバイト 0
R3 ビット 31:24 DB7 [7:0]	データバイト 7
R3 ビット 23:16 DB6 [7:0]	データバイト 6
R3 ビット 15:8 DB5 [7:0]	データバイト 5
R3 ビット 7:0 DB4 [7:0]	データバイト 4
⋮	⋮
Rn ビット 31:24 DBm[7:0]	データバイト m
Rn ビット 23:16 DBm-1[7:0]	データバイト m-1
Rn ビット 15:8 DBm-2[7:0]	データバイト m-2
Rn ビット 7:0 DBm-3[7:0]	データバイト m-3

54.3.6 FDCAN の Tx バッファ要素

Tx バッファセクション (3 つの要素) は Tx FIFO または Tx キューを保持するように設定できます。Tx ハンドラは Tx FIFO と Tx キューを Tx バッファ設定 FDCAN_TXBC.TFQM を使用して区別します。要素サイズは、最大 64 バイトのデータがある CAN FD メッセージの格納について設定されます。

表 581. Tx バッファおよび FIFO 要素

ビット	31	24	23	16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]				
T1	MM[7:0]			EFC	Res.	FDF	BPS	DLC[3:0]
T2	DB3 [7:0]			DB2 [7:0]			DB1 [7:0]	D[7:0]
T3	DB7 [7:0]			DB6 [7:0]			DB5 [7:0]	DB4 [7:0]
⋮	⋮			⋮			⋮	
Tn	DBm[7:0]			DBm-1[7:0]			DBm-2[7:0]	DBm-3[7:0]

表 582. Tx バッファ要素の説明

フィールド	説明
T0 ビット 31 ESI ⁽¹⁾	エラー状態インジケータ – 0 : CAN FD フォーマットの ESI ビットはエラーパッシブフラグにのみ依存します。 – 1 : CAN FD フォーマットの ESI ビットはリセッショで送信されました。
T0 ビット 30 XTD	拡張識別子 – 0 : 11 ビット標準識別子です。 – 1 : 29 ビット拡張識別子です。
T0 ビット 29 RTR ⁽²⁾	リモート送信リクエスト – 0 : データフレームを送信します。 – 1 : リモートフレームを送信します。
T0 ビット 28:0 ID[28:0]	識別子 ビット XTD に応じた標準識別子または拡張識別子です。標準識別子は、ID[28:18] に書き込まれる必要があります。
T1 ビット 31:24 MM[7:0]	メッセージマーカ Tx バッファ設定中に CPU によって書き込まれます。Tx メッセージステータスの識別のために Tx イベント FIFO 要素にコピーされます。
T1 ビット 23 EFC	イベント FIFO 制御 – 0 : Tx イベントを保存しません。 – 1 : Tx イベントを保存します。
T1 ビット 21 FDF	FD フォーマット – 0 : クラシック CAN フォーマットでフレームが送信されます。 – 1 : CAN FD フォーマットでフレームが送信されます。
T1 ビット 20 BRS ⁽³⁾	ビットレート切り替え – 0 : ビットレート切り替えなしで送信された CAN FD フレームです。 – 1 : ビットレート切り替えありで送信された CAN FD フレームです。

表 582. Tx バッファ要素の説明 (続き)

フィールド	説明
T1 ビット 19:16 DLC[3:0]	データ長コード – 0~8 : クラシック CAN + CAN FD : 受信フレームに 0 ~ 8 のデータバイトがあります。 – 9~15 : クラシック CAN : 受信フレームに 8 データバイトがあります。 – 9~15 : CAN FD : 受信フレームに 12/16/20/24/32/48/64 のデータバイトがあります。
T2 ビット 31:24 DB3 [7:0]	データバイト 3
T2 ビット 23:16 DB2 [7:0]	データバイト 2
T2 ビット 15:8 DB1 [7:0]	データバイト 1
T2 ビット 7:0 D[7:0]	データバイト 0
T3 ビット 31:24 DB7 [7:0]	データバイト 7
T3 ビット 23:16 DB6 [7:0]	データバイト 6
T3 ビット 15:8 DB5 [7:0]	データバイト 5
T3 ビット 7:0 DB4[7:0]	データバイト 4
⋮	⋮
Tn ビット 31:24 DBm[7:0]	データバイト m
Tn ビット 23:16 DBm-1[7:0]	データバイト m-1
Tn ビット 15:8 DBm-2[7:0]	データバイト m-2
Tn ビット 7:0 DBm-3[7:0]	データバイト m-3

- 送信バッファの ESI ビットは、エラーパッシブフラグで論理和がとられ、送信される FD フレームでの ESI ビットの値を決定します。CAN FD プロトコルの仕様によって必須とされたとおり、エラーアクティブノードではオプションで ESI ビットリセッスを送信できますが、エラーパッシブノードでは常に ESI ビットリセッスを送信します。
- RTR = 1 のとき、CCCR.FDOE が CAN FD フォーマットで送信を有効にしている場合でも、FDCAN は ISO11898-1 に従ってリモートフレームを送信します。
- ビット ESI、FDF、BRS は CAN FD 動作が有効 (CCCR.FDOE = 1) のときのみ評価されます。ビット BRS はさらに CCCR.BRSE = 1 のときのみ評価されます。

54.3.7 FDCAN Tx イベント FIFO 要素

各要素には、送信されたメッセージに関する情報が保存されています。Tx イベント FIFO を読み出すことで、ホスト CPU はメッセージが送信された順にこの情報を取得します。Tx イベント FIFO に関するステータス情報は、レジスタ TXEFS から取得できます。

表 583. Tx イベント FIFO 要素

ビット	31	24	23	16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]				
E1	MM[7:0]			ET[1:0]	EDL	BRS	DLC[3:0]	TXTS[15:0]

表 584. Tx イベント FIFO 要素の説明

フィールド	説明
E0 ビット 31 ESI	エラー状態インジケータ – 0 : 送信中のノードはエラーアクティブです。 – 1 : 送信中のノードはエラーパッシブです。
E0 ビット 30 XTD	拡張識別子 – 0 : 11 ビット標準識別子です。 – 1 : 29 ビット拡張識別子です。
E0 ビット 29 RTR	リモート送信リクエスト – 0 : データフレームを送信します。 – 1 : リモートフレームを送信します。
E0 ビット 28:0 ID[28:0]	識別子 ビット XTD に応じた標準識別子または拡張識別子です。標準識別子は、ID[28:18] に書き込まれる必要があります。
E1 ビット 31:24 MM[7:0]	メッセージマーカ Tx メッセージステータスの識別のために Tx バッファから Tx イベント FIFO 要素にコピーされます。
E1 ビット 23:22 EFC	イベントタイプ – 00 : 予約済みです。 – 01 : Tx イベント – 10 : キャンセル時でも送信 (DAR モードでの送信では常にセット) – 11 : 予約済みです。
E1 ビット 21 EDL	拡張データ長 – 0 : 標準フレームフォーマットです。 – 1 : FDCAN フレームフォーマット (新しい DLC コーディングおよび CRC)
E1 ビット 20 BRS	ビットレート切り替え – 0 : ビットレート切り替えなしで送信されたフレームです。 – 1 : ビットレート切り替えありで送信されたフレームです。
T1 ビット 19:16 DLC[3:0]	データ長コード 0~8 : 0 ~ 8 のデータバイトのフレームが送信されます。 9~15 : 8 データバイトのフレームが送信されます。
E1 ビット 15:0 TXTS[15:0]	Tx タイムスタンプ フレーム送信の開始時にキャプチャされるタイムスタンプカウンタ値です。分解能はタイムスタンプカウンタプリスケラ TSCC[TCP] の設定に依存します。

54.3.8 FDCAN 標準メッセージ ID フィルタ要素

最大 28 のフィルタ要素を 11 ビット標準 ID に設定できます。標準メッセージ ID フィルタ要素にアクセスするとき、そのアドレスはフィルタリストの標準開始アドレス FLSSA とフィルタ要素のインデックス (0 ~ 27) です。

表 585. 標準メッセージ ID フィルタ要素

ビット	31	24	23	16	15	8	7	0
S0	SFT[1:0]	SFEC[2:0]	SFID1 [10:0]			Res.	SFID2 [10:0]	

表 586. 標準メッセージ ID フィルタ要素フィールドの説明

フィールド	説明
ビット 31:30 SFT[1:0] ⁽¹⁾	標準フィルタタイプ – 00 : SFID1 から SFID2 までの範囲のフィルタ – 01 : SFID1 または SFID2 のデュアル ID フィルタ – 10 : クラシックフィルタ : SFID1 = フィルタ、SFID2 = マスク – 11 : フィルタ要素無効化
ビット 29:27 SFEC[2:0]	標準フィルタ要素の設定 すべての有効化されたフィルタ要素は、標準フレームの受入フィルタに使用されます。受入フィルタは、有効化されたフィルタ要素に最初に一致したとき、またはフィルタの最後に達した場合に停止します。SFEC = 100、101、または 110 の場合、一致すると割込みフラグ IR.HPM がセットされ、有効であれば、割込みが生成されます。この場合、レジスタ HPMS は優先度一致のステータスで更新されます。 – 000 : フィルタ要素の無効化 – 001 : フィルタが一致した場合に Rx FIFO 0 に格納 – 010 : フィルタが一致した場合に Rx FIFO 1 に格納 – 011 : フィルタが一致した場合に ID を拒否 – 100 : フィルタが一致した場合に優先度をセット – 101 : フィルタが一致した場合に優先度をセットして Rx FIFO 0 に格納 – 110 : フィルタが一致した場合に優先度をセットして Rx FIFO 1 に格納 – 111 : 未使用
ビット 26:16 SFID1 [10:0]	標準フィルタ ID 1 標準 ID フィルタ要素の最初の ID
ビット 10:0 SFID2 [10:0]	標準フィルタ ID 2 標準 ID フィルタ要素の 2 番目の ID

1. SFT = 11 でフィルタ要素は無効化され、受入フィルタは続行します (SFEC = 000 と同じ動作)。

注 : 予約済みの値が設定されている場合、フィルタ要素は無効とみなされます。

54.3.9 FDCAN 拡張メッセージ ID フィルタ要素

最大 8 のフィルタ要素を 29 ビット拡張 ID に設定できます。拡張メッセージ ID フィルタ要素にアクセスするとき、そのアドレスはフィルタリストの拡張開始アドレス FLESA にフィルタ要素のインデックス (0 ~ 7) の 2 倍を加えたものです。

表 587. 拡張メッセージ ID フィルタ要素

ビット	31	24	23	16	15	8	7	0
F0	EFEC[2:0]		EFID1 [28:0]					
F1	EFTI[1:0]	Res.	EFID2 [28:0]					

表 588. 拡張メッセージ ID フィルタ要素フィールドの説明

フィールド	説明
F0 ビット 31:29 EFEC[2:0]	<p>拡張フィルタ要素の設定</p> <p>すべての有効化されたフィルタ要素は、拡張フレームの受入フィルタに使用されます。受入フィルタは、有効化されたフィルタ要素に最初に一致したとき、またはフィルタの最後に達した場合に停止します。EFEC = 100、101、または 110 の場合、一致すると割り込みフラグ IR[HPM] がセットされ、有効であれば、割り込みが生成されます。この場合、レジスタ HPMS は優先度一致のステータスで更新されます。</p> <ul style="list-style-type: none"> – 000 : フィルタ要素の無効化 – 001 : フィルタが一致した場合に Rx FIFO 0 に格納 – 010 : フィルタが一致した場合に Rx FIFO 1 に格納 – 011 : フィルタが一致した場合に ID を拒否 – 100 : フィルタが一致した場合に優先度をセット – 101 : フィルタが一致した場合に優先度をセットして Rx FIFO 0 に格納 – 110 : フィルタが一致した場合に優先度をセットして Rx FIFO 1 に格納 – 111 : 未使用
F0 ビット 28:0 EFID1 [28:0]	<p>拡張フィルタ ID 1</p> <p>拡張 ID フィルタ要素の最初の ID</p> <p>Rx FIFO のフィルタ時、このフィールドで格納する拡張メッセージの ID を定義します。受信した識別子は完全一致とし、XIDAM マスキングメカニズムのみです。</p>
F1 ビット 31:30 EFTI[1:0]	<p>拡張フィルタタイプ</p> <ul style="list-style-type: none"> – 00 : EF1ID から EF2ID までの範囲のフィルタ (EF2ID ≥ EF1ID) – 01 : EF1ID または EF2ID のデュアル ID フィルタ – 10 : クラシックフィルタ : EF1ID = フィルタ、EF2ID = マスク – 11 : EF1ID から EF2ID までの範囲のフィルタ (EF2ID ≥ EF1ID)、XIDAM 適用なし
F1 ビット 29	未使用
F1 ビット 28:0 EFID2 [28:0]	<p>拡張フィルタ ID 2</p> <p>標準 ID フィルタ要素の 2 番目の ID</p>

54.4 FDCAN レジスタ

54.4.1 FDCAN コア開放レジスタ (FDCAN_CREL)

アドレスオフセット : 0x0000

リセット値 : 0x3214 1218

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 **REL[3:0]** : 3

ビット 27:24 **STEP[3:0]** : 2

ビット 23:20 **SUBSTEP[3:0]** : 1

ビット 19:16 **YEAR[3:0]** : 4

ビット 15:8 **MON[7:0]** : 12

ビット 7:0 **DAY[7:0]** : 18

54.4.2 FDCAN エンディアンレジスタ (FDCAN_ENDN)

アドレスオフセット : 0x0004

リセット値 : 0x8765 4321

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETV[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **ETV[31:0]** : エンディアンテスト値

エンディアンテスト値は 0x8765 4321 です。

注 : レジスタ読出しでは、エンディアンに問題がないことを確認するために、リセット値を示す必要があります。

54.4.3 FDCAN データビットタイミングおよびプリスケアラレジスタ (FDCAN_DBTP)

アドレスオフセット : 0x000C

リセット値 : 0x0000 : 0A33

このレジスタは、CCCR.CCE および CCCR.INIT がセットされている場合のみ書込みできます。CAN タイムクオンタムは、1~32 FDCAN 周期の範囲でプログラム可能です。tq = (DBRP + 1) FDCAN クロック周期。

DTSEG1 は、Prop_Seg と Phase_Seg1 の合計です。DTSEG2 は、Phase_Seg2 です。そのため、ビット時間の長さは (プログラムされた値) [DTSEG1 + DTSEG2 + 3] tq または (機能の値) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq となります。

情報処理時間 (IPT) は 0 で、次のビットのデータはサンプルポイント後の最初のクロックエッジで使用できることを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]				
								rw			rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DTSEG1 [4:0]					DTSEG2 [3:0]				DSJW[3:0]			
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TDC** : トランシーバ遅延補正

0 : トランシーバ遅延補正は無効です。

1 : トランシーバ遅延補正は有効です。

ビット 22:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **DBRP[4:0]** : データビットレートプリスケアラ

ビット時間のクオンタムを生成するためにオシレータ周波数を分周する値です。ビット時間は、このクオンタムの倍数で構築されます。ボーレートプリスケアラに有効な値は 0~31 です。ハードウェアはこの値をプログラムされた値 +1 として解釈します。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DTSEG1[4:0]** : サンプルポイント前のデータ時間セグメント

有効な値は 0~31 です。ハードウェアによって使用される値はプログラムされたもので、1ずつ増加します。例 : tBS1 = (DTSEG1 + 1) x tq です。

ビット 7:4 **DTSEG2[3:0]** : サンプルポイント後のデータ時間セグメント

有効な値は 0~15 です。ハードウェアによって使用される値はプログラムされたもので、1ずつ増加します。例 : tBS2 = (DTSEG2 + 1) x tq です。

ビット 3:0 **DSJW[3:0]** : 同期ジャンプ幅

有効な値は 0~15 です。ハードウェアによって使用される値はプログラムされたもので、1ずつ増加します。例 : tSJW = (DSJW + 1) x tq です。

注 : 8 MHz の FDCAN クロックで、0x00000A33 のリセット値が FDCAN を 500 kbit/s の高速ビットレートに設定します。

注 : データフェーズのビットレートは、公称ビットレート以上である必要があります。

54.4.4 FDCAN テストレジスタ (FDCAN_TEST)

ビット CCCR[TEST] を 1 にセットすると、このレジスタへの書込みアクセスが有効になります。CCCR[TEST] がリセットされると、すべてのレジスタ機能がリセット値にセットされます。

Tx ピン FDCANx_TX のループバックモードおよびソフトウェア制御はハードウェアテストモードです。TX を 00 以外にプログラムすると、CAN バスでのメッセージ転送を妨害します。

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	TX[1:0]		LBCK	Res.	Res.	Res.	Res.
								r	rW	rW	rW				

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **RX** : 受信ピン

ピン FDCANx_RX の実際の値を監視します。

0 : CAN バスはドミナントです (FDCANx_RX = 0)。

1 : CAN バスはリセッシブです (FDCANx_RX = 1)。

ビット 6:5 **TX[1:0]** : 送信ピンの制御

00 : リセット値、FDCANx_TX TX は CAN コアによって制御され、CAN ビット時間の最後に更新されます。

01 : サンプルポイントはピン FDCANx_TX で監視できます。

10 : ピン FDCANx_TX でドミナント (0) レベルです。

11 : ピン FDCANx_TX でリセッシブ (1) です。

ビット 4 **LBCK** : ループバックモード

0 : リセット値、ループバックモードは無効です。

1 : ループバックモードは有効です ([パワーダウン \(SLEEP モード\)](#) を参照)。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

54.4.5 FDCAN RAM ウォッチドッグレジスタ (FDCAN_RWD)

RAM ウォッチドッグはメッセージ RAM の READY 出力を監視します。メッセージ RAM アクセスは、RWD[WDC] ビットで設定された値でメッセージ RAM ウォッチドッグカウンタを開始します。

メッセージ RAM が READY 出力を有効にすることで完了成功の信号を送信すると、カウンタは RWD[WDC] ビットで再ロードされます。カウンタが 0 までのカウントダウンを完了するまでにメッセージ RAM からレスポンスがない場合、カウンタは停止し、割込みフラグ IR[WDI] ビットがセットされます。RAM ウォッチドッグカウンタは、fdcan_pclk クロックによってクロック供給されます。

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDV[7:0]								WDC[7:0]							
r	r	r	r	r	r	r	r	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **WDV[7:0]** : ウォッチドッグの値

実際のメッセージ RAM ウォッチドッグカウンタの値です。

ビット 7:0 **WDC[7:0]** : ウォッチドッグの設定

メッセージ RAM ウォッチドッグカウンタの開始値です。00 のリセット値で、カウンタは無効になります。これらは書き込み保護 (P) ビットで、FDCAN_CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

54.4.6 FDCAN CC 制御レジスタ (FDCAN_CCCR)

アドレスオフセット : 0x0018

リセット値 : 0x0000 0001

シングルビットのセットおよびリセットの詳細については、[ソフトウェアの初期化](#)を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NISO	TXP	EFBI	PXHD	Res.	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
rW	rW	rW	rW			rW	rW	rW	rW	rW	rW	r	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **NISO** : 非 ISO 動作

このビットがセットされた場合、FDCAN は Bosch CAN FD Specification V1.0 によって指定された CAN FD フレームフォーマットを使用します。

0 : ISO11898-1 に準拠した CAN FD フレームフォーマットです。

1 : Bosch CAN FD Specification V1.0 に準拠した CAN FD フレームフォーマットです。

ビット 14 **TXP** :

このビットがセットされた場合、FDCAN はフレームの送信成功後、次の送信を開始する前に 2 CAN ビット時間、一時停止します。

0 : 無効

1 : 有効

ビット 13 **EFBI** : バス相互接続中のエッジフィルタ

0 : エッジフィルタは無効です。

1 : 2 つの連続したドミナント tq は、ハード同期のためにエッジを検出する必要があります。

ビット 12 **PXHD** : プロトコル例外処理ディセーブル

0 : プロトコル例外処理は有効です。

1 : プロトコル例外処理は無効です。

ビット 11:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **BRSE** : FDCAN ビットレート切り替え

0 : 送信のビットレート切り替えは無効です。

1 : 送信のビットレート切り替えは有効です。

ビット 8 **FDOE** : FD 動作イネーブル

0 : FD 動作は無効です。

1 : FD 動作は有効です。

ビット 7 **TEST** : テストモードイネーブル

0 : 通常動作、レジスタ TEST はリセット値を保持します。

1 : テストモード、レジスタ TEST への書き込みアクセスは有効です。

ビット 6 **DAR** : 自動再送信の無効化

0 : 正常に送信されなかったメッセージの自動再送信は有効です。

1 : 自動再送信は無効です。

ビット 5 **MON** : バスモニタモード

MON ビットは、CCE および INIT の両方が 1 にセットされているときのみソフトウェアによってセットできます。このビットはいつでもホストによってリセットできます。

0 : バスモニタモードは無効です。

1 : バスモニタモードは有効です。

ビット 4 **CSR** : クロック停止リクエスト

0 : クロック停止はリクエストされていません。

1 : クロック停止がリクエストされました。クロック停止がリクエストされたとき、すべてのペンディング中の転送リクエストが完了して CAN バスがアイドルに達した後、まず INIT がセットされ、次に CSA がセットされます。

ビット 3 **CSA** : クロック停止の確認応答

0 : クロック停止の確認応答はありません。

1 : FDCAN は、APB クロックおよびカーネルクロックを停止することで、パワーダウんにセットできます。

ビット 2 **ASM** : ASM 制限動作モード

制限動作モードは、異なる CAN ビットレートに適応するアプリケーションを対象とします。アプリケーションは異なるビットレートをテストし、有効なフレームを受信すると制限動作モードを終了します。オプションの制限動作モードでは、ノードはデータとリモートフレームを送信および受信でき、有効なフレームに確認応答を付与しますが、アクティブエラーフレームまたはオーバーロードフレームは送信しません。エラー状態やオーバーロード状態になった場合、ドミナントビットを送らずに、バスアイドル状態の発生を待って CAN 通信と再同期します。エラーカウンタはインクリメントされません。ASM ビットは、CCE および INIT の両方が 1 にセットされているときのみソフトウェアによってセットできます。このビットはいつでもソフトウェアによってリセットできます。

0 : 通常 CAN 動作です。

1 : 制限動作モードはアクティブです。

ビット 1 **CCE** : 設定変更イネーブル

0 : CPU による保護された設定レジスタへの書き込みアクセスはありません。

1 : CPU による保護された設定レジスタへの書き込みアクセスがあります (CCCR.INIT = 1 のとき)。

ビット 0 **INIT** : 初期化

0 : 通常動作

1 : 初期化開始

注 : 2つのクロックドメインの同期メカニズムによって、INIT に書き込まれた値を読み戻せるようになるまでに遅延が発生する場合があります。そのため、プログラマは INIT を新しい値にセットする前に、INIT を読み出して INIT に書き込まれた前の値が受理されたことを確認する必要があります。

54.4.7 FDCAN 公称ビットタイミングおよびプリスケアラレジスタ (FDCAN_NBTP)

アドレスオフセット : 0x001C

リセット値 : 0x0600 : 0A03

このレジスタは、CCCR[CCE] および CCCR[INIT] がセットされている場合のみ書込みできます。CAN ビット時間は、4~81 tq の範囲でプログラム可能です。CAN タイムクオンタムは、[1~1024] FDCAN カーネルクロック周期の範囲でプログラム可能です。

$tq = (BRP + 1) \text{ FDCAN クロック周期 } f_{\text{dcn_clk}}$

NTSEG1 は、Prop_Seg と Phase_Seg1 の合計です。NTSEG2 は、Phase_Seg2 です。そのため、ビット時間の長さは (プログラムされた値) [NTSEG1 + NTSEG2 + 3] tq または (機能の値) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq となります。

情報処理時間 (IPT) は 0 で、次のビットのデータはサンプルポイント後の最初のクロックエッジで使用できることを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW[6:0]							NBRP[8:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTSEG1 [7:0]							Res.	NTSEG2 [6:0]							
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:25 **NSJW[6:0]** : 公称 (再) 同期ジャンプ幅

有効な値は 0~127 です。この値のハードウェアによる実際の解釈は、使用される値が 1 でインクリメントされたプログラムされた値です。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

ビット 24:16 **NBRP[8:0]** : ビットレートプリスケアラ

ビット時間のクオンタムを生成するためにオシレータ周波数を分周する値です。ビット時間は、このクオンタムの倍数で構築されます。有効な値は 0~511 です。この値のハードウェアによる実際の解釈は、ここでプログラムされた値より 1 つ大きな値が使用されます。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

ビット 15:8 **NTSEG1[7:0]** : サンプルポイント前の公称時間セグメント

有効な値は 0~255 です。この値のハードウェアによる実際の解釈は、プログラムされた値より 1 つ大きな値が使用されます。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **NTSEG2[6:0]** : サンプルポイント後の公称時間セグメント

有効な値は 0~127 です。この値のハードウェアによる実際の解釈は、プログラムされた値より 1 つ大きな値が使用されます。

注 : 48 MHz の CAN カーネルクロックで、0x06000A03 のリセット値が FDCAN を 3 Mbit/s のビットレートに設定します。

54.4.8 FDCAN タイムスタンプカウンタ設定レジスタ (FDCAN_TSCC)

アドレスオフセット : 0x0020

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCP[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSS[1:0]	
														rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **TCP[3:0]** : タイムスタンプカウンタプリスケアラ

タイムスタンプとタイムアウトカウンタの時間単位を CAN ビット時間の倍数 [1~16] で設定します。

この値のハードウェアによる実際の解釈は、ここでプログラムされた値より 1 つ大きな値が使用されます。

CAN FD モードでは、アービトレーションフェーズとデータフェーズで異なる CAN ビット時間のため、内部タイムスタンプカウンタ TCP は一定の時間ベースを提供しません。CAN FD では、タイムスタンプ生成用の外部カウンタが必要です (TSS = 10)。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TSS[1:0]** : タイムスタンプの選択

00 : タイムスタンプカウンタ値は常に 0x0000 です。

01 : タイムスタンプカウンタ値は TCP に従ってインクリメントされます。

10 : TIM3 の値からの外部タイムスタンプカウンタ (tim3_cnt[0:15])。

11 : 00 と同様です。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書き込みアクセス可能です。

54.4.9 FDCAN タイムスタンプカウンタ値レジスタ (FDCAN_TSCV)

アドレスオフセット : 0x0024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSC[15:0]															
rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W	rc_W

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TSC[15:0]** : タイムスタンプカウンタ

内部／外部タイムスタンプカウンタ値は、フレーム受信の開始時 (Rx および Tx の両方) にキャプチャされます。TSCC[TSS] = 01 の場合、タイムスタンプカウンタは、TSCC[TCP] の設定に応じて CAN ビット時間 [1 ~ 16] の倍数でインクリメントされます。ラップアラウンドによって、割込みフラグ IR[TSW] がセットされます。書き込みアクセスによって、カウンタは 0 にリセットされます。

TSCC.TSS = 10 の場合、TSC は外部タイムスタンプカウンタ値を反映します。書き込みアクセスによる影響はありません。

注： 「ラップアラウンド」とは、TSCV への書込みアクセス以外で、0 以外から 0 へのタイムスタンプカウンタ値の変化を指します。

54.4.10 FDCAN タイムアウトカウンタ設定レジスタ (FDCAN_TOCC)

アドレスオフセット：0x0028

リセット値：0xFFFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TOP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOS[1:0]		ETOC
													rw	rw	rw

ビット 31:16 **TOP[15:0]**：タイムアウト期間

タイムアウトカウンタ（ダウンカウンタ）の開始値です。タイムアウト期間を設定します。

ビット 15:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **TOS[1:0]**：タイムアウト選択

連続モードで動作している場合、TOCV への書込みによって、カウンタは TOCC[TOP] で設定された値にプリセットされ、カウントダウンを続行します。タイムアウトカウンタが FIFO のうち 1 つで制御されている場合、空の FIFO は TOCC[TOP] で設定された値にカウンタをプリセットします。カウントダウンは最初の FIFO 要素が格納されたときに開始します。

00：連続動作

01：Tx イベント FIFO によって制御されるタイムアウト

10：Rx FIFO 0 によって制御されるタイムアウト

11：Rx FIFO 1 によって制御されるタイムアウト

これらは書込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書込みアクセス可能です。

ビット 0 **ETOC**：タイムアウトカウンタ有効化

0：タイムアウトカウンタは無効です。

1：タイムアウトカウンタは有効です。

これは書込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ書込みアクセス可能です。

詳細については、[タイムアウトカウンタ](#)を参照してください。

54.4.11 FDCAN タイムアウトカウンタ値レジスタ (FDCAN_TOCV)

アドレスオフセット : 0x002C

リセット値 : 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOC[15:0]															
rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w	rc_w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TOC[15:0]** : タイムアウトカウンタ

タイムアウトカウンタは、TSCC.TCP の設定に応じて CAN ビット時間 [1 ~ 16] の倍数でデクリメントされます。0 までデクリメントされると、割込みフラグ IR.TOO がセットされ、タイムアウトカウンタが停止します。開始およびリセット/再開条件は、TOCC.TOS で設定します。

54.4.12 FDCAN エラーカウンタレジスタ (FDCAN_ECR)

アドレスオフセット : 0x0040

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEL[7:0]							
								rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RP	REC[6:0]							TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **CEL[7:0]** : CAN エラーログ記録

CAN プロトコルエラーによって送信エラーカウンタまたは受信エラーカウンタがインクリメントされるたびにカウンタがインクリメントされます。CEL の読み出しアクセスすることによってリセットされます。カウンタは 0xFF で停止し、TEC または REC の次のインクリメントによって割込みフラグ IR[ELO] がセットされます。

アクセスタイプは RX で、読み出し時にリセットされます。

ビット 15 **RP** : 受信エラーパッシブ

0 : 受信エラーカウンタが 128 のエラーパッシブレベル未満です。

1 : 受信エラーカウンタが 128 のエラーパッシブレベルに達しています。

ビット 14:8 **REC[6:0]** : 受信エラーカウンタ

受信エラーカウンタの実際の状態、値の範囲は 0~127 です。

ビット 7:0 **TEC[7:0]** : 送信エラーカウンタ

送信エラーカウンタの実際の状態、値の範囲は 0~255 です。

CCCR.ASM がセットされている場合、CAN プロトコルエラーが検出されているが、CEL がインクリメントされているときに、CAN プロトコルコントローラは TEC と REC をインクリメントしません。

54.4.13 FDCAN プロトコルステータスレジスタ (FDCAN_PSR)

アドレスオフセット : 0x0044

リセット値 : 0x0000 0707

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[6:0]						
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PXE	REDL	RBR	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	rc_r	rc_r	rc_r	rc_r	rs	rs	rs	r	r	r	r	r	rs	rs	rs

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **TDCV[6:0]** : トランスミッタ遅延補正值

セカンダリサンプルポイントの位置で、FDCAN_TX から FDCAN_RX までで測定された遅延と TDCR.TDCO の合計で定義されます。SSP の位置は、データフェーズでは送信されたビットとセカンダリサンプルポイントの間の最小タイムクオンタム (mtq) の数です。有効な値は 0~127 mtq です。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **PXE** : プロトコル例外イベント

0 : 最後の読出しアクセス以降に発生したプロトコル例外イベントはありません。

1 : プロトコル例外イベントが発生しました。

ビット 13 **REDL** : 受信 FDCAN メッセージ

このビットは、受入フィルタに関係なくセットされます。

0 : このビットが CPU によってリセットされるまでに、受信した FDCAN メッセージはありません。

1 : EDL フラグがセットされた FDCAN フォーマットのメッセージを受信しました。

アクセスタイプは RX で、読出し時にリセットされます。

ビット 12 **RBR** : 最後に受信した FDCAN メッセージの BRS フラグ

このビットは、受入フィルタに関係なく REDL とともにセットされます。

0 : 最後に受信した FDCAN メッセージには BRS フラグがセットされていません。

1 : 最後に受信した FDCAN メッセージには BRS フラグがセットされています。

アクセスタイプは RX で、読出し時にリセットされます。

ビット 11 **RESI** : 最後に受信した FDCAN メッセージの ESI フラグ

このビットは、受入フィルタに関係なく REDL とともにセットされます。

0 : 最後に受信した FDCAN メッセージには ESI フラグがセットされていません。

1 : 最後に受信した FDCAN メッセージには ESI フラグがセットされています。

アクセスタイプは RX で、読出し時にリセットされます。

ビット 10:8 **DLEC[2:0]** : データの最後のエラーコード

BRS フラグをセットした FDCAN フォーマットフレームのデータフェーズで発生した最後のエラーのタイプです。コーディングは LEC と同じです。このフィールドは、BRS フラグをセットした FDCAN フォーマットフレームがエラーなしで転送 (受信または送信) されたときに 0 にクリアされます。

アクセスタイプは RS で、読出し時にセットされます。

ビット 7 **BO** : Bus_Off ステータス

0 : FDCAN は、Bus_Off ではありません。

1 : FDCAN は、Bus_Off 状態です。

ビット 6 **EW** : 警告ステータス

0 : 両方のエラーカウンタが 96 の Error_Warning 制限未満です。

1 : 少なくとも 1 つのエラーカウンタが 96 の Error_Warning 制限に達しました。

ビット 5 EP : エラーパッシブ

0 : FDCAN は、Error_Active 状態です。通常、バス通信に参加して、エラーを検出した場合にアクティブエラーフラグを送信します。

1 : FDCAN は、Error_Passive 状態です。

ビット 4:3 ACT[1:0] : 動作

モジュールの CAN 通信状態を監視します。

00 : 同期中 : ノードが CAN 通信で同期しています。

01 : アイドル : ノードはレシーバでもトランスミッタでもありません。

10 : レシーバ : ノードはレシーバとして動作しています。

11 : トランスミッタ : ノードはトランスミッタとして動作しています。

ビット 2:0 LEC[2:0] : 最終エラーコード

LEC は、CAN バスで発生した最後のエラーのタイプを示します。このフィールドは、メッセージがエラーなしで転送（受信または送信）されたときに 0 にクリアされます。

000 : エラーなし : 正常な受信または送信によって LEC がリセットされて以降に発生したエラーはありません。

001 : スタッエラー : シーケンス上で許容されていない 6 ビット以上の同一レベルの連続ビットが受信メッセージに含まれている

010 : フォームエラー : 受信フレームの固定フォーマット部分に誤ったフォーマットがあります。

011 : Ack(確認応答)エラー : FDCAN によって送信されたメッセージが他のノードによって確認応答されませんでした。

100 : ビット 1 エラー : メッセージ（アービトラージフィールドを除く）送信中、デバイスがリセッシブレベル（論理値のビット 1）の送信を求めたが、監視されたバスの値がドミナントでした。

101 : ビット 0 エラー : メッセージ（または確認応答ビット、アクティブエラーフラグ、オーバーロードフラグ）送信中、デバイスがドミナントレベル（データまたは識別子ビットの論理値 0）の送信を求めたが、監視されたバスの値がリセッシブでした。Bus_Off リカバリ中、11 のリセッシブビットのシーケンスが監視されるたびに、このステータスがセットされます。これにより、CPU が（バスがドミナントでスタックしていない、または継続的に妨害されていないことを示す）Bus_Off リカバリシーケンスの進行を監視できます。

110 : CRC エラー : 受信メッセージの CRC チェックサムが正しくありません。着信メッセージの CRC が、受信したデータから計算された CRC と一致しません。

111 : 変更なし : プロトコルステータスレジスタの読み出しアクセスによって、LEC が 7 に再初期化されます。LEC が値 '7' を示している場合、プロトコルステータスレジスタの最後の CPU 読み出しアクセス以降に CAN バスイベントは検出されませんでした。

アクセスタイプは RS で、読み出し時にセットされます。

注 : FDCAN フォーマットのフレームが、BRS フラグがセットされたデータフェーズに達したとき、次の CAN イベント（エラーまたは有効なフレーム）は LEC の代わりに FLEC で示されます。FDCAN CRC シーケンスの固定スタッフビットでのエラーは、スタッフエラーではなく、フォームエラーとして示されます。

注 : Bus_Off リカバリシーケンス (CAN Specification Rev. 2.0 または ISO11898-1 参照) は、CCCR[INIT] のセットやリセットで短縮できません。デバイスが Bus_Off になると、それ自体で CCCR.INIT をセットし、すべてのバス動作を停止します。CCCR[INIT] が CPU によってクリアされると、デバイスは通常動作を再開する前に、129 のバスアイドルの発生（129 × 11 の連続リセッシブビット）を待ちます。Bus_Off リカバリシーケンスの最後に、エラー管理カウンタがリセットされます。CCCR[INIT] リセット後の待機時間中、11 のリセッシブビットのシーケンスは監視され、ビット 0 エラーコードは PSR[LEC] に書き込まれ、CPU は CAN バスがドミナントで詰まっていないかどうかや、継続的に妨害されていないかをすぐにチェックし、Bus_Off リカバリシーケンスを監視できます。ECR[REC] はこれらのシーケンスをカウントするために使用されます。

54.4.14 FDCAN トランスミッタ遅延補正レジスタ (FDCAN_TDCR)

アドレスオフセット : 0x0048

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDCO[6:0]							Res.	TDCF[6:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:8 **TDCO[6:0]** : トランスミッタ遅延補正オフセット

FDCAN_TX から FDCAN_RX までに測定された遅延とセカンダリサンプルポイントの距離を定義するオフセット値です。有効な値は 0~127 mtq です。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **TDCF[6:0]** : トランスミッタ遅延補正フィルタのウィンドウ長

SSP の位置の最小値を定義し、より早い SSP の位置につながる FDCAN_RX のドミナントエッジは、トランスミッタ遅延測定では無視されます。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

54.4.15 FDCAN 割込みレジスタ (FDCAN_IR)

リストされた条件が検出された場合、このフラグがセットされます (エッジ区別)。フラグはホストがクリアするまでセットされたままとなります。フラグは、対応するビット位置に 1 を書き込むことによってクリアされます。

0 を書き込んでも、ビットの値は変化しません。ハードリセットでレジスタはクリアされます。IE の設定は、割込みが生成されるかどうかを制御します。ILS の設定は、割込みが信号送信される割込みラインを制御します。

アドレスオフセット : 0x0050

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOO	MRAF	TSW	TEFL	TEFF	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1N	RF0L	RF0F	RF0N
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **ARA** : 予約済みアドレスへのアクセス

0 : 予約済みアドレスへのアクセスは発生していません。

1 : 予約済みアドレスへのアクセスが発生しました。

- ビット 22 **PED** : データフェーズでのプロトコルエラー (データビット時間が使用されます)
- 0 : データフェーズでのプロトコルエラーはありませんでした。
 - 1 : データフェーズでのプロトコルエラーが検出されました (0、7 ではない PSR.DLEC)。
- ビット 21 **PEA** : アービトレーションフェーズでのプロトコルエラー (公称ビット時間が使用されます)
- 0 : アービトレーションフェーズでのプロトコルエラーはありませんでした。
 - 1 : アービトレーションフェーズでのプロトコルエラーが検出されました (0、7 ではない PSR.LEC)。
- ビット 20 **WDI** : ウォッチドッグ割込み
- 0 : メッセージ RAM ウォッチドッグイベントは発生していません。
 - 1 : 不足した READY によるメッセージ RAM ウォッチドッグイベントがありました。
- ビット 19 **BO** : Bus_Off ステータス
- 0 : Bus_Off ステータスは変更されません。
 - 1 : Bus_Off ステータスは変更されます。
- ビット 18 **EW** : 警告ステータス
- 0 : Error_Warning ステータスは変更されません。
 - 1 : Error_Warning ステータスは変更されます。
- ビット 17 **EP** : エラーパッシブ
- 0 : Error_Passive ステータスは変更されません。
 - 1 : Error_Passive ステータスは変更されます。
- ビット 16 **ELO** : エラーログ記録ログ記録オーバーフロー
- 0 : CAN エラーログ記録カウンタはオーバーフローしていません。
 - 1 : CAN エラーログ記録カウンタのオーバーフローが発生しました。
- ビット 15 **TOO** : タイムアウト発生
- 0 : タイムアウトなし
 - 1 : タイムアウト到達
- ビット 14 **MRAF** : メッセージ RAM アクセス失敗
- このフラグは、Rx ハンドラが次の場合にセットされます。
- 次のメッセージのアービトレーションフィールドを受信するまでに、受入フィルタまたは許可メッセージの格納を完了しなかった場合です。この場合、受入フィルタまたはメッセージ格納はアボートされ、Rx ハンドラが次のメッセージの処理を開始します。
 - メッセージをメッセージ RAM に書き込めなかった場合です。この場合、メッセージ格納はアボートされます。
- どちらの場合も、FIFO Put インデックスは更新されません。部分的に格納されたメッセージは、次のメッセージがこの位置に格納される際に上書きされます。
- Tx ハンドラがメッセージ RAM から時間内にメッセージを読み出せなかったときもこのフラグがセットされます。この場合、メッセージ送信はアボートされます。Tx ハンドラアクセス失敗時、FDCAN は制限動作モードに切り替わります ([制限動作モード](#)を参照)。制限動作モードを終了するには、ホスト CPU で CCCR.ASM をリセットする必要があります。
- 0 : メッセージ RAM アクセス失敗は発生していません。
 - 1 : メッセージ RAM アクセス失敗が発生しました。
- ビット 13 **TSW** : タイムスタンプラップアラウンド
- 0 : タイムスタンプカウンタのラップアラウンドはありません。
 - 1 : タイムスタンプカウンタのラップアラウンドが発生しました。
- ビット 12 **TEFL** : Tx イベント FIFO 要素の喪失
- 0 : Tx イベント FIFO 要素の喪失はありません。
 - 1 : Tx イベント FIFO 要素の喪失

- ビット 11 **TEFF** : Tx イベント FIFO はフルです。
0 : Tx イベント FIFO はフルではありません。
1 : Tx イベント FIFO はフルです。
- ビット 10 **TEFN** : Tx イベント FIFO 新規エントリ
0 : Tx イベント FIFO は変更されません。
1 : Tx ハンドラが Tx イベント FIFO 要素に書き込みました。
- ビット 9 **TFE** : Tx FIFO はエンプティです。
0 : Tx FIFO はエンプティではありません。
1 : Tx FIFO はエンプティです。
- ビット 8 **TCF** : 送信キャンセルが終了しました。
0 : 終了した送信キャンセルはありません。
1 : 送信キャンセルが終了しました。
- ビット 7 **TC** : 送信が完了しました。
0 : 完了した送信はありません。
1 : 送信が完了しました。
- ビット 6 **HPM** : 高優先メッセージ
0 : 受信した高優先メッセージはありません。
1 : 高優先メッセージを受信しました。
- ビット 5 **RF1L** : Rx FIFO 1 メッセージ喪失
0 : 喪失した Rx FIFO 1 メッセージはありません。
1 : Rx FIFO 1 メッセージ喪失
- ビット 4 **RF1F** : Rx FIFO 1 はフルです。
0 : Rx FIFO 1 はフルではありません。
1 : Rx FIFO 1 はフルです。
- ビット 3 **RF1N** : Rx FIFO 1 新規メッセージ
0 : Rx FIFO 1 に書き込まれた新規メッセージはありません。
1 : 新規メッセージが Rx FIFO 1 に書き込まれました。
- ビット 2 **RF0L** : Rx FIFO 0 メッセージ喪失
0 : 喪失した Rx FIFO 0 メッセージはありません。
1 : Rx FIFO 0 メッセージ喪失
- ビット 1 **RF0F** : Rx FIFO 0 はフルです。
0 : Rx FIFO 0 はフルではありません。
1 : Rx FIFO 0 はフルです。
- ビット 0 **RF0N** : Rx FIFO 0 新規メッセージ
0 : Rx FIFO 0 に書き込まれた新規メッセージはありません。
1 : 新規メッセージが Rx FIFO 0 に書き込まれました。

54.4.16 FDCAN 割込み有効レジスタ (FDCAN_IE)

割込み有効レジスタの設定によって、割込みレジスタでのどのステータス変更が割込みラインで信号送信されるかが決まります。

アドレスオフセット : 0x0054

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1NE	RF0LE	RF0FE	RF0NE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **ARAE** : 予約済みアドレスへのアクセスイネーブル

ビット 22 **PEDE** : データフェーズでのプロトコルエラーイネーブル

ビット 21 **PEAE** : アービトレーションフェーズでのプロトコルエラーイネーブル

ビット 20 **WDIE** : ウォッチドッグの割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 19 **BOE** : Bus_Off ステータス

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 18 **EWE** : 警告ステータスの割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 17 **EPE** : エラーパッシブ割込み有効化

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 16 **ELOE** : エラーログ記録オーバーフローの割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 15 **TOOE** : タイムアウト発生の割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 14 **MRAFE** : メッセージ RAM アクセス失敗の割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 13 **TSWE** : タイムスタンプ・ラップアラウンドの割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

ビット 12 **TEFLE** : Tx イベント FIFO 要素喪失の割込みイネーブル

0 : 割込みは無効です。

1 : 割込みは有効です。

- ビット 11 **TEFFE** : Tx イベント FIFO フルの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 10 **TEFNE** : Tx イベント FIFO 新規エントリの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 9 **TFEE** : Tx FIFO エンプティ割込み有効
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 8 **TCFE** : 送信キャンセル終了の割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 7 **TCE** : 転送完了割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 6 **HPME** : 高優先メッセージの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 5 **RF1LE** : Rx FIFO 1メッセージ喪失の割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 4 **RF1FE** : Rx FIFO 1フルの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 3 **RF1NE** : Rx FIFO 1新規メッセージの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 2 **RF0LE** : Rx FIFO 0メッセージ喪失の割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 1 **RF0FE** : Rx FIFO 0フルの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。
- ビット 0 **RF0NE** : Rx FIFO 0新規メッセージの割込みイネーブル
0 : 割込みは無効です。
1 : 割込みは有効です。

54.4.17 FDCAN 割込みライン選択レジスタ (FDCAN_ILS)

このレジスタは、割込みレジスタの割込みフラグの特定のグループによって生成される割込みを、2つのモジュール割込みラインのうちの1つに割り当てます。割込み生成には、それぞれの割込みラインを ILE[EINT0] および ILE[EINT1] で有効にする必要があります。

アドレスオフセット : 0x0058

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERR	BERR	MISC	TFERR	MSG	RXFIF O1	RXFIF O0
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **PERR** : 以下の割込みをグループ化するプロトコルエラー

ARAL : 予約済みアドレスへのアクセスのライン
 PEDL : データフェーズでのプロトコルエラーのライン
 PEAL : アービトレーションフェーズでのプロトコルエラーのライン
 WDIL : ウォッチドッグの割込みライン
 BOL : Bus_Off ステータス
 EWL : 警告ステータスの割込みライン

ビット 5 **BERR** : 以下の割込みをグループ化するビットおよびラインのエラー

EPL : エラーパッシブの割込みライン
 ELOL : エラーログ記録オーバーフローの割込みライン

ビット 4 **MISC** : 以下の割込みを再グループ化する割込み

TOOL : タイムアウト発生の割込みライン
 MRAFL : メッセージ RAM アクセス失敗の割込みライン
 TSWL : タイムスタンプラップアラウンドの割込みライン

ビット 3 **TFERR** : 以下の割込みをグループ化する Tx FIFO ERROR

TEFLL : Tx イベント FIFO 要素喪失の割込みライン
 TEFFL : Tx イベント FIFO フルの割込みライン
 TEFNL : Tx イベント FIFO 新規エントリの割込みライン
 TFEL : Tx FIFO エンプティの割込みライン

ビット 2 **MSG** : 以下の割込みをグループ化するステータスメッセージビット

TCFL : 送信キャンセル終了の割込みライン
 TCL : 送信完了の割込みライン
 HPML : 高優先メッセージの割込みライン

ビット 1 **RXFIFO1** : 以下の割込みをグループ化する RX FIFO ビット

RF1LL : Rx FIFO 1メッセージ喪失の割込みライン
 RF1FL : Rx FIFO 1フルの割込みライン
 RF1NL : Rx FIFO 1新規メッセージの割込みライン

ビット 0 **RXFIFO0** : 以下の割込みをグループ化する RX FIFO ビット

RF0LL : Rx FIFO 0メッセージ喪失の割込みライン
 RF0FL : Rx FIFO 0フルの割込みライン
 RF0NL : Rx FIFO 0新規メッセージの割込みライン

54.4.18 FDCAN 割込みライン有効レジスタ (FDCAN_ILE)

CPU への 2 つの割込みラインは、それぞれビット EINT0 および EINT1 をプログラムすることで有効化／無効化できます。

アドレスオフセット : 0x005C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EINT1** : 割込みライン 1 有効化

0 : 割込みライン fdcan_intr0_it は無効です。

1 : 割込みライン fdcan_intr0_it は有効です。

ビット 0 **EINT0** : 割込みライン 0 有効化

0 : 割込みライン fdcan_intr1_it は無効です。

1 : 割込みライン fdcan_intr1_it は有効です。

54.4.19 FDCAN グローバルフィルタ設定レジスタ (FDCAN_RXGFC)

メッセージ ID フィルタのグローバル設定です。グローバルフィルタ設定は、[図 778](#) および [図 779](#) に示されたとおり、標準メッセージおよび拡張メッセージのフィルタパスを制御します。

アドレスオフセット : 0x0080

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	LSE[3:0]				Res.	Res.	Res.	LSS[4:0]				
				rw	rw	rw	rw				rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	F0OM	F1OM	Res.	Res.	ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
						rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **LSE[3:0]** : リストサイズ拡張

0 : 拡張メッセージ ID フィルタはありません。

1 から 8 : 拡張メッセージ ID フィルタ要素の数です。

>8 : 8 を超える値は 8 として解釈されます。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **LSS[4:0]** : リストサイズ標準

0 : 標準メッセージ ID フィルタはありません。

1 から 28 : 標準メッセージ ID フィルタ要素の数です。

>28 : 28 を超える値は 28 として解釈されます。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **F00M** : FIFO 0 動作モード (上書きまたはブロック)

これは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 8 **F10M** : FIFO 1 動作モード (上書きまたはブロック)

これは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ANFS[1:0]** : 一致しないフレーム (標準) の受入

フィルタリストの要素と一致しない 11 ビット ID を持つ受信メッセージの処理方法を定義します。

00 : Rx FIFO 0 で受入

01 : Rx FIFO 1 で受入

10 : 拒否

11 : 拒否

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 3:2 **ANFE[1:0]** : 一致しないフレーム (拡張) の受入

フィルタリストの要素と一致しない 29 ビット ID を持つ受信メッセージの処理方法を定義します。

00 : Rx FIFO 0 で受入

01 : Rx FIFO 1 で受入

10 : 拒否

11 : 拒否

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 1 **RRFS** : リモートフレーム (標準) の拒否

0 : 11 ビット標準 ID を持つリモートフレームをフィルタします。

1 : 11 ビット標準 ID を持つすべてのリモートフレームを拒否します。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 0 **RRFE** : リモートフレーム (拡張) の拒否

0 : 29 ビット標準 ID を持つリモートフレームをフィルタします。

1 : 29 ビット標準 ID を持つすべてのリモートフレームを拒否します。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

54.4.20 FDCAN 拡張 ID およびマスクレジスタ (FDCAN_XIDAM)

アドレスオフセット : 0x0084

リセット値 : 0x1FFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EIDM[28:16]												
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIDM[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:0 **EIDM[28:0]** : 拡張 ID マスク

拡張フレームの受入フィルタでは、拡張 ID AND マスクは受信フレームのメッセージ ID と AND でつながれます。SAE J1939 での 29 ビット ID のマスクングを対象とします。すべてのビットのリセット値を 1 にすると、マスクはアクティブではなくなります。

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

54.4.21 FDCAN 高優先メッセージステータスレジスタ (FDCAN_HPMS)

このレジスタは、メッセージ ID フィルタ要素が優先度イベント一致を生成するように設定するたびに更新されます。これを使用して、着信高優先メッセージのステータスを監視し、これらのメッセージへの高速アクセスが可能になります。

アドレスオフセット : 0x0088

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLST	Res.	Res.	FIDX[4:0]				MSI[1:0]		Res.	Res.	Res.	BIDX[2:0]			
r			r	r	r	r	r	r				r	r	r	

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **FLST** : フィルタリスト

一致フィルタ要素のフィルタリストを示します。

0 : 標準フィルタリスト

1 : 拡張フィルタリスト

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **FIDX[4:0]** : フィルタインデックス

一致フィルタ要素のインデックスです。範囲は、0~RXGFC[LSS] - 1 または RXGFC[LSE] - 1 です。

ビット 7:6 **MSI[1:0]** : メッセージストレージインジケータ

- 00 : 選択された FIFO がありません。
- 01 : FIFO オーバーランです。
- 10 : メッセージが FIFO 0 に格納されます。
- 11 : メッセージが FIFO 1 に格納されます。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **BIDX[2:0]** : バッファインデックス

メッセージが格納された Rx FIFO 要素のインデックスです。MSI[1] = 1 の場合のみ有効です。

54.4.22 FDCAN Rx FIFO 0 ステータスレジスタ (FDCAN_RXF0S)

アドレスオフセット : 0x0090

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RF0L	F0F	Res.	Res.	Res.	Res.	Res.	Res.	F0PI [1:0]	
						r	r							r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	F0GI [1:0]		Res.	Res.	Res.	Res.	F0FL [3:0]			
						r	r					r	r	r	r

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **RF0L** : Rx FIFO 0 メッセージ喪失

このビットは割り込みレジスタ IR[RF0L] のコピーです。IR[RF0L] がリセットされると、このビットもリセットされます。

- 0 : 喪失した Rx FIFO 0 メッセージはありません。
- 1 : Rx FIFO 0 メッセージを喪失しました。サイズ 0 の Rx FIFO 0 への書き込み試行後にもセットされます。

ビット 24 **F0F** : Rx FIFO 0 はフルです。

- 0 : Rx FIFO 0 はフルではありません。
- 1 : Rx FIFO 0 はフルです。

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **F0PI[1:0]** : Rx FIFO 0 プットインデックス

Rx FIFO 0 書き込みインデックスポインタで、範囲は 0~2 です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **F0GI[1:0]** : Rx FIFO 0 取得インデックス

Rx FIFO 0 読み出しインデックスポインタで、範囲は 0~2 です。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **F0FL[3:0]** : Rx FIFO 0 フィルレベル

Rx FIFO 0 に格納される要素の数で、範囲は 0~3 です。

54.4.23 CAN Rx FIFO 0 確認応答レジスタ (FDCAN_RXF0A)

アドレスオフセット : 0x0094

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0AI [2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **F0AI[2:0]** : Rx FIFO 0 確認応答インデックス

ホストが Rx FIFO 0 からメッセージまたは一連のメッセージを読み出した後、Rx FIFO 0 から読み出された最後の要素のバッファインデックスを F0AI に書き込む必要があります。これにより、Rx FIFO 0 Get インデックス RXF0S[F0GI] が F0AI + 1 にセットされ、FIFO 0 フィルレベル RXF0S[F0FL] が更新されます。

54.4.24 FDCAN Rx FIFO 1 ステータスレジスタ (FDCAN_RXF1S)

アドレスオフセット : 0x0098

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	Res.	Res.	Res.	Res.	F1PI [1:0]	
						r	r							r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	F1GI [1:0]		Res.	Res.	Res.	Res.	F1FL [3:0]			
						r	r					r	r	r	r

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **RF1L** : Rx FIFO 1 メッセージ喪失

このビットは割込みレジスタ IR[RF1L] のコピーです。IR[RF1L] がリセットされると、このビットもリセットされます。

0 : 喪失した Rx FIFO 1 メッセージはありません。

1 : Rx FIFO 1 メッセージを喪失しました。サイズ 0 の Rx FIFO 1 への書き込み試行後にもセットされます。

ビット 24 **F1F** : Rx FIFO 1 はフルです。

0 : Rx FIFO 1 はフルではありません。

1 : Rx FIFO 1 はフルです。

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **F1PI[1:0]** : Rx FIFO 1 プット インデックス

Rx FIFO 1 書き込みインデックスポインタで、範囲は 0~2 です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **F1GI[1:0]** : Rx FIFO 1 取得インデックス

Rx FIFO 1 読み出しインデックスポインタで、範囲は 0~2 です。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **F1FL[3:0]** : Rx FIFO 1 フィルレベル

Rx FIFO 1 に格納される要素の数で、範囲は 0~3 です。

54.4.25 FDCAN Rx FIFO 1 確認応答レジスタ (FDCAN_RXF1A)

アドレスオフセット : 0x009C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1AI [2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **F1AI[2:0]** : Rx FIFO 1 確認応答インデックス

ホストが Rx FIFO 1 からメッセージまたは一連のメッセージを読み出した後、Rx FIFO 1 から読み出された最後の要素のバッファインデックスを F1AI に書き込む必要があります。これにより、Rx FIFO 1 Get インデックス RXF1S[F1GI] が F1AI + 1 にセットされ、FIFO 1 フィルレベル RXF1S[F1FL] が更新されます。

54.4.26 FDCAN Tx バッファ設定レジスタ (FDCAN_TXBC)

アドレスオフセット : 0x00C0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TFQM** : Tx FIFO/キューモード

0 : Tx FIFO 動作

1 : Tx キュー動作

これは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

ビット 23:0 予約済みであり、リセット値に保持する必要があります。

54.4.27 FDCAN Tx FIFO／キューステータスレジスタ (FDCAN_TXFQS)

Tx FIFO／キューステータスは、レジスタ TXBRP に示されたペンディング中の Tx リクエストに関連します。そのため、追加／キャンセルリクエストの効果は、実行中の Tx スキャン (TXBRP がまだ更新されていない) によって遅延する場合があります。

アドレスオフセット : 0x00C4

リセット値 : 0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQF	Res.	Res.	Res.	TFQPI[1:0]	
										r				r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TFGI[1:0]		Res.	Res.	Res.	Res.	Res.	TFFL[2:0]		
						r	r						r	r	r

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TFQF** : Tx FIFO／キューフル

0 : Tx FIFO／キューはフルではない

1 : Tx FIFO／キューフル

ビット 20:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **TFQPI[1:0]** : Tx FIFO／キュー プット インデックス

Tx FIFO／キュー書き込みインデックスポインタで、範囲は 0～3 です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **TFGI[1:0]** : Tx FIFO Get インデックス

Tx FIFO 読出しインデックスポインタで、範囲は 0～3 です。Tx キュー動作が設定された場合 (TXBC.TFQM = 1)、0 で読み出されます。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TFFL[2:0]** : Tx FIFO フリーレベル

連続するフリー Tx FIFO 要素の数は TFGI から開始し、範囲は 0～3 です。Tx キュー動作が設定された場合 (TXBC[TFQM] = 1)、0 で読み出されます。

54.4.28 FDCAN Tx バッファリクエストペンディングレジスタ (FDCAN_TXBRP)

アドレスオフセット : 0x00C8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRP[2:0]		
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TRP[2:0]** : 送信リクエストペンディングがあります。

各 Tx バッファには独自の送信リクエストペンディングビットがあります。このビットは TXBAR 経由でセットされます。リクエストされた送信が完了した後か、レジスタ TXBCR 経由でキャンセルされた後に、ビットがリセットされます。

TXBRP ビットがセットされた後、Tx スキャンが開始され、最高優先度を持つペンディング中の Tx リクエスト（一番小さいメッセージ ID の Tx バッファ）をチェックします。

キャンセルリクエストで、レジスタ TXBRP の対応する送信リクエストペンディングビットがリセットされます。キャンセルがリクエストされたときに送信をすでに開始していた場合、送信が成功したかどうかに関係なく、送信の最後に実行されます。対応する TXBRP ビットがリセットされた直後に、キャンセルリクエストビットがリセットされます。

キャンセルがリクエストされた後、終了したキャンセルは次のタイミングで TXBCF 経由で信号送信されます。

対応する TXBTO ビットとともに送信が成功した後

キャンセル時点で送信がまだ開始されていなかったとき

ロストアービトレーションによって送信がアボートされたとき

フレーム送信中にエラーが発生したとき

DAR モードでは、すべての送信が成功しなかった場合、自動的にキャンセルされます。すべての成功しなかった送信に対応する TXBCF ビットがセットされます。

0 : 送信リクエストペンディングはありません。

1 : 送信リクエストペンディングがあります。

注 : Tx スキャンが進行中のときにセットされた TXBRP ビットは、この特定の Tx スキャン中は考慮されません。このような Tx バッファに対してキャンセルがリクエストされた場合、この追加リクエストはすぐにキャンセルされ、対応する TXBRP ビットはリセットされます。

54.4.29 FDCAN Tx バッファ追加リクエストレジスタ (FDCAN_TXBAR)

アドレスオフセット : 0x00CC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR[2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **AR[2:0]** : 追加リクエスト

各 Tx バッファには独自の追加リクエストビットがあります。1 を書き込むと、対応する追加リクエストビットがセットされ、0 を書き込んで影響はありません。これにより、ホストは複数の Tx バッファに対する送信リクエストを TXBAR への 1 回の書込みでセットできます。実行中の Tx スキャンがないときはビットがすぐにリセットされ、それ以外の場合は Tx スキャンプロセスが完了するまでビットがセットされたままとなります。

0 : 追加された送信リクエストはありません。

1 : 送信リクエストが追加されました。

注 : 追加リクエストがペンディング中の送信リクエストを持つ Tx バッファに適用される（対応する TXBRP がすでにセットされている）場合、リクエストは無視されます。

54.4.30 FDCAN Tx バッファキャンセルリクエストレジスタ (FDCAN_TXBCR)

アドレスオフセット : 0x00D0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CR[2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **CR[2:0]** : キャンセルリクエスト

各 Tx バッファには独自のキャンセルリクエストビットがあります。1 を書き込むと、対応する CR ビットがセットされ、0 を書き込んでも影響はありません。

これにより、ホストは複数の Tx バッファに対するキャンセルリクエストを TXBCR への 1 回の書込みでセットできます。対応する TXBRP ビットがリセットされるまで、ビットはセットされたままとなります。

0 : ペンディング中のキャンセルはありません。

1 : キャンセルがペンディングされています。

54.4.31 FDCAN Tx バッファ送信発生レジスタ (FDCAN_TXBTO)

アドレスオフセット : 0x00D4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TO[2:0]		
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TO[2:0]** : 送信発生

各 Tx バッファには独自の TO ビットがあります。送信成功後に対応する TXBRP ビットがクリアされたとき、ビットがセットされます。レジスタ TXBAR の対応するビットに 1 を書き込むことによって新しい送信がリクエストされると、ビットがリセットされます。

0 : 発生した送信はありません。

1 : 送信が発生しました。

54.4.32 FDCAN Tx バッファキャンセル終了レジスタ (FDCAN_TXBCF)

アドレスオフセット : 0x00D8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CF[2:0]		
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **CF[2:0]** : キャンセル終了

各 Tx バッファには独自の CF ビットがあります。キャンセルが TXBCR ビットによってリクエストされた後に対応する TXBRP ビットがクリアされたとき、ビットがセットされます。対応する TXBRP ビットがキャンセル時点でセットされなかった場合、CF がすぐにセットされます。レジスタ TXBAR の対応するビットに 1 を書き込むことによって新しい送信がリクエストされると、ビットがリセットされます。

0 : 送信バッファキャンセルはありません。

1 : 送信バッファキャンセルが終了しました。

54.4.33 FDCAN Tx バッファ送信割込み有効レジスタ (FDCAN_TXBTIE)

アドレスオフセット : 0x00DC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE[2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TIE[2:0]** : 送信割込みは有効です。

各 Tx バッファには独自の TIE ビットがあります。

0 : 送信割込みは無効です。

1 : 送信割込みは有効です。

54.4.34 FDCAN Tx バッファキャンセル終了割込み有効レジスタ (FDCAN_TXBCIE)

アドレスオフセット : 0x00E0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFIE[2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **CFIE[2:0]** : キャンセル終了の割込みイネーブル

各 Tx バッファには独自の CFIE ビットがあります。

0 : キャンセル終了割込みは無効です。

1 : キャンセル終了割込みは有効です。

54.4.35 FDCAN Tx イベント FIFO ステータスレジスタ (FDCAN_TXEFS)

アドレスオフセット : 0x00E4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	Res.	Res.	Res.	EFPI[1:0]	
						r	r							r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EFGI[1:0]		Res.	Res.	Res.	Res.	Res.	EFFL[2:0]		
						r	r						r	r	r

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **TEFL** : Tx イベント FIFO 要素の喪失

このビットは割込みレジスタ IR[TEFL] のコピーです。IR[TEFL] がリセットされると、このビットもリセットされます。

0 : Tx イベント FIFO 要素の喪失はありません。

1 : Tx イベント FIFO 要素を喪失しました。サイズ 0 の Tx イベント FIFO への書き込み試行後にもセットされます。

ビット 24 **EFF** : イベント FIFO フル

0 : Tx イベント FIFO はフルではありません。

1 : Tx イベント FIFO はフルです。

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **EFPI[1:0]** : イベント FIFO Put インデックス

Tx イベント FIFO 書き込みインデックスポインタで、範囲は 0~3 です。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **EFGI[1:0]** : イベント FIFO Get インデックス

Tx イベント FIFO 読み出しインデックスポインタで、範囲は 0~3 です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **EFFL[2:0]** : イベント FIFO フィルレベル

Tx イベント FIFO に格納される要素の数で、範囲は 0~3 です。

54.4.36 FDCAN Tx イベント FIFO 確認応答レジスタ (FDCAN_TXEFA)

アドレスオフセット : 0x00E8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EFAI[1:0]	
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **EFAI[1:0]** : イベント FIFO 確認応答インデックス

ホストが Tx イベント FIFO から要素または一連の要素を読み出した後、Tx イベント FIFO から読み出された最後の要素のインデックスを EFAI に書き込む必要があります。これにより、Tx イベント FIFO Get インデックス TXEFS[EFGI] が EFAI + 1 にセットされ、FIFO 0 フィルレベル TXEFS[EFFL] が更新されます。

54.4.37 FDCAN CFG クロック分周回路レジスタ (FDCAN_CKDIV)

アドレスオフセット : 0x0100

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PDIV[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PDIV[3:0]** : 入力クロック分周回路

APB クロックは、CAN サブシステムで使用する前に分周できます。レートは、分周回路出力クロックを使用して計算する必要があります。

0000 : 分周比 1

0001 : 分周比 2

0010 : 分周比 4

0011 : 分周比 6

0100 : 分周比 8

0101 : 分周比 10

0110 : 分周比 12

0111 : 分周比 14

1000 : 分周比 16

1001 : 分周比 18

1010 : 分周比 20

1011 : 分周比 22

1100 : 分周比 24

1101 : 分周比 26

1110 : 分周比 28

1111 : 分周比 30

これらは書き込み保護 (P) ビットで、CCCR レジスタのビット 1 [CCE] およびビット 0 [INIT] が 1 にセットされているときのみ、このビットによる書き込みアクセスが可能です。

54.4.38 FDCAN レジスタマップ

表 589. FDCAN レジスタのマップとリセット値⁽¹⁾

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	FDCAN_CREL	REL[3:0]				STEP[3:0]				SUBSTEP [3:0]				YEAR[3:0]				MON[7:0]							DAY[7:0]								
	リセット値	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	1	0	0	0	1	0
0x0004	FDCAN_ENDN	ETV[31:0]																															
	リセット値	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1
0x0008	予約済みです。	予約済みです。																															
0x000C	FDCAN_DBTP	Res	Res	Res	Res	Res	Res	Res	Res	TDC	Res	Res	DBRP[4:0]				Res	Res	Res	DTSEG1 [4:0]				DTSEG2 [3:0]			DSJW[3:0]						
	リセット値									0	0	0	0	0	0	0	0				0	1	0	1	0	0	0	1	1	0	0	1	1
0x0010	FDCAN_TEST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RX	TX [1:0]		LBCK	Res	Res	Res	Res	
	リセット値																								0	0	0	0					
0x0014	FDCAN_RWD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDV[7:0]							WDC[7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	FDCAN_CCCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NISO	TXP	EFBI	PXHD	Res	Res	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INT
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x001C	FDCAN_NBTP	NSJW[6:0]						NBRP[8:0]								NTSEG1 [7:0]							Res	NTSEG2 [6:0]									
	リセット値	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1
0x0020	FDCAN_TSCC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TCP[3:0]				Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TSS [1:0]
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	FDCAN_TSCV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TSC[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	FDCAN_TOCC	TOP[15:0]															Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TOS [1:0]	ETOC		
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	FDCAN_TOCV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TOC[15:0]															
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x0030 - 0x003C	予約済みです。	予約済みです。																															
0x0040	FDCAN_ECR	Res	Res	Res	Res	Res	Res	Res	Res	CEL[7:0]							RP	REC[6:0]						TEC[7:0]									
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	FDCAN_PSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDCV[6:0]							Res	PXE	REDL	RBRRES1	RESI	DLEC[2:0]		BO			EW	EP	ACT[1:0]		LEC[2:0]	
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

表 589. FDCAN レジスタのマップとリセット値⁽¹⁾ (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0048	FDCAN_TDCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDCO[6:0]						Res	TDCF[6:0]								
	リセット値																		0	0	0	0	0	0		0	0	0	0	0	0	0	0
0x004C	予約済みです。	予約済みです。																															
0x0050	FDCAN_IR	Res	Res	Res	Res	Res	Res	Res	Res	ARA	PED	PEA	WDI	BO	EW	EP	ELO	TOO	MRAF	TSW	TEFL	TEFF	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1N	RFOL	RF0F	RF0N
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	FDCAN_IE	Res	Res	Res	Res	Res	Res	Res	Res	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1NE	RFOLE	RF0FE	RF0NE
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0058	FDCAN_ILS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PERR	BERR	MISC	TFERR	SMMSG	RXFIFO1	RXFIFO0
	リセット値	0																									0	0	0	0	0	0	0
0x005C	FDCAN_ILE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EINT1	EINT0	
	リセット値																														0	0	
0x0060 - 0x007C	予約済みです。	予約済みです。																															
0x0080	FDCAN_RXGFC	Res	Res	Res	Res	LSE[3:0]				Res	Res	LSS[4:0]				Res	Res	Res	Res	Res	Res	Res	Res	F0OM	F1OM	Res	Res	ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
	リセット値					0	0	0	0				0	0	0	0	0							0	0			0	0	0	0	0	0
0x0084	FDCAN_XIDAM	Res	Res	Res	EIDM[28:0]																												
	リセット値				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0088	FDCAN_HPMS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FLST	Res	Res	FIDX[4:0]				MSI [1:0]		Res	Res	Res	Res	BIDX [2:0]			
	リセット値																0			0	0	0	0	0	0	0	0				0	0	0
0x0090	FDCAN_RXF0S	Res	Res	Res	Res	Res	Res	RFOL	FOF	Res	Res	Res	Res	Res	Res	F0PI [1:0]	Res	Res	Res	Res	Res	Res	Res	F0GI [1:0]	Res	Res	Res	Res	F0FL [3:0]				
	リセット値							0	0							0	0							0	0					0	0	0	0
0x0094	FDCAN_RXF0A	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	F0AI [2:0]		
	リセット値																													0	0	0	
0x0098	FDCAN_RXF1S	Res	Res	Res	Res	Res	Res	RF1L	F1F	Res	Res	Res	Res	Res	Res	F1PI [1:0]	Res	Res	Res	Res	Res	Res	Res	F1GI [1:0]	Res	Res	Res	Res	F1FL [3:0]				
	リセット値							0	0							0	0							0	0					0	0	0	0
0x009C	FDCAN_RXF1A	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	F1AI [2:0]		
	リセット値																													0	0	0	
0x00A0 - 0x00BC	予約済みです。	予約済みです。																															

表 589. FDCAN レジスタのマッピングとリセット値⁽¹⁾ (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00C0	FDCAN _TXBC	Res	Res	Res	Res	Res	Res	Res	TFQM	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値								0	0																								
0x00C4	FDCAN _TXFQS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TFQF	Res	Res	Res	Res	TFQPI[1:0]	Res	Res	Res	Res	Res	Res	Res	TFGI[1:0]	Res	Res	Res	Res	Res	Res	TFFL[2:0]		
	リセット値											0					0	0						0	0						0	1	1	
0x00C8	FDCAN _TXBRP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TRP2	TRP1	TRP0	
	リセット値																													0	0	0		
0x00CC	FDCAN _TXBAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AR2	AR1	AR0	
	リセット値																													0	0	0		
0x00D0	FDCAN _TXBCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CR2	CR1	CR0	
	リセット値																													0	0	0		
0x00D4	FDCAN _TXBTO	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TO2	TO1	TO0	
	リセット値																													0	0	0		
0x00D8	FDCAN _TXBCF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CF2	CF1	CF0	
	リセット値																													0	0	0		
0x00DC	FDCAN _TXBTIE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIE2	TIE1	TIE0	
	リセット値																													0	0	0		
0x00E0	FDCAN _TXBCIE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CFIE2	CFIE1	CFIE0
	リセット値																													0	0	0		
0x00E4	FDCAN _TXEFS	Res	Res	Res	Res	Res	Res	TEFL	EFF	Res	Res	Res	Res	Res	Res	EFPI[1:0]	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EFFL[2:0]	
	リセット値							0	0							0	0							0	0						0	0	0	
0x00E8	FDCAN _TXEFA	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EFAI[1:0]	
	リセット値																														0	0		
0x0100	FDCAN _CKDIV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PDIV[3:0]				
	リセット値																													0	0	0	0	

1. R = 読み出し、S = 読み出し時にセット、X = 読み出し時にリセット、W = 書き込み、P = 書き込み保護、p = セット保護、C = 書き込み時にクリア/リセット

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

55 ユニバーサルシリアルバスフルスピードホスト/デバイスインタフェース (USB)

55.1 概要

USB ペリフェラルは、フルスピードの USB 2.0 バスと APB2 バスの間のインタフェースを実装します。
USB サスペンド/レジュームがサポートされるため、節電のためにデバイスのクロックを停止できます。

55.2 USB の主な機能

- USB 仕様バージョン 2.0 フルスピード準拠
- ホストモードとデバイスモードの両方に対応
- 1 から 8 まで設定可能なエンドポイント数
- 2048 バイトの専用パケットバッファメモリ (SRAM)
- 巡回冗長検査 (CRC) の生成/チェック、NRZI (Non-return-to-zero Inverted) 方式のエンコード/デコード、およびビットスタッフィング
- アイソクロナス転送サポート
- ダブルバッファバルク/アイソクロナスエンドポイント/チャネルに対応
- USB サスペンド/レジューム操作
- フレームロッククロックパルス生成
- USB 2.0 リンク電源管理サポート (デバイスモードのみ)
- バッテリ充電仕様リビジョン 1.2 のサポート (デバイスモードのみ)
- USB 接続/切断機能 (USB DP ライン上の制御可能な組み込みプルアップレジスタ)

55.3 USB の実装

表 590 に、デバイスでの USB 実装を示します。

表 590. STM32H563/H573および STM32H562 USB の実装

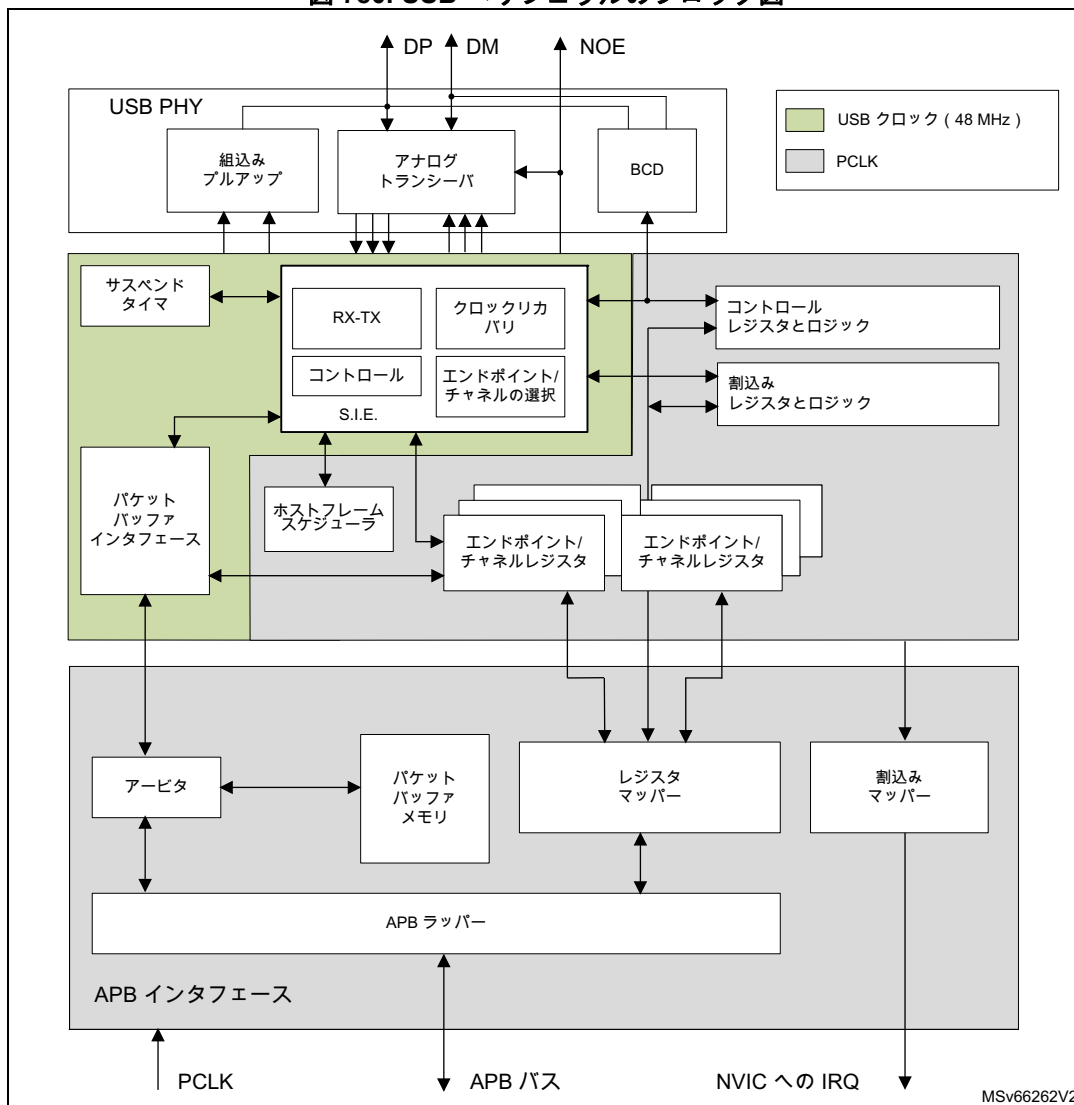
USB の機能 ⁽¹⁾	USB
ホストモード	X
エンドポイント数	8
専用パケットバッファメモリ SRAM のサイズ	2048 バイト
専用パケットバッファメモリ SRAM のアクセス方式	32 ビット
デバイス内での USB 2.0 リンク電源管理 (LPM) のサポート	X
デバイス用のバッテリ充電検出 (BCD) のサポート	X
USB_DP ライン上の組み込みプルアップレジスタ	X

1. X = サポート。

55.4 USB の機能説明

図 780 に、USB ペリフェラルのブロック図を示します。

図 780. USB ペリフェラルのブロック図



概要とデバイスモードの機能

USB ペリフェラルを使用すると、マイクロコントローラによって実装される機能と、ホスト PC と USB デバイスを兼ねる外部 USB 機能との間の USB 準拠接続が可能になります。外部 USB ホストまたはデバイスとシステムメモリ間のデータ転送は、USB ペリフェラルによって直接アクセスされる専用パケットバッファメモリを通じて行われます。この専用メモリのサイズは 2048 バイトであり、最大 16 の単方向または 8 つの双方向エンドポイントを使用できます。USB ペリフェラルは外部 USB ホストまたはデバイスとインタフェースして、トークンパケットを検出し、データの送受信を処理し、USB 標準の要求に従ってハンドシェイクパケットを処理します。トランザクションのフォーマット化は、CRC の生成とチェックも含めて、ハードウェアによって行われます。

各エンドポイント/チャネルは、エンドポイント/チャネル関連のメモリ領域の位置、その大きさ、または送信バイト数を示すバッファ記述ブロックに関連付けられています。有効な機能/エンドポイント

のペアのトークンが USB ペリフェラルによって検出されると、関連するデータ転送が行われます（必要であり、エンドポイント/チャンネルが設定されている場合）。USB ペリフェラルによってバッファされたデータは、内部 16 ビットレジスタに配置され、専用バッファへのメモリアクセスが行われます。すべてのデータが転送されると、必要な場合には、転送方向に従って、適切なハンドシェイクパッケージが USB 経由で生成され、または転送の方向によって期待されます。

トランザクションの終了時には、エンドポイント/チャンネル固有の割込みが生成され、ステータスレジスタを読み出すか、別の割込み応答ルーチンを使用します。マイクロコントローラは、以下を決めることができます。

- 処理するエンドポイント/チャンネル
- エラーが発生した場合に行うトランザクションのタイプ（ビットスタッフィング、フォーマット、CRC、プロトコル、ACK の欠如、オーバーラン/アンダーランなど）

アイソクロナス転送と高スループットバルク転送に対する特別なサポートが提供され、ダブルバッファ使用が実装されることにより、USB ペリフェラルが使用できるバッファが常にあり、マイクロコントローラ用のバッファが別にあります。

レジスタ USB_ISTR の特別ビット THR512 を使用すると、512 バイトの通知をバッファとの間で送受信することができます。このビットは、データの早期開始または読出し/書き込みを促進する働きを持つため、長い ISO パケットで（512 ~ 1023 バイト）使用する必要があります。このように、最初の 512 バイトは、ダブルバッファモードの使用を回避しながら、ソフトウェアで処理することができます。このビットは、1 つの ISO エンドポイントのみが設定されている場合に有効です。

必要なときには、制御レジスタに書き込むことによって、ユニットを低電力モード（SUSPEND モード）にできます。このとき、すべての静的消費電流が回避され、USB クロックをスローダウンまたは停止できます。低電力モード中に USB 入力でアクティビティが検出されると、デバイスは非同期でウェイクアップします。特殊な割込みソースをウェイクアップラインに直接接続することにより、システムは通常のクロック生成をただちに再開し、直接クロックの開始/停止をサポートできます。

ホストモードと専用機能

レジスタ USB_CNTR の単一の HOST ビット を使用して、ホストモードを有効化できます。ホストモード機能により、USB はリモートペリフェラルと通信できます。サポートされる機能は、デバイスモードに対応し、同じレジスタ構造を使用してバッファを管理します。ホストモードでは同数のエンドポイントをサポートできますが、ホストモードでは「チャンネル」という用語がよく使用されます。実際、各チャンネルは接続されたデバイスとそのデバイス上のエンドポイントの組み合わせであるためです。パケット送受信の基本メカニズムは、デバイスモードでサポートされているメカニズムと同様です。

ホストモードでの動作時、USB にはバスの役割があり、その役割を果たすために、有効な周期的/非周期的エンドポイントに対応したトランザクションリクエストを発行する必要があります。ホストフレームスケジューラでは、フレームを効率的に使用することが前提となります。ハブへの接続がサポートされています。直接接続とハブ経由の接続の両方で、ロースピードデバイスへの接続が可能です。

デバイスモードの項で説明したように、バルクチャンネルとアイソクロナスチャンネルの両方で、ダブルバッファモードもホストモードでサポートされています。THR512 機能も（ただし、デバイスモードと同様に）ISO トラフィックに対してのみサポートされています。

注： デバイスモードではバッテリー充電機能が検出されますが（高速充電を促進するために）、ホストモードではバッテリー充電容量（標準の CDP または DCP ケース）を示すための統合サポートはなく、ホストポートが常にデフォルトの標準データポート（SDP）として提示されます。

注： LPM（リンク電源管理）の場合、ホストモードではこの機能がサポートされていません。

55.4.1 デバイスモードとホストモードの両方で使用される USB ブロックの説明

USB ペリフェラルは USB インタフェースに関するすべての機能を実装し、これには以下のブロックが含まれます。

- **USB 物理インタフェース (USB PHY) :** このブロックは、外部 USB ホストへの電気インタフェースを維持します。ディファレンシャルアナログトランシーバ本体、制御可能な組み込みプルアップレジスタ (USB_DP ラインに接続)、およびバッテリ充電検出 (BCD) のサポート、同じ USB_DP および USB_DM ラインでの多重化を含みます。アナログトランシーバの出力有効制御信号 (アクティブロー) は、USB_NOE により外部から供給されます。これを使用して、アクティビティ LED を駆動したり、実際の通信方向に関する情報を他の回路に提供することができます。
- **シリアルインタフェースエンジン (SIE) :** このブロックの機能には、同期パターン認識、ビットスタッフィング、CRC 生成およびチェック、PID 確認/生成、およびハンドシェイク評価が含まれます。USB トランシーバとインタフェースし、ローカルデータストレージ用のパケットバッファインタフェースによって提供される仮想バッファを使用する必要があります。このユニットは、フレーム開始 (SOF)、USB_Reset、データエラーなどの USB ペリフェラルイベントと、送信終了、パケットの正しい受信などのエンドポイント関連イベントに従って信号を生成し、これらの信号は割込みの生成に使用されます。
- **タイマ :** このブロックは、フレーム開始ロッククロックパルスを生成し、トラフィックが 3 ms の間、受信されなかったときにはグローバルサスペンドを (ホストから) 検出します。
- **パケットバッファインタフェース :** このブロックは、ローカルメモリを管理して、送信と受信の両方のためのバッファのセットを柔軟に実装します。SIE から着信したリクエストに従って適切なバッファを選び、エンドポイント/チャネルレジスタによって示されるメモリアドレスに配置します。パケットの終わりまで、各バイトの交換後にアドレスをインクリメントして、交換バイト数を追跡し、バッファが最大容量までオーバーランするのを防ぎます。
- **エンドポイント/チャネル関連レジスタ :** 各エンドポイント/チャネルには、エンドポイント/チャネルのタイプと現在のステータスを含んだレジスタが関連付けられます。単方向/シングルバッファエンドポイントの場合、1 つのレジスタで 2 つのエンドポイントを実装できます。レジスタの数は 8 つであり、最大 16 の単方向/シングルバッファまたは最大 7 つのダブルバッファエンドポイントを任意の組み合わせで使用できます。たとえば、USB ペリフェラルをプログラムして、4 つのダブルバッファエンドポイントと 8 つのシングルバッファ/単方向エンドポイントを使用できます。
- **制御レジスタ :** これらは USB ペリフェラル全体のステータスに関する情報を含むレジスタであり、レジュームやパワーダウンなどの USB イベントを強制するために使用されます。
- **割込みレジスタ :** これらは割込みマスクとイベントのレコードを含みます。割込み理由、割込みステータスの問い合わせや、保留中の割込みのステータスをクリアするために使用できます。

注 : * エンドポイント/チャネル 0 は、常にシングルバッファモードでの転送の制御に使用されます。

USB ペリフェラルは APB2 バスに APB2 インタフェースを通じて接続され、以下のブロックを含みます。

- **パケットメモリ :** これは、パケットバッファを物理的に含むローカルメモリです。パケットバッファインタフェースによって使用して、データ構造を作成し、アプリケーションソフトウェアから直接アクセスできます。パケットメモリのサイズは、2048 バイトであり、32 ビットの 512 個のワードで構成されます。
- **アービタ :** このブロックは、APB2 バスおよび USB インタフェースから着信するリクエストを受け入れます。APB2 アクセスに優先順位を付けることによってコンフリクトを解決するとともに、常にメモリ帯域幅の半分をすべての USB 転送を完了するために確保しています。このタイムデュプレックス方式によって仮想デュアルポート SRAM が実装され、USB トランザクション中のメモリアccessを可能にします。この方式により、任意の長さのマルチワード APB2 転送も可能です。

- レジスタマッパー：このブロックは、USB ペリフェラルのさまざまなバイト長およびビット長のレジスタを APB2 によってアドレス指定され構造化された 32 ビット長のワードセットに集めます。
- APB2ラッパー：これは、メモリとレジスタに APB2 へのインタフェースを提供します。また、USB ペリフェラル全体を APB2 アドレス空間にマップします。
- 割込みマッパー：このブロックは、USB イベントによって割込みを生成する方法を選択し、それらを NVIC にマップするために使用されます。

55.4.2 ホストモードに固有のホストフレームスケジューラ (HFS) の説明

ホストフレームスケジューラは、USB の優先順位とバンド幅のアクセス規則に従ってバスのホストチャンネルリクエストを送信する働きを持つハードウェアマシンです。

ホストチャンネルは次の 2 つのカテゴリに分類されます。

- 周期的チャンネル：アイソクロナスおよび割込みトラフィックタイプ。保証されたバンド幅アクセスを使用します。
- 非周期的チャンネル：バルクおよび制御トラフィックタイプ。ベストエフォート型サービスを使用します。

ホストフレームスケジューラは、次の 3 つのシーケンシャルウィンドウでフルスピードフレームを構成します。

- 周期的サービスウィンドウ
- 非周期的サービスウィンドウ
- ブラックセキュリティウィンドウ

新しいフレームの開始時にホストスケジューラは次のように機能します。

1. まず、フレームの開始時に有効だった (STAT ビットが VALID) すべての周期的チャンネルについて検討します。
2. 周期的チャンネルのサービスの単一ラウンドである周期的サービスウィンドウを、CH#1 から CH#8 の優先順位で実行します。双方向チャンネルの場合、OUT 方向を最初に実行します。
3. 周期的ラウンドが終了すると、一部の周期的チャンネルが最有効化された場合や、SOF の後に一部の新しいチャンネルが有効化された場合でも、HFS によって周期的サービスウィンドウが閉じられ、周期的トラフィックのサービスが停止されます。
4. CH#1 から CH#8 のハードウェア優先順位で現在有効になっている (STAT ビットが VALID) 非周期的チャンネルのすべてのサービスを開始します。双方向チャンネルの場合、OUT 方向を最初に実行します。
5. ほぼフレームの終了に達するまで、非周期的チャンネルの複数のラウンドロビンサービスサイクルを実行します。
6. 非周期的トラフィックはいつでもリクエストでき、HFS によりベストエフォートの遅延でサービスされます。ただし、バブルを避けるために新しいインジェクトリクエストが次のフレームまで直接延期されるフレームの最後のブラックセキュリティウィンドウは例外です。これは、セキュリティウィンドウの前にサービスされていない保留中のトランザクションにも適用されます。

55.5 デバイスモードおよびホストモードでのプログラミングの考慮事項

以下のセクションでは、アプリケーションソフトウェア開発を容易にするために、USB ペリフェラルとアプリケーションプログラムの間の予期される相互作用について説明します。

55.5.1 汎用 USB デバイスのプログラミング

このパートでは、USB 準拠動作を得るためにアプリケーションソフトウェアに必要な主なタスクについて説明します。ほとんどの汎用 USB イベントに関連するアクションが考慮に入れられており、ダブルバッファエンドポイントとアイソクロナス転送という特殊ケースについて、それぞれのパラグラフで説明します。システムリセットを別にすると、アクションは常に USB ペリフェラルによって開始され、以下に述べる USB イベントの 1 つによって駆動されます。

55.5.2 システムリセットとパワーオンリセット

システムリセットおよびパワーオンリセットの際、アプリケーションソフトウェアが最初に行うべき操作は、必要なクロック信号のすべてを USB ペリフェラルに与えることであり、その後、リセット信号をネゲートして、レジスタにアクセスできるようにしなければなりません。初期化シーケンス全体は、以下のとおりです。

最初のステップとして、アプリケーションソフトウェアは、デバイスクロック管理ロジックによって提供される関連の制御ビットを使用して、レジスタマクロセルクロックを有効にし、マクロセル固有のリセット信号をネゲートする必要があります。

その後、CNTR レジスタの PDWN ビットを使用して、USB トランシーバに関連するデバイスのアナログ部分をオンにする必要があります。これには特別な処理が必要です。このビットは、ポートトランシーバに供給される内部電圧基準をオンにすることを目的としています。この回路にはスタートアップ時間が定義されており（データシートで指定されている $t_{STARTUP}$ ）、この間の USB トランシーバの動作は定義されていません。したがって、CNTR レジスタの PDWN ビットをセットした後、この時間だけ待つ必要があります。その後、USB 部分のリセット条件を解除します（CNTR レジスタの USBRST ビットをクリアすることによって）。ISTR レジスタをクリアすると、他のマクロセル動作が有効になる前に、誤って保留中の割り込みが削除されます。

システムリセット時には、マイクロコントローラは必要なすべてのレジスタとパケットバッファ記述テーブルを初期化して、USB ペリフェラルが割り込みとデータ転送を正しく生成できるようにする必要があります。エンドポイント/チャネル固有ではないレジスタはすべて、アプリケーションソフトウェアの必要に応じて初期化する必要があります（有効にする割り込みの選択、パケットバッファのアドレスの選択など）。USB リセットの場合、その後、プロセスが続行します（以下のパラグラフを参照）。

デバイスモードでの USB バスリセット (RST_DCON 割り込み)

このイベントが発生すると、USB ペリフェラルは、前のパラグラフで説明した初期化後のシステムリセットと同じ条件になります。すなわち、すべてのエンドポイントレジスタで通信が無効になります（USB ペリフェラルはパケットに応答しません）。USB リセットイベントに対して、USB 機能を有効にして、USB アドレスを 0 にし、デフォルトのコントロールエンドポイントのみを実装します（エンドポイントアドレスも 0）。これは、USB_DADDR レジスタの機能有効 (EF) ビットをセットし、CHEP0R レジスタと関連するパケットバッファを初期化することによって行われます。USB エニューメレーションプロセス時、ホストは一意的なアドレスをこのデバイスに割り当て、これが USB_DADDR レジスタの ADD[6:0] ビットに書き込まれなければならない、他に必要なエンドポイントを設定する必要があります。

RST_DCON 割り込みが受信されると、アプリケーションソフトウェアは、割り込みをトリガしたリセットシーケンスの終了から 10 ms 以内に、USB 機能 0 のデフォルトのエンドポイントを再び有効にする必要があります。

ホストモードでの USB バスリセット

ホストモードでバスリセットを有効化するには、USB_CNTR レジスタの USBRST ビットをセットします。標準からの最小有効リセット時間が順守されると、ソフトウェアによってクリアされます。

パケットバッファの構造と用途

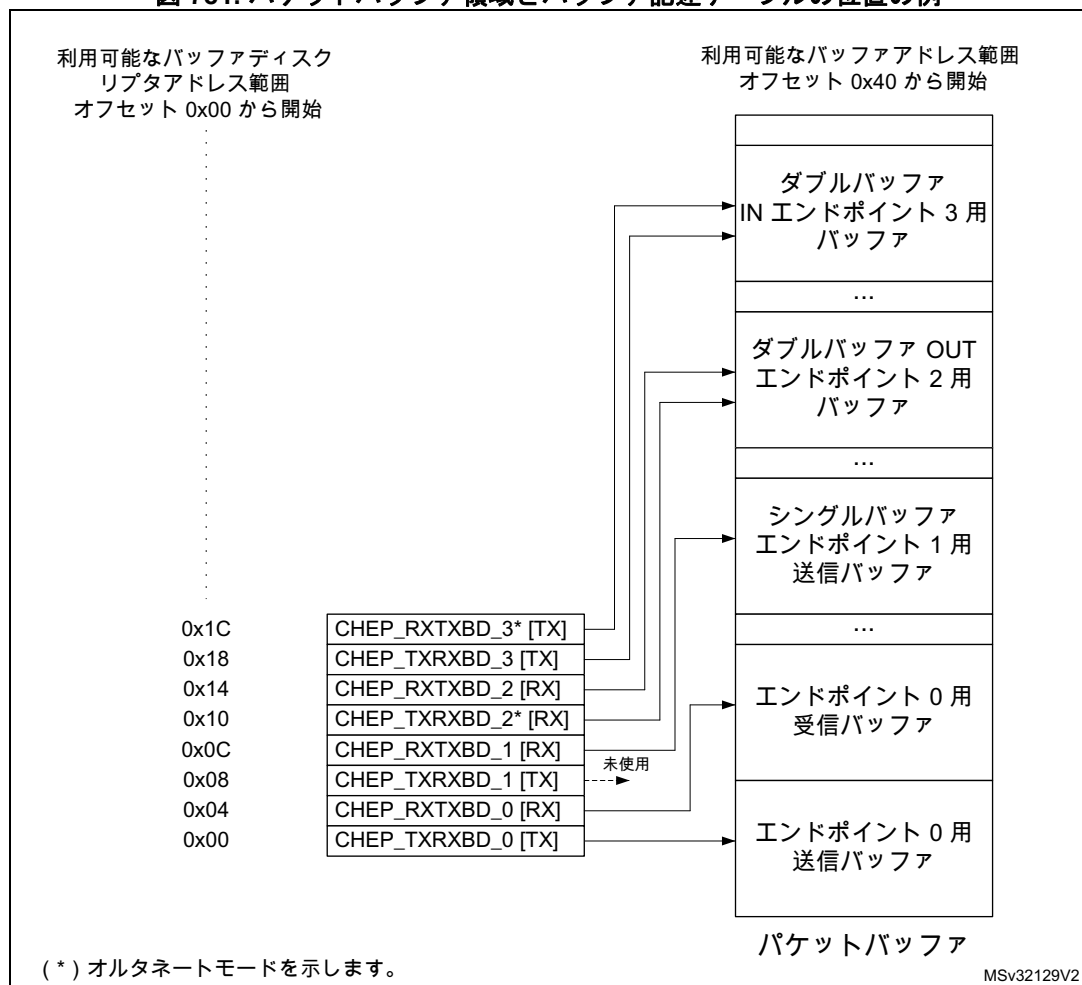
各双方向エンドポイントは、バス上でデータを送受信できます。受信データは、そのエンドポイント用に予約された専用のメモリバッファに格納され、別のメモリバッファにはエンドポイントによって送信されるデータが格納されます。このメモリへのアクセスは、パケットバッファインタフェースブロックによって行われ、これはメモリアクセスメモリにアクセスする必要があるため、アクセスコンフリクトを避けるためにアービトレーションロジックが使用され、APB2 サイクルの半分がマイクロコントローラのアクセスに使用され、残りの半分が USB ペリフェラルのアクセスに使用されます。このように、パケットメモリがデュアルポート SRAM であるかのように両方のエージェントが動作でき、マイクロコントローラが連続アクセスを実行中でも、コンフリクトを気にする必要がありません。USB ペリフェラルロジックは、専用クロックを使用します。この専用クロックの周波数は、USB 標準の要件により 48 MHz に固定されており、これは APB2 バスへのインタフェースに使用されるクロックと異なる場合があります。APB2 クロック周波数が USB ペリフェラルのクロック周波数より高い、または低い場合、さまざまなクロック設定が可能です。

注： USB データレートとパケットメモリインタフェース要件により、APB2 クロックは、データのオーバーラン/アンダーラン問題を回避するために、少なくとも 12 MHz 以上の周波数でなければなりません。

各エンドポイントには 2 つのパケットバッファが関連付けられます（通常は、送信用に 1 つ、受信用に 1 つ）。バッファの位置とサイズはバッファ記述テーブルで指定され、このテーブルもパケットメモリ内に配置されるので、バッファはパケットメモリ内の任意の場所に配置できます。各テーブルエントリがエンドポイントレジスタに関連付けられ、2 つの 32 ビットワードで構成されるため、テーブル開始アドレスは常に 8 バイトの境界に揃える必要があります。バッファディスクリプタテーブルのエントリについては、[セクション 55.6.2：バッファディスクリプタテーブル](#)で説明します。エンドポイントが単方向であり、アイソクロナスでもダブルバッファバルクでもない場合、必要なパケットバッファは 1 つだけです（サポートされる転送方向に応じた 1 つのバッファ）。サポートされない転送方向または未使用のエンドポイントに関連するテーブル位置は、ユーザが使用できます。アイソクロナスおよびダブルバッファバルクエンドポイントには、パケットバッファの特別な取り扱いがあります（それぞれ、[セクション 55.5.5：デバイスモードでのアイソクロナス転送](#) および [セクション 55.5.3：ダブルバッファエンドポイントとデバイスモードでの使用](#)を参照）。バッファ記述テーブルのエントリとパケットバッファ領域の関係を [図 781](#) に示します。

ホストモードについては、別のセクション、特に[セクション 55.5.6：ホストモードでのアイソクロナス転送](#)と[セクション 55.5.4：ダブルバッファチャネル：ホストモードでの使用](#)でバッファ使用モデルについて説明します。

図 781. パケットバッファ領域とバッファ記述テーブルの位置の例



各パケットバッファは、始めから終わりまで、受信または送信のいずれかに使用されます。USB ペリフェラルは、割り当てられたメモリバッファに隣接するメモリ位置の内容を変えることはありません。割り当てられたバッファ長を超えるパケットが受信された場合（バッファオーバーラン条件）、使用可能な最期の位置までしか、データはメモリにコピーされません。

エンドポイントの初期化

エンドポイントを初期化するための最初のステップは、CHEP_TXBD_n および CHEP_RXBD_n レジスタ（SRAM 内）の ADDRn_TX/ADDRn_RX フィールドに適切な値を書き込んで、送信データがすでに使用可能になったことと、受信データをバッファできることを USB ペリフェラルに知らせることです。USB_CHEPnR レジスタの UTYPE ビットをエンドポイントのタイプに応じてセットし、最終的には EPKIND ビットを使用して、特に必要な機能を有効にする必要があります。送信側では、USB_CHEPnR レジスタの STATTX ビットを使用してエンドポイントを有効にする必要があります、COUNTn_TX を初期化する必要があります。受信の場合、STATRX ビットをセットして受信を有効にする必要があります、BLSIZE および NUM_BLOCK フィールドを使用して、割り当てられたバッファサイズを COUNTn_RX に書き込む必要があります。単方向エンドポイントは、アイソクロナスおよびダブルバッファバルクエンドポイントを除き、サポートされる方向に関するビットとレジスタのみを初期化する必要があります。送信または受信が有効になると、レジスタ USB_CHEPnR と位置 ADDRn_TX/ADDRn_RX、COUNTn_TX/COUNTn_RX（それぞれ）をアプリケーションソフトウェアで変更しないでください。これらの値はハードウェアが直接変更できます。データ転送操作が完了し、

CTR 割込みイベントによって通知されると、再びアクセスできるようになり、新しい操作を再び有効にできます。

デバイスモードでのデータ送信 (IN パケット)

IN トークンパケットを受信するとき、受信したアドレスが設定済みの有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の CHEP_TXBD_n (フィールド ADDRn_TX および COUNTn_TX) にアクセスします。これらの位置の内容は、内部 16 ビットレジスタ ADDR および COUNT に格納されます (ソフトウェアからはアクセスできません)。パケットメモリが再びアクセスされ、最初の送信バイトが読出され (2554 ページのパケットバッファの構造と用途を参照)、USB ペリフェラルにより、USB_CHEPnR のビット DTOGTX に従って DATA0 または DATA1 PID の送信が開始されます。PID が完了すると、バッファメモリから読み出された最初のバイトが出力シフトレジスタにロードされ、USB バスで送信されます。最後のデータバイトが送信された後、計算された CRC が送信されます。アドレス指定されたエンドポイントが有効でなかった場合は、USB_CHEPnR レジスタの STATTX ビットに従って、NAK または STALL ハンドシェイクパケットがデータパケットの代わりに送信されます。

内部レジスタ CHEP_TXBD_n の ADDRn_TX フィールドは、現在のバッファメモリ位置を指すポインタとして使用され、COUNT は、残りの送信バイト数をカウントするために使用されます。パケットバッファメモリから読み出された各ハーフワードは、最下位バイトから順に USB バスで送信されます。送信バッファメモリは、COUNTn_TX/4 ワードの場合、ADDRn_TX によって示されるアドレスから順に読み出されます。送信パケットが奇数個のバイトで構成される場合、最後にアクセスされたハーフワードの下位半分のみが使用されます。

ホストが ACK を受信した場合、USB_CHEPnR レジスタは次のように更新されます。DTOGTX ビットがトグルされ、STATTX = 10 (NAK) をセットすることによってエンドポイントが無効化され、ビット VTTX がセットされます。アプリケーションソフトウェアは、まず、USB_ISTR レジスタの IDN および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。VTTX イベントの処理は、割込みビットをクリアすることから始まります。次に、アプリケーションソフトウェアは、送信データでいっぱい別のバッファを準備し、COUNTn_TX テーブル位置を次の転送時に送信されるバイト数で更新し、最後に、STATTX を 11 (VALID) にセットして、送信を再び有効にします。STATTX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された IN リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再試行します。CTR 割込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の IN トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

ホストモードでのデータ送信 (OUT パケット)

ホストモードでのデータ送信は、デバイスモードの場合と同じ一般原則に従います。主な違いは、プロトコルによるものです。たとえば、ホストは送信を開始しますが、デバイスは着信トークンに応答します。

パケットの送信用に予約されたパケットメモリ内の位置に、ADDRn_TX をセットする必要があります。これにより、OUT パケットの内容がパケットメモリ内のそのアドレスに書き込まれ、パケット内のバイト数を示すために (必要に応じて) COUNTn_TX を更新する必要があります。

適切なエンドポイント用に DEVADDR を書き込む必要があり、送信をトリガするために STATTX を 11 (VALID) にセットする必要があります。これにより、HFS によって送信がスケジュールされます。

送信が正常に行われると、CTR 割込み (適切な転送) がトリガされます。IDN および DIR を検査することにより、該当するチャンネルと方向が特定できます。示されるチャンネルに基づいて、STATTX フィールドは DISABLE に移行しています。NAK の受信中の場合 (ペリフェラルが準備完了でない場合)、STATTX は NAK の状態です。STALL 応答の場合、STATTX は STALL の状態です。最後の場合、バスのリセットが必要です。

デバイスが ACK を受信した場合、USB_CHEPnR レジスタは次のように更新されます。DTOGTX ビットがトリガされます。

次の場合、ビット VTTX および ERR_TX を通じてエラー条件が通知されます。

- ハンドシェイクを時間内に受信しない場合
- 偽の EOP
- ビットスタッフィングエラー
- 無効なハンドシェイク PID

デバイスモードでのデータ受信 (OUT および SETUP パケット)

この 2 つのトークンは、USB ペリフェラルによっておおむね同じように扱われます。SETUP パケットの取り扱いにおける違いについては、制御転送に関する以下のパラグラフで詳しく述べます。OUT/SETUP PID を受信するとき、アドレスが有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の ADDRn_RX および COUNTn_RX フィールドの内容にアクセスします。ADDRn_RX フィールドの内容は、内部レジスタ ADDR に直接格納されます。内部レジスタ COUNT がリセットされ、BLSIZE および NUM_BLOCK ビットフィールドの値が USB_CHEP_RXBD_n 内に読み出されると、それを使用して BUF_COUNT が初期化され、これは、バッファオーバーラン条件のチェックに使用される内部 16 ビットカウンタです (これらの内部レジスタはすべて、ソフトウェアからはアクセスできません)。その後、USB ペリフェラルによって受信されるデータバイトは、ハーフワード単位でパケット化され (最初の受信バイトが最下位バイトとして格納されます)、内部 ADDR レジスタに含まれるアドレスから順にパケットバッファに転送され、各バイト転送により、BUF_COUNT はデクリメントされ、COUNT はインクリメントされます。DATA パケットの終わりが検出されると、受信された CRC の正しさがテストされ、受信時にエラーが発生しなかった場合のみ、ACK ハンドシェイクパケットが送信元ホストに送り返されます。

CRC が正しくなかった場合、または他の種類のエラー (ビットスタッフィング違反、フレームエラーなど) が発生した場合、データバイトは、少なくともエラー検出ポイントまではパケットメモリにコピーされますが、ACK パケットは送信されず、USB_ISTR レジスタの ERR ビットがセットされます。ただし、この場合、通常はソフトウェアのアクションは不要です。USB ペリフェラルは受信エラーから回復し、次のトランザクションを処理する準備ができた状態を保ちます。アドレス指定されたエンドポイントが有効でなかった場合は、USB_CHEPnR レジスタの STATRX ビットに従って、NAK または STALL ハンドシェイクパケットが ACK の代わりに送信され、データは受信メモリバッファに書き込まれません。

受信メモリバッファの位置は、ADDRn_RX に含まれるアドレスから受信データパケット長に対応するバイト数だけ、または、BLSIZE および NUM_BLOCK によって定義された割り当て済みの最後のメモリ位置のいずれか先に来る方まで書き込まれます。このように、USB ペリフェラルは、割り当てられた受信メモリバッファ領域の終わりを越えて書き込むことはありません。データパケットペイロードの長さ (アプリケーションによって使用される実際のバイト数) が割り当てられたバッファを超える場合、USB ペリフェラルによってバッファオーバーラン条件が検出されます。この場合、通常の ACK の代わりに STALL ハンドシェイクが送信されて問題がホストに通知され、割り込みは生成されず、トランザクションはエラーとみなされます。

ACK ハンドシェイクパケットを送信することにより、トランザクションが正常に完了すると、内部 COUNT レジスタがバッファ記述テーブルエントリ内の COUNTn_RX 位置にコピーされ、影響を受けなかった BLSIZE および NUM_BLOCK フィールドはそのままに保たれ (通常は再書き込みの必要はありません)、USB_CHEPnR レジスタが次のように更新されます。DTOGRX ビットがトリガされ、STATRX = 10 (NAK) をセットすることによってエンドポイントが無効化され、ビット VTRX がセットされます。エラーまたはバッファオーバーランによってトランザクションが失敗した場合、上記のアクションはいずれも実行されません。アプリケーションソフトウェアは、まず、USB_ISTR レジスタの IDN および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。VTRX イベントの処理は、まず、トランザクションのタイ

ブを決定することによって行われます (USB_CHEPnR レジスタの SETUP ビット)。アプリケーションソフトウェアは、割込みフラグビットをクリアして、処理されるエンドポイントに関連するバッファ記述テーブルエントリ内の COUNTn_RX 位置を読み出して、受信バイト数を取得する必要があります。受信データが処理された後、アプリケーションは USB_CHEPnR の STATRX ビットを 11 (VALID) にセットして、次のトランザクションを有効にする必要があります。STATRX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された OUT リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再試行します。CTR 割込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の OUT トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

ホストモードでのデータ受信 (IN パケット)

ホストモードでのデータ受信は、デバイスモードの場合と同じ一般原則に従います。この場合も、主な違いはプロトコルによるものです。デバイスで、前の操作後の準備状況によって、データの受信が可能な場合と可能でない場合がありますが、ホストでは、準備が完了してデータが格納できる場合のみ、受信データのリクエストをします。

パケットの送信用に予約されたパケットメモリ内の位置に、ADDRn_TX をセットする必要があります。IN トークンパケットへのデータフェーズ応答で受信した内容は、パケットメモリ内のアドレスに書き込まれ、パケット内のバイト数を示すため、このプロセス中に COUNTn_TX がハードウェアによって更新されます。

適切なエンドポイント用に DEVADDR を書き込む必要があり、受信をトリガするために STATRX を VALID にセットする必要があります。これにより、HFS によって受信がスケジュールされます。

受信が正常に行われると、割込み CTR (適切な転送) がトリガされます。IDN および DIR を検査することにより、該当するチャンネルと方向が特定できます。示されるチャンネルに基づいて、STATRX フィールドは DISABLE に移行しています。NAK の受信中の場合 (ペリフェラルが準備完了でない場合)、STATRX は NAK の状態です。STALL 応答の場合、STATRX は STALL の状態です。最後の場合、バスのリセットが必要です。次の場合、IN パケット中にビット VTRX および ERR_RX を通じてエラー条件が通知されます。

- 偽の EOP
- ビットスタッフィングエラー
- CRC の誤り

デバイスモードでの制御転送

制御転送は、SETUP トランザクション、すべて同じ方向のゼロ個以上のデータステージ、およびステータスステージ (反対方向のゼロバイトの転送) で構成されます。SETUP トランザクションは制御エンドポイントのみによって処理され、OUT トランザクション (データ受信) と非常に似ていますが、制御転送を初期化するために、アドレス指定されたエンドポイントレジスタの DTOGTX および DTOGRX ビットの値がそれぞれ 1 と 0 にセットされる点と、ソフトウェアが後続のトランザクションが IN か OUT かを SETUP の内容に応じて決定できるように、STATTX と STATRX の両方が 10 (NAK) にセットされる点が異なります。制御エンドポイントは VTRX イベントごとに USB_CHEPnR レジスタの SETUP ビットをチェックして、通常の OUT トランザクションと SETUP トランザクションを区別する必要があります。USB デバイスは、SETUP ステージで転送されたデータを解釈することによって、データステージの数と方向を判断でき、エラーの場合にはトランザクションを STALL する必要があります。そのためには、最後以外のすべてのデータステージで、未使用の方向を STALL にセットして、ホストが転送方向を反転するのが速過ぎた場合にはステータスステージとして STALL されるようにする必要があります。

最後のデータステージを有効にするときには、反対方向を NAK にセットして、ホストが (ステータスステージを実行するために) 転送方向をすぐに反転した場合、制御操作の完了を待つようにする必要があります。制御操作が正常に完了した場合、ソフトウェアは NAK を VALID に変更し、そうでな

い場合は STALL に変更します。同時に、ステータスステージが OUT の場合、STATUS_OUT (USB_CHEPnR レジスタの EPKIND) ビットをセットして、ステータストランザクションが非ゼロのデータで実行された場合はエラーが生成されるようにする必要があります。ステータストランザクションが処理されると、アプリケーションは STATUS_OUT ビットをクリアし、STATRX を VALID にセットし (新しいコマンドを受け入れるため)、STATTX を NAK にセットします (次のセットアップの直後のステータスステージを遅延するため)。

USB 仕様では、SETUP パケットには ACK 以外のハンドシェイクでは応答できず、結果として、以前許可されていないアボートして新しいコマンドを開始すると規定されているため、USB ロジックでは、コントロールエンドポイントはホストから受信した SETUP トークンに対して NAK または STALL パケットで応答することはできません。

STATRX ビットが 01 (STALL) または 10 (NAK) にセットされ、SETUP トークンが受信されると、USB はデータを受け入れ、要求されたデータ転送を実行し、ACK ハンドシェイクを返送します。そのエンドポイントに以前に発行された VTRX リクエストがあり、アプリケーションによってまだ確認応答されていない場合 (例: VTRX ビットが以前に完了した受信からセットされたままになっている場合)、USB は SETUP トランザクションを破棄し、状態にかかわらずハンドシェイクパケットで応答せず、受信エラーをシミュレートして、ホストが SETUP トークンを再び送信しなければならないようにします。これは、VTRX 割込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された SETUP トランザクションの通知が失われるのを避けるために行われます。

ホストモードでの制御転送

制御転送は、SETUP トランザクション、すべて同じ方向のゼロ個以上のデータステージ、およびステータスステージ (反対方向のゼロバイトの転送) で構成されます。SETUP トランザクションは、制御エンドポイントによってのみ処理されます。制御エンドポイントで USB_CHEPnR レジスタの SETUP ビットをセットする必要があります。アドレス指定されたエンドポイントレジスタの DTOGTx および DTOGRx ビットの値が 0 にセットされます。制御書込みか制御読出ししかによって、ホストフレームスケジューラを通じて制御転送をトリガするため、STATTX または STATRX が 11 (ACTIVE) に設定されます。

CTR 割込みの受信時に、IDN および DIR ビットを検査することにより、チャネル (デバイスアドレスとエンドポイント) を特定できます。パケットに確認が不能になるような破損がない場合、制御のたびにデバイスで NAK がセットされます。この状態は STATTX の値に反映されます。

エラーが発生している場合、ERR ビットがセットされます。たとえば、デバイスで CRC エラーが発生している場合、ホストに ACK は返されません。ホストが ACK を受信しない場合、適切な遅延の後に、ERR_TX がセットされて (割込みを生成可能) タイムアウトエラーが生成されます。

55.5.3 ダブルバッファエンドポイントとデバイスモードでの使用

USB 標準によって定義されているすべてのエンドポイントタイプは、さまざまなトラフィックモデルを表し、各種のデータ転送操作の典型的な要件を記述しています。データの大部分をホスト PC と USB 機能の間で転送する必要があるときには、バルクエンドポイントタイプが最適のモデルです。ホストはバルクトランザクションをスケジューリングして、フレーム内の使用可能な帯域幅のすべてをいっぱいにして、アドレス指定された USB 機能がバルクトランザクションを処理できる限り、実際の転送速度を最大化します。次のトランザクションが着信したときに、USB 機能が前のトランザクションでビジーであった場合は、NAK ハンドシェイクで応答し、ホスト PC は、USB 機能が処理できるようになるまで、同じトランザクションを再び発行するので、再送信によって帯域幅が占有されるため、実際の転送速度は低下します。このため、バルクエンドポイントでは、「ダブルバッファリング」と呼ばれる専用機能を使用できます。

「ダブルバッファリング」が有効なときには、データトグルシーケンシングが使用されて、USB ペリフェラルが要求されたデータ転送を実行するために使用するバッファが選択され、送信と受信の両方のパケットメモリ領域を使用して、トランザクションの成功ごとのバッファスワッピングを管理する

ため、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルは他方のバッファをいっぱいにします。たとえば、「受信」ダブルバッファバルクエンドポイント宛での OUT トランザクション時、1 つのバッファが USB ホストから着信する新しいデータで満たされている間、もう 1 つのバッファはマイクロコントローラソフトウェアが使用できます（「送信」ダブルバッファバルクエンドポイントと IN トランザクションでも同様です）。

スワップバッファ管理には、アドレスポイントと割り当て済みメモリバッファの長さをホストする 4 つすべてのバッファ記述テーブル位置を使用する必要があるため、ダブルバッファバルクエンドポイントの実装に使用される USB_CHEPnR レジスタは、強制的に単方向として使用されます。したがって、1 つの STAT ビットのみを 00 (DISABLED) 以外の値にセットする必要があります。ダブルバッファバルクエンドポイントが受信について有効な場合は STATRX、ダブルバッファバルクエンドポイントが送信について有効な場合は STATTX。ダブルバッファバルクエンドポイントを受信と送信の両方について有効にする必要がある場合は、2 つの USB_CHEPnR レジスタを使用する必要があります。

ダブルバッファリング機能を利用して、可能な限り高い転送速度を達成するには、前の章で説明したエンドポイントフロー制御構造を変更して、トランザクションの正常官僚の旅ではなく、USB ペリフェラルとアプリケーションの間でバッファコンフリクトが発生したときのみ、エンドポイントのステータスを NAK に切り替える必要があります。USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関連する DTOG ビットによって定義されます。すなわち、「受信」ダブルバッファバルクエンドポイントの場合は DTOGRX (USB_CHEPnR レジスタのビット 14)、または「送信」ダブルバッファバルクエンドポイントの場合は DTOGTX (USB_CHEPnR レジスタのビット 6) です。新しいフロー制御方式を実装するには、USB ペリフェラルはアプリケーションソフトウェアによって使用されているパケットバッファを知り、コンフリクトに注意する必要があります。USB_CHEPnR レジスタには 2 つの DTOG ビットがありますが、データおよびバッファシーケンシングのために USB ペリフェラルによって使用されるのは 1 つだけであり（ダブルバッファリング機能に必要な単方向成約のため）、もう 1 つはアプリケーションソフトウェアが、使用中のバッファを示すために使用できます。この新しいバッファフラグは、SW_BUF と呼ばれます。次の表で、「送信」および「受信」ダブルバッファバルクエンドポイントの場合の USB_CHEPnR レジスタのビットと DTOG/SW_BUF 定義の対応を説明します。

表 591. ダブルバッファリングバッファフラグの定義

バッファフラグ	「送信」エンドポイント	「受信」エンドポイント
DTOG	DTOGTX (USB_CHEPnR ビット 6)	DTOGRX (USB_CHEPnR ビット 14)
SW_BUF	USB_CHEPnR ビット 14	USB_CHEPnR ビット 6

USB ペリフェラルによって現在使用されているメモリバッファは、DTOG バッファフラグによって定義され、アプリケーションソフトウェアによって現在使用されているバッファは、SW_BUF バッファフラグによって示されます。バッファフラグの値と使用されているパケットバッファの関係は、どちらの場合も同じであり、次の表に示されています。

表 592. バルクダブルバッファリングメモリバッファの用途 (デバイスモード)

エンド ポイント タイプ	DTOG	SW_BUF	USB ペリフェラルによって 使用されるバッファ	アプリケーションソフトウェアに よって使用されるバッファ
送信 (IN)	0	1	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置
	1	0	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
受信 (OUT)	0	1	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	1	0	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。

1. NAK ステータスのエンドポイント。

バルクエンドポイントのダブルバッファリング機能は、次のようにして有効化されます。

- USB_CHEPnR レジスタ UTYPE ビットフィールドに 00 を書き込み、エンドポイントをバルクとして定義します。
- 同じレジスタの EPKIND ビットを 1 (DBL_BUF) にセットします。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG および SW_BUF ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。DBL_BUF のセット後に発生した初めてのトランザクションの終了により、ダブルバッファバルクエンドポイントの特殊なフロー制御がトリガされ、DBL_BUF がセットされたままである限り、このエンドポイントにアドレス指定された他のすべてのトランザクションについて、これが使用されます。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB_CHEPnR レジスタの VTRX または VTTX ビットがセットされます。同時に、USB_CHEPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、USB ペリフェラルのバッファスワッピングを完全にソフトウェアから独立させます。通常のトランザクション、および DBL_BUF のセット後の初めてのトランザクションと違って、STAT ビットペアはトランザクション終了の影響を受けず、値は 11 (VALID) のままです。ただし、新しいトランザクションのトークンパッケージが受信されると、実際のエンドポイントステータスは、USB ペリフェラル

とアプリケーションソフトウェアの間のバッファコンフリクトが検出されたときには 10 (NAK) としてマスクされます (この条件は DTOG と SW_BUF が同じ値を持つことによって示されます。2561 ページの表 592 を参照)。アプリケーションソフトウェアは、CTR イベント通知に応答して、割込みフラグをクリアし、完了したトランザクションの要求された処理を開始します。アプリケーションパケットバッファの使用が終了すると、ソフトウェアは SW_BUF ビットに 1 を書き込んでトグルし、そのバッファが使用可能になったことを USB ペリフェラルに通知します。このように、NAK されるトランザクションの数は、トランザクションデータのアプリケーションエラーレーション時間によってのみ制限されます。エラーレーション時間が USB バスでのトランザクションの完了に必要な時間より短かった場合、フロー制御による再送信は行われず、実際の転送速度はホスト PC によってのみ制限されます。

アプリケーションソフトウェアは、11 (VALID) とは異なる明示的なステータスを、関連する USB_CHEPnR レジスタの STAT ビットペアに書き込むことによって、ダブルバッファバルクエンドポイントのために実装される特殊なフロー制御をオーバーライドできます。この場合、USB ペリフェラルは、バッファ使用条件に関係なく、プログラムされたエンドポイントステータスを常に使用します。

55.5.4 ダブルバッファチャネル：ホストモードでの使用

ホストモードのダブルバッファチャネルで使用される送受信方法は、デバイスモードで説明した方法と同様です。

デバイスモードの表と同様に、以下の新しい表表 593：バルクダブルバッファリングメモリバッファの用途 (ホストモード) に OUT および IN トークンのプログラム設定を示します。

表 593. バルクダブルバッファリングメモリバッファの用途 (ホストモード)

エンドポイントタイプ	DTOG	SW_BUF	USB ペリフェラルによって使用されるパケットバッファ	アプリケーションソフトウェアによって使用されるパケットバッファ
送信 (OUT)	0	1	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置
	1	0	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。

表 593. バルクダブルバッファリングメモリバッファの用途 (ホストモード) (続き)

エンド ポイント タイプ	DTOG	SW_BUF	USB ペリフェラルによって 使用されるパケットバッファ	アプリケーションソフトウェアに よって使用されるパケットバッファ
受信 (IN)	0	1	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	1	0	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。

1. NAK ステータスのエンドポイント。

55.5.5 デバイスモードでのアイソクロナス転送

USB 標準は、固定された正確なデータ生成/消費周波数を必要とするフルスピードペリフェラルをサポートし、この種類のトラフィックを「アイソクロナス」として定義しています。このデータの典型的な例は、オーディオサンプル、圧縮ビデオストリーム、および一般に、供給される周波数の正確さについて厳密な要件を持つ、あらゆる種類のサンプルデータです。エニュメレーションフェーズでエンドポイントが「アイソクロナス」として定義されると、ホストは必要な帯域幅をフレームに割り当て、エンドポイントの方向に応じて、各フレームに正確に 1 つの IN または OUT パケットを供給します。帯域幅要件を制限するために、アイソクロナストラフィックについては、失敗したトランザクションの再送信はできません。このため、アイソクロナストランザクションにはハンドシェイクフェーズはなく、ACK パケットは予期されず、データパケット後に送信されません。同じ理由で、アイソクロナス転送はデータトルゲンシーケンシングをサポートせず、常に DATA0 PID を使用してデータパケットを開始します。

エンドポイントのアイソクロナス動作は、USB_CHEPnR レジスタの UTYPE ビットを 10 にセットすることによって選択されます。ハンドシェイクフェーズがないため、STATRX/STATTX ビットの唯一の正当な値は 00 (DISABLED) と 11 (VALID) であり、他の値の場合は、USB 標準に準拠しない結果になります。アイソクロナスエンドポイントはダブルバッファリングを実装して、アプリケーションソフトウェア開発を容易にし、「送信」と「受信」の両方のパケットメモリ領域を使用して、トランザクションの成功のたびにバッファスワッピングを管理し、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルはもう 1 つのバッファを満たします。

USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関する DTOG ビットによって、表 594 に従って定義されます (「受信」アイソクロナスエンドポイントの場合は DTOGRX、「送信」アイソクロナスエンドポイントの場合は DTOGTX であり、両方とも、関連する USB_CHEPnR レジスタ)。

表 594. アイソクロナスメモリバッファの使用

エンドポイント タイプ	DTOG ビット の値	USB ペリフェラルによって 使用されるバッファ	アプリケーションソフトウェアに よって使用されるバッファ
送信 (IN)	0	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置
	1	USB_CHEP_RXTXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置	USB_CHEP_TXRXBD_0 (ADDR_TX / COUNT_TX) バッファ記述テーブル位置。
受信 (OUT)	0	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。
	1	USB_CHEP_TXRXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置	USB_CHEP_RXTXBD_0 (ADDR_RX / COUNT_RX) バッファ記述テーブル位置。

ダブルバッファアイソクロナスエンドポイントで発生すると、アイソクロナスエンドポイントの実装に使用される USB_CHEPnR レジスタは強制的に単方向として使用されます。アイソクロナスエンドポイントを受信と送信の両方について有効にする必要がある場合は、2 つの USB_CHEPnR レジスタを使用する必要があります。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB_CHEPnR レジスタの VTRX または VTTX ビットがセットされます。同時に、USB_CHEPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、バッファスワッピングを完全にソフトウェアから独立させます。STAT ビットペアはトランザクションの完了に影響を受けません。アイソクロナス転送ではハンドシェイクフェーズがないため、フロー制御は不可能であり、エンドポイントは常に 11 (VALID) のままです。CRC エラーまたはバッファオーバーラン条件がアイソクロナス OUT 転送時に発生した場合、常に正しいトランザクションとみなされ、常に VTRX イベントをトリガします。ただし、CRC エラーが発生すると、USB_ISTR レジスタの ERR ビットがセットされて、データ破損の可能性をソフトウェアに通知します。

55.5.6 ホストモードでのアイソクロナス転送

ホストの観点から、ホストフレームスケジューラにより、フレームごとに 1 つずつアイソクロナスパケットが発行、またはリクエストされます。NAK/ACK プロトコルはなく、データやトークンの再送もありません。

メカニズムは、デバイスモードの表に示したものと非常に類似しています。DTOG ビットバッファとバッファの使用との関係については、以下の表 595 を参照してください。

表 595. アイソクロナスメモリバッファの使用

エンドポイントタイプ	DTOGビットの値	USB ペリフェラルによって使用される パケットバッファ	アプリケーションソフトウェアによって使用されるパケットバッファ
送信 (OUT)	0	USB_CHEP_TXRXBD_0 (ADDRn_TX / COUNTn_TX) バッファ記述テーブル位置。	USB_CHEP_RXTXBD_0 (ADDRn_TX / COUNTn_TX) バッファ記述テーブル位置
	1	USB_CHEP_RXTXBD_0 (ADDRn_TX / COUNTn_TX) バッファ記述テーブル位置	USB_CHEP_TXRXBD_0 (ADDRn_TX / COUNTn_TX) バッファ記述テーブル位置。
受信 (IN)	0	USB_CHEP_RXTXBD_0 (ADDRn_RX / COUNTn_RX) バッファ記述テーブル位置。	USB_CHEP_TXRXBD_0 (ADDRn_RX / COUNTn_RX) バッファ記述テーブル位置。
	1	USB_CHEP_TXRXBD_0 (ADDRn_RX / COUNTn_RX) バッファ記述テーブル位置	USB_CHEP_RXTXBD_0 (ADDRn_RX / COUNTn_RX) バッファ記述テーブル位置。

エンドポイントのアイソクロナス動作は、USB_EPnR レジスタの UTYPE ビットを 10 にセットすることによって選択されます。ハンドシェイクフェーズがないため、STATRX/STATTX ビットの唯一の正当な値は 00 (DISABLED) と 11 (VALID) となります。

デバイスモードの場合と同様、このメカニズムでは DTOG ビットの自動トグルが可能です。ただし、ホストモードではこのトグルと同時に、完了したバッファの STATTX または STATRX が自動的に DISABLED にセットされるため、今後のバッファのアクセスは、11 (VALID) にセットして最有効化しなくても行えます。

55.5.7 サスペンド/レジュームイベント

USB 標準では、SUSPEND と呼ばれる特殊なペリフェラル状態が定義され、USB バスからの平均電流は 2.5 mA より大きくなってはなりません。この要件はバスパワードデバイスにとって基本的な重要性を持ちますが、セルフパワードデバイスは、この厳格な消費電力成約を順守する必要はありません。サスペンドモードでは、ホスト PC は 3 ms を超えて USB バスにトラフィックを送信しないことによって通知を送信します。SOF パケットは、通常動作では 1 ms ごとに送信されなければならないので、USB ペリフェラルは 3 回連続の SOF パケットの欠落をホスト PC からのサスペンドリクエストとして検出し、USB_ISTR レジスタの SUSP ビットを 1 にセットし、有効な場合は割込みが生成されます。デバイスがサスペンドされると、いわゆる RESUME シーケンスによって通常操作が復元され、これはホスト PC から、またはペリフェラルそのものから直接開始できますが、常にホスト PC によって終了されます。サスペンドされた USB ペリフェラルは、常に RESET シーケンスを検出して、通常の USB リセットイベントとして、このイベントに対応できなければなりません。

USB ペリフェラルをサスペンドするために使用される実際の手順は、デバイスの構成によって合計消費量を削減するために必要なアクションが異なるため、デバイスに依存します。

USB ペリフェラルの SUSP 通知に応答するアプリケーションソフトウェアルーチンの USB 関連の局面を中心にして、典型的なサスペンド手順の簡単な説明を以下に示します。

1. USB_CNTR レジスタの SUSPEN ビットを 1 にセットします。このアクションによって、USB ペリフェラル内のサスペンドモードが有効になります。サスペンドモードが有効になると、USB がサスペンドされている間に SUSP 割込みが発行されるのを回避するために、SOF 受信時のチェックが無効になります。
2. USB ペリフェラルとは異なるブロックの静的消費電力を除去または削減します。

3. USB_CNTR レジスタの SUSPRDY ビットを 1 にセットして、アナログ USB トランシーバの静的消費電力を除去しつつ、レジュームアクティビティを検出できるままにしておきます。
4. オプションで、外部オシレータとデバイス PLL をオフにして、デバイス内部のアクティビティを停止します。

デバイスが SUSPEND モード中に USB イベントが発生したときには、RESUME 手順が呼び出されて、通常のチェックと復元し、通常の USB 動作を復元する必要があります。ウェイクイベントが USB リセットシーケンスである時には、このプロセスが 10 ms 以内で終了するように特に注意が必要です (詳細については、「ユニバーサルシリアルバス仕様」を参照してください)。USB ペリフェラルのサスペンド中にレジュームまたはリセットシーケンスが開始されると、USB_CNTR レジスタの SUSPRDY ビットが非同期でクリアされます。このイベントによって WKUP 割込みがトリガされた場合でも、システムクロックの再開による長い遅延のため、割込み応答ルーチンの使用は慎重に評価する必要があります。公称クロックの再有効化の前の遅延を短くするには、サスペンド手順の終了直後にレジューム手順を行って、システムクロックが再開したらすぐにコードが実行されるようにします。システムをウェイクアップする際の ESD 放電またはその他の種類のノイズを防止するために (サスペンドモードの終了は、非同期イベントです)、サスペンド中にデータラインステータスの適切なアナログフィルタを有効にします。フィルタ幅は約 70 ns です。

以下は、レジューム手順で行うべきアクションのリストです。

1. オプションで、外部オシレータおよびデバイス PLL をオンにします。
2. USB_CNTR レジスタの SUSPEN ビットをクリアします。
3. レジュームをトリガしたイベントを識別する必要がある場合は、USB_FNR レジスタのビット RXDP および RXDM を [表 596](#) に従って使用でき、この表には、あらゆる場合のソフトウェアのアクションもリストされています。必要な場合には、上記のビットのステータスを監視することによって、レジュームまたはリセットシーケンスの終了を検出でき、これらが「10」設定 (アイドルバス状態を表します) に達したことをチェックします。さらに、リセットシーケンスの終了時には、USB_ISTR レジスタの RST_DCON ビットが 1 にセットされ、有効な場合は割込みが発行され、通常通りに処理されます。

表 596. レジュームイベント検出

[RXDP,RXDM] ステータス	ウェイクアップイベント	必要なレジュームソフトウェアアクション
"00"	ルートのリセット	なし
"10"	なし (バス上のノイズ)	サスペンドモードに戻る
"01"	ルートのレジューム	なし
"11"	不可 (バス上のノイズ)	サスペンドモードに戻る

デバイスは、USB プロトコルに直接関係しない特定のイベントに対する応答として、サスペンドモードを終了しなければならない場合があります (たとえば、マウスの移動はシステム全体をウェイクアップします)。この場合、USB_CNTR レジスタの L2RES ビットを 1 にセットし、1 ms から 15 ms までのインターバル後に 0 にリセットすることによって、レジュームシーケンスを開始できます (このインターバルは、システムクロックが公称周波数で動作しているときには 1 ms 周期で発生する ESOF 割込みによって計測できます)。L2RES ビットがクリアされると、レジュームシーケンスはホスト PC によって完了され、その終了は USB_FNR レジスタの RXDP および RXDM ビットを使用して監視できます。

注： L2RES ビットは、USB ペリフェラルがサスペンドモードにされた (USB_CNTR レジスタの SUSPEN ビットを 1 にセットした) 後でのみ使用する必要があります。

ホストモードでのサスペンドとレジューム

前のセクションでは、サスペンドメカニズムとレジュームメカニズムの基本について説明しました。

ホストの観点から見ると、サスペンドを開始するには、USB_CNTR の SUSPEN ビットを書き込みます。サスペンドのエントリが確認されると、SUSPRDY (同じく USB_CNTR 内) がセットされます。

サスペンドの状態のとき、アプリケーションでバスのレジュームを行うには、USB_CNTR の L2RES ビットを 1 にセットします。

以下の表 597 には、ウェイクアップイベント後に推奨される別のアクションを示します。ウェイクアップイベント後の別のラインの状態に従って、イベントの解釈と提案される動作を示します。ただし、ホストはフルスピードデバイスとロースピードデバイスの両方に対応し、サスペンドとレジュームの両方に異なるラインの状態が使用されることがあるため、この表は前に示したデバイスの表と比べて多少長くなります。

表 597. ホスト用のレジュームイベント検出

[RXDP,RXDM] ステータス	ウェイクアップイベント	必要なレジュームソフトウェアアクション
"00"	不可 (バス上のノイズ)	サスペンドモードに戻る
"10"	フルスピード対応デバイス : 不可 (バス上のノイズ) ロースピードデバイス : デバイスのリモートウェイクアップレジューム	なし
"01"	フルスピード対応デバイス : デバイスのリモートウェイクアップレジューム ロースピードデバイス : 不可 (バス上のノイズ)	なし
"11"	不可 (バス上のノイズ)	サスペンドモードに戻る

55.6 USB および USB SRAM レジスタ

USB ペリフェラルレジスタは、以下のグループに分けることができます。

- 共通レジスタ：割込みおよび制御レジスタ
- エンドポイント/チャネルレジスタ：エンドポイント/チャネル設定およびステータス

USB SRAM レジスタは以下をカバーします。

- バッファディスクリプタテーブル：データバッファの配置に使用されたパケットメモリの位置 (USB SRAM のベースアドレスは、[セクション 2.3：メモリ構成](#)を参照してください)。

すべてのレジスタアドレスは、USB ペリフェラルレジスタのベースアドレスに対するオフセットとして表されますが、USB SRAM ベースアドレスから始まるバッファディスクリプタテーブル位置は除外されます。

レジスタの説明で使用されている略語のリストについては、[100ページのセクション 1.2](#)を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

55.6.1 共通レジスタ

これらのレジスタは USB ペリフェラルの全般的動作に影響し、動作モード、割込み処理、デバイスアドレス、およびホスト PC によって更新される現在のフレーム番号へのアクセス付与を定義します。

USB 制御レジスタ (USB_CNTR)

アドレスオフセット：0x40

リセット値：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HOST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDISCM	THR 512M
rw														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRM	PMA OVRM	ERRM	WKUP M	SUSP M	RST_D CONM	SOFM	ESOF M	L1REQ M	Res.	L1RE S	L2RE S	SUS PEN	SUSP RDY	PDWN	USB RST
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	r	rw	rw

ビット 31 **HOST** : HOST モード

HOST ビットでは、ホストまたはデバイス USB 操作モードを選択します。このビットは、機能イネーブルビットによって USB ペリフェラルを有効化する前にセットする必要があります。

0 : USB デバイス機能

1 : USB ホスト機能

ビット 30:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DDISCM** : デバイス切断マスク

– ホストモード

0 : デバイス切断割込みは無効です

1 : デバイス切断割込みは有効です

ビット 16 **THR512M** : 512 バイト閾値割込みマスク

0 : 512 バイト閾値割込みは無効です

1 : 512 バイト閾値割込みは有効です

ビット 15 **CTRM** : 正しい転送割込みマスク

0 : 正しい転送 (CTR) 割込みは無効です。

1 : CTR 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 14 **PMAOVRM** : パケットメモリ領域のオーバー/アンダーラン割込みマスク

0 : PMAOVR 割込みは無効です。

1 : PMAOVR 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 13 **ERRM** : エラー割込みマスク

0 : ERR 割込みは無効です。

1 : ERR 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 12 **WKUPM** : ウェイクアップ割込みマスク

0 : WKUP 割込みは無効です。

1 : WKUP 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 11 **SUSPM** : サスペンドモード割込みマスク

0 : サスペンドモードリクエスト (SUSP) 割込みは無効です。

1 : SUSP 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 10 **RST_DCONM** : USB リセットリクエスト (デバイスモード) またはデバイス接続/切断 (ホストモード) 割込みマスク

0 : RESET 割込みは無効です。

1 : RESET 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 9 **SOFM** : フレーム開始割込みマスク

0 : SOF 割込みは無効です。

1 : SOF 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 8 **ESOFM** : 予期されるフレーム開始割込みマスク

0 : 予期されるフレーム開始 (ESOF) 割込みは無効です。

1 : ESOF 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 7 **L1REQM** : LPM L1 状態リクエスト割込みマスク

0 : LPM L1 状態リクエスト (L1REQ) 割込みは無効です。

1 : L1REQ 割込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割込みリクエストが生成されます。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **L1RES** : L1 リモートウェイクアップ/レジュームドライバ

– デバイスモード

LPM L1 50 μ s リモートウェイクアップ信号をホストに送信するために、ソフトウェアによってこのビットがセットされます。このビットは、信号の送信終了後にハードウェアによってクリアされます。

0 : 影響なし。

1 : 50 μ s リモートウェイクアップ信号をホストに送信します。

ビット 4 **L2RES** : L2 リモートウェイクアップ/レジュームドライバ

– デバイスモード

マイクロコントローラはこのビットをセットして、ホストにリモートウェイクアップ信号を送信できます。USB 仕様に従って、ホスト PC がレジュームシーケンスを終了まで駆動する準備ができてから 1 ms 以上 15 ms 以下で有効にする必要があります。

– ホストモード

レジューム信号をデバイスに送信するために、ソフトウェアによってこのビットがセットされます。レジューム終了をデバイスに送信してSOF生成を再開するために、ソフトウェアによってこのビットがクリアされます。

リモートウェイクアップのコンテキストでは、WAKEUP 割込みに続いてこのビットをセットします。

0 : 影響なし。

1 : L2 レジューム信号をデバイスに送信します

ビット 3 **SUSPEN** : サスペンド状態の有効化

– 状態 : デバイスモード

ソフトウェアは、SUSP 割込みが受信されたときにこのビットをセットできます。この割込みは、USB ペリフェラルが 3 ms 間、トラフィックを受信しなかったときに発行されます。ソフトウェアは、高低応答が送信されて L1REQ 割込みを受信した場合にも、このビットをセットできます。

サスペンド状態が内部的に伝播されるとすぐにデバイスのすべてのアクティビティが停止し、USB クロックがゲートされ、USB トランシーバが低電力モードにセットされ、SUSPRDY ビットがハードウェアによってセットされます。USB クロックソースを停止し、マイクロコントローラを STOP モードに移行することにより、デバイスアプリケーションの電力消費をより積極的に節約したい場合は、バスパワーデバイスアプリケーションの場合と同様、数サイクル待って、SUSPRDY = 1 によりサスペンドリクエストに対して確認応答されることを確認する必要があります。

このビットは、WAKEUP フラグがセットされるのと同時にハードウェアによってクリアされます。

0 : 影響なし。

1 : L1/L2 サスペンドを開始します

– 状態 : ホストモード

次のフレームに対してホストアプリケーションに何もスケジュールされていなくて、長期的な省電力を開始したい場合、ソフトウェアによってこのビットをセットすることができます。このビットをセットすると、ただちに SOF 生成と他のホストのアクティビティが停止され、USB クロックがゲートされ、トランシーバが低電力モードにセットされます。USB トランザクションの実行中に SUSPEN がセットされた場合、現在のトランザクションが終了した時点でサスペンドが開始されます。

サスペンド状態が内部で伝播されて有効になるとすぐに、SUSPRDY ビットがセットされます。USB クロックソースを停止し、マイクロコントローラを STOP モードに移行することにより、ホストアプリケーションの電力消費をより積極的に節約したい場合は、数サイクル待って、SUSPRDY = 1 によりサスペンドリクエストに対して確認応答されることを確認する必要があります。

このビットは、WAKEUP フラグがセットされるのと同時にハードウェアによってクリアされます。

0 : 影響なし。

1 : L1/L2 サスペンドを開始します

ビット 2 **SUSPRDY** : サスペンド状態が有効

SUSPEN 制御が内部的に有効になることでサスペンド状態が開始されるとすぐに、ハードウェアによってこのビットがセットされます。この状態では、USB のアクティビティはサスペンドされ、USB クロックはゲートされ、トランシーバは作動レシーバを無効化することにより低電力モードにセットされます。リモートウェイクアップまたはレジュームイベントを検出するため、非同期ウェイクアップロジックとシングルエンドレシーバのみが動作状態を維持します。

STOP モードが開始される前にビットがセットされていることを確認するため、ソフトウェアによりこのビットをポーリングする必要があります。

このビットは、WAKEUP フラグがセットされるのと同時にハードウェアによってクリアされます。

0 : 通常動作

1 : サスペンド状態

ビット 1 PDWN : パワーダウン

このビットは、何らかの理由で USB ペリフェラルを完全に無効にする必要がある場合、USB 関連のすべてのアナログ部品を完全にオフにするために使用されます。このビットがセットされると、USB ペリフェラルはトランシーバから切断され、使用できなくなります。

0 : パワーダウンを終了します

1 : パワーダウンモードに入ります

ビット 0 USBRST : USB リセット**– 状態 : デバイスモード**

USB で RESET 信号を受信するときと同様に、USB コアをリセットするために、ソフトウェアによってこのビットをセットすることができます。USB ペリフェラルは、RESET に対応して、内部プロトコル状態マシンをリセットします。RST_DCON ビットがクリアされるまで、受信と送信は無効です。すべての設定レジスタがリセットされるわけではありません。マイクロコントローラはこれらのレジスタを明示的にクリアする必要があります（これは、RST_DCON 割込みの安全な配信を確保し、RESET の直前のトランザクションを完了できるようにするためです）。機能アドレスおよびエンドポイントレジスタは、USB リセットイベントによってリセットされます。

0 : 影響なし。

1 : USB コアがリセットされます。

– 状態 : ホストモード

バスで USB リセット状態を駆動し、デバイスを初期化するため、ソフトウェアによってこのビットがセットされます。ソフトウェアによってこのビットがクリアされるとすぐに、USB リセットが終了します。

0 : 影響なし。

1 : USB リセット駆動

USB 割込みステータスレジスタ (USB_ISTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

このレジスタはすべての割込みソースのステータスを含むためアプリケーションソフトウェアは割込みリクエストの原因になったイベントを判断できます。

このレジスタの上位部分はシングルビットを含み、そのそれぞれが特定のイベントを表します。これらのビットは、関連するイベントが発生したときにハードウェアによってセットされます。USB_CNTR レジスタの対応するビットがセットされている場合、汎用割込みリクエストが生成されます。各ビットを調べる割込みルーチンは、必要なすべてのアクションを実行し、最後に処理済みビットをクリアします。そのいずれかがクリアされていない場合、割込みはまだ保留中であるとみなされ、割込みラインは再びハイに保たれます。複数のビットが同時にセットされた場合、1 つの割込みだけが生成されます。

エンドポイント/チャネルトランザクションの完了は、割込み応答遅延を軽減するために、さまざまな方法で処理できます。CTR ビットは、エンドポイント/チャネルがトランザクションを正常に完了するとすぐにハードウェアによってセットされ、USB_CNTR の対応するビットがセットされていた場合は汎用割込みリクエストが生成されます。エンドポイント/チャネル専用割込み条件は、USB_CNTR レジスタの CTRM ビットとは無関係に有効化されます。どちらの割込み条件も、ソフトウェアが対応する USB_CHEPnR レジスタの保留中のビットをクリアするまでアクティブなままです（CTR ビットは実際には読出し専用ビットです）。エンドポイント/チャネル関連の割込みの場合、ソフトウェアは、トランザクション方向 (DIR) および IDN 読出し専用ビットを使用して、最後の割込みリクエストを発行して対応する割込みサービスルーチンを呼び出したエンドポイント/チャネルを特定できます。

ユーザは、ソフトウェアが割込みサービスルーチンで USB_ISTR ビットをチェックする順序を指定することによって、同時に保留中の USB_ISTR ビットの相対的優先順位を選ぶことができます。イ

メントに関連し、処理済みのビットだけがクリアされます。サービスルーチンの終了時、残りの条件を処理するために、別の割込みがリクエストされます。

ビットの誤ったクリアを避けるために、ロード命令付きでクリアすることが推奨され、その場合、変更してはならないすべてのビットに 1 が書き込まれ、クリアすべきすべてのビットに 0 が書き込まれます（これらのビットはソフトウェアによってのみクリアできます）。読出し-変更-書込みのサイクルは避けるべきです。読出し操作と書込み操作の間に、別のビットがハードウェアによってセットされる可能性があり、デバイスにイベントを処理する時間ができる前に、次の書込みによってクリアされる可能性があるためです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LS_DCON	DCON_STAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDISC	THR 512
	r	r												rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMA_OVR	ERR	WKUP	SUSP	RST_DCON	SOE	ESOF	L1REQ	Res.	Res.	DIR	IDN[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **LS_DCON** : 低速デバイスが接続されています

– ホストモード :

このビットは、LS デバイス接続が検出されたときに、ハードウェアによってセットされます。デバイス接続は、未接続状態から連続で 22 サイクルの USB クロック (48 MHz) の間、LS J 状態がサンプリングされた後で信号送信されます。

ビット 29 **DCON_STAT** : デバイス接続ステータス

– ホストモード :

このビットには、デバイス接続ステータスに関する情報が格納されます。このビットはLS/FS デバイスがホストに取り付けられるとハードウェアによってセットされ、デバイスが取り外されるとリセットされます。

0 : デバイスが接続されていません

1 : FS または LS デバイスがホストに接続されています

ビット 28:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **DDISC** : デバイス接続

– ホストモード

このビットは、デバイス接続が検出されたときにセットされます。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んで効果はありません。

ビット 16 **THR512** : 512 バイト閾値割込み

このビットは、アイソクロナス転送中に 512 バイトの送受信が行われたときに、ハードウェアによって 1 にセットされます。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んで効果はありません。関連するチャネル/エンドポイントを示すための情報はありますが、実際にはこのような大きなパケットを持つ 1 つの ISO エンドポイント/チャネルのみがサポート可能です。

ビット 15 **CTR** : ホストモードでの転送完了

このビットは、エンドポイント/チャネルがトランザクションを正常に完了したことを示すために、ハードウェアによってセットされます。DIR および IDN ビットを使用して、ソフトウェアは割込みをリクエストしたエンドポイント/チャネルを判断できます。このビットは読出し専用です。

ビット 14 **PMAOVR** : パケットメモリ領域のオーバー/アンダーラン

このビットは、マイクロコントローラに USB メモリリクエストに対応する時間がなくなった場合にセットされます。USB ペリフェラルは、このイベントを次のように処理します。受信時、ACK ハンドシェイクパケットは送信されず、送信時、送信ストリームのビットスタッフエラーが強制的に発行されます。いずれの場合も、ホストはトランザクションを再試行します。通常操作では、PMAOVR 割込みは発生しません。失敗したトランザクションはホストによって再試行されるので、アプリケーションソフトウェアには、この割込み処理中にデバイス操作をスピードアップして、次のトランザクションの差異施行に備える機会があります。ただし、アイソクロナス転送では（アイソクロナストランザクションは再試行されません）、データの喪失につながるため、これは行われません。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 13 **ERR** : 誤差

このフラグは、下記にリストされているエラーのいずれかが発生したときにセットされます。

NANS : 応答なし (No ANSwer)。ホスト応答がタイムアウトになりました。

CRC : 巡回冗長検査 (Cyclic Redundancy Check) エラー。トークンまたはデータで受信された CRC の 1 つが正しくありませんでした。

BST : ビットスタッフイングエラー。ビットスタッフイングエラーが、PID、データ、または CRC で検出されました。

FVIO : フレーミングフォーマット違反。非標準のフレームが受信されました（正しい位置にない EOP、正しくないトークンシーケンスなど）。

USB ペリフェラルと PC ホストは、エラーの場合、完全にトランスペアレントな方法で再送信を管理するため、USB ソフトウェアは、通常、エラーを無視できます。この割込みは、ソフトウェア開発フェーズで、または USB バスでの送信品質の監視に役立ち、問題の可能性をユーザに知らせることができます（コネクタの緩み、ノイズが多い環境、USB ケーブルの導体の破損など）。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 12 **WKUP** : ウェイクアップ

このビットは、サスペンドモード中に、USB ペリフェラルをウェイクアップするアクティビティが検出されたときにハードウェアによって 1 にセットされます。このイベントは CTLR レジスタの SUSPRDY ビットを非同期でクリアし、USB_WAKEUP ラインを有効にし、レジュームプロセスの開始に関して残りのデバイス（ウェイクアップユニットなど）に通知するために使用されます。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 11 **SUSP** : サスペンドモードリクエスト

- デバイスモード

このビットは、3 ms にわたってトラフィックが受信されず、USB バスからのサスペンドモードリクエストを示したときに、ハードウェアによってセットされます。サスペンド条件チェックは、USB リセットの直後に有効化され、サスペンドモードがアクティブなとき（SUSPEN=1）、レジュームシーケンスの終了まで、ハードウェアによって無効にされます。このビットは読出し/書込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 10 **RST_DCON** : USB リセットリクエスト（デバイスモード）またはデバイス接続/切断（ホストモード）

- デバイスモード

このビットは、ホストによって USB リセットが解放され、バスがアイドル状態に戻るときに、ハードウェアによってセットされます。USB リセット状態は、連続する 60 回の SE0 サイクルのサンプリング後に、内部で検出されます。

- ホストモード

このビットは、デバイス接続またはデバイス切断が検出されたときに、ハードウェアによってセットされます。デバイス接続の信号送信は、未接続状態から 22 サイクル連続で J 状態がサンプリングされた後で行われます。デバイス切断の信号送信は、接続状態から 22 ビット時間連続で SE0 状態が検出された後で行われます。

ビット 9 SOF : フレーム開始

このビットは、新しい USB フレームの開始を知らせ、SOF パケットが USB バス経由で着信したときにセットされます。割込みサービスルーチンによって SOF イベントを監視して、USB ホストへの 1 ms の同期イベントを実行し、SOF パケット受信時に更新される USB_FNR レジスタを安全に読み出すことができます (これはアイソクロナスアプリケーションに便利です)。このビットは読出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 8 ESOF : 予期されたフレーム開始

– デバイスモード

このビットは、予期された SOF パケットが受信されなかったときにハードウェアによってセットされます。ホストから 1 ms ごとに SOF パケットが送信されますが、デバイスでこのパケットが正しく受信されない場合、サスペンドタイムによってこの割込みが発行されます。3 回連続して ESOF 割込みが生成され (たとえば、3 つの SOF パケットが失われ)、その間にトラフィックが発生しなかった場合、SUSP 割込みが生成されます。このビットは、サスペンドタイムがまだロックされていないときに SOF パケットの喪失が発生したときにセットされます。このビットは読出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 7 L1REQ : LPM L1 状態リクエスト

– デバイスモード

このビットは、L1 状態を入力する LPM コマンドが正常に受信され、確認応答された場合に、ハードウェアによってセットされます。このビットは読出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 6:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 DIR : トランザクションの方向

このビットは、割込みリクエストを生成した成功したトランザクションの方向に従って、ハードウェアによって書き込まれます。

DIR ビット = 0 の場合、割込みエンドポイントに関連して USB_CHEPnR レジスタで VTTX ビットがセットされます。割込みの原因となったトランザクションは、IN タイプです (USB ペリフェラルからホスト PC へのデータ送信)。

DIR ビット = 1 の場合、割込みをかけたエンドポイントに関連する USB_CHEPnR レジスタの VTRX ビットまたは VTTX と VTRX の両方がセットされます。割込みの原因となったトランザクションは OUT タイプであるか (ホスト PC からのデータを USB ペリフェラルが受信)、または 2 つの保留中のトランザクションが処理を待っています。

これは割込みを保留中にした方向を表すので、アプリケーションソフトウェアはこの情報を使用して、トリガしたトランザクションに関連する USB_CHEPnR ビットにアクセスできます。このビットは読出し専用です。

ビット 3:0 IDN[3:0] : デバイスエンドポイント/ホストチャネルの識別番号

これらのビットは、割込みリクエストを生成したホストチャネルまたはデバイスエンドポイントの番号に従って、ハードウェアによって書き込まれます。いくつかのエンドポイント/チャネルのトランザクションが保留中である場合、次のように定義された優先順位の高いエンドポイント/チャネルに関する識別番号がハードウェアによって書き込まれます。まず、優先順位に従って 2 つのレベルが定義され、アイソクロナスおよびダブルバッファバブルチャネル/エンドポイントが最初に検討されてから、それ以外が検査されます。同じセットの複数のエンドポイント/チャネルが割込みをリクエストしている場合、最下位リクエストレジスタに従って、USB_ISTR レジスタの IDN ビットが割り当てられ、CHEP0R が最も高い優先順位を持ち、その後 CHEP1R が続き、以下同様です。アプリケーションソフトウェアは、この優先順位方式するチャネルに従って、レジスタを各エンドポイント/チャネルに割り当てて、同時発生したエンドポイント/チャネルリクエストを適切に順序づけることができます。これらのビットは読出し専用です。

USB フレーム番号レジスタ (USB_FNR)

アドレスオフセット : 0x48

リセット値 : 0x0000 0XXX (X は未定義)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **RXDP** : 受信データ + ラインステータス

このビットを使用して、受信データプラスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

ビット 14 **RXDM** : 受信データ - ラインステータス

このビットを使用して、受信データマイナスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

ビット 13 **LCK** : ロック

– デバイスモード

このビットは、USB リセット条件の終了後、または USB レジュームシーケンスの終了後に、少なくとも 2 つ連続した SOF パケットが受信されたときに、ハードウェアによってセットされます。ロックされると、USB リセットまたは USB サスペンドイベントが発生するまで、フレームタイマはこの状態に保たれます。

ビット 12:11 **LSOF[1:0]** : SOF の喪失

– デバイスモード

これらのビットは、ESOF 割込みが生成されたときに、ハードウェアによって書き込まれ、失われた連続 SOF パケットの数をカウントします。SOF パケットの受信時、これらのビットはクリアされます。

ビット 10:0 **FN[10:0]** : フレーム番号

このビットフィールドは、最後に受信された SOF パケットに含まれる 11 ビットのフレーム番号を含みます。フレーム番号は、ホストによって送信されたフレームごとにインクリメントされ、アイソクロナス転送の場合に便利です。このビットフィールドは、SOF 割込みの生成時に更新されます。

USB デバイスアドレス (USB_DADDR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD[6:0]						
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **EF** : 機能有効

このビットは、USB デバイスを有効にするために、ソフトウェアによってセットされます。このデバイスのアドレスは、次の ADD[6:0] ビットに含まれます。このビットが 0 の場合、USB_CHEPnR レジスタの設定にかかわらず、トランザクションは処理されません。

ビット 6:0 **ADD[6:0]** : デバイスアドレス

– デバイスモード

これらのビットは、エニュメレーションプロセス時にホスト PC によって割り当てられた USB 機能を含みます。必要なエンドポイントへのトランザクションを処理するためには、関連する USB_CHEPnR レジスタのこのフィールドとエンドポイント/チャネルアドレス (EA) フィールドの両方とも、USB トークンに含まれる情報に一致する必要があります。

– ホストモード

これらのビットには、LPM トランザクションと一緒に送信されたアドレスが格納されます。

LPM 制御およびステータスレジスタ (USB_LPMCSR)

アドレスオフセット : 0x54

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]				REM WAKE	Res.	LPM ACK	LPM EN
								r	r	r	r	r		rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **BESL[3:0]** : BESL 値

– デバイスモード

これらのビットは、最後に ACK された LPM トークンとともに受信した BESL 値を含みます。

ビット 3 **REM WAKE** : bRemoteWake 値

– デバイスモード

このビットは、最後に ACK された LPM トークンとともに受信した bRemoteWake 値を含みます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **LPMACK** : LPM トークン確認応答有効化

– デバイスモード :

0 : 有効な LPM トークンは NYET です。

1 : 有効な LPM トークンは ACK です。

NYET/ACK は LPM トランザクションが成功した場合にのみ返されます。

EXT トークンと LPM トークンの両方にエラーがない場合 (それ以外の場合は ERROR)

有効な bLinkState = 0001B (L1) を受信した場合 (それ以外の場合は STALL)

ビット 0 **LPMEN** : LPM サポートイネーブル

– デバイスモード

このビットは、USB デバイス内の LPM サポートを有効にするために、ソフトウェアによってセットされます。このビットが 0 の場合は、LPM トランザクションは処理されません。

バッテリー充電検出回路 (USB_BCDR)

アドレスオフセット : 0x58

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPPU_DPD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DC DET	SDEN	PDEN	DCD EN	BCD EN
rw								r	r	r	r	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DPPU_DPD** : DP Pull-up/DPDM Pull-down

– デバイスモード

このビットは、DP ライン上の組み込みプルアップを有効にするために、ソフトウェアによってセットされます。0 にクリアすると、ユーザのソフトウェアが必要とするときに、ホストとの接続が切断されたことを通知します。

– ホストモード

このビットは、DP および DM ライン上の組み込みプルダウンを有効にするために、ソフトウェアによってセットされます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **PS2DET** : DM プルアップ検出ステータス

– デバイスモード

このビットは、PD 中のみアクティブになって DM 電圧レベルと V_{LGC} 閾値の比較の結果を示します。通常の状況では、DM レベルは閾値を下回る必要があります。閾値を上回る場合は、DM が外部的にハイレベルにプルアップされることを意味します。これは、PS2 ポート (DP および DM ラインの両方をプルアップする) への接続、または BCD 仕様に準拠しない一部の独自の充電器への接続によって発生します。

0 : 通常のポートが検出されました (SDP、ACA、CDP、または DCP に接続されたポート)。

1 : PS2 ポートまたは独自の充電器が検出されました。

ビット 6 **SDET** : 2 次検出 (SD) ステータス

– デバイスモード

このビットは、SD の結果を示します。

0 : CDP が検出されました。

1 : DCP が検出されました。

ビット 5 **PDET** : 1 次検出 (PD) ステータス

– デバイスモード

このビットは、PD の結果を示します。

0 : BCD サポートは検出されませんでした (SDP または独自デバイスに接続)。

1 : BCD のサポートが検出されました (ACA、CDP、または DCP に接続)。

ビット 4 **DCDET** : データ接触検出 (DCD) ステータス

– デバイスモード

このビットは、DCD の結果を示します。

0 : データラインの接触は検出されませんでした。

1 : データラインの接触が検出されました。

ビット 3 **SDEN** : 2 次検出 (SD) モードイネーブル

– デバイスモード

このビットは、BCD を SD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 2 **PDEN** : 1 次検出 (PD) モードイネーブル

– デバイスモード

このビットは、BCD を PD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 1 **DCDEN** : データ接触検出 (DCD) モードイネーブル

– デバイスモード

このビットは、BCD を DCD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 0 **BCDEN** : バッテリ充電検出 (BCD) イネーブル

– デバイスモード

このビットは、USB デバイス内の BCD サポートを有効にするために、ソフトウェアによってセットされます。有効にすると、USB PHY は BCD によって完全に制御され、通常の通信では使用できなくなります。BCD の検出完了後は、通常の USB 動作を可能にするために、このビットを 0 にクリアして BCD を OFF モードにします。

ホストチャンネル固有/デバイスエンドポイント固有のレジスタ

これらのレジスタの数は、USB ペリフェラルが処理する設計になっているエンドポイントまたはホストチャンネルの数に応じて変化します。USB ペリフェラルは、最大 8 つの双方向エンドポイントまたはホストチャンネルをサポートします。各 USB デバイスは、制御エンドポイント/チャンネルをサポートしなければならない、そのアドレス (EA ビット) は 0 にセットされなければなりません。複数のエンドポイントが有効であり、同じエンドポイント/チャンネル番号値を持つ場合、USB ペリフェラルの動作は未定義です。各エンドポイントについて、USB_CHEPnR レジスタはエンドポイント/チャンネル固有の情報を格納できます。

USB エンドポイント/チャンネルの n レジスタ (USB_CHEPnR)

アドレスオフセット : $0x00 + 0x4 \cdot n$ 、($n = 0 \sim 7$)

リセット値 : $0x0000\ 0000$

USB リセットが USB バスから受信されたとき、または CTRL レジスタのビット USBRST を通じて強制されたときにもリセットされますが、USB リセットイベントの直前の正しいパケット通知の喪失を回避するために、VTRX および VTTX ビットは変更されません。各エンドポイント/チャンネルに USB_CHEPnR レジスタがあり、 n はエンドポイント/チャンネル識別子です。

これらのレジスタでは読出し-変更-書込みのサイクルは避ける必要があります。読出し操作と書込み操作の間に、いくつかのビットがハードウェアによってセットされる可能性があり、CPU に変更を

検出する時間ができる前に、次の書込みによって変更される可能性があるためです。この目的のために、この問題の影響を受けるすべてのビットが「不変」の値を持ち、変更が不要なときに使用される必要があります。これらのレジスタはロード命令付きで変更することが推奨され、この場合、ハードウェアによって変更できるすべてのビットには「不変」値が書き込まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	THREE_ERR_RX[1:0]		THREE_ERR_TX[1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]						
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	rc_w0	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTRX	DTOG_RX	STATRX[1:0]		SETUP	UTYPE[1:0]		EP_KIND	VTTX	DTOG_TX	STATTX[1:0]		EA[3:0]			
rc_w0	t	t	t	r	rw	rw	rw	rc_w0	t	t	t	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **THREE_ERR_RX[1:0]** : IN トランザクションの 3 個のエラー

– ホストモード

このビットは、IN トランザクションに対して USB バスでトランザクションエラーが 3 回連続して発生した場合に、ハードウェアによってセットされます。アイソクロナストランザクションでは、THREE_ERR_RX は生成されません。このビットは、ソフトウェアのみがクリアできます。

受信したエラーのコーディング :

00 : 受信したエラーが 3 個未満。

01 : 受信したエラーが 3 個を超え、最後のエラーがタイムアウトエラー。

10 : 受信したエラーが 3 個を超え、最後のエラーがデータエラー (CRC エラー)。

11 : 受信したエラーが 3 個を超え、最後のエラーがプロトコルエラー (無効な PID、偽の EOP、ビットスタッフィングエラー、SYNC エラー)。

ビット 28:27 **THREE_ERR_TX[1:0]** : OUT または SETUP トランザクションの 3 つのエラー

– ホストモード

このビットは、OUT トランザクションに対して USB バスでトランザクションエラーが 3 回連続して発生した場合に、ハードウェアによってセットされます。アイソクロナストランザクションでは、THREE_ERR_TX は生成されません。このビットは、ソフトウェアのみがクリアできます。

受信したエラーのコーディング :

00 : 受信したエラーが 3 個未満。

01 : 受信したエラーが 3 個を超え、最後のエラーがタイムアウトエラー。

10 : 受信したエラーが 3 個を超え、最後のエラーがデータエラー (CRC エラー)。

11 : 受信したエラーが 3 個を超え、最後のエラーがプロトコルエラー (無効な PID、偽の EOP、ビットスタッフィングエラー、SYNC エラー)。

ビット 26 **ERR_RX** : 受信した IN トランザクションのエラー

– ホストモード

このビットは、このチャンネルで IN トランザクション中にエラー (デバイスからの応答なし、CRC エラー、ビットスタッフィングエラー、フレーミングフォーマット違反など) が発生したときに、ハードウェアによってセットされます。このビットは、ソフトウェアのみがクリアできます。USB_CNTR レジスタの ERRM ビットがセットされた場合、チャンネル関連のフラグ (常に有効化されます) とともに汎用割り込み条件が生成されます。

ビット 25 **ERR_TX** : 受信した OUT/SETUP トランザクションのエラー

– ホストモード

このビットは、このチャンネルで OUT または SETUP トランザクション中にエラー (デバイスからの応答なし、CRC エラー、ビットスタッフィングエラー、フレーミングフォーマット違反など) が発生したときに、ハードウェアによってセットされます。このビットは、ソフトウェアのみがクリアできます。USB_CNTR レジスタの ERRM ビットがセットされた場合、チャンネル関連のフラグ (常に有効化されます) とともに汎用割り込み条件が生成されます。

ビット 24 **LS_EP** : ロースピードエンドポイントーハブのみを持つホスト

– ホストモード

このビットは、対応するエンドポイントに LS トランザクションを送信するために、ソフトウェアによってセットされます。

0 : フルスピードエンドポイント

1 : ロースピードエンドポイント

ビット 23 **NAK** :

– ホストモード

このビットは、デバイスが NAK により応答したときに、ハードウェアによってセットされます。ソフトウェアでは、このビットを使用して、デバイスから受信した NAK の数を監視することができます。

ビット 22:16 **DEVADDR[6:0]** :

– ホストモード

エニューメレーションプロセス時にエンドポイントに割り当てられたデバイスアドレス。

ビット 15 **VTRX** : USB 有効トランザクションの受信

– デバイスモード

このビットは、このエンドポイントで OUT/SETUP トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件（常に有効化されます）とともに汎用割り込み条件が生成されます。発生したトランザクションのタイプ（OUT または SETUP）は、下記の SETUP ビットから決めることができます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

– ホストモード

このビットは、このチャンネルで IN トランザクションが正常に完了したときに、ハードウェアによってセットされます。このビットは、ソフトウェアのみがクリアできます。USB_CNTR レジスタの CTRM ビットがセットされた場合、チャンネル関連のフラグ（常に有効化されます）とともに汎用割り込み条件が生成されます。

- トランザクションが NAK で終了した場合、このビットがセットされ、このレジスタの STATRX フィールドから NAK 状態が読み出されてアプリケーションに NAK 応答がレポートされます。NAK 応答されたトランザクションは保留中となり、次のフレームでホストによって自動的に再試行されます。または、STATRX 状態を VALID にリセットすることにより、ホストですぐに再試行することもできます。

- トランザクションが STALL ハンドシェイクで終了した場合、このビットがセットされ、このレジスタの STATRX フィールドから STALL 状態が読み出されてアプリケーションに STALL 応答がレポートされます。最終的に、ホストアプリケーションによりチャンネルが無効化され、再エニューメレーションされます。

- トランザクションが ACK ハンドシェイクで終了した場合、このビットがセットされます。

ダブルバッファリングが無効になっている場合、このレジスタの STATRX フィールドから DISABLE 状態が読み出され、アプリケーションにより ACK 応答がレポートされます。ホストアプリケーションにより、受信したデータが USBRAM から読み出され、このレジスタの STATRX フィールドに VALID を書き込むことにより、チャンネルが再設定されます。

ダブルバッファリングが有効になっている場合、このレジスタの STATRX フィールドから VALID 状態が読み出され、アプリケーションにより ACK 応答がレポートされます。ホストアプリケーションにより、受信したデータが USBRAM から読み出され、このレジスタの DTOGTX ビットがトグルされます。

- トランザクションがエラーで終了した場合、このビットがセットされます。

エラーは ERR_RX ビットを通じて確認できます（ホストモードのみ）。

このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 14 **DTOGRX** : 受信転送の場合のデータトグル

エンドポイント/チャネルがアイソクロナスでない場合、このビットには次の受信データパケットのデータトグルビット (0 = DATA0、1 = DATA1) の予期される値が格納されます。一致するデータ PID 値を持つデータパケット受信に続いて ACK ハンドシェイクが送信されると、ハードウェアによってこのビットがトグルされます。エンドポイントが制御エンドポイントとして定義される場合、ホストから受信した SETUP PID の受信時にハードウェアによってこのビットがクリアされ (デバイスモード)、デバイスによって SETUP トランザクションが確認応答される場合はハードウェアによってこのビットが 1 にセットされます (ホストモード)。

エンドポイント/チャネルがダブルバッファリングを使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます ([セクション 55.5.3 : ダブルバッファエンドポイントとデバイスモードでの使用](#)を参照)。

エンドポイント/チャネルがアイソクロナスの場合、この種類のチャネル/エンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはデータ転送用のパケットバッファスワッピングをサポートするためにのみ使用されます ([セクション 55.5.5 : デバイスモードでのアイソクロナス転送](#)を参照)。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット受信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため (エンドポイントが制御エンドポイントでないときは必須)、または特定のデータトグル/パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOGRX の値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。

ビット 13:12 **STATRX[1:0]** : 受信転送の場合のステータスビット

– デバイスモード

これらのビットには、表 598 : 2586 ページの受信ステータスエンコードにリストされているエンドポイントのステータスに関する情報が格納されます。これらのビットは、ソフトウェアによってトグルして、値を初期化できます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このエンドポイントにアドレス指定された OUT または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (VTRX = 1)、STATRX ビットを NAK にセットするので、ソフトウェアは新しいトランザクションに確認応答する前に受信データを処理する時間があります。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します (セクション 55.5.3 : ダブルバッファエンドポイントとデバイスモードでの使用を参照)。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみなので、ハードウェアはトランザクションの成功後にエンドポイントのステータスを変更できません。ソフトウェアがアイソクロナスエンドポイントの STATRX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読出し/書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

– ホストモード

これらのビットは、チャンネルによって駆動されるホストトランザクションを開始、再試行、または中断するためのホストアプリケーション制御です。

これらのビットには、前回の IN チャンネルトランザクションに対するデバイス応答の情報も格納され、これらのビットでは次の状態の STATRX テーブルに従ってチャンネルの現在のステータスをレポートします。

- DISABLE

シングルバッファチャンネルで ACK 確認応答を受信した場合、DISABLE 値がレポートされます。DISABLE 状態の場合、チャンネルは未使用または非アクティブの状態で、VALID を書き込むことにより、アプリケーションで再起動されるのを待っています。アプリケーションは VALID のチャンネルを DISABLE にリセットすることにより、トランザクションを中断することができます。この場合、トランザクションはすぐにホスト実行リストから削除されます。中断するトランザクションがすでに実行中になっている場合、USB 上で定期的に終了しますが、相対 VTRX 割込みは生成されません。

- VALID

ホストチャンネルは、VALID 状態のときのみ、積極的にデバイスに USB トランザクションを送信しようとしします。VALID 状態はソフトウェアにより、または新しいフレームの開始時に NAKED チャンネルでハードウェアにより自動的にセットできます。VALID にセットされると、ホストチャンネルはホスト実行キューに入り、設定されたトランザクションを送信するためにホストフレームスケジューラからの許可を待ちます。

ダブルバッファチャンネルで ACK 確認応答を受信した場合、VALID 値もレポートされます。この場合、チャンネルはオルタネートバッファ上でアクティブな状態のままとなりますが、アプリケーションでは現在のバッファを読み出して DTOGTX をトグルする必要があります。ソフトウェアでの読出しに時間がかかり、オルタネートバッファが準備完了になっていない場合、ホストチャンネルはアプリケーションの関与なしで自動的にサスペンドされます。サスペンドされたダブルバッファチャンネルは、遅延から復旧し、DTOGTX がトグルされるとすぐに再有効化されます。

- NAK

NAK 値は、NAK 確認応答を受信した場合にレポートされます。NAK 状態の場合、チャンネルはサスペンドされ、送信は行いません。NAK 状態は、次のフレームの開始時にハードウェアにより VALID に移行されます。あるいは、ソフトウェアにより VALID として書き込むと、すぐに送信を再試行するように変更でき、DISABLE として書き込むと、無効化し、トランザクションを中断することができます。

- STALL

STALL 値は、STALL 確認応答を受信した場合にレポートされます。STALL 状態の場合、チャンネルは無効として動作します。アプリケーションにより送信の再試行は行われませんが、USB がリセットされ、再エニュメレーションされます。

ビット 11 **SETUP** : セットアップトランザクション完了

– デバイスモード

このビットは読み出し専用であり、最後に完了したトランザクションが **SETUP** のときにハードウェアによってセットされます。このビットはコントロールエンドポイントについてのみ値を変更します。成功した受信トランザクションの場合 (VTRX イベント)、これを調べて、発生したトランザクションのタイプを判断する必要があります。割り込みサービスルーチンを次の着信トークンによる **SETUP** ビットの変更から保護するために、このビットは VTRX ビットが 1 の間は停止され、VTRX が 0 のときに状態が変更されます。このビットは読み出し専用です。

– ホストモード

このビットは、制御エンドポイントで **SETUP** トランザクションを送信するために、ソフトウェアによってセットされます。このビットはコントロールエンドポイントについてのみ値を変更します。**SETUP** トランザクションが確認応答され、VTTX 割り込みが生成されると、ハードウェアによってクリアされます。

ビット 10:9 **UTYPE[1:0]** : トランザクションの USB タイプ

これらのビットは、表 599: エンドポイント/チャネルタイプのエンコードに記載されているように、このエンドポイント/チャネルの動作を設定します。チャネル 0/エンドポイント 0 は常に制御エンドポイント/チャネルでなければならない、各 USB 機能にはアドレス 0 を持つ制御エンドポイント/チャネルが少なくとも 1 つ必要ですが、必要な場合は、他の制御チャネル/エンドポイントを使用することもできます。制御チャネル/エンドポイントだけが **SETUP** トランザクションを処理し、他の種類のエンドポイントでは無視されます。**SETUP** トランザクションは NAK または STALL で応答することはできません。制御エンドポイント/チャネルが NAK として定義された場合、**SETUP** トランザクションが受信されたときには、受信方向では USB ペリフェラルは応答せず、受信エラーをシミュレートします。制御エンドポイント/チャネルが受信方向で STALL として定義された場合、**SETUP** パケットは受け入れられ、データを送信し、CTR 割り込みを発行します。エンドポイント/チャネルが制御エンドポイントの場合でも、OUT トランザクションの受信は通常通りに処理されます。

バルクおよび割り込みエンドポイントの動作はほぼ同様ですが、EPKIND 設定ビットを使用して使用可能な特殊な機能のみが異なります。

アイソクロナスチャネル/エンドポイントの使用については、セクション 55.5.5: デバイスモードでのアイソクロナス転送で説明します。

ビット 8 **EPKIND** : エンドポイント/チャネルの種類

このビットの意味は、UTYPE ビットによって設定されたエンドポイント/チャネルのタイプに依存します。表 600 はさまざまな意味を要約します。

DBL_BUF : このビットは、このバルクエンドポイントのダブルバッファリング機能を有効にするために、ソフトウェアによってセットされます。ダブルバッファバルクエンドポイントの使用については、セクション 55.5.3: ダブルバッファエンドポイントとデバイスモードでの使用で説明します。

STATUS_OUT : このビットは、ステータスアウトトランザクションが予期されることを示すためにソフトウェアによってセットされます。この場合、1 個以上のデータバイトを含んでいるすべての OUT トランザクションは、ACK ではなく STALL で応答されます。このビットを使用して、制御転送時のプロトコルエラーに対するアプリケーションの堅牢性を向上させることができ、その使用はコントロールエンドポイントのみを対象としています。STATUS_OUT がリセットされると、OUT トランザクションは必要に応じて任意の数のバイトを持つことができます。

ビット 7 **VTTX** : 送信される有効な USB トランザクション

– デバイスモード

このビットは、このエンドポイントで IN トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件 (常に有効化されます) とともに汎用割り込み条件が生成されます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し/書き込みですが、0 のみを書き込むことができます。

– ホストモード

VTRX の動作と同様ですが、USB OUT および **SETUP** トランザクション用です。

ビット 6 DTOGTX : 送信転送の場合のデータトグル

エンドポイント/チャンネルがアイソクロナスでない場合、このビットには次の送信データパケットのデータトグルビット (0 = DATA0、1 = DATA1) の必要な値が格納されます。データパケットの送信後、USB ホストから ACK ハンドシェイクが受信されると、ハードウェアはこのビットをトグルします。エンドポイント/チャンネルが制御エンドポイント/チャンネルとして定義されている場合、このエンドポイントにアドレス指定された SETUP PID の受信時 (デバイスモード)、またはデバイスによって SETUP トランザクションが確認応答されたときに (ホストモード)、このビットはハードウェアによって 1 にセットされます。

エンドポイント/チャンネルがダブルバッファ機能を使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます ([セクション 55.5.3 : ダブルバッファエンドポイントとデバイスモードでの使用](#)を参照)。

エンドポイント/チャンネルがアイソクロナスの場合、この種類のエンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはパケットバッファスワッピングをサポートするために使用されます ([セクション 55.5.5 : デバイスモードでのアイソクロナス転送](#)を参照)。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット送信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため (エンドポイント/チャンネルが制御エンドポイント/チャンネルでないときは必須)、または特定のデータトグル/パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOGTX 値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し/書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

ビット 5:4 STATTX[1:0] : 送信転送の場合のステータスビット**– デバイスモード**

これらのビットは、[表 601](#)にリストされているエンドポイントのステータスに関する情報を含みます。これらのビットは、ソフトウェアによってトグルして、値を初期化できます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このチャンネル/エンドポイントにアドレス指定された IN または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (VTTX = 1)、STATTX ビットを NAK にセットします。次に、ソフトウェアが次の送信データセットを準備するのを待ちます。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します ([セクション 55.5.3 : ダブルバッファエンドポイントとデバイスモードでの使用](#)を参照)。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみです。したがって、ハードウェアは、トランザクションの成功後にチャンネル/エンドポイント/チャンネルのステータスを変更することはできません。ソフトウェアがアイソクロナスチャンネル/エンドポイントの STATTX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読み出し/書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

– ホストモード

STATTX ビットにはチャンネルのステータスに関する情報が格納されます。詳細については、[表 601](#) (「ホストモード」の説明) を参照してください。デバイスモードではこれらのビットには次のトランザクションで割り当てられるステータスが格納されますが、ホストモードではデバイスから前回受信したステータスが格納されます。NAK を受信した場合、STATTX には NAK を示す値が格納されます。

ビット 3:0 EA[3:0] : エンドポイント/チャンネルアドレス**– デバイスモード**

ソフトウェアは、このエンドポイント宛でのトランザクションを識別するために使用される 4 ビットのアドレスをこのフィールドに書き込む必要があります。値は、対応するエンドポイントを有効にする前に書き込まれる必要があります。

– ホストモード

ソフトウェアは、ホストトランザクションによってアドレス指定されたチャンネルを識別するために使用される 4 ビットのアドレスをこのフィールドに書き込む必要があります。

表 598. 受信ステータスエンコード

STATRX[1:0]	意味
00	DISABLED : このエンドポイント/チャンネルにアドレス指定されたすべての受信リクエストは無視されます。
01	STALL : デバイスモード : エンドポイントは停止していて、すべての受信リクエストは STALL ハンドシェイクで終了します。 ホストモード : デバイスでチャンネルが停止していることを示します。
10	NAK : デバイスモード : エンドポイントは NAK され、すべての受信リクエストは NAK ハンドシェイクで終了します。 ホストモード : デバイスで受信リクエストが NAK していることを示します。
11	VALID : このエンドポイント/チャンネルは受信に有効です。

表 599. エンドポイント/チャンネルタイプのエンコード

UTYPE[1:0]	意味
00	BULK
01	CONTROL
10	ISO
11	INTERRUPT

表 600. エンドポイント/チャンネルの種類の意味

UTYPE[1:0]		EPKIND の意味
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT
10	ISO	SBUF_ISO : このビットは、アイソクロナスエンドポイントのシングルバッファリング機能を有効にするために、ソフトウェアによってセットされます
11	INTERRUPT	未使用

表 601. 送信ステータスエンコード

STATTX[1:0]	意味
00	DISABLED : このエンドポイント/チャンネルにアドレス指定されたすべての送信リクエストは無視されます。
01	STALL : デバイスモード : エンドポイントは停止していて、すべての送信リクエストは STALL ハンドシェイクで終了します。 ホストモード : デバイスでチャンネルが停止していることを示します。

表 601. 送信ステータスエンコード (続き)

STATTX[1:0]	意味
10	NAK : デバイスモード: エンドポイントは NAK され、すべての送信リクエストは NAK ハンドシェイクで終了します。 ホストモード: デバイスで送信リクエストが NAK していることを示します。
11	VALID : このエンドポイント/チャネルは送信に有効です。

55.6.2 バッファディスクリプタテーブル

注： バッファディスクリプタテーブルは、別の「USB SRAM」アドレス空間のポケットバッファメモリ内に位置しています。

バッファディスクリプタテーブルはポケットバッファメモリ（「USB SRAM」領域）内にありますが、そのエントリは、USB マクロセルとデバイス間でデータを交換するためのポケットバッファの位置とサイズを設定する追加レジスタとみなすことができます。

最初のポケットメモリ位置は、USB SRAM ベースアドレスにあります。USB_CHEPnR レジスタに関連するバッファディスクリプタテーブルのエントリについて、以下に説明します。メモリはワード（32 ビット）アクセスを使用してアドレス指定する必要があります。

ポケットバッファとバッファディスクリプタテーブルの使用に関する詳細については、[2554 ページのポケットバッファの構造と用途](#)を参照してください。

チャネル/エンドポイント送信バッファディスクリプタ n (USB_CHEP_TXRXBD_n)

アドレスオフセット：n*8

このレジスタの説明が適用されるのは、該当する CHEPnR レジスタにより、受信モードで動作するダブルバッファリングの使用がプログラムされない場合です（それ以外の場合は、次のレジスタの説明を参照してください）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	COUNT_TX[9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_TX[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:16 **COUNT_TX[9:0]**：送信バイトカウント

これらのビットには、アドレス指定された次の IN トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって送信されるバイト数が格納されます。

ビット 15:0 **ADDR_TX[15:0]**：送信バッファアドレス

これらのビットは、アドレス指定された次の IN トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって送信されるデータが格納されたポケットバッファの開始アドレスを指します。ポケットメモリはワード長であり、すべてのポケットバッファがワードで揃えられている必要があります。そのため、ビット 1 および 0 には常に 00 が書き込まれる必要があります。

チャネル/エンドポイント受信バッファディスクリプタ n [オルタネート] (USB_CHEP_TXRXBD_n)

アドレスオフセット：n*8

このレジスタの説明が適用されるのは、該当する CHEPnR レジスタにより、ダブルバッファリングの使用がプログラムされ、受信バッファが有効化される場合です（それ以外の場合は、前のレジスタの説明を参照してください）。

このテーブル位置は、両方ともポケット受信時に必要とされる 2 つの値を格納するために使用されます。最上位ビットは割り当てられるバッファサイズの定義を含み、バッファオーバーフロー検出を可能にし、この位置の最下位部分には、受信終了時に USB ペリフェラルによって実際の受信バイト数が書き込まれます。使用可能なビット数が制限されているため、バッファサイズは割り当てられたメ

メモリブロック数を使用して表され、細かい粒度／小さいバッファと粗い粒度／大きなバッファの間のトレードオフにより、ブロックサイズを選択できます。割り当てられたバッファのサイズは、エンドポイント / チャネルディスクリプタの一部であり、通常、エニュメレーションプロセス時に maxPacketSize パラメータ値に従って定義されます（「ユニバーサルシリアルバス仕様」を参照）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLSIZE	NUM_BLOCK[4:0]					COUNT_RX[9:0]									
r/w	r/w	r/w	r/w	r/w	r/w	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_RX[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 BLSIZE : ブロックサイズ

このビットは、割り当てられたバッファ領域を定義するために使用されるメモリブロックのサイズを選択します。

- BLSIZE =0 の場合、メモリブロックは 2 バイト長であり、これは、ハーフワード長のメモリで可能な最小ブロックです。このブロックサイズでは、割り当てられるバッファサイズは 2 ～ 62 バイトです。
- BLSIZE =1 の場合、メモリブロックは 32 バイト以上であり、USB 仕様によって定義された最大パケット長に達することができます。このブロックサイズでは、割り当てられるバッファサイズは、理論上、32 ～ 1024 バイトであり、USB 標準仕様で可能な最長のパケットサイズです。ただし、適用可能なサイズは、使用可能なバッファメモリによって制限されます。

ビット 30:26 NUM_BLOCK[4:0] : ブロック数

これらのビットは、このパケットバッファに割り当てられるメモリブロック数を定義します。実際の割り当てメモリの量は、表 602 に示されているように、BLSIZE の値に依存します。

ビット 25:16 COUNT_RX[9:0] : 受信バイトカウント

これらのビットには、USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって、アドレス指定された最後の OUT/SETUP トランザクション中に受信されたバイト数が格納されます。

ビット 15:0 ADDR_RX[15:0] : 送信バッファアドレス

これらのビットは、アドレス指定された次の OUT/SETUP トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって受信されるデータが格納されたパケットバッファの開始アドレスを指します。パケットメモリはワード長であり、すべてのパケットバッファがワードで揃えられている必要があるため、ビット 1 および 0 には常に 00 が書き込まれる必要があります。

表 602. 割り当てられるバッファメモリの定義

NUM_BLOCK[4:0] の値	BLSIZE =0 のとき割り当てられるメモリ	BLSIZE =1 のとき割り当てられるメモリ
0 (00000)	設定禁止	32 バイト
1 (00001)	2 バイト	64 バイト
2 (00010)	4 バイト	96 バイト
3 (00011)	6 バイト	128 バイト
.....
14 (01110)	28 バイト	480 バイト
15 (01111)	30 バイト	
16 (10000)	32 バイト	
.....
29 (11101)	58 バイト

表 602. 割り当てられるバッファメモリの定義 (続き)

NUM_BLOCK[4:0] の値	BLSIZE = 0 のとき割り当てられるメモリ	BLSIZE = 1 のとき割り当てられるメモリ
30 (11110)	60 バイト	992 バイト
31 (11111)	62 バイト	1023 バイト

チャネル/エンドポイント受信バッファディスクリプタ n (USB_CHEP_RXTXBD_n)

アドレスオフセット : $n \times 8 + 4$

このレジスタの説明が適用されるのは、該当する CHEPnR レジスタにより、送信モードでダブルバッファリングの使用がプログラムされない場合です（それ以外の場合は、次のレジスタの説明を参照してください）。

このテーブル位置は、両方ともパケット受信時に必要とされる 2 つの値を格納するために使用されます。最上位ビットは割り当てられるバッファサイズの定義を含み、バッファオーバーフロー検出を可能にし、この位置の最下位部分には、受信終了時に USB ペリフェラルによって実際の受信バイト数を書き込まれます。使用可能なビット数が制限されているため、バッファサイズは割り当てられたメモリブロック数を使用して表され、細かい粒度/小さいバッファと粗い粒度/大きなバッファの間のトレードオフにより、ブロックサイズを選択できます。割り当てられたバッファのサイズは、エンドポイント / チャネルディスクリプタの一部であり、通常、エニュメレーションプロセス時に maxPacketSize パラメータ値に従って定義されます（「ユニバーサルシリアルバス仕様」を参照）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLSIZE	NUM_BLOCK[4:0]					COUNT_RX[9:0]									
rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_RX[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 BLSIZE : ブロックサイズ

このビットは、割り当てられたバッファ領域を定義するために使用されるメモリブロックのサイズを選択します。

- BLSIZE = 0 の場合、メモリブロックは 2 バイト長であり、これは、ハーフワード長のメモリで可能な最小ブロックです。このブロックサイズでは、割り当てられるバッファサイズは 2 ~ 62 バイトです。
- BLSIZE = 1 の場合、メモリブロックは 32 バイト以上であり、USB 仕様によって定義された最大パケット長に達することができます。このブロックサイズでは、割り当てられるバッファサイズは、理論上、32 ~ 1024 バイトであり、USB 標準仕様で可能な最長のパケットサイズです。ただし、適用可能なサイズは、使用可能なバッファメモリによって制限されます。

ビット 30:26 **NUM_BLOCK[4:0]** : ブロック数

これらのビットは、このパケットバッファに割り当てられるメモリブロック数を定義します。実際の割り当てメモリの量は、表 602 に示されているように、BLSIZE の値に依存します。

ビット 25:16 **COUNT_RX[9:0]** : 受信バイトカウント

これらのビットには、USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって、アドレス指定された最後の OUT/SETUP トランザクション中に受信されたバイト数が格納されます。

ビット 15:0 **ADDR_RX[15:0]** : 送信バッファアドレス

これらのビットは、アドレス指定された次の OUT/SETUP トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって受信されるデータが格納されたパケットバッファの開始アドレスを指します。パケットメモリはワード長であり、すべてのパケットバッファがワードで揃えられている必要があるため、ビット 1 および 0 には常に 00 が書き込まれる必要があります。

チャネル/エンドポイント送信バッファディスクリプタ n [オルタネート] (USB_CHEP_RXTXBD_n)

アドレスオフセット : $n \times 8 + 4$

このレジスタの説明が適用されるのは、該当する CHEPnR レジスタにより、ダブルバッファリングの使用がプログラムされ、送信バッファが有効化される場合です（それ以外の場合は、前のレジスタの説明を参照してください）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	COUNT_TX[9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_TX[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:16 **COUNT_TX[9:0]** : 送信バイトカウント

これらのビットには、アドレス指定された次の IN トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって送信されるバイト数が格納されます。

ビット 15:0 **ADDR_TX[15:0]** : 送信バッファアドレス

これらのビットは、アドレス指定された次の IN トークンで USB_CHEPnR レジスタに関連するエンドポイント/チャネルによって送信されるデータが格納されたパケットバッファの開始アドレスを指します。パケットメモリはワード長であり、すべてのパケットバッファがワードで揃えられている必要があるため、ビット 1 および 0 には常に 00 が書き込まれる必要があります。

55.6.3 USB レジスタマップ

次の表に、USB レジスタマップとリセット値を示します。

表 603. USB レジスタのマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	USB_CHEP0R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	USB_CHEP1R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	USB_CHEP2R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	USB_CHEP3R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	USB_CHEP4R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	USB_CHEP5R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	USB_CHEP6R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	USB_CHEP7R	Res.	THREE_ERR_RX [1:0]		THREE_ERR_TX [1:0]		ERR_RX	ERR_TX	LS_EP	NAK	DEVADDR[6:0]							VTRX	DTGRX	STATRX [1:0]	SETUP	UTYPE [1:0]		EPKIND	VTTX	DTGTX	STATTX [1:0]	EA[3:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20～ 0x3F	予約済みです。																																	
0x40	USB_CNTR	HOST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDISCM	THR512M	CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RST_DCONM	SOFM	ESOFM	L1REQM	Res.	L1RES	L2RES	SUSPEN	SUSPRDY	PDWN	USBRST
	リセット値	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		

表 603. USB レジスタのマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44	USB_ISTR	Res.	LS_DCON	DCON_STAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDISC	THR512	CTR	PMAOVR	ERR	WKUP	SUSP	RST_DCON	SOF	ESOF	L1REQ	Res.	Res.	DIR	IDN[3:0]			
	リセット値		0	0												0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0
0x48	USB_FNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDP	RXDM	LCK	LSOF [1:0]	FN[10:0]												
	リセット値																0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x
0x4C	USB_DADDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD[6:0]							
	リセット値																								0	0	0	0	0	0	0	0	0
0x54	USB_LPMCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]			REMWAKE	Res.	LPMACK	LPMEN		
	リセット値																								0	0	0	0	0	0	0	0	0
0x58	USB_BCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPDP_DP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSDET	SDET	PDET	DCDET	SDEN	PDEN	DCDEN	BCDEN	
	リセット値																0								0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

56 USB Type-C[®]/USB Power Delivery インタフェース (UCPD)

56.1 概要

USB Type-C/USB Power Delivery インタフェースの構成要素：

- Universal Serial Bus Type-C ケーブルおよびコネクタ仕様：リリース 2.2、2022 年 10 月
- Universal Serial Bus Power Delivery 仕様：
 - リビジョン 2.0、バージョン 1.3、2017 年 1 月 12 日
 - リビジョン 3.1、バージョン 1.6、2022 年 10 月

Type-C ケーブル動作の、CC 信号手法 (VBUS なし) による Power Delivery (PD) 仕様の物理層を統合します。

56.2 UCPD の主な機能

- USB Type-C 仕様リリース 2.2 に準拠
- USB Power Delivery 仕様リビジョン 2.0 および 3.1 に準拠
 - PPS などの高度なアプリケーションを有効化 (プログラム可能な電源機器)
- STOP モードの低電力動作のサポート
- 内蔵アナログ PHY
 - USB Type-C プルアップ (Rp、全ての値) およびプルダウン (Rd) レジスタ
 - 『デッドバッテリー』Rd サポート
 - USB Power Delivery のメッセージの送受信
 - FRS (ファストロールスワップ) Rx のサポート
- デジタル・コントローラ
 - BMC (バイフェーズマークコーディング) のエンコードおよびデコード
 - 4b5b でのエンコードおよびデコード
 - 割込みが生成されるデバウンスによる USB Type-C レベル検出
 - FRS 信号
 - 割込みが生成される FRS 検出
 - 割込みが生成される USB Power Delivery ペイロードの DMA 互換のバイトレベルインタフェース
 - USB Power Delivery クロックプリスケアラ/分周回路
 - CRC 生成/チェック
 - 順序集合 (受信時のプログラム可能な順序集合のマスクを含む) のサポート
 - 受信 Rx ストリームからのクロックリカバリ

56.3 UCPD の実装

デバイスには、1 つの USB タイプ C ポートに対応する 1 つの UCPD コントローラが装備されています。

表 604. UCPD の実装⁽¹⁾

UCPD の機能	UCPD1
UCPDx_DBCC1 および UCPDx_DBCC2 外部信号によるデッドバッテリーのサポート	X
オルタネート機能ピンとしての UCPDx_FRSTX	X
全自動のトリミング	_(2)
USB PD レシーバのハードウェアフィルタ制御	X
ディスクリートコンポーネントである PHY のサポート	-

1. “X” はサポートされています。“-” はサポートされていません。

2. [セクション 56.5.5 : UCPD ソフトウェアのトリミング](#)で説明するソフトウェアのトリミングを適用します。

以下の表に、不揮発性メモリに格納されたトリムデータのメモリ位置を示します。これは、[セクション 56.5.5:UCPD ソフトウェアのトリミング](#)で説明するソフトウェアのタイミング手順で使用するものです。

表 605. UCPD ソフトウェアの trim データ

名前	不揮発性メモリ位置	
	アドレス	ビット
3A0_CC1 [3:0]	0x4002 242C	7:4
3A0_CC2 [3:0]	0x4002 242C	15:12
1A5_CC1 [3:0]	0x08FF F844	3:0
1A5_CC2 [3:0]	0x08FF F844	19:16
Rd_CC1 [3:0]	0x4002 242C	3:0
Rd_CC2 [3:0]	0x4002 242C	11:8

56.4 UCPD の機能詳細

UCPD ペリフェラルでは、USB Power Delivery 制御インタフェース仕様のためのハードウェアサポートを行うため、その目的で特別に設計された I/O を使用します。

内蔵 PHY では、Type-C 電圧レベルを直接検出し、Power Delivery BIST キャリアモード 2 (Tx のみ)、BIST テストデータ (Tx および Rx)、Power Delivery Rx FRS 信号をサポートします。

Power Delivery FRS Tx 信号の場合、UCPD_FRSTX ピン (オルタネート機能)、CC ラインでの低抵抗プルダウンを保証する外部 NMOS トランジスタを通してデバイスの制御を設定できます。

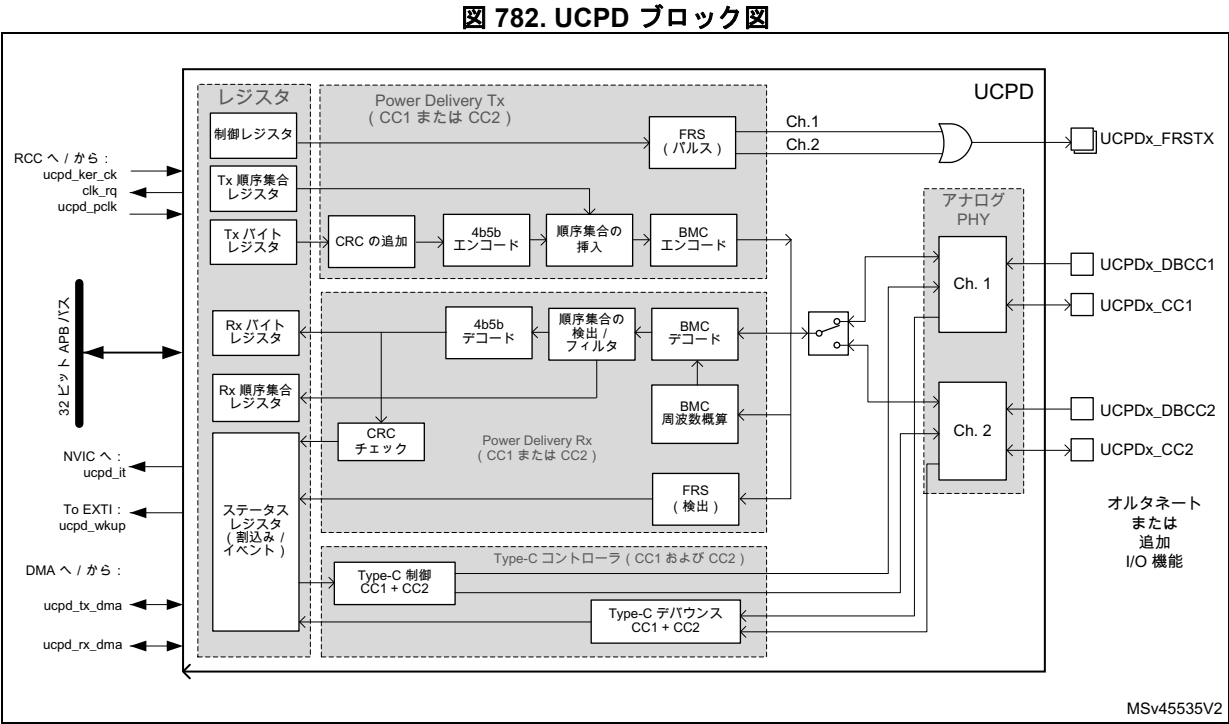
UCPD トランスミッタ BMC (バイフェーズマーク) はデータ、たとえばプリアンブル、SOP、プロトコル層からのペイロードデータ (4b5b エンコーディング後)、CRC、および Type-C コネクタ CC ライン上の EOP をエンコードし、送信します。自動的にフレーム間ギャップを挿入し、「Hard Reset」を実行します。

UCPD レシーバでは SOP を検出し、受信ストリームを BMC デコードし、プリアンブルを復元し、ペイロードデータを 4b5b デコードし、EOP を検出し、CRC をチェックします。5 つの K コード SOP

と 2 つのリセット順序集合、2 つのソフトウェア定義のパターンの検出も自動的に行います（標準に定義されているとおり、4 つの K コードのうち 3 つのみを正しく受信する）。

STOP モードでは、ペリフェラルは受信 USB Power Delivery メッセージと、低電力動作を可能にする FRS 信号を検出する能力を維持します。

56.4.1 UCPD ブロック図



以下の表に、外部信号（オルタネートまたは追加の I/O 機能）を示します。

表 606.ピンの UCPD 信号

ピン名	信号タイプ	説明
UCPDx_FRSTX	出力	USB Type-C ファストロールスワップ (FRS) 信号制御で、DRP によりのみ適用されます。信号 (アクティブハイ) により外部 NMOS トランジスタが駆動され、アクティブ CC ラインがプルダウンされます。代表的なアプリケーションには 2 つのトランジスタ (CC ラインごとに 1 つずつ) があり、いずれかの NMOS を駆動するための個別の I/O が用意されています。最初、I/O は低駆動 GPIO として設定されます。検出時、Type-C ステートマシンを通して、アクティブ CC ラインを決定するケーブル設置の向き、アクティブ CC ラインの I/O を UCPDx_FRSTX オルタネート機能にセットし、非アクティブ CC ラインの I/O は低駆動 GPIO としてセットする必要があります。
UCPDx_CC1	入力／出力	USB Type-C 設定制御ライン 1 で、USB Type-C コネクタ CC1 端子にルーティングする必要があります。
UCPDx_CC2	入力／出力	USB Type-C 設定制御ライン 2 で、USB Type-C コネクタ CC2 端子にルーティングする必要があります。

表 606. ピンの UCPD 信号 (続き)

ピン名	信号タイプ	説明
UCPDx_DBCC1	入力	USB Type-C 設定制御ライン 1 のデッドバッテリー信号で、デッドバッテリーのサポートが必要な場合、USB Type-C コネクタ CC1 端子にルーティングする必要があります。
UCPDx_DBCC2	入力	USB Type-C 設定制御ライン 2 のデッドバッテリー信号で、デッドバッテリーのサポートが必要な場合、USB Type-C コネクタ CC2 端子にルーティングする必要があります。

次の表に、主要な内部信号を示します。

表 607. UCPD 内部信号

内部信号名	信号タイプ	説明
ucpd_pclk	入力	レジスタ用の APB クロック
ucpd_ker_ck	入力	カーネルクロック
ucpd_tx_dma	入力/出力	Rx DMA 確認応答/リクエスト
ucpd_rx_dma	入力/出力	Tx DMA 確認応答/リクエスト
ucpd_it	出力	NVIC に接続された割り込みリクエスト (すべての割り込みの論理和)
ucpd_wkup	出力	EXTI に接続されたウェイクアップリクエスト
clk_rq	出力	RCC に接続されたクロックリクエスト

56.4.2 UCPD のリセットおよびクロック

ペリフェラルはシングルリセット信号 (APB バスリセット) を備えています。

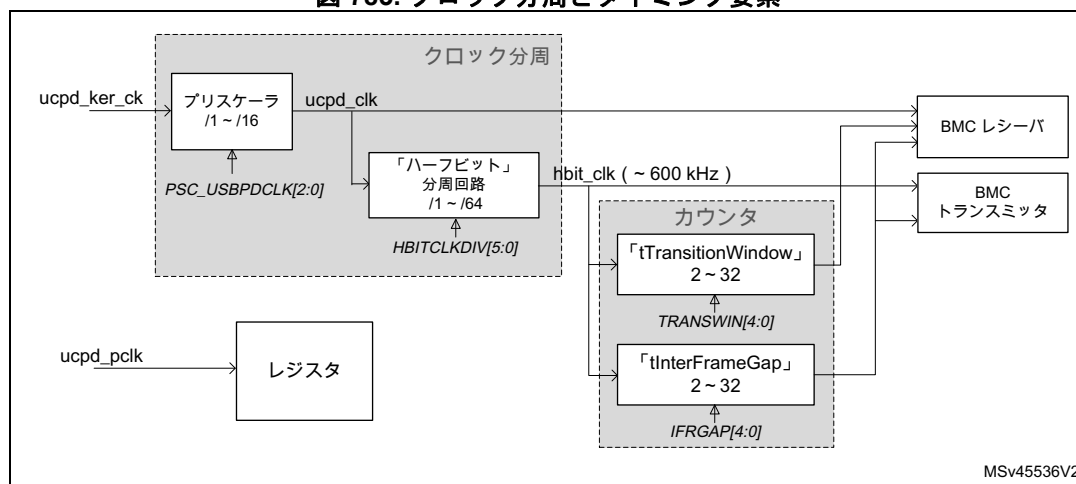
レジスタセクションは、APB クロック (ucpd_pclk) によりクロック供給されます。

トランスミッタのメイン機能部分は、ucpd_clk クロックによりクロック供給され、UCPD_CFGR1 レジスタの PSC_USBDCLK[2:0] ビットフィールドに従って ucpd_ker_ck (HSI16) クロックからプリスケールされます。レシーバのメイン機能部分は、受信ビットストリームから復元された ucpd_rx_clk によりクロック供給されます。

レシーバは、6 ~ 18 MHz のクロック周波数範囲で動作するように設計されています。ただし、最適なパフォーマンスが得られるのは、9 ~ 18 MHz の範囲内です。

下図に、UCPD ペリフェラルのクロック供給およびタイミングの要素を示します。

図 783. クロック分周とタイミング要素



適切な遅延をセットするには、USB PD 仕様を参照してください。tTransitionWindow と、特に tInterFrameGap では、どのような場合でも指定したタイミングを満たすことができるよう、クロック周波数の不確定性を考慮する必要があります。

56.4.3 物理層のプロトコル

物理層は、USB Power Delivery 仕様の基礎となる信号送信をカバーします。

トランスミッタ側では、主な機能は、定義されたパケットフォーマットに従ってパケットを形成することです。一般的に、以下の内容が含まれます。

- プリアンブルは
- パケットの開始 (SOP、順序集合)
- ペイロード・ヘッダ
- ペイロード・データ
- 巡回冗長検査 (CRC) 情報
- パケットの終了 (EOP)

CC ラインに送られる前に、指定されたタイミング制限について、データストリームが BMC エンコードされます。

受信側の基本タスクは、以下の内容を実行することです。

- パケットの開始 (SOP、順序集合) 情報の抽出
- ペイロード・ヘッダの抽出
- ペイロード・データの抽出
- CRC の受信と確認
- パケットの終了 (EOP) の受信

受信は基本的に送信プロセスの逆であるため、BMC データストリームのデコードから開始されます。

記号のエンコーディング

プリアンブルを除いて、記号はすべて以下の表に示す仕様に従って、4b5b スキームでエンコードされます。

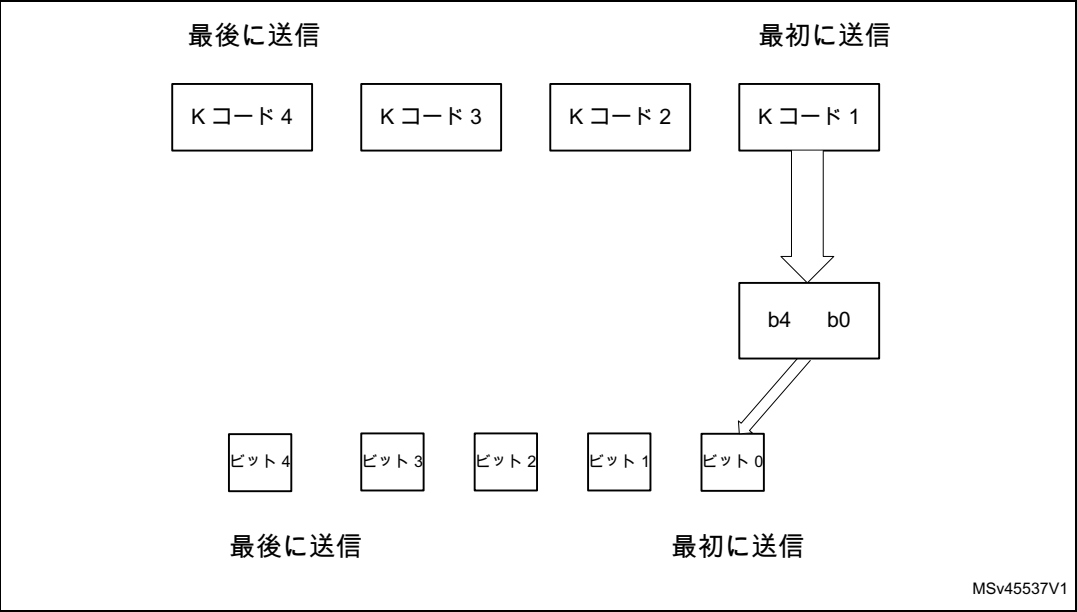
表 608.4b5b 記号エンコード表

名前	4b	5b	記号の説明
0	0000	11110	16 進数データ 0
1	0001	01001	16 進数データ 1
2	0010	10100	16 進数データ 2
3	0011	10101	16 進数データ 3
4	0100	01010	16 進数データ 4
5	0101	01011	16 進数データ 5
6	0110	01110	16 進数データ 6
7	0111	01111	16 進数データ 7
8	1000	10010	16 進数データ 8
9	1001	10011	16 進数データ 9
A	1010	10110	16 進数データ A
B	1011	10111	16 進数データ B
C	1100	11010	16 進数データ C
D	1101	11011	16 進数データ D
E	1110	11100	16 進数データ E
F	1111	11101	16 進数データ F
Sync-1	K コード	11000	Startsynch #1
Sync-2	K コード	10001	Startsynch #2
RST-1	K コード	00111	Hard Reset #1
RST-2	K コード	11001	Hard Reset #2
EOP	K コード	01101	EOP
予約済みです。	誤差	00000	使用不可
予約済みです。	誤差	00001	使用不可
予約済みです。	誤差	00010	使用不可
予約済みです。	誤差	00011	使用不可
予約済みです。	誤差	00100	使用不可
予約済みです。	誤差	00101	使用不可
Sync-3	K コード	00110	Startsynch #3
予約済みです。	誤差	01000	使用不可
予約済みです。	誤差	01100	使用不可
予約済みです。	誤差	10000	使用不可
予約済みです。	誤差	11111	使用不可

順序集合

順序集合は、下図に示す 4 つの K コードから成ります。

図 784. K コード送信



以下の表に、発生しうる SOP* シーケンスを含め、定義済みの順序集合を示します。

物理層では、Hard Reset は他の順序集合よりも優先順位が高くなるため、進行中の Tx メッセージに割込みが発生することがあります。

表 609.順序集合

順序集合名	K コード #1	K コード #2	K コード #3	K コード #4
SOP	Sync-1	Sync-1	Sync-1	Sync-2
SOP'	Sync-1	Sync-1	Sync-3	Sync-3
SOP''	Sync-1	Sync-3	Sync-1	Sync-3
Hard Reset	RST-1	RST-1	RST-1	RST-2
Cable Reset	RST-1	Sync-1	RST-1	Sync-3
SOP'_Debug	Sync-1	RST-2	RST-2	Sync-3
SOP''_Debug	Sync-1	RST-2	Sync-3	Sync-2

受信時、物理層は、以下の表に示すように、4 つのうちの 3 つの正しい K コードの任意の組み合わせにより順序集合を受け入れる必要があります。

表 610. 順序集合の検証

ステータス	1 番目のコード	2 番目のコード	3 番目のコード	4 番目のコード
有効	破壊	K コード	K コード	K コード
有効	K コード	破壊	K コード	K コード
有効	K コード	K コード	破壊	K コード
有効	K コード	K コード	K コード	破壊
有効 (完全)	K コード	K コード	K コード	K コード
無効 (例)	K コード	破壊	K コード	破壊

送信時のビット順

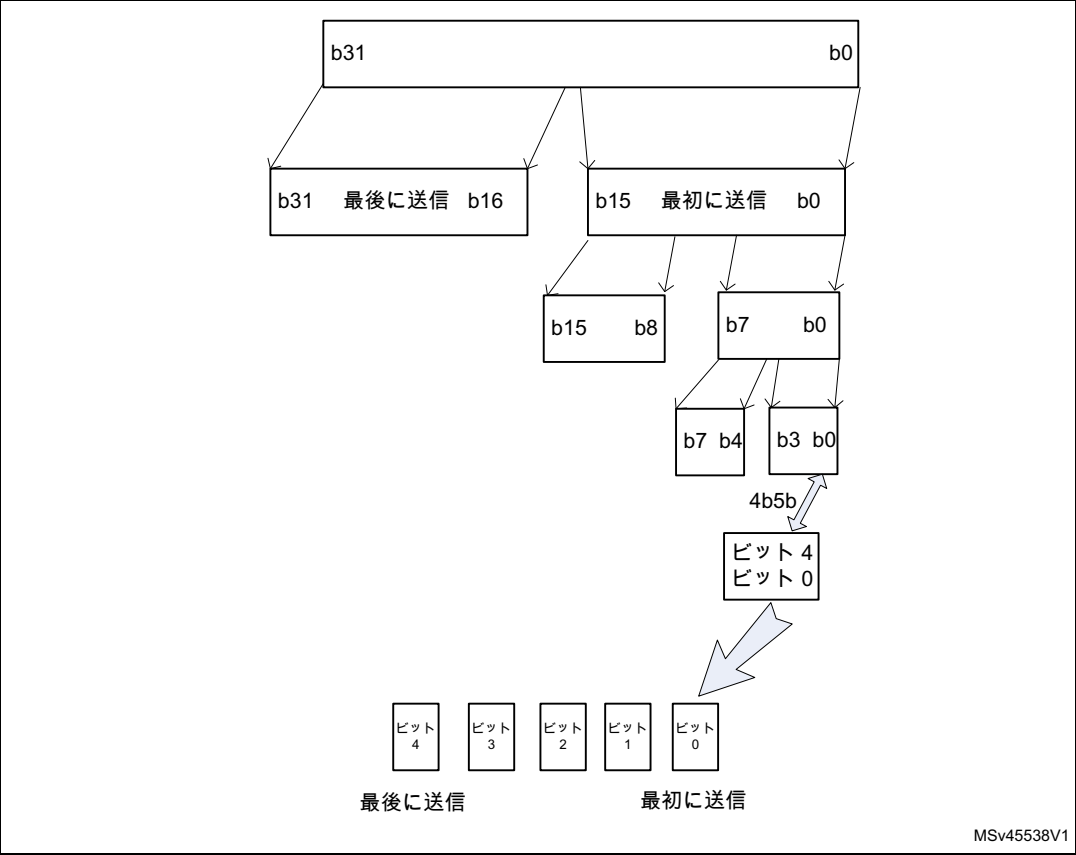
許容される送信データ単位/データサイズを以下の表に示します。

表 611. データサイズ

データユニット	エンコードされない場合	エンコードされる場合
バイト	8 ビット	10 ビット
ワード	16 ビット	20 ビット
ダブルワード	32 ビット	40 ビット

下図にビット送信順序を示します。

図 785. さまざまなデータサイズの送信順

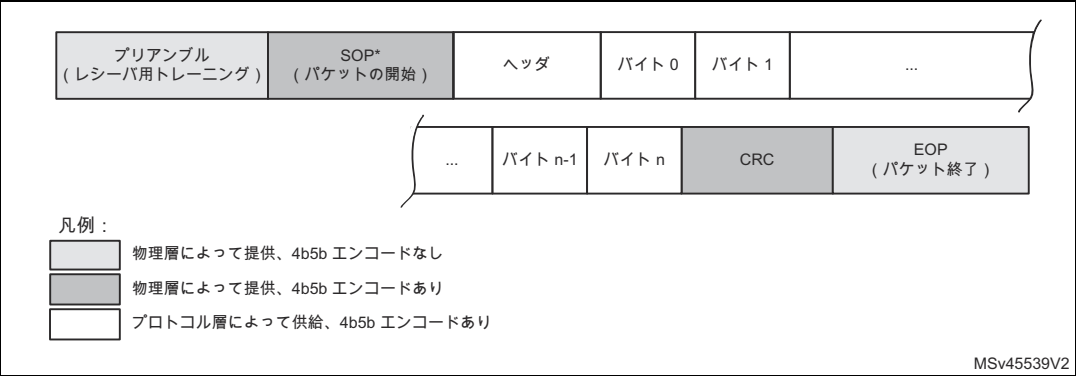


パケットフォーマット

Hard Reset と Cable Reset 以外のメッセージ

パケットフォーマットを、4b5b エンコードとデータソースに関する情報と一緒に下図に示します。

図 786. パケットフォーマット



Hard Reset

物理層では、Hard Reset 信号は、進行中の転送への割込みが行えるように高い優先順位を持っているため、他のタイプのメッセージとは違う方法で処理されます。

Tx メッセージの送信中の場合、物理層の仕様は次のようなシーケンスとなります。

1. EOP K コードを送信してメッセージを終了させ、メッセージの残りを破棄します。
2. $t_{InterFrameGap}$ 時間の間待ちます。
3. CC ラインがアイドル状態でない場合、アイドル状態になるまで待ちます。
4. プリアンブル、Hard Reset 信号の 4 個の K コードの順に送信します。
5. CC チャネルを無効にし（送受信を停止）、物理層をリセットし、プロトコル層に物理層がリセットされたことを通知します。
6. プロトコル層にリクエストされたときは、チャネルを再度有効にします。

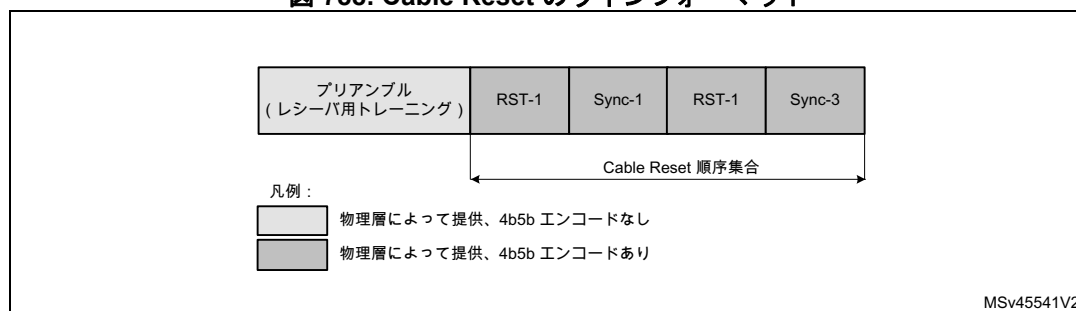
図 787. Hard Reset のラインフォーマット



Cable Reset

下図に示す Cable Reset は Hard Reset とフォーマットが似ていますが、Hard Reset とは違い、特に優先して処理する必要はありません。

図 788. Cable Reset のラインフォーマット



衝突の回避

物理層では、Tx メッセージの最後に送信されたビットの終了時から次のメッセージの最初のビットの間に発生する $t_{InterFrameGap}$ 遅延が考慮されます。

また、送信開始前に CC ラインのアイドル状態についても確認されます。CC ラインは、 $t_{TransitionWindow}$ (12 ~ 20 μ s) 内での遷移が 3 回 ($n_{TransitionCount}$) 未満の場合、アイドル状態とみなされます。Power Delivery 仕様リビジョン 3.1 では、Rp 値（ソース）を管理し、（シンクでの）これらの Rp 変更の Type-C 電圧レベルを監視する必要があります。

物理層の信号方式

ビットはバイフェーズマークコーディング (BMC) により信号送信されます。

BIST

プロトコル層が要求する BIST アクションに応じて、以下のいずれかを実行できます。

- Tx BIST パターンテスト。TXMODE および TXSEND を書き込むことで達成されます。
- Rx BIST パターンテスト。RXBIST の正しい値に RXMODE を書き込むことで達成されます。

UCPD では、「BMC」モードに対応する) 次の 2 つのパターンをサポートできます。

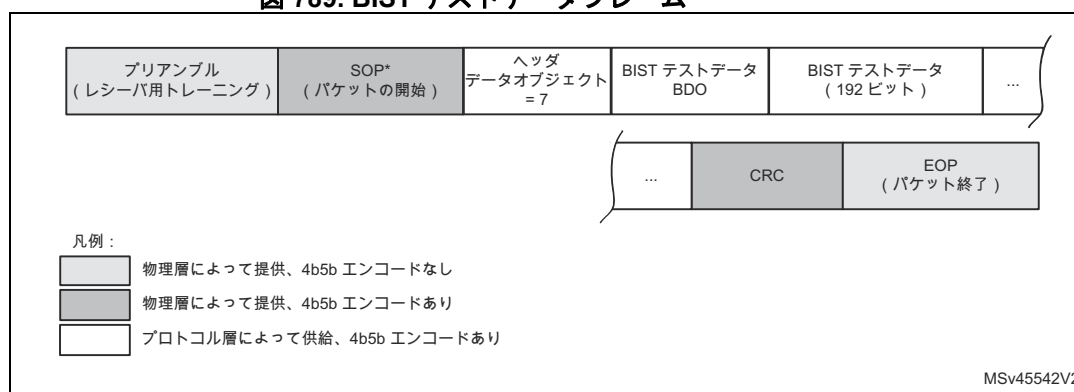
- BIST テストデータ (192 ビットパターン)。Tx および Rx に適用されます。Rx の場合は、メッセージを受信します (ただし、プロトコル層には渡されず破棄されます。その場合でも、確認応答として GoodCRC Tx メッセージが生成されます)。
- BIST キャリアモード 2 (単一パターン、無限長のメッセージ)。Tx のみに適用されます。Tx とは異なり、この状態のとき、このモードのレシーバは単に CC ラインを無視します。

BIST テストデータパターン

テストデータパターンは、UCPD では特殊なケースとみなされません。

BIST テストデータの packets フォーマットを下図に示します。

図 789. BIST テストデータフレーム



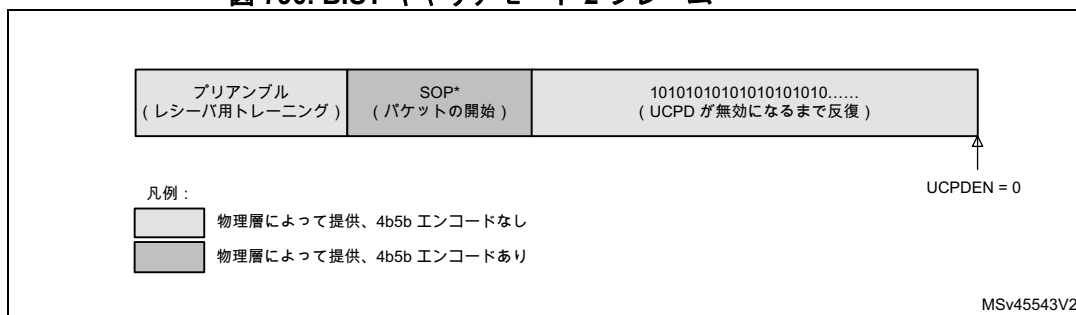
固定長のテストデータパターンです。実際、図 786: パケットフォーマットに示されている通常の packets フォーマットとの違いを表す点はヘッダの内容のみです。UCPD は Tx ヘッダの内容をプログラミングによって受信するため (単にペイロードの一部とみなされている)、BIST テストデータ packets から通常の packets を区別するのはこのプログラミングのみ (ブロックの動作ではない) です。

BIST キャリアモード 2

必要な場合は、この BIST テストモードから反復する 1010 の代替パターンが送信されます。このモードは信号分析用であるため、一定の長さを持たない安定した状態です (USB PD 仕様の V1.0)。USB PD 仕様の V1.1 以降では、このモードを終了するタイミングを示すカウンタをプロトコル層で定義します。

無限の 1010 シーケンスを停止するには (USB PD 仕様の要件に従って)、UCPDEN ビットにより UCPD ペリフェラルを無効化します。

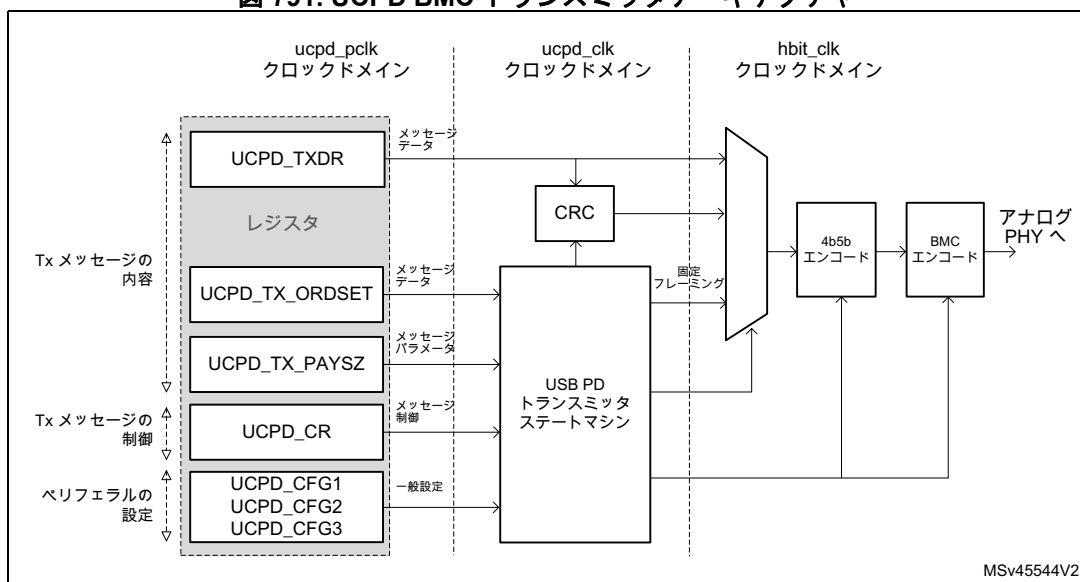
図 790. BIST キャリアモード 2 フレーム



56.4.4 UCPD BMC トランスミッタ

BMC トランスミッタでは、下図に示すように 4b5b エンコード、CRC 生成、BMC エンコードが行われます。その出力はチャンネルスイッチを経由してアナログ PHY に送られます。

図 791. UCPD BMC トランスミッタアーキテクチャ



BMC エンコーダ

バイフェーズマークコーディングメソッドは、「IEC 60958-1 デジタルオーディオインタフェースパート 1 総合版 3.0 2008-09 www.iec.ch」仕様で定義されています。

ハーフビットクロック hbit_clk は、UCPD_CFGFR1 レジスタの HBITCLKDIV[5:0] ビットフィールドによって制御されるシンプルな分周回路を通して ucpd_clk から生成されます。これにより、ハイとローのハーフビット周期の同じ期間（立上りエッジと立下りエッジの異なる期間による、またはジッタによる小さな差を無視できる場合）、および同じビット期間（ジッタを無視できる場合）が保証されます。

トランスミッタのタイミングと衝突の回避

衝突の回避に対するハードウェアのサポートは、トランスミッタのハーフビット時間の機能の 1 つです。2 つのカウンタが実装されています。

- *tInterFrameGap* : IFRGAP 經由 (事前定義値、變更可能)
- *tTransitionWindow* : TRANSWIN 經由 (事前定義値、變更可能)

この2つのカウンタを一度正しくセットすると、フレーム間ギャップを生成します。

トランスミッタにおける Hard Reset

Hard Reset の生成を容易にするために、TXMODE フィールドの特殊コードが使用されます。他のフィールドに書き込む必要はありません。

正しいコードを書き込むと、ハードウェアは正しい（最適な）タイミングで実行中の Tx メッセージに強制的に Hard Reset Tx を適用します。このメッセージは（まだ進行中である場合）、現行のシーケンスを切り詰め、EOP K コードシーケンスを直接付加することで、クリーンに終了します。特定の割込みがこの切詰めイベントに関連して発生することはありません。

エラー時のトランスミッタの動作

アンダーラン状態（TXUND 割込み）が偶発的に発生する可能性があり、その場合、UCPD では（正しい）Tx ペイロードが足りなくなって Tx メッセージを正しく完成することができなくなる可能性があります。これは重大なエラーです（これが生じるということは、ソフトウェアが適時応答できていないということ）。結果として、ハードウェアはメッセージの最後で CRC が不正であることを確認し、これによりレシーバ側でメッセージを破棄することを保証します。

56.4.5 UCPD BMC レシーバ

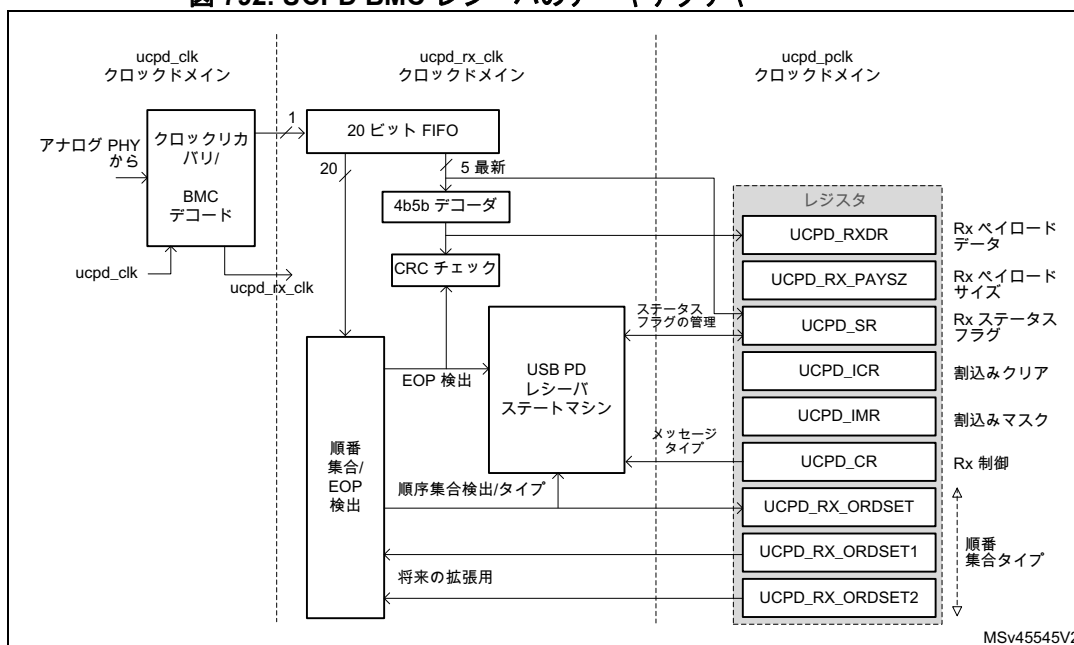
UCPD BMC レシーバの処理内容：

- クロックリカバリ
- プリアンブル検出／タイミング導出
- BMC デコード
- 4b5b でのデコード
- K コードの順序集合の認識
- CRC チェック
- SOP 検出
- EOP 検出

レシーバは UCPD ペリフェラルが（UCPDEN を通じて）有効化されるとすぐに有効化されますが、メッセージの受信を開始する前に CC ラインのアイドル状態を待ちます。

下図に、UCPD BMC レシーバのハイレベルアーキテクチャを示します。

図 792. UCPD BMC レシーバのアーキテクチャ



CRC チェッカー

受信したビットは CRC チェッカーに供給され、受信したペイロードビットストリーム中に 32 ビットの状態が進展します。最後には、CRC の 32 ビットもロジックに供給されます。

EOP 検出 (5 ビット) はプロセスを停止し、固定残差状態のチェックを行ってペイロードを正しく受信したことを確認します (実際には残差は 0xC704DD78 です)。

この時点で UCPD によって割込み RXMSGEND が発生します。CRC が不正な場合は RXERR が真にセットされ、受信データは破棄する必要があります。

通常動作時には、この割込みは以前に認識され、クリアされています。そうでない場合は、別の割込み RXOVR が RXMSGEND の代わりに生成されます。

順序集合の検出

この機能では、それぞれ 4 つの 5 ビット K コードから成るさまざまな順序集合を検出します。

プリアンブルに入ったら、順序集合のスライディングウィンドウ検出を開始します（5 ビットの 4 つのワード）。

検出された順序集合にはすべての SOP* コード (SOP、SOP'、および SOP'') が含まれますが、Hard Reset、Cable Reset、SOP'_Debug、SOP''_Debug のほか、UCPD_RX_ORDEXT1 および UCPD_RX_ORDEXT2 レジスタで定義された 2 つの拡張機能も含まれます。

EOP 検出および Hard Reset の例外処理

EOP はメッセージの終了を表す固定の 5 ビット K コードです。

トランスミッタで Hard Reset の送信が必要となるのは（前回のメッセージの送信がまだ進行中の場合に）、この前回のメッセージが EOP によって早めに切捨てられているためです。

Hard Reset が無視された場合、EOP 検出は予想された時間でのみ実行できます。ただし、この Hard Reset 問題により、EOP 検出は Rx メッセージの着信中はアクティブでなければなりません。「早期 EOP」が検出されると、切捨てられた Rx メッセージはすぐに破棄されます。

切捨てられたメッセージまたは破損したメッセージの例外

順序集合が検出されると、メッセージによっては CRC と EOP で完成するデータバイトを受信する必要がある場合があります。これらのフェーズのどこかで、次のようなエラー条件が発生します。

- 「1 UI」を十分上回る時間の間、ラインが静的になります（この条件の正確な閾値は重要ではありませんが、例外は 3 UI より前に発生する必要があります）。または
- メッセージは終了しますが、認識されません（例：EOP の破損）。

どちらの場合でも、レシーバは RXMSGEND および RXERR フラグを立ち上げ、現在のメッセージを停止します。

短いプリアンブルまたは未完成の順序集合の例外

レシーバが確認する予期したプリアンブルが半分に満たない例外的な事例では、正しい BMC デコードを可能にする周波数の概算を算出できません。プリアンブル全体を確認している場合でも、周波数の概算は可能ですが、ラインが静的になる前にすべての順序集合を受信できず、レシーバステートマシンは開始されません。

どちらの場合でもクロックリカバリ/BMC デコーダは再開し、最初に IDLE 条件、次にプリアンブルを確認し、続いて周波数を概算します。

56.4.6 UCPD Type-C のプルアップ (Rp) およびプルダウン (Rd)

UCPD ではこれらのレジスタを ANAMODE および ANASUBMODE[1:0] によって簡単に制御できます。使用する CC ラインが 1 つのみである場合、CCENABLE[1:0] ビットフィールドを使用して他のラインの制御を無効にすることで、消費電力を最適化することができます。

MCU が電源オフの場合でも、UCPDx_DBCC1 ピンと UCPDx_DBCC2 ピンが、それぞれ UCPDx_CC1 ピンと UCPDx_CC2 ピンに接続されている場合は、「デッドバッテリー」Rd が示されます。

デッドバッテリー動作が必要ない場合は（ソースのみを対象とした製品など）、UCPDx_DBCC1 ピンと UCPDx_DBCC2 ピンの両方をアース接続してください。

電力が供給され、MCU が起動されたら、ANAMODE と ANASUBMODE[1:0] に望ましい動作（ソースなど）をプログラムする必要があります。PWR_CR3 レジスタの UCPD_DBDIS ビットをセットする前に、デッドバッテリープルダウンレジスタを削除し、プログラムされたばかりの値を有効にします。

切断されている状態のシンクの場合は、STANDBY 低電力モードを使用できます。

56.4.7 UCPD Type-C 電圧監視とデバウンス

Type-C ステートマシンを正しく動作させ、ケーブル方向を検出するには、ピークなどの高速イベントを無視しながら、CC1/2 ラインの電圧レベルを監視する必要があります。

CC1/2 ラインの電圧レベル間の閾値は、PHY 閾値検出設定に基づいて決まります。

TYPE_C_VSTATE_CC1/2[1:0] ビットフィールドは、ピークなどの高速ラインイベントを抑制するハードウェアデバウンスフィルタにより処理される CC1/2 ラインレベルを反映します。PHYCCSEL ビットでは、Power Delivery 信号に使用される CC1 または CC2 ラインを選択します。

消費電力を最小限に抑えるため、Type-C 検出器を常にオンにしておくのではなく、ポーリングの瞬間だけ Type-C 検出器をオンにするポーリング方法を使用し、CC1/2 ラインイベント時にデバイスを STOP モードからウェイクアップさせることを推奨します。

56.4.8 UCPD ファストロールスワップ (FRS)

FRS 信号

FRS 接続（特定の長さのパルス）は、FRSTX ビットのセット時に生成されます。

FRS 接続の期間、UCPD_FRSTX として設定された I/O（オルタネート機能）は、アクティブ CC ラインをプルダウンする外部 NMOS トランジスタのゲートをハイレベルで制御します。

FRS 検出

FRS 監視を有効にするには、検出されたケーブル方向に基づいてアクティブ CC ラインを選択する PHYCCSEL を書き込んだ後、ビット FRSRXEN をセットします。

56.4.9 UCPD DMA インタフェース

DMA は UCPD に実装され、有効にすると、UCPD1_TXDR および UCPD1_RXDR レジスタ（Tx と Rx のデータレジスタ、それぞれ 1 バイト）を処理するバイトレベルの割込みは不要になります。

TXDMAEN や RXDMAEN ビットを有効にすることで、Tx や Rx の機能に対して個別に DMA を有効にできます。

56.4.10 STOP モードからのウェイクアップ

消費電力を最適化するには、STOP モードを使用して、CC ラインでのイベントを待って MCU をウェイクアップできると便利です。

これが機能するためには、最初に WUPEN に 1 を書き込んで有効にする必要があります。

ウェイクアップを引き起こすイベントは次のとおりです。

- BMC レシーバのイベント（RXORDDDET、RXHRSTDET）、ハードウェアによる PHYRXEN の有効化
- FRS 検出器のイベント（FRSEVT）、ハードウェアによる FRSRXEN の有効化
- Type-C 検出器のイベント（TYPECEVT1、TYPECEVT2）、ハードウェアによる CC1TCDIS、CC2TCDIS の有効化

56.5 UCPD プログラミングシーケンス

UCPD ユニットを使用する正常なシーケンスは次のとおりです。

1. UCPD を設定します。
2. UCPD を有効化します。
3. 同時に
 - プロトコル層からの要求に応じて、Tx メッセージを送信します。
 - 関連する Rx メッセージを取得し（ポーリングまたは割込みを待ち）、プロトコル層に引き渡す詳細を復元します。

最後の項目を無限に繰り返します。

56.5.1 初期化フェーズ

次のシーケンスを使用して、クリーンに起動します。

1. UCPD_CFG レジスタを書き込んで、すべての初期クロック分周値を準備します。
2. UCPDEN ビットをセットすることによって、ユニットを有効にします。
3. UCPD_CFGR2 レジスタの RXAFILTEN ビットを使用して、各 CC ラインのアナログ Rx フィルタを有効にします。

56.5.2 Type-C ステートマシンの処理

ソース、シンク、デュアルロールポート（ソースとシンクの代替となる最後のもの）の一般的な用途では、ソフトウェアにより該当する USB Type-C ステートマシンを実装する必要があります。基本的なコードを以下の表に示します。

表 612.ANAMODE と ANASUBMODE のコーディングおよび TYPEC_VSTATE_CCx とのリンク

ANAMODE	ANASUBMODE[1:0]	注	TYPEC_VSTATE_CCx[1:0]			
			00	01	10	11
0 : 転送元	00 : 無効	無効	N/A			
	01 : デフォルト USB Rp	-	vRa[Def]	vRd[Def]	vOPEN[Def]	N/A
	10 : 1.5A Rp	-	vRa[1.5]	vRd[1.5]	vOPEN[1.5]	
	11 : 3.0A Rp	-	vRa[3.0]	vRd[3.0]	vOPEN[3.0]	
1 : シンク	xx	-	vRa	vRd-USB	vRd-1.5	vRd-3.0

CCENABLE[1:0] ビットフィールドにより、CC ラインの 1 つでプルアップ/プルダウンを無効にできます。

注 : Type-C ステートマシンは CC ラインのレベルだけでなく、VBUS の存在検出（シンクモード）にも依存します。ソースモードの場合は、VCONN 生成と VBUS 状態（オン/オフ/電圧レベル、放電）を、決定します。UCPD は VBUS の生成回路や VCONN 負荷スイッチ（CC ラインに接続する VCONN 供給ジェネレータを有効化）を直接制御しませんが、アプリケーションでは正常に機能するためにこれらの入力や制御を必要とします。

一般的なプログラミングシーケンス（UCPD を設定し、有効化）

1. ANAMODE と ANASUBMODE[1:0] を USB Type-C ステートマシンの現在の位置に（ソースの場合は現在の通知にも）基づいてセットします。これにより、CC ラインの適切なプルアップ/プルダウンがオンになり、TYPEC_VSTATE フィールドが表す電圧レベルが定義されます。プログラミングの前には、PHY が事実上オフである点に注意してください。
2. TYPEC_VSTATE_CC1/2 を読み込んで、Type-C の初期状態を決定します（ローカルソースがリモートシンクに接続されているかどうかなど）。
3. 接続されていない場合は、接続イベントを待ちます。
4. 接続が検出され、ローカルの Power Delivery 機能が実装されたと仮定して、Power Delivery のメッセージの送受信を開始します。
5. 安定した電圧の変更を示す PHYEVT1/2 に新しい割込み/イベントが発生した場合は、影響を再評価して、この入力を Type-C ステートマシンに提供します。

設定可能な 3 つの Rp 値 (Default-USB / 1.5A / 3.0A) のいずれかに変更する必要のあるソースと、これに接続されたシンクの例

- [ソース] 単純に ANASUBMODE[1:0] を再プログラムします。
- [その時点からのシンクの動作] PHYEVT1/2 が発生し、TYPEC_VSTATE1/2 がこれに合わせて変化します。

ソースからシンクにトグルするデュアルロールポート (DRP) のプログラミング:

- 新しい動作を開始するために、単純に ANAMODE と ANASUBMODE[1:0] を再プログラムします。

詳細なプログラミングシーケンス (例):

表 613.Type-C シーケンス (転送元: 3A)、ケーブル/シンク接続済み (CC1 は Rd、CC2 は Ra)

Type-C ステート	ANAMODE、 ANASUBMODE [1:0]	CCENABLE	PHYCCSEL	RDCH	CC[x] VCONNEN (1)	イベント=> 次の行に 移動	コメント
切断。 SRC	0：転送元、 11：Rp3A0	11： 両方有効	0（無視）		00： [どちらもで ない]	PHYEVT1： [VRd-3A0]	シンク接続検出待ち、CC1 [EVT1] に出現
接続待ち。 SRC						PHYEVT2： [VRa]	接続待ち開始（100 ～ 200ms）、Ra => VCONN の リクエストも参照
接続。 SRC [VCONN => CC2]	0：転送元、 11：Rp3A0 [SinkTxOK]	01： CC2 無効 （外部 VCONN スイッチに より可能／ 推奨）	0 [CC1 は Rd]	0： [通常]	10： [CC2 アクティブ]	タイマ （100 ms）、 PHYEVTx なし	PHY から切断されたロー カルの CC2（VCONN ス イッチは VCONN ソース を外部的に CC2 に接続） PHYEVT1 の監視を継続
	0：転送元、 10：Rp1A5 [SinkTxNG]					SW タイマ （SinkTxNG）	ソースがメッセージシー ケンスの初期化を希望 （SinkTxNG 条件を最初に セット）
	0：転送元、 11：Rp3A0 [SinkTxOK]					PHYEVT1： [VOpen-3A0]	ソースがメッセージシー ケンスを終了（その後 SinkTxOK 条件を設定）
							シンクの切断待ち（CC1 は Vopen）
切断待ち。 SRC	0：転送元、 11：Rp3A0	11： 両方有効	0：（無視）	1： [放電]	00： [どちらもで ない]	> 0.8V 検出	能動的に VCONN [CC2] を 放電 [Rdch]、< 0.8V まで
切断。 SRC			0： [通常]			[詳細は表の最初の行を 参照]	

1. 外部負荷スイッチコンポーネントを通して VCONN を有効化する 2 つの GPIO

56.5.3 USB PD の送信

プロトコル層のメッセージの受信時 (つまり、送信)、UCPD_TX_ORDSET および UCPD_TX_PAYSZ レジスタを書き込むことによって、Tx メッセージコンテンツを準備します。

メッセージ送信をトリガするには、TXMODE ビットフィールドの適切な値で TXSEND ビットをセットします。

データバイトの送信時は、UCPD_TXDR レジスタへの新しいデータ書き込みをリクエストするため、TXIS フラグが立てられます。

データのペイロード全体が送信されるまで、これが繰り返されます。

CRC パケットの送信時、TXMSGSENT フラグがセットされ、メッセージ送信の完了が示されます。

Hard Reset の送信

Hard Reset を送信する必要があるとわかったとき、すぐに UCPD_CR レジスタの TXHRST ビットをセットすると、内部ステートマシンで強制的に正しいシーケンスを生成できます。UCPD_TX_ORDSET の値は、この特定のケースでは更新は不要です (Hard Reset の正しいコードは UCPD によって送信されます)。

USB Power Delivery 仕様では、メッセージ送信中に Hard Reset を優先することを規定しています。この場合、UCPD では現在のメッセージのペイロードを切捨てるなどして、最後に EOP を付加します。レジスタによる通知は利用できません (TXMSGSEND フラグによる通知など)。これは、Hard Reset が以前のすべてのアクティビティに優先されるという事実によって裏付けられています (このため、完了したかどうかを知ることは重要ではなくなります)。

送信用の DMA の使用

送信について DMA (Direct Memory Access) を有効にするには、UCPD_CR レジスタの TXDMAEN ビットをセットします。

各メッセージに対して：

- メッセージ全体をメモリ内で準備します (2 つのヘッダバイトから始まります)。
- 2 つのヘッダバイトに対応する長さと、データワードの数の 4 倍に対応する多数のデータバイトにより、DMA の動作をプログラムします。
- メッセージ転送を開始するには、TXSEND を書き込みます。
- TXMSGDISC の場合は、前のラインに戻ります (TXSEND)。
- DMA 転送完了割込みを待ちます (つまり、UCPD への最後の Tx バイトの書き込み)。
- 後続の TXMSGSENT 割込みが発生していないかダブルチェックします。

56.5.4 USB PD の受信

受信メッセージシーケンスの開始の通知は、UCPD_SR (ビット RXORDDDET) の割込みによってトリガされます。

情報は、以下の読出しによって復旧されます。

- UCPD_RX_SOP (割込み RXORDDDET)
- UCPD_RXDR (割込み RXNE、バイトごとに繰り返される)
- UCPD_RXPAYSZ (割込み RXMSGEND)

上記の UCPD_RXDR から前に読み出したデータは、RXERR フラグがセットされている場合、この時点で破棄する必要があります。

CRC が有効な場合、受信データはプロトコル層に転送されます。

デバッグのため、受信した不正な K コード数の統計を追跡することが望まれます (仕様が定義されているとおり、K コードの 3/4 が有効である場合にのみ実行します)。これは、以下を使用して容易に行えます。

- 少なくとも 1 つの無効な K コードがあることを示す RXSOP3OF4 ビット
- 順序集合内の無効な K コードの順序を特定する RXSOPKINVALID ビットフィールド

受信用の DMA の使用

受信について DMA (Direct Memory Access) を有効にするには、UCPD_CR レジスタの RXDMAEN ビットをセットします。

Rx メッセージを受信する可能性があるときには、以下を実行します。

- DMA 受信動作をプログラムします (また、最大有効メッセージより少し長くしてバッファを割きます)。長さは拡張メッセージサポートに依存します。
- RXORDDDET を受信した後、DMAは、バックグラウンドで動作を開始します。
- RXMSGEND 割込みの受信時に RXPAYSZ を読み出します。
- RXPAYSZ を DMA Rx のバイト数に対してダブルチェックします (一致するはずですが、RXDR の DMA 読出しは RXDR が最後のバイトを取得した少しあとになります)。
- DMA Rx バッファを処理します。
- できるだけ速やかに次の Rx DMA バッファを用意して備えます。

56.5.5 UCPD ソフトウェアのトリミング

CC プルアップ (Rp) およびプルダウン (Rd) デバイスは、必要な精度を得るために各部をトリミングする必要があります。トリミング値は不揮発性メモリに保存されます。

ソフトウェアによって CC プルアップおよびプルダウンデバイスをトリミングするには、次の手順を実行します。

1. 不揮発性メモリから TRIM 値を取得します (表 605:UCPD ソフトウェアの trim データを参照)。
2. 初期化時に、次のように UCPD_CFGR3 レジスタのビットフィールドに TRIM 値を書き込みます。
 - TRIM_CC1_RP[3:0] に 3A0_CC1[3:0]
 - TRIM_CC2_RP[3:0] に 3A0_CC2[3:0]
 - TRIM_CC1_RD[3:0] に Rd_CC1[3:0]
 - TRIM_CC2_RD[3:0] に Rd_CC2[3:0]
3. 各 ANASUBMODE を 1A5 または 3A0 にセットし、UCPD_CFGR3 レジスタビットフィールドに次のようにトリミング値を書き込みます。
 - TRIM_CC1_RP[3:0] にそれぞれ 1A5_CC1[3:0] または 3A0_CC1[3:0]
 - TRIM_CC2_RP[3:0] にそれぞれ 1A5_CC2[3:0] または 3A0_CC2[3:0]

56.6 UCPD 低電力モード

低電力モードの概要を以下の表 614 : 低電力モードが UCPD に与える影響に示します。

表 614.低電力モードが UCPD に与える影響

モード	説明
SLEEP	影響なし。
STOP	イベントの検出 (Type-C、BMC Rx、FRS 検出) 動作は続行し、MCU をウェイクアップする可能性があります。
STANDBY	UCPD は動作しておらず、MCU をウェイクアップすることはできません。プルダウンは、設定されている場合はアクティブなままです。
電源オフ	デッドバッテリープルダウンはアクティブなままです。

UCPD は、次のいずれかの対応するイベントを認識したときに、MCU を STOP モードからウェイクアップできます。

- TYPEC_VSTATE_CCx に表示される、CC ラインの電圧範囲の変化に関連する Type-C イベント
- RXORDSET を読み出すことで表示される、順序集合が RXORDSETEN[8:0] に応じてフィルタリングされた順序集合と一致する Power Delivery 受信メッセージ

STOP モードからのウェイクアップを有効にするには、UCPD_CFG2 レジスタの WUPEN ビットをセットします。

UCPD レベルでは、STOP モード中にカーネルクロックアクティビティを必要とする次の 3 つのタイプのイベントが発生する場合があります。

- 後で Type-C 仕様で定義された電圧範囲間の一定の変化であると確認できるアナログ PHY 電圧閾値検出器のアクティビティ
- 後で Rx メッセージイベント (RXORDSET) を生成可能な Power Delivery BMC レシーバのアクティビティ (選択した CC ラインから入力される)
- 後で FRS 信号検出イベント (FRSEVT) を生成可能な Power Delivery FRS 検出器のアクティビティ

以下の非同期アクティビティがある場合、RCC とともに正常に機能するために、クロックリクエスト信号が有効化されます (WUPEN の条件付き)。

- Type-C 電圧閾値検出器 (いずれかの CC ラインから入力される)
- Power Delivery レシーバ信号 (選択した CC ラインから入力される)
- FRS 検出信号 (選択した CC ラインから入力される)

56.7 UCPD 割込み

以下の表に、UCPD イベントフラグのほか、関連するフラグクリアビットと割込みイネーブルビットを示します。

表 615. UCPD 割込みリクエスト

割込みイベント	イベントフラグ	イベントフラグ/割込みのクリア方法	割込み有効制御ビット
FRS 検出	FRSEVT	FRSEVTCF をセット	FRSEVTIE
CC2 での Type-C の電圧レベル変化	TYPECEVT2	TYPECEVT2CF をセット	TYPECEVT2IE
CC1 での Type-C の電圧レベル変化	TYPECEVT1	TYPECEVT1CF をセット	TYPECEVT1IE
Rx メッセージ受信	RXMSGEND	RXMSGENDCF をセット	RXMSGENDIE
Rx データオーバーフロー	RXOVR	RXOVRCF をセット	RXOVR
Rx Hard Reset 検出済み	RXHRSTDET	RXHRSTDETCF をセット	RXHRSTDETIE
Rx 順序集合 (4 つの K コード) 検出済み	RXORDDDET	RXORDDDETCF をセット	RXORDDDETIE
受信データレジスタノットエンプティ	RXNE	UCPD_RXDR のデータを読み出す	RXNEIE
Tx データアンダーラン	TXUND	TXUND CF をセット	TXUNDIE
Hard Reset 送信済み	HRSTSENT	HRSTSENTCF をセット	HRSTSENTIE
Hard Reset 破棄済み	HRSTDISC	HRSTDISCCF をセット	HRSTDISCIE
送信メッセージがアボート	TXMSGABT	TXMSGABTCF をセット	TXMSGABTIE
送信メッセージ送信済み	TXMSGSENT	TXMSGSENTCF をセット	TXMSGSENTIE

表 615. UCPD 割込みリクエスト (続き)

割込みイベント	イベントフラグ	イベントフラグ/割込みのクリア方法	割込み有効制御ビット
送信メッセージ破棄済み	TXMSGDISC	TXMSGDISCCF をセット	TXMSGDISCIE
送信データが必要	TXIS	UCPD_TXDR レジスタにデータを書き込む	TXISIE

UCPD からの割込みが受信された場合、ソフトウェアによって、UCPD_SR レジスタを読み出すことで、割込みの転送元を確認する必要があります。

どのビットが 1 であるかによって、ISR はこの条件を処理し、UCPD_ICR レジスタの適切なビットに書き込んでこのビットをクリアする必要があります。

56.8 UCPD レジスタ

56.8.1 UCPD 設定レジスタ 1 (UCPD_CFGR1)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

ペリフェラルの一般設定このレジスタへの書き込みが可能なのは、UCPD が無効 (UCPDEN = 0) の場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UCPDEN	RXDMAEN	TXDMAEN	RXORDSETEN[8:0]								PSC_USBDCLK[2:0]			Res.	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANSWIN[4:0]					IFRGAP[4:0]					HBITCLKDIV[5:0]					
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 UCPDEN : UCPD ペリフェラルの有効化

UCPD ペリフェラルの全般的有効化

0 : 無効化

1 : イネーブル

無効化すると、ペリフェラルではすぐに実行中のアクティビティが停止し、制御ビットとビットフィールドはすべてデフォルトでリセット値になります。ペリフェラルが無効状態から有効状態に移るたびに、望ましい値をセットする必要があります。

ビット 30 RXDMAEN : 受信 DMA モードの有効化

このビットをセットすると、DMA モードの受信が有効になります。

0 : 無効化

1 : イネーブル

ビット 29 TXDMAEN : 送信 DMA モードの有効化

このビットをセットすると、DMA モードの送信が有効になります。

0 : 無効化

1 : イネーブル

ビット 28:20 **RXORDSETEN[8:0]** : レシーバの順序集合の有効化

このビットフィールドで、レシーバが検出する順序集合のタイプを規定します。セット/クリアすると、各ビットにより特定の機能が有効化/無効化されます。

0bXXXXXXXX1 : SOP 検出有効
 0bXXXXXXXX1X : SOP' 検出有効
 0bXXXXXXXX1XX : SOP" 検出有効
 0bXXXXX1XXX : Hard Reset 検出有効
 0bXXXX1XXXX : Cable Detect リセット有効
 0bXXX1XXXXX : SOP' _Debug 有効
 0bXX1XXXXXX : SOP" _Debug 有効
 0bX1XXXXXXX : SOP 拡張 #1 有効
 0b1XXXXXXX : SOP 拡張 #2 有効

ビット 19:17 **PSC_USBPDCLK[2:0]** : ucpd_clk を生成するためのプリスケアラ分周比

このビットフィールドで、UCPD ベリフェラルクロックを生成するカーネルクロックのプリスケアラの分周比を規定します (ucpd_clk)。

0x0 : 1 (バイパス)
 0x1 : 2
 0x2 : 4
 0x3 : 8
 0x4 : 16

プリスケアラの使用では、ucpd_clk 周波数を 6 ~ 9 MHz の範囲でセットできるようにすることを推奨します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15:11 **TRANSWIN[4:0]** : 遷移ウィンドウ期間

このビットフィールドで、 $t_{TransitionWindow}$ 間隔を生成する hbit_clk 分周回路の分周比 (ビットフィールド値 - 1) を規定します。

0x00 : サポートされません
 0x01 : 2
 0x09 : 10 (推奨値)
 0x1F : 32

ucpd_clk 周波数と HBITCLKDIV[5:0] ビットフィールドの設定を考慮して、12 ~ 20 us の間隔を生成する値をセットします。

ビット 10:6 **IFRGAP[4:0]** : フレーム間ギャップタイマクロックを生成するための分周比

このビットフィールドで、フレーム間ギャップタイマクロック ($t_{InterFrameGap}$) を生成する ucpd_clk 分周回路の分周比 (ビットフィールド値 - 1) を規定します。

0x00 : サポートされません
 0x01 : 2
 0x0D : 14
 0x0E : 15
 0x0F : 16
 0x1F : 32

分周比 15 は USB PD 2.0 仕様の正常値の Tx クロックが対象となります。15 未満の分周比は正常未満の Tx クロックが対象となり、15 を超える分周比は正常範囲を超える Tx クロックが対象となります。

ビット 5:0 **HBITCLKDIV[5:0]** : ハーフビットクロックを生成するための分周比

このビットフィールドで、ハーフビットクロック (hbit_clk) を生成する ucpd_clk 分周回路の分周比 (ビットフィールド値 - 1) を規定します。

0x00 : 1 (バイパス)
 0x1A : 27
 0x3F : 64

56.8.2 UCPD 設定レジスタ 2 (UCPD_CFGR2)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

UCPD Rx 信号のフィルタリングの設定。このレジスタへの書込みが可能なのは、UCPD が無効 (UCPDEN = 0) の場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXAFILT EN	Res.	Res.	Res.	Res.	WUPEN	FORCECLK	RXFILT2N3	RXFILTDIS
							rw					rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 RXAFILTEN : Rx アナログフィルタの有効化

このビットをセットすると、最適な Power Delivery の受信に必要な Rx アナログフィルタが有効になります。

0 : 無効化

1 : イネーブル

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 WUPEN : STOP モードからのウェイクアップの有効化

このビットをセットすると、UCPD_ASYNC_INT 信号が有効になります。

0 : 無効化

1 : イネーブル

ビット 2 FORCECLK : ClkReq クロックのリクエストの強制実行

0 : 強制的にクロックリクエストを行わない

1 : 強制的にクロックリクエストを行う

ビット 1 RXFILT2N3 : BMC デコーダ Rx プリフィルタのサンプリング方式

新しい値を確認応答する前の適合連続サンプルの数

0 : 3 サンプル

1 : 2 サンプル

ビット 0 RXFILTDIS : BMC デコーダ Rx プリフィルタの有効化

0 : イネーブル

1 : 無効化

このサンプリングクロックはレシーバのものです (つまり、プリスケラ後)。

56.8.3 UCPD 設定レジスタ 3 (UCPD_CFGR3)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

CC プルアップおよびプルダウンデバイスの UCPD トリミングの設定。トリミングは、ソフトウェアにより最初にこのレジスタに書き込まれるまで、ハードウェアによって管理されます。

レジスタは全自動のトリミングをサポートするデバイス用に予約されています（書き込むことはできません）。表 604 : UCPD の実装を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TRIM_CC2_RP [3:0]				Res.	Res.	Res.	Res.	Res.	TRIM_CC2_RD [3:0]			
			rw	rw	rw	rw						rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TRIM_CC1_RP [3:0]				Res.	Res.	Res.	Res.	Res.	TRIM_CC1_RD [3:0]			
			rw	rw	rw	rw						rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:25 **TRIM_CC2_RP[3:0]** : CC2 ライン上の Rp 電流ソース用の SW TRIM 値

ビット 24:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **TRIM_CC2_RD[3:0]** : CC2 ライン上の Rd レジスタ用の SW TRIM 値

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:9 **TRIM_CC1_RP[3:0]** : CC1 ライン上の Rp 電流ソース用の SW TRIM 値

ビット 8:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **TRIM_CC1_RD[3:0]** : CC1 ライン上の Rd レジスタ用の SW TRIM 値

56.8.4 UCPD 制御レジスタ (UCPD_CR)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2TCDIS	CC1TCDIS	Res.	RDCH	FRSTX	FRSRXEN
										rw	rw		rw	rs	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CCENABLE[1:0]		ANAMODE	ANASUBMODE[1:0]		PHYCSEL	PHYRXEN	RXMODE	TXHRST	TXSEND	TXMODE[1:0]	
				rw	rw	rw	rw	rw	rw	rw	rw	rs	rs	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **CC2TCDIS** : CC2 Type-C 検出器ディスエーブル

このビットで、CC2 ライン上で Type-C 検出器を無効化します。

0 : イネーブル

1 : 無効化

有効な場合、CC2 用の Type-C 検出器は ANAMODE と ANASUBMODE[1:0] を使用して設定されます。

ビット 20 **CC1TCDIS** : CC1 Type-C 検出器ディスエーブル

このビットで、CC1 ライン上で Type-C 検出器を無効化します。

0 : イネーブル

1 : 無効化

有効な場合、CC1 用の Type-C 検出器は ANAMODE と ANASUBMODE[1:0] を使用して設定されます。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **RDCH** : Rdch 条件の実行

このビットは、ソース専用の *UnattachedWait.SRC* 状態中に、Type-C ステートを考慮して、残りのセットにより、PHYCCSEL ビット（そのため、VCONN に関連）を通して選択された CC ラインの Rdch 条件を実行します。「ソース VCONN 放電の USB Type-C ECN」を参照してください。CCENABLE[1:0] ビットフィールドもそれに合わせてセットする必要があります。

0 : 影響なし。

1 : Rdch 条件の実行

ビット 17 **FRSTX** : FRS Tx 信号の有効化。

このビットをセットすると、FRS Tx 信号が有効になります。

0 : 影響なし。

1 : イネーブル

このビットは、USB Power Delivery 仕様リビジョン 3.1 を考慮した遅延後、ハードウェアによってクリアされます。

ビット 16 **FRSRXEN** : FRS イベントの検出の有効化

このビットをセットすると、PHYCCSEL ビットを使用して選択された CC ラインでの FRS Rx イベント (FRSEVT) 検出が有効になります。0 : 無効化

1 : イネーブル

デバイスが FRS 不対応ソース/シンクに設置されている場合は、このビットをクリアします。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **CCENABLE[1:0]** : CC ラインの有効化

このビットフィールドでは、ANAMODE と ANASUBMODE[1:0] の設定に従って CC1 および CC2 ラインのアナログ PHY（プルアップおよびプルダウン）を有効化します。

0x0 : 両方の PHY を無効化します

0x1 : CC1 PHY を有効化します

0x2 : CC2 PHY を有効化します

0x3 : CC1 および CC2 PHY を有効化します

単一ライン PHY を有効化できるのは、外部 VCONN スイッチを使用して他のラインが VCONN によって駆動されている場合などです。両方の PHY を有効化するのは、シンク/ソースの通常の使用法です。

ビット 9 **ANAMODE** : アナログ PHY 動作モード

0 : 転送元

1 : シンク

CC1 と CC2 の使用は CCENABLE に基づきます。ANASUBMODE[1:0] と一緒に使用した場合のこのビットフィールドの効果については、[表 612 : ANAMODE と ANASUBMODE のコーディングおよび TYPEC_VSTATE_CCx とのリンク](#)を参照してください。

ビット 8:7 **ANASUBMODE[1:0]** : アナログ PHY のサブモード

このビットフィールドの効果については、表 612 : **ANAMODE** と **ANASUBMODE** のコーディングおよび **TYPEC_VSTATE_CCx** とのリンクを参照してください。

ビット 6 **PHYCCSEL** : USB Power Delivery 信号のための CC1/CC2 ラインセクタ

0 : Power Delivery 通信で CC1 IO を使用します。

1 : Power Delivery 通信で CC2 IO を使用します。

選択は、設置時に検知されるケーブルの方向によって決まります。

ビット 5 **PHYRXEN** : USB Power Delivery レシーバの有効化

0 : 無効化

1 : イネーブル

このビットをクリアすると、CC1 と CC2 の両方のレシーバが無効になります。このビットをセットすると、PHYCCSEL ビットを使用して選択した CC レシーバのみが有効になります。

ビット 4 **RXMODE** : レシーバモード

レシーバのモードを決定します。

0 : 通常受信モード

1 : BIST 受信モード (BIST テストデータモード)

このビットがセットされると、RXORDSET は通常どおりに動作します。RXDR はバイトを受信しなくなりますが、CRC チェックは通常のメッセージとして続行します。このモードではヘッダ (MessageID を含む) が受信されないため、このテスト中にまだ送信する必要がある GoodCRC 確認応答に含めるために、受信した MessageID のカウンタをソフトウェアにより自動でインクリメントする必要があります。

ビット 3 **TXHRST** : Tx Hard Reset を送信するためのコマンド

0 : 影響なし。

1 : Tx Hard Reset メッセージを開始します

このビットは、メッセージの送信が開始されるか、破棄されるとすぐに、ハードウェアによりクリアされます。

ビット 2 **TXSEND** : Tx パケットを送信するためのコマンド

0 : 影響なし。

1 : Tx パケットの送信を開始します

このビットは、パケットの送信が開始されるか、破棄されるとすぐに、ハードウェアによりクリアされます。

ビット 1:0 **TXMODE[1:0]** : Tx パケットタイプ

このビットフィールドに書き込むと、値に応じて、次のようなアクションがトリガされます。

0x0 : 他のレジスタで前に定義された Tx パケットの送信

0x1 : Cable Reset シーケンス

0x2 : BIST テストシーケンス (BIST キャリアモード 2)

その他 : 無効

USB PD 仕様の V1.1 以降は、BIST キャリアモード 2 の期間に定義されたカウンタが設けられています。このモードを (「tBISTContMode」遅延後に) 正しく終了するには、ペリフェラルを無効化します (UCPDEN = 0)。

56.8.5 UCPD 割込みマスクレジスタ (UCPD_IMR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVTIE	Res.	Res.	Res.	Res.
											r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2IE	TYPECEVT1IE	Res.	RXMSGENDIE	RXOVRIE	RXHRSTDETIE	RXORDDDETIE	RXNEIE	Res.	TXUNDIE	HRSTSENTIE	HRSTDISCIE	TXMSGABTIE	TXMSGSENTIE	TXMSGDISCIE	TXISIE
rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **FRSEVTIE** : FRSEVT 割込み有効

0 : 無効化

1 : イネーブル

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TYPECEVT2IE** : TYPECEVT2 割込み有効

0 : 無効化

1 : イネーブル

ビット 14 **TYPECEVT1IE** : TYPECEVT1 割込み有効

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **RXMSGENDIE** : RXMSGEND 割込み有効

0 : 無効化

1 : イネーブル

ビット 11 **RXOVRIE** : RXOVR 割込み有効

0 : 無効化

1 : イネーブル

ビット 10 **RXHRSTDETIE** : RXHRSTDET 割込み有効

0 : 無効化

1 : イネーブル

ビット 9 **RXORDDDETIE** : RXORDDDET 割込み有効

0 : 無効化

1 : イネーブル

ビット 8 **RXNEIE** : RXNE 割込み有効

0 : 無効化

1 : イネーブル

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TXUNDIE** : TXUND 割込み有効

0 : 無効化
1 : イネーブル

ビット 5 **HRSTSENTIE** : HRSTSENT 割込み有効

0 : 無効化
1 : イネーブル

ビット 4 **HRSTDISCIE** : HRSTDISC 割込み有効

0 : 無効化
1 : イネーブル

ビット 3 **TXMSGABTIE** : TXMSGABT 割込み有効

0 : 無効化
1 : イネーブル

ビット 2 **TXMSGSENTIE** : TXMSGSENT 割込み有効

0 : 無効化
1 : イネーブル

ビット 1 **TXMSGDISCIE** : TXMSGDISC 割込み有効

0 : 無効化
1 : イネーブル

ビット 0 **TXISIE** : TXIS 割込み有効

0 : 無効化
1 : イネーブル

56.8.6 UCPD ステータスレジスタ (UCPD_SR)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

フラグ (単一ビットのステータスビットフィールド) は割込みリクエストと関連付けられます。UCPD_IMR レジスタの対応するビットによって有効化された場合に、割込みが生成されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVT	TYPEC_VSTATE_CC2 [1:0]		TYPEC_VSTATE_CC1 [1:0]	
											r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2	TYPECEVT1	RXERR	RXMSGEND	RXOVR	RXHRSTDET	RXORDET	RXNE	Res.	TXUND	HRSTSENT	HRSTDISC	TXMSGABT	TXMSGSENT	TXMSGDISC	TXIS
r	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **FRSEVT** : FRS 検出イベント

フラグをクリアするには、FRSEVTCF ビットをセットします。

0 : 新規イベントはありません。

1 : 新しい FRS 受信イベントが発生しました

ビット 19:18 **TYPEC_VSTATE_CC2[1:0]** : CC2 ラインの電圧レベル

このステータスビットフィールドは、定常状態の CC2 ラインの電圧レベルを示します。

0x0 : 最低

0x1 : 低

0x2 : 高

0x3 : 最高

USB PD メッセージ中の BMC PHY 変調による CC2 ラインの電圧の変化は、ビットフィールド値に影響しません。

ビット 17:16 **TYPEC_VSTATE_CC1[1:0]** :

このステータスビットフィールドは、定常状態の CC1 ラインの電圧レベルを示します。

0x0 : 最低

0x1 : 低

0x2 : 高

0x3 : 最高

USB PD メッセージ中の BMC PHY 変調による CC1 ラインの電圧の変化は、ビットフィールド値に影響しません。

ビット 15 **TYPECEVT2** : CC2 ラインでの Type-C の電圧レベルイベント

このフラグは、新しい Type-C イベントに対応する TYPEC_VSTATE_CC2[1:0] ビットフィールド値の変化を示します。クリアするには、TYPECEVT2CF ビットをセットします。

0 : 新規イベントはありません。

1 : 新しい Type-C イベント

ビット 14 **TYPECEVT1** : CC1 ラインでの Type-C の電圧レベルイベント

このフラグは、新しい Type-C イベントに対応する TYPEC_VSTATE_CC1[1:0] ビットフィールド値の変化を示します。クリアするには、TYPECEVT2CF ビットをセットします。

0 : 新規イベントはありません。

1 : 新しい Type-C イベント

ビット 13 **RXERR** : 受信メッセージエラー

このフラグは、不適切な CRC や切捨てられたメッセージ（EOP が満たされる前に静的になるライン）など、(RXMSGEND を使用して) 宣言された最後の Rx メッセージのエラーを示します。RXMSGEND フラグがセットされると必ずアサートされます。

0 : エラーの検出なし

1 : エラー検出

ビット 12 **RXMSGEND** : Rx メッセージ受信

このフラグは、CRC 値に関係なく、メッセージ（Hard Reset メッセージを除く）が受信されたかどうかを示します。このフラグをクリアするには、RXMSGENDCF ビットをセットします。

0 : 新しい Rx メッセージを受信していません

1 : 新しい Rx メッセージを受信しました

RXMSGEND フラグがハイになったときにセットされる RXERR フラグは、最後に受信したメッセージのエラーを示します。

ビット 11 **RXOVR** : Rx データオーバーフロー検出

このフラグは Rx データバッファのオーバーフローを示します。クリアするには、RXOVRCF ビットをセットします。

0 : オーバーフローなし

1 : オーバーフロー

バッファのオーバーフローは、受信データの読み出し速度が十分でない場合に発生する可能性があります。

ビット 10 **RXHRSTDET** : Rx Hard Reset 受信検出

このフラグは、有効な Hard Reset メッセージを受信したことを示します。クリアするには、RXHRSTDETCF ビットをセットします。

- 0 : Hard Reset を受信していません
- 1 : Hard Reset を受信しました

ビット 9 **RXORDDDET** : Rx 順序集合 (4 つの K コード) 検出

このフラグは、順序集合が検出されたことを示します。関連情報は、UCPD_RX_ORDSET レジスタの RXORDSET[2:0] ビットフィールドに保存されます。クリアするには、RXORDDDETCF ビットをセットします。

- 0 : 順序集合が検出されていません
- 1 : 新しい順序集合が検出されました

ビット 8 **RXNE** : 受信データレジスタノットエンプティ検出

このフラグは、UCPD_RXDR レジスタが空でないことを示します。UCPD_RXDR の読出し時に自動的にクリアされます。

- 0 : Rx データレジスタは空です。
- 1 : Rx データレジスタは空ではありません。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TXUND** : Tx データアンダーラン検出

このフラグは、送信メッセージが正常に実行されるように時間内に Tx データレジスタ (UCPD_TXDR) が書き込まれなかったことを示します。クリアするには、TXUND CF ビットをセットします。

- 0 : Tx データアンダーランが検出されていません
- 1 : Tx データアンダーランが検出されました

ビット 5 **HRSTSENT** : Hard Reset メッセージ送信済み

このフラグは、Hard Reset メッセージが送信されることを示します。このフラグをクリアするには、HRSTSENT CF ビットをセットします。

- 0 : Hard Reset メッセージは送信されません
- 1 : Hard Reset メッセージが送信されます

ビット 4 **HRSTDISC** : Hard Reset 破棄済み

このフラグは、Hard Reset メッセージが破棄されることを示します。このフラグをクリアするには、HRSTDISCCF ビットをセットします。

- 0 : Hard Reset は破棄されません
- 1 : Hard Reset が破棄されます

ビット 3 **TXMSGABT** : 送信メッセージのアボート

このフラグは、後続の Hard Reset メッセージ送信リクエストが送信中に優先されるために、Tx メッセージがアボートされることを示します。クリアするには、TXMSGABT CF ビットをセットします。

- 0 : 送信メッセージのアボートなし
- 1 : 送信メッセージのアボート

ビット 2 **TXMSGSENT** : メッセージ送信の完了

このフラグは、パケット送信が完了したことを示します。クリアするには、TXMSGSENT CF ビットをセットします。

- 0 : Tx メッセージが完了していません
- 1 : Tx メッセージが完了しました

Hard Reset によってメッセージ送信の割込みが発生した場合、このフラグは立てられません。

ビット 1 **TXMSGDISC** : メッセージ送信破棄

このフラグは、メッセージ送信が破棄されたことを示します。このフラグをクリアするには、TXMSGDISCCF ビットをセットします。

0 : Tx メッセージが破棄されていません

1 : Tx メッセージが破棄されました

同時に受信が行われている場合や、ラインに過剰なノイズがある場合は、メッセージの送信が破棄されることがあります。Tx メッセージの破棄後、CC ラインがアイドル状態になった場合のみ、フラグが立てられます。

ビット 0 **TXIS** : 送信割込みステータス

このフラグは、UCPD_TXDR レジスタが空で、新しいデータの書込みが必要であることを示します（送信されたデータの量が、TXPAYSZ ビットフィールドで定義されたペイロードサイズに達していないため）。UCPD_TXDR レジスタにデータが書き込まれると、このフラグはクリアされます。

0 : 新しい Tx データ書込みは不要です

1 : 新しい Tx データ書込みが必要です

56.8.7 UCPD 割込みクリアレジスタ (UCPD_ICR)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVTCF	Res.	Res.	Res.	Res.
											w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPECEVT2CF	TYPECEVT1CF	Res.	RXMSGENDCF	RXOVRCF	RXHRSTDETCF	RXORDDETCF	Res.	Res.	TXUNDCF	HRSTSENTCF	HRSTDISCCF	TXMSGABTCF	TXMSGSENTCF	TXMSGDISCCF	Res.
w	w		w	w	w	w			w	w	w	w	w	w	

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **FRSEVTCF** : FRS イベントフラグ (FRSEVT) クリア

このビットをセットすると UCPD_SR レジスタの FRSEVT フラグがクリアされます。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TYPECEVT2CF** : Type-C CC2 ラインイベントフラグ (TYPECEVT2) クリア

このビットをセットすると UCPD_SR レジスタの TYPECEVT2 フラグがクリアされます

ビット 14 **TYPECEVT1CF** : Type-C CC1 イベントフラグ (TYPECEVT1) クリア

このビットをセットすると UCPD_SR レジスタの TYPECEVT1 フラグがクリアされます

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **RXMSGENDCF** : Rx メッセージ受信フラグ (RXMSGEND) クリア

このビットをセットすると UCPD_SR レジスタの RXMSGEND フラグがクリアされます。

ビット 11 **RXOVRCF** : Rx オーバーフローフラグ (RXOVR) クリア

このビットをセットすると UCPD_SR レジスタの RXOVR フラグがクリアされます。

- ビット 10 **RXHRSTDETCF** : Rx Hard Reset 検出フラグ (RXHRSTDET) クリア
このビットをセットすると UCPD_SR レジスタの RXHRSTDET フラグがクリアされます。
- ビット 9 **RXORDDETCF** : Rx 順序集合検出フラグ (RXORDDET) クリア
このビットをセットすると UCPD_SR レジスタの RXORDDET フラグがクリアされます。
- ビット 8:7 予約済みであり、リセット値に保持する必要があります。
- ビット 6 **TXUNDCF** : Tx アンダーフローフラグ (TXUND) クリア
このビットをセットすると UCPD_SR レジスタの TXUND フラグがクリアされます。
- ビット 5 **HRSTSENTCF** : Hard Reset 送信フラグ (HRSTSENT) クリア
このビットをセットすると UCPD_SR レジスタの HRSTSENT フラグがクリアされます。
- ビット 4 **HRSTDISCCF** : Hard Reset 破棄フラグ (HRSTDISC) クリア
このビットをセットすると UCPD_SR レジスタの HRSTDISC フラグがクリアされます。
- ビット 3 **TXMSGABTCF** : Tx メッセージアバートフラグ (TXMSGABT) クリア
このビットをセットすると UCPD_SR レジスタの TXMSGABT フラグがクリアされます。
- ビット 2 **TXMSGSENTCF** : Tx メッセージ送信フラグ (TXMSGSENT) クリア
このビットをセットすると UCPD_SR レジスタの TXMSGSENT フラグがクリアされます。
- ビット 1 **TXMSGDISCCF** : Tx メッセージ破棄フラグ (TXMSGDISC) クリア
このビットをセットすると UCPD_SR レジスタの TXMSGDISC フラグがクリアされます。
- ビット 0 予約済みであり、リセット値に保持する必要があります。

56.8.8 UCPD Tx 順序集合タイプレジスタ (UCPD_TX_ORDSETR)

アドレスオフセット : 0x01C

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっていて、パケット送信が実行中でない (TXSEND および TXHRST ビットが両方ともロー) 場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXORDSET[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXORDSET[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **TXORDSET[19:0]** : 送信対象の順序集合

このビットフィールドで、それぞれ 5 ビットの 4 つの K コードからなり、送信パケットを定義する、送信対象のフル 20 ビットシーケンスを規定します。ビット 0 (K コード 1 のビット 0) を最初に受信し、ビット 19 (K コード 4 のビット 4) を最後に受信します。

56.8.9 UCPD Tx ペイロードサイズレジスタ (UCPD_TX_PAYSZR)

アドレスオフセット : 0x020

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TXPAYSZ[9:0]									

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **TXPAYSZ[9:0]** : これから送信するペイロードサイズ

このビットフィールドは、ソフトウェアとハードウェアによって変更されます。これから送信するペイロード (ヘッダは含まれますが、CRC は含まれません) のバイト数で構成され、データバイトが UCPD_TXDR レジスタに書き込まれるたびに、ビットフィールド値が 0 に達した場合を除いて、ビットフィールド値がデクリメントされ、TXIS ビットがセットされます。エnumレーションした値は、送信開始前の標準ペイロードサイズです。

0x2 : 2 バイト - プロトコル層からの制御メッセージのサイズ

0x6 : 6 バイト - プロトコル層で許容される最短のデータメッセージ

0x1E : 30 バイト - プロトコル層で許容される最長の非拡張データメッセージ

0x106 : 262 バイト - 許容される最長の拡張メッセージ

0x3FF : 1024 バイト - 許容される最長のペイロード (将来の拡張用)

56.8.10 UCPD Tx データレジスタ (UCPD_TXDR)

アドレスオフセット : 0x024

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが有効 (UCPDEN = 1) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXDATA[7:0]** : 送信対象のデータバイト

56.8.11 UCPD Rx 順序集合レジスタ (UCPD_RX_ORDSETR)

アドレスオフセット : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPKINVALID[2:0]		RXSOP3OF4		RXORDSET[2:0]		
									r	r	r	r	r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **RXSOPKINVALID[2:0]** :

このビットフィールドは、デバッグ目的でのみ使用されます。

0x0 : 壊れた K コードはありません

0x1 : 最初の K コードが壊れています

0x2 : 2 番目の K コードが壊れています

0x3 : 3 番目の K コードが壊れています

0x4 : 4 番目の K コードが壊れています

その他 : 無効

ビット 3 **RXSOP3OF4** :

このビットは、正常な K コードの数を示します。デバッグ専用です。

0 : 4 つの K コードのうち、4 つが正常です

1 : 4 つの K コードのうち、3 つが正常です

ビット 2:0 **RXORDSET[2:0]** : Rx 順序集合コード検出

0x0 : レシーバで SOP コードを検出しました。

0x1 : レシーバで SOP' コードを検出しました。

0x2 : レシーバで SOP" コードを検出しました。

0x3 : レシーバで SOP'_Debug コードを検出しました。

0x4 : レシーバで SOP"_Debug コードを検出しました。

0x5 : レシーバで Cable Reset を検出しました。

0x6 : レシーバで SOP 拡張 #1 を検出しました。

0x7 : レシーバで SOP 拡張 #2 を検出しました。

56.8.12 UCPD Rx ペイロードサイズレジスタ (UCPD_RX_PAYSZR)

アドレスオフセット : 0x02C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RXPAYSZ[9:0]									
						r	r	r	r	r	r	r	r	r	r

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **RXPAYSZ[9:0]** : Rx ペイロードサイズ受信

このビットフィールドには、受信したペイロード（ヘッダは含まれますが、CRC は含まれません）のバイト数が格納されます。UCPD_RXDR レジスタで新しいデータバイトを受信するたびに、ビットフィールド値がインクリメントされ、RXMSGEND フラグがセットされます（そして、有効な場合は割込みが生成されます）。

0x2 : 2 バイト - プロトコル層からの制御メッセージのサイズ

0x6 : 6 バイト - プロトコル層で許容される最短のデータメッセージ

0x1E : 30 バイト - プロトコル層で許容される最長の非拡張データメッセージ

0x106 : 262 バイト - 許容される最長の拡張メッセージ

0x3FF : 1024 バイト - 許容される最長のペイロード（将来の拡張用）

このビットフィールドでは、バイト受信の実行中の場合（RXMSGEND フラグがロー）、不正な値が返されることがあります。

56.8.13 UCPD 受信データレジスタ (UCPD_RXDR)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXDATA[7:0]** : データバイト受信済み

56.8.14 UCPD Rx 順序集合拡張レジスタ 1 (UCPD_RX_ORDEXTR1)

アドレスオフセット : 0x034

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが無効 (UCPDEN = 0) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPX1[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXSOPX1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **RXSOPX1[19:0]** : 順序集合 1 受信済み

このビットフィールドには、それぞれ 5 ビットの 4 つの K コードからなる、受信済みのフル 20 ビットシーケンスが格納されます。ビット 0 (K コード 1 のビット 0) を最初に受信し、ビット 19 (K コード 4 のビット 4) を最後に受信します。

56.8.15 UCPD Rx 順序集合拡張レジスタ 2 (UCPD_RX_ORDEXTR2)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

このレジスタへの書込みが可能なのは、ペリフェラルが無効 (UCPDEN = 0) になっている場合のみです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPX2[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXSOPX2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **RXSOPX2[19:0]** : 順序集合 2 受信済み

このビットフィールドには、それぞれ 5 ビットの 4 つの K コードからなる、受信済みのフル 20 ビットシーケンスが格納されます。ビット 0 (K コード 1 のビット 0) を最初に受信し、ビット 19 (K コード 4 のビット 4) を最後に受信します。

56.8.16 UCPD レジスタマップ

表 616. UCPD レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	UCPD_CFGR1	UCPDEN	RXDMAEN	TXDMAEN	RXORDSETEN[8:0]										PSC_USBDPCLK[2:0]			Res.	TRANSWIN[4:0]				IFRGAP[4:0]				HBITCLKDIV[5:0]							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	UCPD_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUPEN	FORCECLK	RXFILT2N3	RXFILT1DIS	
	リセット値																								0					0	0	0	0	
0x008	UCPD_CFGR3	Res.	Res.	Res.	TRIM_CC2_RP [3:0]					Res.	Res.	Res.	Res.	Res.	TRIM_CC2_RD [3:0]				Res.	Res.	Res.	TRIM_CC1_RP [3:0]					Res.	Res.	TRIM_CC1_RD [3:0]					
	リセット値				0	0	0	0							0	0	0	0				0	0	0	0					0	0	0	0	
0x00C	UCPD_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2TCDIS	CC1TCDIS	Res.	RDCH	FRSTX	FRSRXEN	Res.	Res.	Res.	Res.	CCENABLE[1:0]		ANAMODE		ANASUBMODE[1:0]		PHYCCSEL	PHYRXEN	RXMODE	TXHRST	TXSEND	TXMODE[1:0]
	リセット値												0	0		0	0	0		Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0	0
0x010	UCPD_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVTIE	Res.	Res.	Res.	TYPECEVT2IE	TYPECEVT1IE	Res.	Res.	RXMSGENDIE	RXOVRIE	RXHRSTDETIE	RXORDDETIE	RXNEIE	Res.	TXUNDIE	HRSTSENTIE	HRSTDISCIE	TXMSGABTIE	TXMSGSENTIE	TXMSGDISCIE	TXISIE
	リセット値												0					0	0			0	0	0	0	0		0	0	0	0	0	0	0
0x014	UCPD_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVT	TYPECV_STATE_CC2 [1:0]		TYPECV_STATE_CC1 [1:0]		TYPECEVT2	TYPECEVT1	RXERR	RXMSGEND	RXOVR	RXHRSTDET	RXORDDET	RXNE	Res.	TXUND	HRSTSENT	HRSTDISC	TXMSGABT	TXMSGSENT	TXMSGDISC	TXIS
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x018	UCPD_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRSEVTCF	Res.	Res.	Res.	TYPECEVT2CF	TYPECEVT1CF	Res.	RXMSGENDCF	RXOVR	RXHRSTDETCF	RXORDDETCF	Res.	Res.	TXUNDCF	HRSTSENTCF	HRSTDISCCF	TXMSGABTCF	TXMSGSENTCF	TXMSGDISCCF	Res.	
	リセット値													0				0	0		0	0	0	0	0		0	0	0	0	0	0	0	
0x01C	UCPD_TX_ORDSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXORDSET[19:0]																				
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 616. UCPD レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
0x020	UCPD_TX_PAYSZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXPAYSZ[9:0]																
	リセット値																							0	0	0	0	0	0	0	0	0	0							
0x024	UCPD_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]													
	リセット値																									0	0	0	0	0	0	0	0							
0x028	UCPD_RX_ORDSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPKINVALID[2:0]				RXSOP3OF4				RXORDSET[2:0]				
	リセット値																										0	0	0	0	0	0	0	0						
0x02C	UCPD_RX_PAYSZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXPAYSZ[9:0]															
	リセット値																								0	0	0	0	0	0	0	0	0	0						
0x030	UCPD_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]													
	リセット値																										0	0	0	0	0	0	0	0						
0x034	UCPD_RX_ORDEXTR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPX1 [19:0]																										
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x038	UCPD_RX_ORDEXTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSOPX2 [19:0]																										
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x03C - 0x3FF	予約済みです。	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

57 Ethernet (ETH) : メディアアクセス制御 (MAC)、DMA コントローラ付き

57.1 Ethernetの概要

Portions Copyright (c) Synopsys, Inc. All rights reserved. 掲載許可取得済み。

Ethernetペリフェラルでは、IEEE 802.3-2008 標準に準拠した Ethernet 経由でデータを送受信できます。ペリフェラルは、さまざまなコンシューマや産業アプリケーションのニーズに合わせて設定できます。

57.2 Ethernet の主な機能

Ethernet ペリフェラルは、ダイレクトメモリインタフェースの専用 DMA、複数のフォーマットをサポートするメディアアクセスコントローラ (MAC) および PHY インタフェースブロックを内蔵しています。

57.2.1 準拠規格

Ethernet ペリフェラルは、以下の標準に準拠しています。

- IEEE 802.3-2008 (Ethernet MAC)、メディア独立インタフェース (MII)
- IEEE 1588-2008 (高精度ネットワーク化クロック同期 (PTP))
- IEEE 802.3az-2010 (省電力型 Ethernet (EEE))
- AMBA 2.0 (AHB マスタおよび AHB スレーブポート)
- RMII 仕様バージョン 1.2 (RMII コンソーシアム)

57.2.2 MAC の機能

MAC Tx および Rx の共通機能

- アプリケーションへの個別の送信、受信、および制御インタフェース
- 以下の PHY インタフェースでの 10、100 Mbps のデータ転送速度
 - IEEE 802.3 準拠の MII インタフェースで、外部 Fast Ethernet PHY と通信します。
 - RMII インタフェースで、外部 Fast Ethernet PHY と通信
- 半二重動作 :
 - CSMA/CD プロトコルサポート
 - バックプレッシャを使用したフロー制御 (実装固有のホワイト・ペーパーと UNH Ethernet Clause 4 MAC テストスイート - 付録 D に基づく)
- 標準 IEEE 802.3az-2010 (MII PHY での省電力型 Ethernet)
- アプリケーション側での 32 ビットデータ転送インタフェース
- 全二重フロー制御動作 (IEEE 802.3x ポーズパケットおよび優先フロー制御)
- RMON または MIB カウンタによるネットワーク統計 (RFC2819/RFC2665 を部分的にサポート)
- IEEE 1588-2002 および IEEE 1588-2008 に示された Ethernet パケットのタイムスタンプ (PTP パケットの Tx または Rx ステータスで示される 64 ビットタイムスタンプ)。Tx 方向でワンステップおよびツーステップのタイムスタンプ両方をサポート
- Pulse-Per-Second (PPS) 出力信号を柔軟に制御 (eth_ptp_pps_out および ETH_PPS_OUT)
- PHY デバイスの設定と管理のための MDIO (Clause 22 および Clause 45) マスタインタフェース

MAC Tx の機能

- プリアンブルとフレーム開始データ (SFD) の挿入
- アプリケーションから送信される各パケット用の個別の 32 ビットステータス
- フレームごとに制御可能な CRC とパッドの自動生成。
- 標準または最大 16 KB までのジャンボ Ethernet パケットをサポートするプログラム可能なパケット長
- プログラム可能なパケット間ギャップ (40 ~ 96 ビット時間、8 ビット時間単位で)
- フロー制御入力がアサートからネゲートに遷移するとき、IEEE 802.3x フロー制御でのゼロクオンタムポーズパケットの自動送信 (全二重モード)
- ソースアドレスフィールド挿入または書換え、およびパケット単位やスタティックグローバル制御のある送信されたパケットでの VLAN の挿入、書き換え、削除
- 最大 2 つの VLAN タグの挿入、書換え、削除
- 全二重モードでの削減されたプリアンブルサイズによるパケット送信のオプション
- キュー / チャンネルベースの VLAN タグの挿入、書換え、削除

MAC Rx の機能

- 自動パッドおよび CRC ストリッピングオプション
- 自動 CRC チェックの無効化オプション
- プリアンブルおよび SFD 削除
- 個別の 112 ビットまたは 128 ビットステータス
- プログラム可能なウォッチドッグタイムアウト制限
- 柔軟なアドレスフィルタモード :
 - 各バイトのマスク付きで 4 つの 48 ビット完全アドレス (DA) フィルタ
 - 各バイトのマスク付きで 4 つの 48 ビット SA アドレス比較チェック
 - マルチキャストおよびユニキャスト (DA) アドレス用 64 ビットハッシュフィルタ
- すべてのマルチキャストアドレスパケットを通過させるオプション
- ネットワーク監視にフィルタリングを使用せずにすべてのパケットを通過させるプロミスキャスモード
- ステータスレポート付きで、すべての着信パケットを (フィルタに従って) 通過
- 追加パケットフィルタ :
 - VLAN タグベース : 外部または内部 VLAN タグに基づいたフィルタによる完全一致およびハッシュベースフィルタ
 - レイヤ 3 およびレイヤ 4 ベース : IPv4 または IPv6 での TCP または UDP
- IEEE 802.1Q VLAN タグ検知および受信パケットで VLAN タグを削除できるオプション
- リモートウェイクアップパケットと AMD マジックパケットの検出
- 受信したポーズパケットをアプリケーションに転送するオプション (全二重モード)
- 受信パケットのレイヤ 3 / レイヤ 4 チェックサムオフロード
- 最大 2 つの VLAN タグのストリップングとステータスでのタグの提示

57.2.3 トランザクションレイヤ (MTL) の機能

MTL Tx および Rx の共通機能

- 32 ビットトランザクションレイヤブロック (アプリケーションと MAC をブリッジ)
- パケットデリミタ付きパケット指向転送の最適化
- EQOS-MTL 設定でのバーストデータ転送をサポートする、最大で MTL Rx キューまたは Tx キューの半分のサイズのプログラマブルバースト長
- 各キューのための閾値プログラム機能 (デフォルトは 64 バイト)

MTL Tx の機能

- 閾値プログラム機能付きの 2048 バイトの送信 FIFO
- MAC への送信に対するストアアンドフォワードメカニズムまたは閾値モード (カットスルー)
- 半二重モードでの衝突パケットの自動再送信
- 遅れ衝突、過剰衝突、過剰遅延、適切なステータスを持つアンダーラン状態に関する破棄パケット
- スストアアンドフォワードモードで送信されたフレームに、IPv4 ヘッダチェックサムと TCP、UDP、または ICMP チェックサムを計算して挿入するモジュール
- Tx FIFO で (アンダーフローが原因で) ドロップされるパケットのためにパルスを生成することによる統計
- 以下のためのパケットレベルの制御
 - VLAN タグの挿入または書換え
 - Ethernet ソースアドレスの挿入
 - レイヤ 3/レイヤ 4 のチェックサム挿入制御
 - ワンステップタイムスタンプ
 - タイムスタンプ制御
 - CRC およびパッド制御

MTL Rx の機能

- 閾値の設定が可能な 2048 バイトの受信 FIFO
- 閾値 (またはカットスルー) モードのプログラム可能な Rx キュー閾値 (デフォルトは 64 バイト固定)
- スストアアンドフォワードモードで、受信時にすべてのエラーパケットをフィルタして、アプリケーションに転送しないオプション
- アンダーサイズの良いパケットを転送するオプション
- Rx FIFO でドロップされるパケットのためにパルスを生成することによる統計
- Rx キューフィルレベルに基づくポーズパケット制御または MAC へのバックプレッシャ信号の自動生成

57.2.4 DMA ブロックの機能

DMA ブロックは、ペリフェラルとシステムメモリでデータを交換します。DMA 転送はソフトウェアディスクリプタ構造によって駆動されます。アプリケーションはレジスタのセット ([セクション 57.11.2 : Ethernet DMA レジスタ](#)を参照) を使用して DMA の動作を制御できます。DMA ブロックは、次の機能をサポートしています。

- 32 ビットデータ転送
- 送信パスと受信パスでの個別の DMA
- パケットデリミタ付きパケット指向 DMA 転送の最適化
- データバッファサポートのためのバイト整列アドレッシング。
- デュアルバッファ (リング) ディスクリプタのサポート
- 最小の CPU 介入で大きなデータブロックの転送を可能にするディスクリプタアーキテクチャ (各ディスクリプタは最大 32 KB のデータを転送できます)
- 正常動作と転送エラーについての包括的なステータスレポート
- 最適なホストバス利用のために、Tx DMA および Rx DMA エンジンのバースト長を個別にプログラム可能
- さまざまな動作条件に対応するプログラム可能な割込みオプション。
- パケットごとの送信または受信完了割込み制御
- 受信エンジンと送信エンジン間のラウンドロビン方式または固定優先順位のアービトレーション
- START モードおよび STOP モード
- ホスト制御 (AHB) アクセス用およびホストデータインタフェース用の個別のポート
- TCP セグメンテーションオフロード (TSO) 機能を有効化した Tx DMA チャンネル
- スループットの向上を目的とした送信ディスクリプタのポスト書き込み用のプログラム可能な制御

57.2.5 バスインタフェースの機能

AHB マスタインタフェース

AHB マスタインタフェース機能は次のとおりです。

- AHB を通じたアプリケーションとのインタフェース
- AHB マスタポートでの 32 ビットデータ
- スプリット、再試行、エラーの AHB レスポンス
- AHB 1 KB 境界バーストリプリット
- ソフトウェア選択による AHB バーストのタイプ (固定バースト、可変バースト、または両方の混合)

AHB マスタインタフェースは次のものを生成しません。

- ラップバースト
- 転送のロックまたは保護

AHB スレーブインタフェース

AHB スレーブインタフェースは、次の機能をサポートしています。

- AHB を通じたアプリケーションとのインタフェース
- CSR アクセス用 AHB スレーブインタフェース (32 ビット)
- すべての AHB バーストタイプ

AHB スレーブインタフェースは次のレスポンスを生成しません。

- スプリット
- 再試行
- 誤差

57.3 Ethernet ピンおよび内部信号

表 617 表 617 にパッケージのピンまたはボールに接続された Ethernet 入出力信号を記載します。アクティブなピンは、選択された PHY (MII または RMII) とデバイス設定によって決まります。

表 618 に、内部 Ethernet 信号を示します。

表 617. Ethernet ペリフェラルピン

ポート名	デジタルポートのタイプ	説明
ETH_COL	入力	衝突検出信号、MII のみ
ETH_CRS	入力	キャリアセンス信号、MII のみ
ETH_REF_CLK	入力	RMII リファレンスクロック
ETH_RX_CLK	入力	Rx データ転送用の MII 基準タイミング
ETH_RXD[3:0]	入力	受信データ MII 用に 4 つのピン、RMII 用に 2 つのピン。
ETH_RX_DV	入力	受信データが有効
ETH_CRS_DV	入力	RMII : キャリアセンス (CRS) と RX データ有効 (RX DV) をオルタネートクロックサイクルで多重化 10 Mbit/s モードでは、10 クロックサイクルごとに切り替わります。
ETH_RX_ER	入力	受信エラー
ETH_TX_CLK	入力	Tx データ転送用の MII 基準タイミング
ETH_TXD[3:0]	出力	送信データ MII 用に 4 つのピン、RMII 用に 2 つのピン。
ETH_TX_EN	出力	送信データが有効
ETH_TX_ER	出力	送信エラー
ETH_MDC	出力	管理データクロック
ETH_MDIO	入力/出力	管理データ
ETH_PHY_INTN	入力	PHY 割込み
ETH_PPS_OUT	出力	PTP の PPS (Pulse-per-second) 出力

表 618. Ethernet 内部入力／出力信号

信号名	信号タイプ	説明
eth_hclk	デジタル入力	AHB クロック
eth_sbd_intr_it	デジタル出力	メイン Ethernet 割込み
lpi_intr_o	デジタル出力	トランスミッタまたはレシーバが LPI 状態の移行時または終了時に生成されるサイドバンド信号
pmt_intr_o	デジタル出力	有効なリモートウェイクアップパケットが受信されたときに生成されるサイドバンド信号
eth_mii_tx_clk	デジタル入力	MII Tx カーネルクロック
eth_mii_rx_clk	デジタル入力	MII Rx カーネルクロック
eth_rmii_ref_clk	デジタル入力	RMII リファレンスカーネルクロック
eth_ptp_pps_out	デジタル出力	PTP の PPS (Pulse-per-second) 信号
mac_speed_o[1:0]	デジタル出力	RCC によって使用される MAC 速度情報
clk_ptp_ref_i	デジタル入力	PTP リファレンスクロック入力
eth_ptp_trig[4:1]	デジタル入力	PTP システム時間の補助スナップショット用のトリガ入力

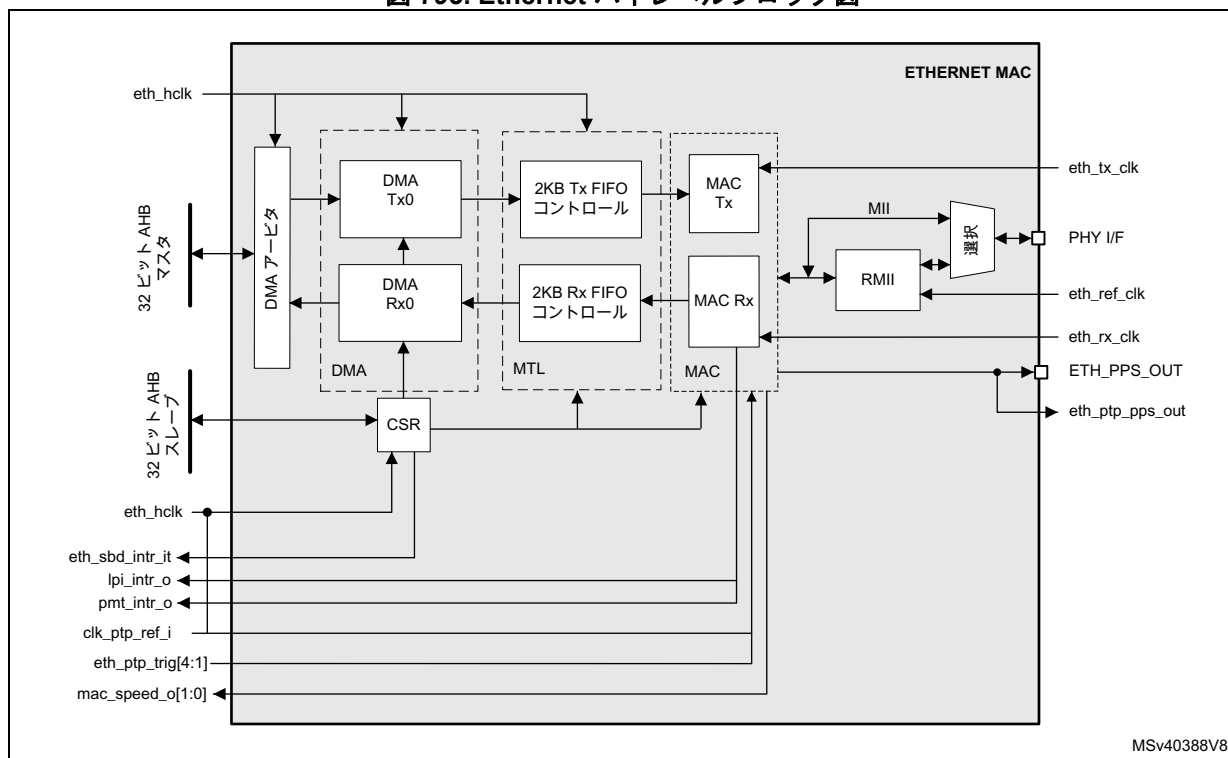
57.4 Ethernet アーキテクチャ

Ethernet ペリフェラルは 4 つの主な機能モジュールで構成されます。

- **制御およびステータスレジスタモジュール (CSR)** は、AHB 32 ビットスレーブインタフェースを通じてレジスタアクセスを制御します。
- **ダイレクトメモリアccessインタフェース (DMA)**
これは、受信に 1 つ、送信に 1 つの物理チャネルを持つ論理 DMA モジュールです。AMBA AHB 32 ビットマスタインタフェースを通じて MAC とシステムメモリでのデータ転送を制御します。
- **メディアアクセス制御モジュール (MAC)** は、Ethernet プロトコルを実装する役割を持っています。
- **MAC トランザクションレイヤ (MTL)** は、アプリケーションと MAC のデータフローを制御する役割を持っています。

RMII PHY メディア独立インタフェースをサポートするには、プロトコル適応モジュールが追加されます。

図 793. Ethernet ハイレベルブロック図



1. 内部信号の定義については、表 618 を参照してください。
2. Ethernet クロックアーキテクチャの詳細な説明については、RCC の「Ethernet のクロック分配」を参照してください。

57.4.1 DMA コントローラ

DMA には、独立した送信 (Tx) エンジンと受信 (Rx) エンジンがあります。Tx エンジンは、データをシステムメモリから MAC トランザクションレイヤ (MTL) に転送し、Rx エンジンは、データをデバイスポート (PHY) からシステムメモリに転送します。

コントローラは、ディスクリプタを使用して、データを転送元から転送先に最小限のアプリケーション CPU 介入によって効率的に移動します。DMA は、Ethernet のパケットなど、パケット指向のデータ転送を行うように設計されています。コントローラは、パケット送受信完了やその他の正常またはエラー条件発生でアプリケーション CPU に割り込むようにプログラムすることができます。

DMA データ構造

DMA とアプリケーションは、次の 2 つのデータ構造を使用して通信します。

- 制御およびステータスレジスタ (CSR)
- ディスクリプタリストおよびデータバッファ

DMA は、MAC によって受信したデータパケットをシステムメモリ内の Rx バッファに、Tx データパケットをシステムメモリ内の Tx バッファから転送します。システムメモリ内に存在するディスクリプタは、これらのバッファに対するポインタを含みます。

各リストのベースアドレスは、それぞれ Tx および Rx レジスタに書き込まれます (チャンネル Tx ディスクリプタリストアドレスレジスタ (ETH_DMACCTXDLAR) およびチャンネル Rx ディスクリプタリストアドレスレジスタ (ETH_DMACRXDLAR))。

ディスクリプタリストは次にリンクされ、次のディスクリプタは常に現在のものに対する固定オフセットで考慮されます。リスト内のディスクリプタの数は、それぞれ Tx/Rx (チャンネル Tx ディスクリプタリング長レジスタ (ETH_DMACTXRLR) およびチャンネル Rx ディスクリプタリング長レジスタ (ETH_DMACRXRLR)) でプログラムされます。

DMA がリストの最終ディスクリプタを処理すると、自動的にリストアドレスレジスタのディスクリプタにジャンプし、ディスクリプタリングを作成します。ディスクリプタリストは、アプリケーションの物理メモリアドレス空間にあります。各ディスクリプタは、最大 2 つのバッファを指し示すことができます。これにより、メモリ内で隣接したバッファでなくても、物理的にアドレス指定された 2 つのバッファを使用できます。

データバッファは、アプリケーション物理メモリ空間に置かれ、パケット全体または一部で構成されますが、1 パケットの長さを超えることはできません。バッファは、データのみを含みます。バッファのステータスは、ディスクリプタ内に保存されます。データ連結とは、複数のデータバッファにまたがるパケットを指します。ただし、1 つのディスクリプタが複数のパケットにまたがることはできません。DMA は、EOP が検出されると、次のパケットのデータバッファにスキップします。

ディスクリプタについては、[セクション 57.10 : ディスクリプタ](#)で指定されています。

DMA アービトレーション

DMA モジュールには、AHB マスタインタフェースの Tx チャンネルアクセスと Rx チャンネルアクセスの間のアービトレーションを行うアービタが組み込まれています。次の 2 種類のアービトレーションをサポートしており、[DMA モードレジスタ \(ETH_DMAMR\)](#) で選択できます。

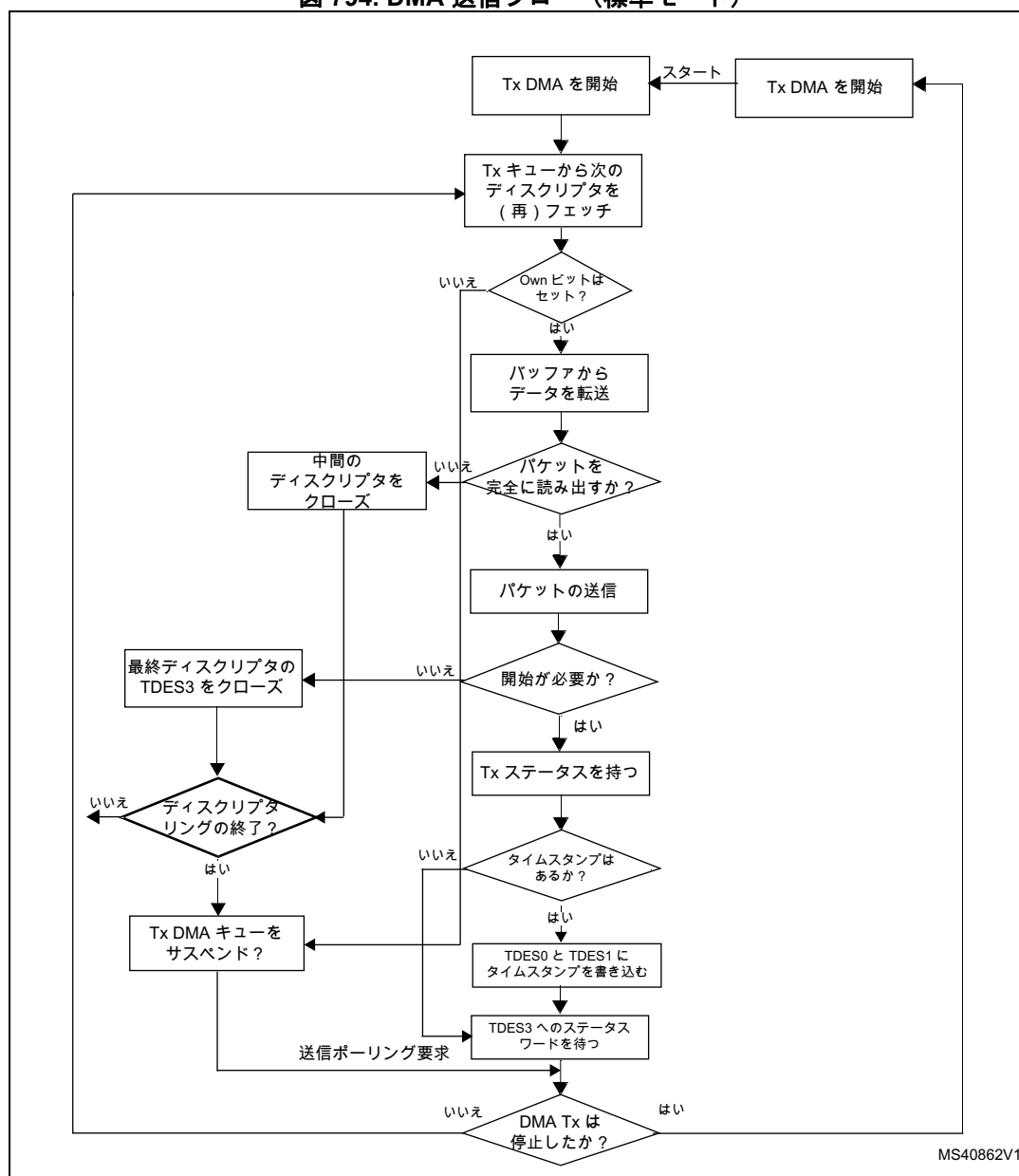
- ラウンドロビン方式アービトレーション : ETH_DMAMR のビット [14:12] でセットされた比率でアービタが Rx と Tx のデータバスを割り当てます。
- 固定優先順位アービトレーション : デフォルトで、Rx DMA は、データアクセスの際、常に Tx DMA より優先されます。ETH_DMAMR レジスタのビット 11 をセットすると、Tx DMA が優先されます。

デフォルトモードでの DMA 送信

デフォルトモードの Tx DMA エンジン、次のように動作します。

1. アプリケーションは、送信ディスクリプタ (TDES0~TDES3) をセットアップし、対応するデータバッファを Ethernet パケットデータでセットアップした後、Own ビット (TDES0[31]) をセットします。
2. アプリケーションは、送信チャネルのディスクリプタのテイルポインタオフセット値をシフトします。
3. DMA は、アプリケーションメモリからディスクリプタをフェッチします。
4. DMA が次の条件のうち 1 つを検出した場合、そのチャネルからの送信はサスペンドされ、対応する DMA チャネルステータスレジスタのビット 2 および 16 がセットされ、Tx エンジンは手順 11 に進みます。
 - ディスクリプタがアプリケーションによって所有されているというフラグが立てられた場合 (TDES3[31] = 0)
 - ディスクリプタのテイルポインタがリングディスクリプタリストモードの現在のディスクリプタと同じ場合
 - エラー条件が発生した場合
5. 取得したディスクリプタに DMA の所有であるというフラグが立てられていた場合 (TDES3[31] = 1)、DMA は、取得したディスクリプタから送信データバッファアドレスをデコードします。
6. DMA は、システムメモリから送信データをフェッチして、送信のために MTL にデータを転送します。
7. Ethernet パケットが複数のディスクリプタのデータバッファ経由で格納された場合、DMA は、中間のディスクリプタをクローズして、次のディスクリプタをフェッチします。Ethernet パケットデータの終わりが MTL に転送されるまで、手順 3~7 が繰り返されます。
8. パケット送信が完了すると、IEEE 1588 タイムスタンプ機能がそのパケットに対して有効な場合 (Tx ステータスに示される)、MTL から取得したタイムスタンプの値が EOP バッファを含む Tx ディスクリプタ (TDES0 および TDES1) に書き込まれます。ステータス情報はこの Tx ディスクリプタ (TDES3) に書き込まれます。Own ビットがこの手順中にクリアされるため、アプリケーションがこのディスクリプタを所有するようになります。タイムスタンプ機能がこのパケットに対して無効である場合、DMA は TDES0 と TDES1 の内容を変更しません。
9. 最終ディスクリプタで完了時割込み (TDES2[31]) がセットされているパケットの送信を完了した後、**チャンネルステータスレジスタ (ETH_DMCSR)** のビット 0 がセットされます。DMA エンジンは手順 3 に戻ります。
10. サスペンド状態では、DMA がディスクリプタの再取得を試みます (その結果、手順 3 に戻ります)。送信ポーリング要求の受け取りと、アンダーフロー割込みステータスビットがクリアされると、任意の値を**チャンネル Tx ディスクリプタテイルポインタレジスタ (ETH_DMACTXDTPR)** に書き込むことでポーリング要求コマンドがトリガされます。アプリケーションが対応する DMA チャネルの送信制御レジスタのビット 0 をクリアすることで DMA を停止させた場合、DMA は停止状態に入ります。

図 794. DMA 送信フロー (標準モード)



MS40862V1

OSP (2 番目のパケットに対する動作) モードでの DMA 送信

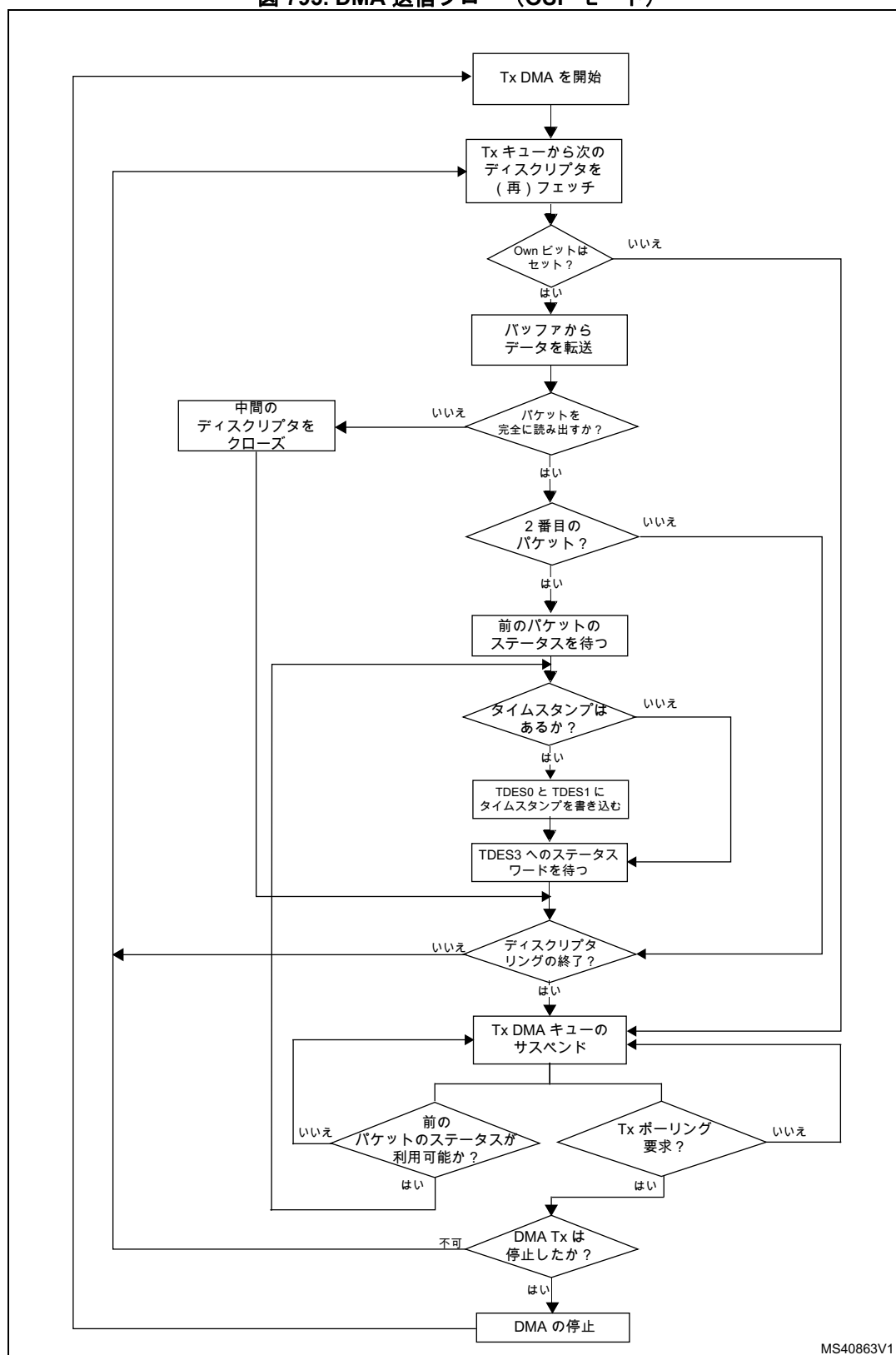
実行状態では、**チャンネル送信制御レジスタ (ETH_DMACTXCR)** でビット 4 がセットされている場合、送信プロセスで最初のパケットのステータスディスクリプタをクローズせずに 2 つのパケットを同時に取得できます。送信プロセスが最初のパケットの転送を完了すると、ただちに 2 番目のパケットの送信ディスクリプタリストにポーリングします。2 番目のパケットが有効な場合、送信プロセスは最初のパケットのステータス情報を書き込む前に、このパケットを転送します。

OSP モードでは、実行状態の DMA 送信は、以下のシーケンスに示すとおり動作します。

1. DMA は、デフォルトモードで DMA 送信シーケンスの手順 1~7 を実行します ([セクション : デフォルトモードでの DMA 送信](#)を参照)。
2. DMA は、前のパケットの最終ディスクリプタをクローズせずに、次のディスクリプタをフェッチします。
3. DMA が取得したディスクリプタを所有している場合、DMA は、このディスクリプタ内の送信バッファアドレスをデコードします。DMA がディスクリプタを所有していない場合、DMA は、サスペンドモードに入り、手順 7 にジャンプします。
4. DMA は、システムメモリから送信パケットをフェッチして、EOP データが転送されるまで、MTL にパケットを転送し、このパケットが複数のディスクリプタに分割されている場合は、中間のディスクリプタをクローズします。
5. DMA は、前のパケットの送信ステータスとタイムスタンプを待ちます。ステータスが入手可能なときには、DMA は、タイムスタンプがキャプチャされていた場合は (ステータスビットで示されます)、タイムスタンプを TDES0 と TDES1 に書き込みます。DMA は、Own ビットがクリアされたステータスを、対応する TDES3 に書き込み、ディスクリプタをクローズします。タイムスタンプ機能が前のパケットに対して有効でなかった場合、DMA は、TDES2 と TDES3 の内容を変更しません。
6. 送信割込みがセットされます (有効時)。DMA は、次のディスクリプタをフェッチしてから、手順 3 へ進みます (ステータスが正常な場合)。前の送信ステータスがアンダーフローエラーを示している場合、DMA はサスペンドモードに移行します (手順 7)。
7. サスペンドモードでは、ペンディングステータスおよびタイムスタンプが MTL から受信される場合、DMA は次の操作を実行します。
 - a) DMA は、TDES2 と TDES3 にタイムスタンプ (現在のパケットに対して有効な場合) を書き込みます。
 - a) DMA は対応する TDES3 にステータスを書き込みます。
 - a) DMA は、該当する割込みをセットして、サスペンドモードに戻ります。ペンディング中のステータスがなく、アプリケーションが対応する DMA チャネルの送信制御レジスタのビット 0 をクリアすることで DMA を停止させた場合、DMA は停止状態に入ります。
8. DMA は、対応するチャネルの送信ディスクリプタのテイルポインタレジスタで送信ポーリング要求を受け取った後にのみ、サスペンドモードから抜けて、実行状態に入ることができます (ペンディングステータスに応じて、手順 1 または手順 2 へ移行します)。

OSP モードでの基本的な DMA 送信フローの説明については、[図 796 : 受信 DMA フロー](#)を参照してください。

図 795. DMA 送信フロー (OSP モード)



MS40863V1

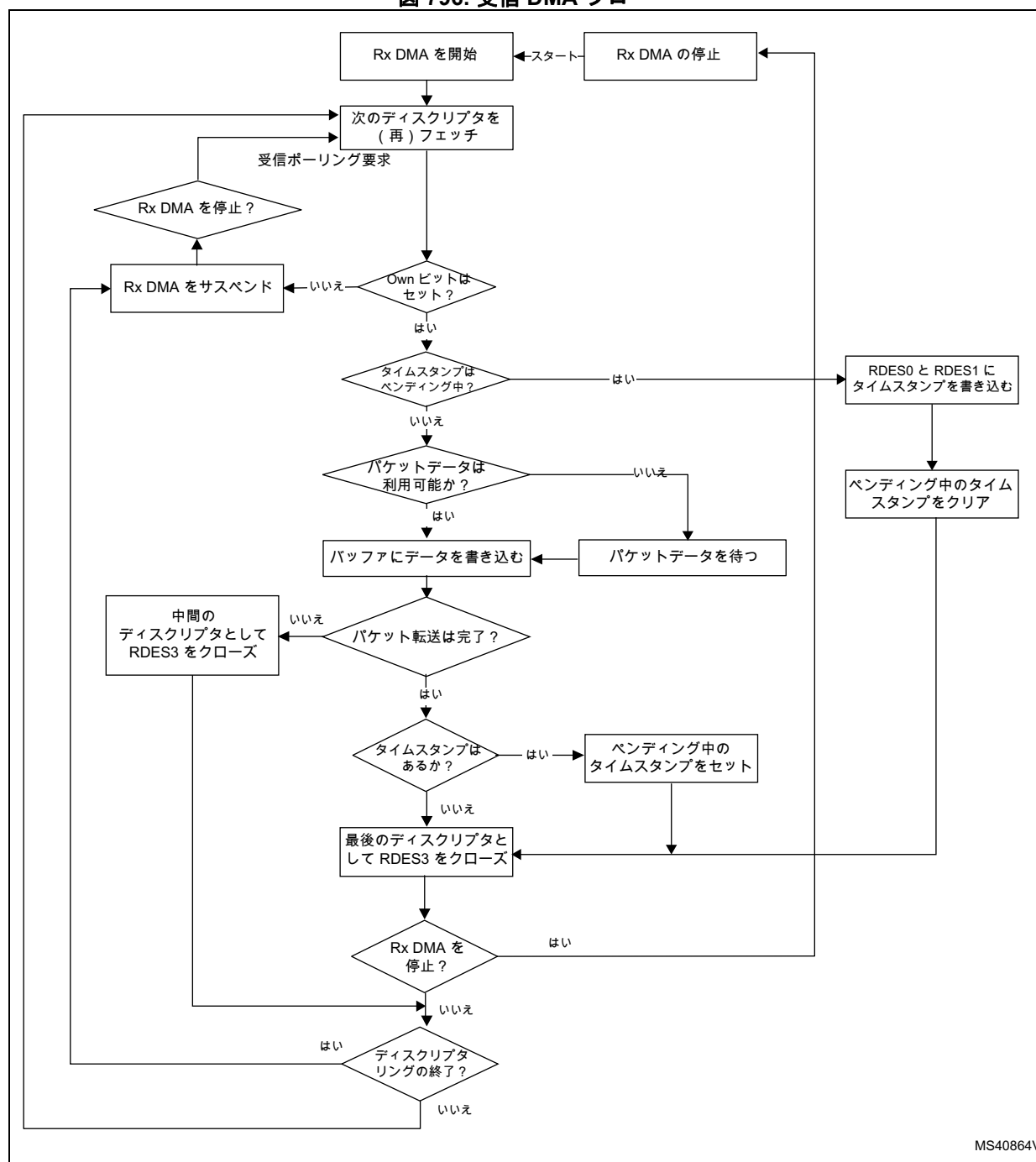
DMA 受信

受信パスで、DMA は MTL 受信キューからパケットを読み出し、対応する DMA チャンネルのパケットデータバッファに書き込みます。

Rx DMA エンジンの受信シーケンスは、次のとおりです (図 796 : 受信 DMA フローも参照)。

1. アプリケーションは、Rx ディスクリプタ (RDES0-RDES3) と Own ビット (RDES3[31]) をセットアップします。アプリケーションは、対応する DMA チャンネルの受信ディスクリプタのテイルポインタレジスタに正しい値をセットする必要があります。
2. **チャンネル受信制御レジスタ (ETH_DMARXCR)** のビット 0 がセットされると、DMA は実行状態に入ります。DMA は Rx 現行ディスクリプタおよびディスクリプタのテイルポインタレジスタ値に基づいて、空いているディスクリプタを探します。空いているディスクリプタがない場合、DMA チャンネルはサスペンド状態に入り、手順 11 に進みます。
3. DMA は、リングで次に使用できるディスクリプタをフェッチし、取得したディスクリプタから受信データバッファアドレスをデコードします。
4. IEEE 1588 タイムスタンプが有効で、タイムスタンプが前のパケットに使用できる場合、DMA はタイムスタンプ (使用できる場合) を現在のディスクリプタの RDES0 および RDES1 に書き込み、CTXT フィールド (RDES3[30]) をセットします。
5. DMA は着信パケットを処理して、取得したディスクリプタのデータバッファに格納します。
6. 現在のパケット転送をまだ完了していない場合、DMA は現在のディスクリプタを中間ディスクリプタとしてクローズして、手順 10 に進みます。
7. DMA は、MTL から受信フレームのステータスを取り出し、そのステータスワードを、Own ビットがクリアされ、最終ディスクリプタビットがセットされている現在のディスクリプタに書き込みます。
8. DMA はフレーム長を RDES3 に、VLAN タグを RDES0 に書き込みます。また、DMA は MAC 制御フレーム OP コード、OAM 制御フレームコード、拡張ステータス情報 (使用できる場合) を、最終ディスクリプタの RDES1 に書き込みます。
9. DMA はタイムスタンプ (使用できる場合) を保存します。DMA は、現在のパケットの最終ディスクリプタの後、コンテキストディスクリプタを (次に使用できるディスクリプタに) 書き込みます。
10. 追加のディスクリプタが Rx DMA ディスクリプタリングで使用できる場合、手順 3 に進みます。そうしないとサスペンド状態 (手順 11) に移行します。
11. 受信 DMA は、受信ポーリング要求が提示され、アプリケーションがチャンネル受信テイルポインタレジスタをインクリメントすると、サスペンド状態から抜けます。エンジンは手順 2 に進み、次のディスクリプタを再びフェッチします。

図 796. 受信 DMA フロー



Tx DMA および Rx DMA 優先度スキーム

DMA アービタは、ディスクリプタやデータバッファにアクセスするため、DMA チャンネル 0 の Tx パスと Rx パスの間のアービトレーションを行います。DMA アービタでは、固定優先順位と加重ラウンドロビンの 2 種類のアービトレーションがサポートされています。[DMA モードレジスタ \(ETH_DMAMR\)](#) の DA ビットでは、所定チャンネルの Tx および Rx DMA の間のアービトレーション構成（固定または加重ラウンドロビン）を指定します。

所定チャンネルの Tx DMA および Rx DMA が有効になっている場合、チャンネルによってパスが制御される場合にバスを取得する DMA を指定する必要があります。対応する Tx DMA と Rx DMA の間の優先順位は、[DMA モードレジスタ \(ETH_DMAMR\)](#) の TXPR フィールドで設定することができます。ラウンドロビン方式アービトレーションの場合、Tx DMA および Rx DMA の間の加重優先順位は、[DMA モードレジスタ \(ETH_DMAMR\)](#) の PR フィールドを通して設定されます。[表 619](#) に、Tx DMA と Rx DMA の間の優先度スキームに関する情報を示します。

表 619. Tx DMA および Rx DMA 優先度スキーム

DMA モードレジスタ (ETH_DMAMR)			優先度スキーム		
PR[2:0]			TXPR	DA	
x	x	x	0	1	Rx は常に Tx より優先されます。
0	0	0	0	0	Tx と Rx は等しい優先度です。同時にリクエストすると、最初に Rx がアクセスされます。
0	0	1	0	0	Rx は 2:1 の比率で Tx より優先されます。
0	1	0	0	0	Rx は 3:1 の比率で Tx より優先されます。
0	1	1	0	0	Rx は 4:1 の比率で Tx より優先されます。
1	0	0	0	0	Rx は 5:1 の比率で Tx より優先されます。
1	0	1	0	0	Rx は 6:1 の比率で Tx より優先されます。
1	1	0	0	0	Rx は 7:1 の比率で Tx より優先されます。
1	1	1	0	0	Rx は 8:1 の比率で Tx より優先されます。
x	x	x	1	1	Tx は常に Rx より優先されます。
0	0	0	1	0	Tx と Rx は等しい優先度です。同時にリクエストすると、最初に Tx がアクセスされます。
0	0	1	1	0	Tx は 2:1 の比率で Rx より優先されます。
0	1	0	1	0	Tx は 3:1 の比率で Rx より優先されます。
0	1	1	1	0	Tx は 4:1 の比率で Rx より優先されます。
1	0	0	1	0	Tx は 5:1 の比率で Rx より優先されます。
1	0	1	1	0	Tx は 6:1 の比率で Rx より優先されます。
1	1	0	1	0	Tx は 7:1 の比率で Rx より優先されます。
1	1	1	1	0	Tx は 8:1 の比率で Rx より優先されます。

57.4.2 MTL

MAC トランザクションレイヤ (MTL) は、アプリケーションシステムメモリと MAC 間のパケットを緩和して調整するための FIFO メモリインタフェースを備えています。また、アプリケーションクロックドメインと MAC クロックドメインの間でデータを転送できるようにします。MTL レイヤでは、送信パスと受信パスの 2 つの 32 ビット幅データパスを備えています。

- 送信パス

アプリケーションまたは内部 DMA は、アプリケーションやシステムメモリから読み出された Ethernet パケットを Tx FIFO にプッシュします。キュー閾値に達した場合 (閾値モード) または完了パケットがキューにある場合 (ストアアンドフォワードモード)、パケットはポップアウトされ、MAC に転送されます。EOP が転送されると、送信ステータスが MAC から取り出されて、アプリケーションまたは内部 DMA に返送されます。Tx キューサイズは 2048 バイトです。

- 受信パス

MTL の Rx モジュールは、MAC からパケットを受信して、Rx キューにプッシュします。キューのステータス (フィルレベル) が設定された受信閾値 (Rx キュー動作モードレジスタ (ETH_MTLRXQOMR) で定義された RTC ビット[1:0]) を超えるか完了パケットを受信すると、そのことがアプリケーションまたは DMA に示されます。また、MTL はキューのフィルレベルを示して、DMA はマスタインタフェースに対して、事前設定されたバースト転送を開始できます。Rx キューサイズは 2048 バイトです。

57.4.3 MAC

MAC は Ethernet プロトコルを処理します。送信モードでは、MTL からデータを受信してから、PHY インタフェースに転送します。受信モードでは、MAC が PHY インタフェースからデータを受信してから、MTL モジュールの Rx FIFO に転送します。

このセクションでは、送受信シーケンスについて簡単に説明します。

MAC 送信

送信シーケンスは、次のとおりです。

1. MTL アプリケーションが SOP (パケットの開始) 信号をアサートしてデータをプッシュすると送信が開始されます。
2. SOF 信号が検出されると、MAC はデータを受け入れて、MII への送信を開始します。
3. EOP (パケットの終了) が MAC に転送されると、MAC は次のいずれかを実行します。
 - MAC は通常送信を完了し、MTL に送信ステータスを提供します。
 - 送信中に通常の衝突 (半二重モード) が発生した場合、MAC は、再試行ビットをセットして送信ステータスを MTL に提供します。MAC は、次のいずれかに当てはまるまで再試行リクエストを提供します。

パケットが正常に送信されました。

再試行リクエストの最大数に達しています。この場合、MAC は過剰衝突送信ステータスのパケット送信をアボートします。MAC は、次の SOP を受信するまですべての追加データを受け入れてドロップします。MAC から (ステータス内で) 再試行リクエストが観測されたときには、MTL ブロックは同じパケットを SOF から再送信しなければなりません。

- 次のいずれかが発生した場合、MAC はパケット送信をアボートします。

キャリアなし (半二重モード)

キャリアの喪失 (半二重モード)

過剰遅延 (半二重モード)

遅れ衝突 (半二重モード)

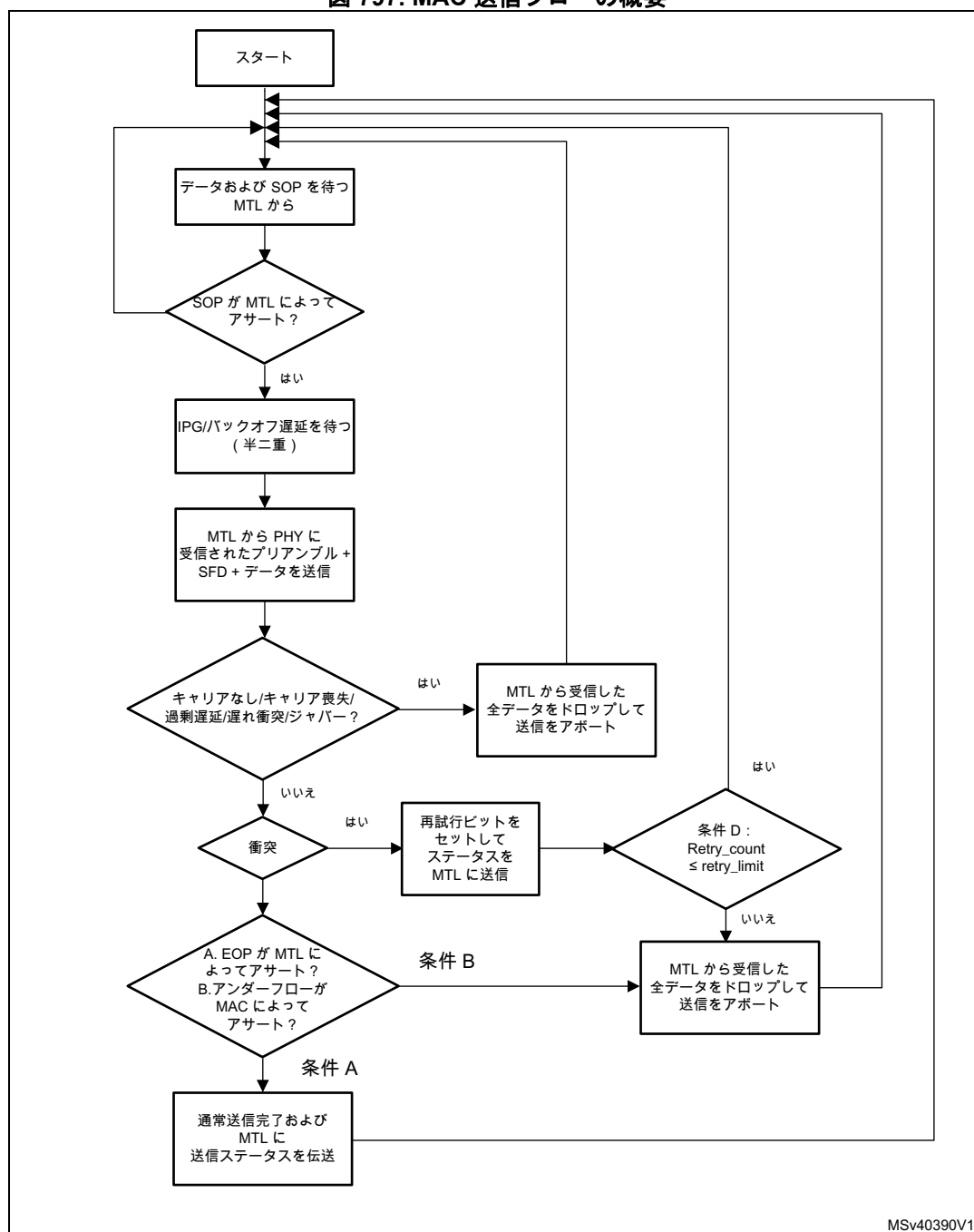
ジャバー

MAC は、次の SOP を受信するまですべての追加データを受け入れてドロップします。

4. MTL が送信中にデータを連続的に供給できなかった場合、MAC は、アンダーフローステータスを発行します。MAC は、次の SOP を受信するまですべての追加データを受け入れてドロップします。
5. MTL からのパケットの通常転送中に、MAC が前のパケットの EOP を受信していないのに SOP を受信した場合、SOP を無視して、新しいパケットを前のパケットの続きとみなします。

図 797 : MAC 送信フローの概要に、MAC 送信プロセスフローの図を示します。

図 797. MAC 送信フローの概要



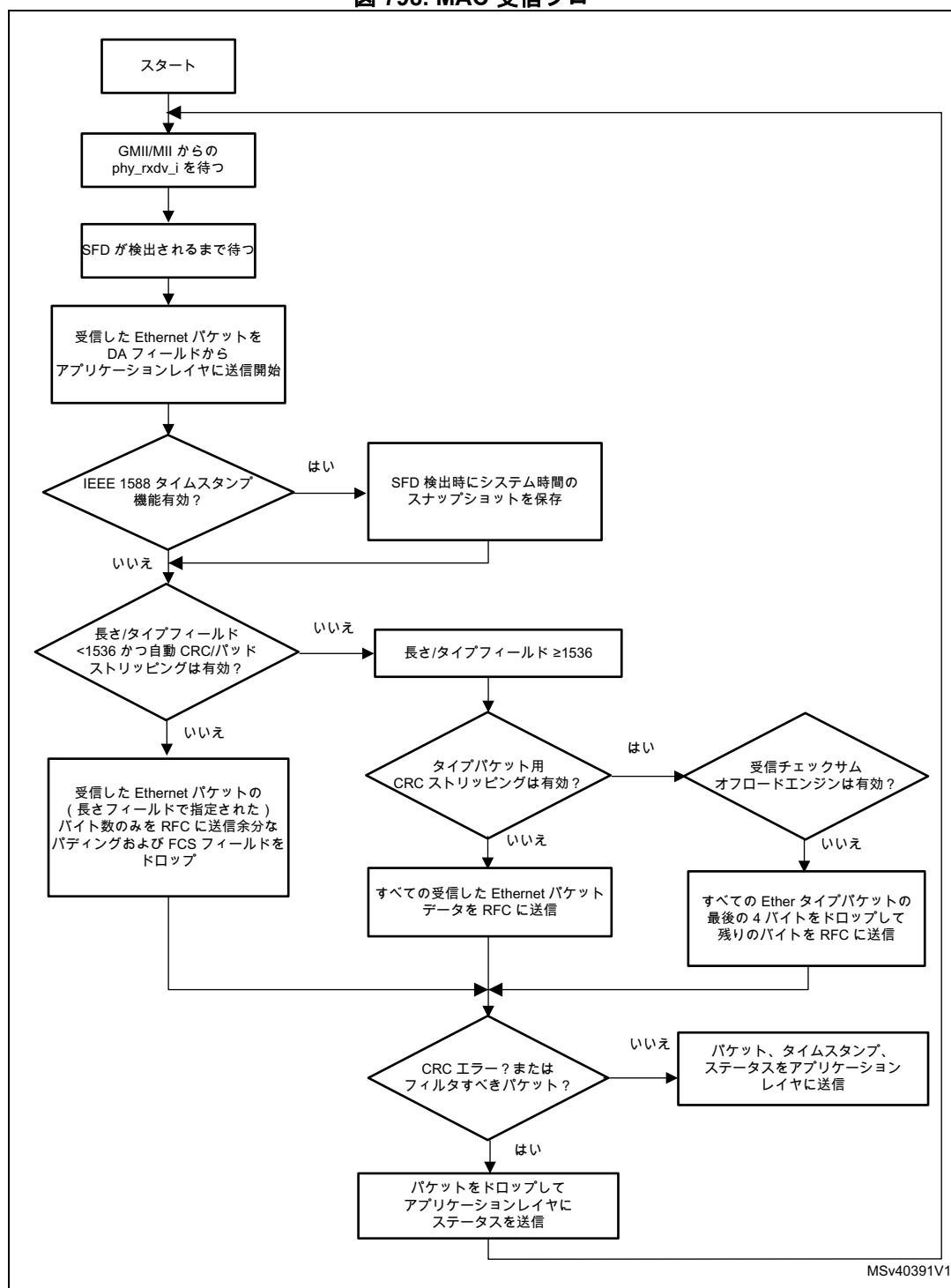
MAC 受信

受信動作は、MAC が MII で SFD を検出したときに開始されます。MAC は、パケットの処理を進める前に、プリアンブルと SFD を削除します。ヘッダフィールドはフィルタリングのためにチェックされ、FCSフィールドはパケットのCRCを検証するために使用されます。受信したパケットは、アドレスフィルタが実行されるまで浅いバッファに格納されます。アドレスフィルタでエラーが発生した場合、そのパケットは MAC でドロップされます。

受信シーケンスは、次のとおりです。

1. MII の受信データ有効信号 (RxDV) がアクティブになると、受信ステートマシン (RSM) は SFD フィールドの検索を開始します (0xD ニブル (訳注: 4ビット))。
ステートマシンは SFD を検出するまで受信パケットをドロップします。
2. SFD が検出されると、ステートマシンは Ethernet パケットのデータの RPC モジュールへの送信を開始し、最初のバイトから始めて、次に SFD (デスティネーションアドレス) という順序で開始します。
3. IEEE 1588 タイムスタンプ機能が有効な場合、任意のパケットの SFD が MII で検出されたとき、MAC はシステム時間のスナップショットを取得します。このパケットが MAC フィルタ中にドロップされない場合、タイムスタンプはアプリケーションに渡されます。MAC は、受信したニブルデータをバイトに変換し、有効なパケットデータを RFC モジュールに転送します。
4. 受信ステートマシンは、受信した Ethernet パケットの長さ/タイプフィールドをデコードします。
この長さ/タイプフィールドが 1,536 未満の場合と、MAC の自動 CRC/パッドストリッピング (動作モード設定レジスタ (ETH_MACCCR) のビット 20) がプログラムされていた場合、ステートマシンは、パケットデータを長さ/タイプフィールドで指定されたカウントまで送信し、バイトのドロップ (FCS フィールドを含む) を開始します。ステートマシンは、長さ/タイプフィールドをデコードして、長さ解釈をチェックします。
5. 長さ/タイプフィールドが 1,536 以上の場合、RPE モジュールは、動作モード設定レジスタ (ETH_MACCCR) のビット 21 でタイプパケットの CRC ストリッピングを有効にしていなくても、受信したすべての Ethernet パケットデータを RFC モジュールに送信します。ただし、CRC ストリッピングがタイプパケットに対して有効で、受信チェックサムオフロードエンジンに有効ではない場合、MAC は Ether タイプのすべてのパケットの最後の 4 バイトを削除してドロップしてから、パケットをアプリケーションに転送します。
6. デフォルトでは、MAC はウォッチドッグタイマを有効にするようプログラムされており、2,048 (ジャンボパケットが有効の場合は 10,240) バイト以上のパケット (DA + SA + LT + データ + パッド + FCS) は RPE モジュールで切り捨てられます。さらに、プログラム可能なウォッチドッグタイマ (ウォッチドッグタイムアウトレジスタ (ETH_MACWTR) のビット 16) を使用して、2,048 バイトまたは 10,240 バイトの固定タイムアウトを上書きできます。ウォッチドッグタイマは、動作モード設定レジスタ (ETH_MACCCR) のビット 19 をプログラムすることで無効にできます。ただし、ウォッチドッグタイマが無効な場合でも、32 KB を超えるパケットは切り捨てられ、ウォッチドッグタイムアウトステータスが与えられます。

図 798. MAC 受信フロー



57.5)Ethernet 機能の説明 : MAC

57.5.1 ダブル VLAN 処理

Ethernet ペリフェラルは、MAC が最大 2 つの VLAN タグ（内部および外部）を処理できるダブル VLAN（仮想 LAN）タグ機能をサポートします。

MAC は、次の機能をサポートしています。

- 送信パスでの最大 2 つの VLAN タグの挿入、書き換え、または削除
- 受信パスでの 2 つの VLAN タグのうち 1 つに基づいたパケットフィルタおよびストリッピング。
受信ステータスの一部としての受信パスでの最大 2 つの VLAN タグのストリッピングと提示

送信パス

表 620 : Tx パスでのダブル VLAN 処理機能 表 620 : 「Tx パスでのダブル VLAN 処理機能」には、送信側で MAC によってサポートされる機能を記載しています。

表 620. Tx パスでのダブル VLAN 処理機能

機能	説明
C-VLAN および S-VLAN タグタイプのサポート	<p>内部または外部 VLAN タグは、C-VLAN および S-VLAN タイプにできます。VLAN タイプは、それぞれ VLAN 内包レジスタ (ETH_MACVIR) および 内部 VLAN 内包レジスタ (ETH_MACIVIR) の CSVL ビットを通じて指定されます。</p> <p>Ethernet ペリフェラルは、外部および内部 VLAN タグのあらゆるシーケンスの処理をサポートします。ただし、C-VLAN S-VLAN シーケンスはサポートされません。</p> <p>MACは、アプリケーションによって提供されたパケットが有効なVLANタグタイプのシーケンスを有するか、挿入または置換操作が無効なVLANタグタイプのシーケンスであるかをチェックしません。そのため、アプリケーションは VLAN タグタイプの正しいシーケンスを提供し、送信パケットで VLAN タグタイプのシーケンスが正しくなるように MAC をプログラムする必要があります。アプリケーションは次の内容を確認する必要があります。</p> <ul style="list-style-type: none"> – 外部 C-VLAN タグ挿入が有効である場合、内部タグは S-VLAN にしてはなりません。 – 内部 S-VLAN タグ挿入が有効である場合、外部タグは C-VLAN にしてはなりません。 – 外部タグを C-VLAN に書き換える必要がある場合、内部タグは S-VLAN にしてはなりません。 – 内部タグを S-VLAN に書き換える必要がある場合、外部タグは C-VLAN にしてはなりません。
VLAN タグの削除	<p>VLAN タグの削除は、それぞれ VLAN 内包レジスタ (ETH_MACVIR) または 内部 VLAN 内包レジスタ (ETH_MACIVIR) の VLC フィールドを通じて外部または内部タグに対して有効にできます。VLAN 削除が有効である場合、MAC は対応する位置にあるタグを削除します。パケットに 1 つだけタグがある場合は、外部タグとみなされます。内部タグ削除が有効で、パケットに 1 つだけタグがある場合は、MAC はタグを削除しません。</p>
VLAN タグの挿入または書き換え	<p>VLAN タグの挿入または書き換えは、それぞれ VLAN 内包レジスタ (ETH_MACVIR) または 内部 VLAN 内包レジスタ (ETH_MACIVIR) の VLC フィールドを通じて外部または内部タグに対して有効にできます。VLAN タグの挿入または書き換えが有効である場合、前のレジスタの VLTi ビットを使用して、VLAN タグをそのレジスタまたは制御ワードから取得すべきかを決定します。</p>

受信パス

表 621 : Rx パスでのダブル VLAN 処理 には、受信側で MAC によってサポートされる機能および VLAN タグレジスタ (ETH_MACVTR) で対応するビットを記載しています。

表 621. Rx パスでのダブル VLAN 処理

機能	説明
外部または内部 VLAN タグベースフィルタ	MAC は、ERIVLT ビットを通じて外部または内部 VLAN タグに基づいてパケットをフィルタできます。
C-VLAN または S-VLAN タグベースフィルタ	MAC は、ERSVLM ビットに基づく C-VLAN または S-VLAN タイプに基づいてパケットをフィルタできます。
外部または内部 VLAN タグストリッピング	MAC は、EVLS および EIVLS ビットに基づく受信フレームから外部および内部 VLAN タグを削除できます。
Rx ステータスでの 16 ビット外部または内部 VLAN タグおよびタイプ	MAC は、それぞれ EVLRXS および EIVLRXS ビットに基づく Rx ステータスで、16 ビット外部および内部 VLAN タグおよびタイプを提供できます。
外部 VLAN タグタイプのチェックの無効化またはスキップ	MAC は、DOVLTC ビットに基づく C-VLAN または S-VLAN に合わせるために、外部 VLAN タグタイプのチェックを無効化またはスキップできます。

57.5.2 ソースアドレスおよび VLAN の挿入、書換え、または削除

ソースアドレスの挿入または書き換え

ソフトウェアは、SA (ソースアドレス) の挿入または書き換え機能を使用して、MAC に Tx パケットに対して次の操作を行うように命令できます。

- SA フィールドに MAC アドレスレジスタの内容を挿入
- SA フィールドの内容を MAC アドレスレジスタの内容に書き換え

SA の挿入が有効である場合、アプリケーションは MAC に送信されるパケットに SA フィールドがないことを確認する必要があります。MAC は、送信パケットに SA フィールドが存在するかチェックせず、SA フィールドに MAC アドレスレジスタの内容を挿入します。同様に、SA の書き換えが有効である場合、アプリケーションは MAC に送信されるパケットに SA フィールドがないことを確認する必要があります。MAC は、送信パケットのデスティネーションアドレスフィールドに続く 6 バイトを、MAC アドレスレジスタの内容に書き換えます。

SA の挿入または書き換え機能は、すべての送信パケットまたは選択パケットに対して有効にできます。

- すべてのパケットに対する SA の挿入または書き換えの有効化
すべてのパケットに対してこの機能を有効にするには、[動作モード設定レジスタ \(ETH_MACCCR\)](#) の SARC フィールドをプログラムします。
- 選択パケットに対する SA の挿入または書き換えの有効化
選択パケットに対してこの機能を有効にするには、パケットの最初の送信ディスクリプタで次を使用して SA 挿入制御フィールド (送信ディスクリプタワード 3/TDES3 のビット [25:23]、[セクション 57.10.3 : 送信ディスクリプタ](#)を参照) をプログラムします。TDES3 のビット 25 がセットされた場合、SA 挿入制御フィールドは MAC アドレス 1 レジスタによる挿入または書き換えを示します。TDES3 のビット 25 がリセットされた場合、MAC アドレス 0 レジスタによる挿入または書き換えを示します。

MAC アドレス 1 レジスタが有効ではない場合、MAC アドレス 0 レジスタは SA 挿入制御フィールドの最上位ビットの値の挿入または書き換えに使用されます。

VLAN の挿入、書き換え、または削除

ソフトウェアは、VLAN の挿入、書き換え、または削除機能を使用して、MAC に Tx パケットに対して次の操作を行うように命令できます。

- VLAN タイプおよび VLAN タグフィールドの削除
- VLAN タイプおよび VLAN タグフィールドの挿入または書き換え

挿入または書き換えは、表 622 : VLTi ビットに基づいた VLAN の挿入または書き換えに示された VLAN 内包レジスタ (ETH_MACVIR) の VLTi ビットの設定に基づいて実行されます。

表 622. VLTi ビットに基づいた VLAN の挿入または書き換え

状態	説明
VLTi ビットがセットされている場合	MAC は、次の内容を挿入または書き換えます。 VLAN タイプフィールド (VLAN 内包レジスタ (ETH_MACVIR) の CSVL ビットによって示される C-VLAN または S-VLAN) パケットの送信コンテキストディスクリプタの VT フィールドがある VLAN タグフィールド
VLTi ビットがリセットされている場合	MAC は、次の内容を挿入または書き換えます。 VLAN タイプフィールド (VLAN 内包レジスタ (ETH_MACVIR) の CSVL ビットによって示される C-VLAN または S-VLAN) VLAN 内包レジスタ (ETH_MACVIR) の VLT フィールドがある VLAN タグフィールド

VLAN の書き換えまたは削除が有効である場合、MAC は送信パケットの DA および SA フィールドの後に VLAN タイプフィールド (0x8100 または 0x88A8) が存在しているかどうかチェックします。VLAN タイプフィールドが DA および SA フィールドに続く 2 バイトで検出されない場合、書き換えまたは削除の操作は発生しません。ただし、VLAN の挿入が有効である場合、MAC は送信パケットでの VLAN タイプフィールドの存在をチェックせず、ただ VLAN タイプと VLAN タグフィールドを挿入します。

すべての Tx パケットまたは選択パケットに対して VLAN の挿入、書き換えまたは削除を有効にできます。

- すべてのパケットに対してこの機能を有効にするには、VLAN 内包レジスタ (ETH_MACVIR) の VLP および VLP フィールドをプログラムします。
- 選択パケットに対してこの機能を有効にするには、TDES2 通常ディスクリプタの VTIR フィールドをプログラムします (表 655: TDES2 通常ディスクリプタ (読出しフォーマット) を参照)。
さらに、設定に応じて MAC がホストから制御入力を取得するために VLAN 内包レジスタ (ETH_MACVIR) (外部 VLAN の場合) および内部 VLAN 内包レジスタ (ETH_MACVIR) (内部 VLAN の場合) で VLP (VLAN 優先度制御) ビットをリセットする必要があります。

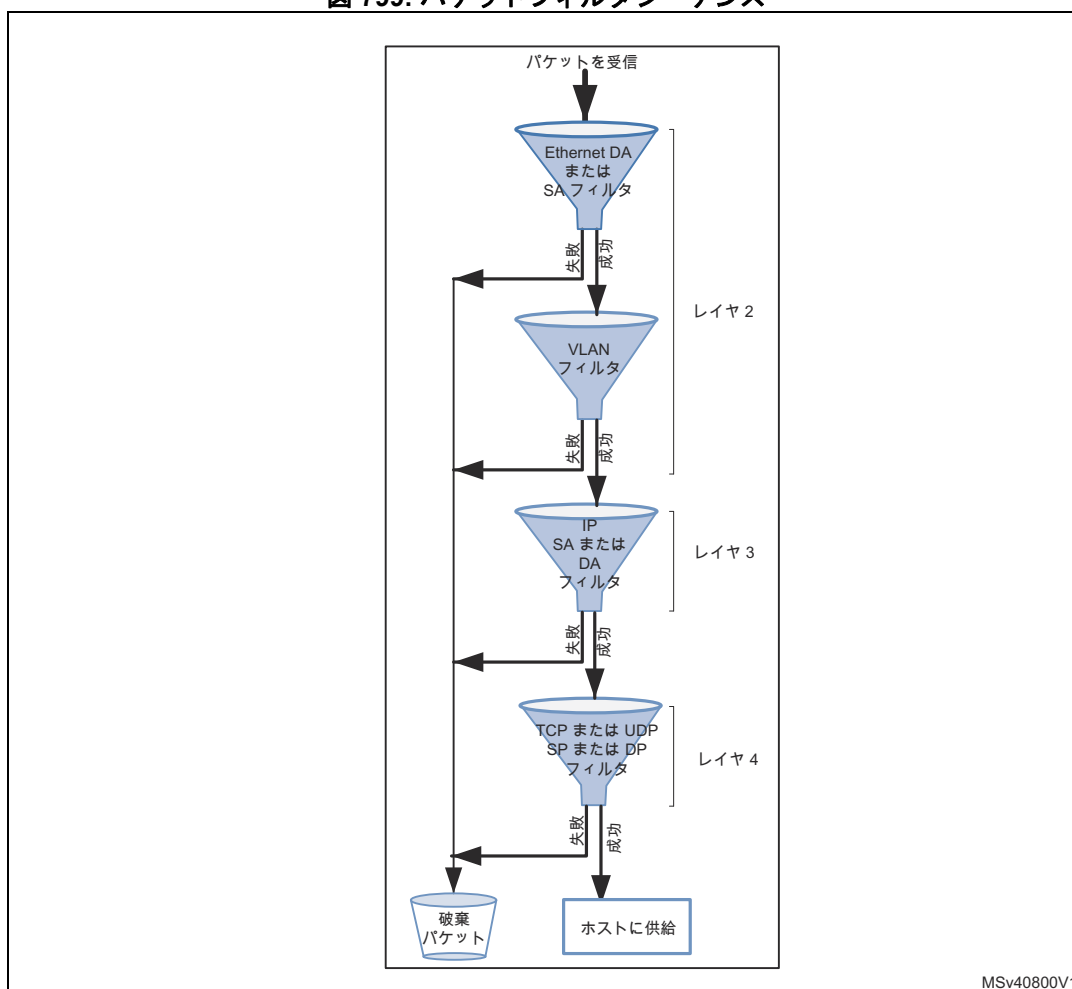
57.5.3 パケットフィルタ

MAC は Rx パケットの次のタイプのフィルタをサポートします。

- **MAC ソースアドレスまたはデスティネーションアドレスフィルタリング** : アドレスフィルタモジュール (AFM) が、各着信パケットのソースアドレスフィールドおよびデスティネーションアドレスフィールドをチェックします。
- **VLAN フィルタ** : MAC が VLAN タグベースおよび VLAN ハッシュフィルタをサポートします。
- **レイヤ 3 およびレイヤ 4 フィルタ** : レイヤ 3 フィルタは、IP ソースアドレスおよびデスティネーションアドレスのフィルタを参照します。レイヤ 4 フィルタは、ソースポートおよびデスティネーションポートのフィルタを参照します。

3 つのフィルタタイプは直列可能です。図 799 に、Rx パケットのフィルタシーケンスを示します。

図 799. パケットフィルタシーケンス



MSv40800V1

すべてのフィルタ (L2、VLAN、L3、L4) がアクティブになっている場合、図 799 に示すシーケンスが有効になります。有効になっていないレイヤフィルタがある場合、そのフィルタはバイパスされ、その後のフィルタが適用されます。いずれかのフィルタで障害が発生するパケットは破棄されます。ただし、破棄されたパケットは、レジスタ制御に基づいてホストに転送することができます。

たとえば、**パケットフィルタ制御レジスタ (ETH_MACPFR)** の RA ビットが 1 にセットされている場合、パケットステータスが特定のフィルタ障害のものを除いて、破棄されたすべてのパケットがホストに転送されます。RA ビットが 0 にクリアされると、VLAN フィルタおよびレイヤ 3~4 フィルタで障害が発生するパケットを破棄するかホストに転送する必要がある場合、**パケットフィルタ制御レジスタ (ETH_MACPFR)** の VTFE および IPFE ビットが制御されます。

MAC ソースアドレスまたはデスティネーションアドレスフィルタリング

MAC アドレスフィルタモジュールは、各着信パケットのソースアドレス (SA) フィールドおよびデスティネーションアドレス (DA) フィールドをチェックします。

ユニキャストデスティネーションアドレスフィルタ

MAC は、ユニキャストの完全なフィルタリングのために、4 つの MAC アドレスをサポートします。完全フィルタリングが選択された場合 (**パケットフィルタ制御レジスタ (ETH_MACPFR)** の HUC ビットがリセット)、MAC は、受信したユニキャストアドレスの 48 ビットすべてを、プログラムさ

れた MAC アドレスと比較して、一致しているかどうかを確認します。デフォルトの MacAddr0 は常に有効です。

MacAddr1~MacAddr3 のアドレスは、個別のイネーブルビットで選択されます。各バイトは、[MAC アドレス x 上位レジスタ \(ETH_MACAxHR\)](#) の対応するマスクバイト制御ビットをセットすることによって、対応する受信 DA バイトとの比較時にマスクできます。これは、DA のグループアドレスフィルタを可能にします。

ハッシュフィルタリングモード (HUC ビットがセットされているとき) では、MAC は、64 ビットのハッシュテーブルを使用して、ユニキャストアドレスの不完全フィルタリングを実行します。ハッシュフィルタリングの場合、MAC は、受信したデスティネーションアドレスの CRC の上位 6 ビットを使用して、ハッシュテーブルの内容をインデックスします。値 00000 は、選択されたレジスタのビット 0 を選択し、値 11111 は、ハッシュテーブルレジスタのビット 63 を選択します。対応するビット (6 ビット CRC によって示される) が 1 にセットされた場合、ユニキャストパケットはハッシュフィルタを通過したとみなされ、そうでない場合、パケットはハッシュフィルタ通過できなかったとみなされます。

マルチキャストデスティネーションアドレスフィルタ

すべてのマルチキャストパケットを通過する MAC をプログラムするには、[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) の PM ビットをセットします。PM ビットがリセットされた場合、MAC は、[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) の HMC ビットに基づいて、マルチキャストアドレスのフィルタリングを実行します。

完全フィルタリングモードでは、マルチキャストアドレスが、プログラムされた MAC デスティネーションアドレスレジスタと比較されます。グループアドレスフィルタリングもサポートされています。

ハッシュフィルタリングモードでは、MAC は、64 ビットのハッシュテーブルを使用して、不完全フィルタリングを実行します。MAC は、受信したマルチキャストアドレスの CRC の上位 6 ビットを使用して、ハッシュテーブルの内容をインデックスします。値 000000 は、選択されたレジスタのビット 0 を選択し、値 111111 は、ハッシュテーブルレジスタのビット 63 を選択します。対応するビットが 1 にセットされた場合は、マルチキャストパケットがハッシュフィルタを通過したとみなされます。そうでない場合は、パケットがハッシュフィルタを通過できなかったとみなされます。

ハッシュまたは完全アドレスフィルタ

DA がハッシュフィルタまたは完全フィルタに一致するときパケットを通過させるために DA フィルタを設定するには、[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) の HPF ビットおよび対応する HUC または HMC ビットをセットします。これは、ユニキャストパケットとマルチキャストパケットの両方に適用できます。HPF ビットがリセットされた場合は、1 つのフィルタ (ハッシュまたは完全) だけが受信パケットに適用されます。

ブロードキャストアドレスフィルタ

デフォルトでは、MAC は、ブロードキャストパケットをフィルタしません。すべてのブロードキャストパケットを拒否する MAC をプログラムするには、[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) の DBF ビットをセットします。

ユニキャストソースアドレスフィルタ

MAC は、受信パケットのソースアドレスフィールドに基づいて、完全フィルタリングを実行できます。デフォルトでは、MAC は、SA フィールドを、SA レジスタでプログラムされた値と比較します。[MACアドレス x 上位レジスタ \(ETH_MACAxHR\)](#) のビット 30 をセットすることによって、比較のために DA の代わりに SA を使用するように MAC アドレスレジスタを設定することができます。

MAC は SA でのグループフィルタリングもサポートします。アドレスの 1 つ以上のバイトをマスクすることで、アドレスのグループをフィルタできます。SAF ビットが[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) でセットされている場合、MAC は SA フィルタに失敗したパケットをドロップします。そうでない場合、SA フィルタの結果は、受信ステータスワード内のステータスビットとして

与えられます (表 624 を参照)。SAF ビットがセットされた場合、SA フィルタと DA フィルタの結果が AND (論理積) されて、パケットを転送する必要があるかどうかが決まります。つまり、いずれかのフィルタに失敗した場合、パケットはドロップされます。パケットが両方のフィルタを順に通過した場合のみ、パケットはアプリケーションに転送されます。

逆フィルタ

DA および SA フィルタでは、**パケットフィルタ制御レジスタ (ETH_MACPFR)** の DAIF および SAIF ビットをセットすることで、最終出力でフィルタ一致結果を反転させることができます。DAIF ビットは、ユニキャストとマルチキャストの両方の DA パケットに適用可能です。このモードでは、ユニキャストまたはマルチキャストデスティネーションアドレスフィルタの結果が反転されます。同様に、SAIF ビットがセットされると、ユニキャスト SA フィルタの結果が反転されます。

表 623 および表 624 に、受信パケットのタイプに基づく DA と SA のフィルタリングを示します。

注: **パケットフィルタ制御レジスタ (ETH_MACPFR)** の RA ビットがセットされている場合、すべてのパケットが、Rx ステータスのアドレスフィルタリングの正しい結果と一緒にシステムに転送されます。

表 623. デスティネーションアドレスフィルタリング

パケットタイプ	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA フィルタ動作
ブロードキャスト	1	X	X	X	X	X	X	成功
	0	X	X	X	X	X	0	成功
	0	X	X	X	X	X	1	失敗
ユニキャスト	1	X	X	X	X	X	X	すべてのパケットを通過。
	0	X	0	0	X	X	X	完全/グループフィルタ一致で通過。
	0	X	0	1	X	X	X	完全/グループフィルタ一致で失敗。
	0	0	1	0	X	X	X	ハッシュフィルタ一致で通過。
	0	0	1	1	X	X	X	ハッシュフィルタ一致で失敗。
	0	1	1	0	X	X	X	ハッシュまたは完全/グループフィルタ一致で通過。
	0	1	1	1	X	X	X	ハッシュまたは完全/グループフィルタ一致で失敗。
マルチキャスト	1	X	X	X	X	X	X	すべてのパケットを通過。
	X	X	X	X	X	1	X	すべてのパケットを通過。
	0	X	X	0	0	0	X	完全/グループフィルタ一致で通過、PCF = 0x の場合はポーズパケットをドロップ。
	0	0	X	0	1	0	X	ハッシュフィルタ一致で通過、PCF = 0x の場合はポーズパケットをドロップ。
	0	1	X	0	1	0	X	ハッシュまたは完全/グループフィルタ一致で通過、PCF = 0x の場合はポーズパケットをドロップ。
マルチキャスト	0	X	X	1	0	0	X	完全/グループフィルタ一致で失敗、PCF = 0x の場合はポーズパケットをドロップ。
	0	0	X	1	1	0	X	ハッシュフィルタ一致で失敗、PCF = 0x の場合はポーズパケットをドロップ。
	0	1	X	1	1	0	X	ハッシュまたは完全/グループフィルタ一致で失敗、PCF = 0x の場合はポーズパケットをドロップ。

表 624. ソースアドレスフィルタリング

パケットタイプ	PR	SAIF	SAF	SA フィルタ動作
ユニキャスト	1	X	X	すべてのパケットを通過。
	0	0	0	完全またはグループフィルタ一致で通過ステータスだが、失敗したパケットをドロップしない。
	0	1	0	完全またはグループフィルタ一致で失敗ステータスだが、パケットをドロップしない。
	0	0	1	完全またはグループフィルタ一致で通過、失敗したパケットをドロップする。
	0	1	1	完全またはグループフィルタ一致で失敗、失敗したパケットをドロップする。

VLAN フィルタ

MAC は、完全およびハッシュ VLAN フィルタをサポートします。詳細なプログラミング手順については、[セクション 57.9.14 : 受信時の VLAN フィルタに関するプログラミングガイドライン](#)を参照してください。

VLAN タグ完全フィルタ

VLAN タグ完全フィルタでは、MAC は受信パケットの VLAN タグを比較して、VLAN パケットステータスをアプリケーションに提供します。プログラムされたモードに基づいて、MAC は受信した VLAN タグの下位 12 ビットまたは全 16 ビットを比較して完全一致を判断します。

VLAN タグ完全フィルタが有効である場合、MAC は VLAN タグ一致ステータスとともに VLAN タグ付きパケットを転送し、一致しない VLAN パケットをドロップします。[VLAN タグレジスタ \(ETH_MACVTR\)](#) の VTIM ビットをセットすることで、VLAN パケットの反転一致も有効にできます。さらに、[VLAN タグレジスタ \(ETH_MACVTR\)](#) の ESVL ビットをセットすることで、デフォルトの C-VLAN タグ付きパケットと合わせて S-VLAN タグ付きパケットの一致を有効にできます。VLAN パケットステータスビット (RDES0 のビット 10) は、一致したパケットの VLAN タグ一致ステータスを示します。

注 : ソースまたはデスティネーションアドレス (有効時) には、VLAN タグフィルタにわたって優先度があります。これは、ソースまたはデスティネーションアドレスフィルタに失敗したパケットが VLAN タグフィルタの結果に関係なくドロップされることを示します。

VLAN タグハッシュフィルタ

VLAN タグに基づくグループアドレスのフィルタリングには、16 ビットの VLAN ハッシュテーブルが使用されます。VLAN タグのハッシュフィルタ機能を有効化するには、[VLAN タグレジスタ \(ETH_MACVTR\)](#) の VTHM (VLAN タグハッシュテーブル一致の有効化) ビットを使用します。VTHM ビットがセットされている場合、VLAN タグの CRC-32 の最上位 4 ビットを使用して、VLAN ハッシュテーブルレジスタの内容をインデックスします。インデックスに対応する VLAN ハッシュテーブルレジスタでの値 1 は、パケットの VLAN タグが一致し、パケットを転送しなければならないことを示します。値 0 は、VLAN タグ付きパケットをドロップしなければならないことを示します。

注 : VLAN タグの 16 ビットまたは 12 ビットは、ETH_MACVTR レジスタの ETV ビットに基づいた CRC-32 計算に考慮されます。

ETV ビットがリセットされている場合、VLAN タグの CRC-32 の最上位 4 ビットが反転して使用され、[VLAN ハッシュテーブルレジスタ \(ETH_MACVHTR\)](#) の内容をインデックスします。

ETV ビットがセットされている場合、VLAN タグの CRC-32 の最上位 4 ビットは直接使用され、[VLAN タグレジスタ \(ETH_MACVTR\)](#) の内容をインデックスします。

MAC は、VLAN パケットの反転一致もサポートします。反転一致モードでは、パケットの VLAN タグが完全またはハッシュフィルタに一致するとき、パケットをドロップする必要があります。VLAN 完全および VLAN ハッシュ一致が有効である場合、VLAN ハッシュまたは VLAN 完全フィルタのいずれ

れかに一致すれば、パケットは一致したものとみなされます。反転一致がセットされると、完全およびハッシュフィルタが不一致を示した場合のみパケットが転送されます。

表 625 表 625 には、VLAN 一致について異なる可能な内容と最終的な LAN 一致ステータスを示しています。パケットフィルタ制御レジスタ (ETH_MACPFR) の RA ビットがセットされている場合、すべてのパケットが受信され、VLAN 一致ステータスが RDES2 通常ディスクリプタ (書き戻しフォーマット) の VF ビットに示されます。パケットフィルタ制御レジスタ (ETH_MACPFR) で RA ビットがセットされておらず、VTFE ビットがセットされているとき、VLAN 一致ステータスが失敗の場合、パケットはドロップされます。表 625 で、値 X はこの列にどの値でも入ることを示します。

VLAN タグレジスタ (ETH_MACVTR) の VL フィールドで VLAN VID が 0 にプログラムされているとき、すべての VLAN タグ付きパケットは完全一致とみなされますが、VLAN ハッシュ一致のステータスは VLAN タグレジスタ (ETH_MACVTR) の VTHM および VTIM ビットに依存します。

表 625. VLAN 一致ステータス⁽¹⁾

VID	VLAN 完全フィルタ一致の結果	VTHM ビット	VLAN ハッシュフィルタ一致の結果	VTIM ビット	最終的な VLAN 一致ステータス
VID = 0	成功	0	X	X	成功
	成功	1	X	0	成功
	成功	1	失敗	1	成功
	成功	1	成功	1	失敗
VID != 0	成功	X	X	0	成功
	失敗	0	X	0	失敗
	失敗	1	失敗	0	失敗
	失敗	1	成功	0	成功
	失敗	0	X	1	成功
	成功	X	X	1	失敗
	失敗	1	成功	1	失敗
	失敗	1	失敗	1	成功

1. この表では、'X' は任意の値を示します。

レイヤ 3 およびレイヤ 4 フィルタ

MAC は、レイヤ 3 およびレイヤ 4 ベースのパケットフィルタをサポートします。レイヤ 3 フィルタは IPv4 または IPv6 パケットでの IP ソースまたはデスティネーションアドレスフィルタを参照し、レイヤ 4 フィルタは TCP または UDP でのソースまたはデスティネーションのポート番号フィルタを参照します。

レイヤ 3 およびレイヤ 4 パケットフィルタ機能は、受信側で IPC フルチェックサムオフロードエンジンを自動的に有効にします。レイヤ 3 およびレイヤ 4 フィルタの操作には、動作モード設定レジスタ (ETH_MACCCR) の IPC ビットをセットして、Rx チェックサムオフロードエンジンを有効にする必要があります。

レイヤ 3 およびレイヤ 4 フィルタが有効である場合、パケットは次のようにフィルタされます。

- 一致するパケット

MAC は、すべての有効化されたフィールドに一致するパケットをステータスとともにアプリケーションに転送します。動作モード設定レジスタ (ETH_MACCR) の IPC ビットがセットされ、次のいずれかの条件が当てはまる場合のみ、MAC は一致フィールドステータスを付与します。

- すべての有効化されたレイヤ 3 およびレイヤ 4 フィールドが一致する。
- 少なくとも 1 つの有効化されたフィールドが一致し、その他のフィールドはバイパスされるか無効化される。

複数のレイヤ 3 およびレイヤ 4 フィルタが有効である場合、どのフィルター一致でも一致とみなされます。複数のフィルタが一致する場合、MAC はフィルタ 0 が最小フィルタで、フィルタ 3 が最高フィルタであれば、最小フィルタのステータスを提供します。たとえば、フィルタ 0 とフィルタ 1 が一致する場合、MAC はフィルタ 0 に対応するステータスを付与します。

注 : ソースまたはデスティネーションアドレスおよび VLAN タグフィルタ (有効時) には、レイヤ 3 およびレイヤ 4 フィルタにわたって優先度があります。これは、ソースまたはデスティネーションアドレスまたは VLAN タグフィルタに失敗したパケットがレイヤ 3 およびレイヤ 4 フィルタの結果に関係なくドロップされることを示します。

- 一致しないパケット

MAC は、どの有効化されたフィールドにも一致しないパケットをドロップします。反転一致機能を使用して、IP フィールドにまたがる特定の TCP または UDP でパケットをブロックまたはドロップでき、他のすべてのパケットを転送できます。アボートされたパケットや部分的なパケットは MTL Rx FIFO でドロップされます。Rx FIFO が閾値 (カットスルー) モードで動作していて、閾値が小さな値にプログラムされている場合、障害となったレイヤ 3 およびレイヤ 4 フィルタの結果が使用可能になる前にアプリケーションへのこのようなパケット転送が開始され、アプリケーションが適切なアボートステータスを持つ部分的なパケットを受信することがあります。

- 非 TCP または非 UDP の IP パケット

デフォルトでは、すべての非 TCP または非 UDP の IP パケットは、レイヤ 3 およびレイヤ 4 フィルタからバイパスされます。オプションで MAC をプログラムして、IP パケットにまたがる非 TCP または非 UDP をすべてドロップすることができます。

レイヤ 3 フィルタ

MAC は、IP ソースアドレスおよびデスティネーションアドレスの完全一致または反転一致をサポートします。さらに、完全な IP アドレスを一致させたり、下位ビット一致をマスク、つまり指定した下位マスクビットを除いたアドレスのすべてのビットを比較したりできます。

IPv6 パケットフィルタでは、レジスタセットの最後の 4 つのデータレジスタを有効にして、128 ビットの IP ソースアドレスや IP デスティネーションアドレスを含めることができます。IP ソースアドレスやデスティネーションアドレスは、IPv6 仕様で定義された順でプログラムしなければなりません。つまり、受信パケットの IP ソースアドレスやデスティネーションアドレスの最初のバイトはレジスタの上位バイトであり、それに続くレジスタは同じ順番に従います。

IPv4 パケットフィルタでは、レジスタセットの 2 番目と 3 番目のデータレジスタを有効にして、32 ビットの IP ソースアドレスや IP デスティネーションアドレスを含めることができます。残りの 2 つのデータレジスタは予約済みとなります。IP ソースアドレスやデスティネーションアドレスは、IPv4 仕様で定義された順でプログラムしなければなりません。つまり、受信パケットの IP ソースアドレスやデスティネーションアドレスの最初のバイトは各レジスタの上位バイトとなります。

レイヤ 4 フィルタ

MAC は、TCP または UDP のソースおよびデスティネーションのポート番号の完全一致または反転一致をサポートします。ただし、一度に 1 つのタイプ (TCP または UDP) のみプログラムできます。最初のデータレジスタには、TCP または UDP の 16 ビットのソースおよびデスティネーションの

ポート番号が含まれます。つまり、ソースのポート番号の下位 16 ビットとデスティネーションのポート番号の上位 16 ビットです。

TCP または UDP のソースおよびデスティネーションのポート番号は、TCP や UDP の仕様で定義された順でプログラムしなければなりません。つまり、受信パケットの TCP または UDP のソースポート番号やデスティネーションポート番号の最初のバイトはレジスタの上位バイトとなります。

レイヤ 3 およびレイヤ 4 フィルタのレジスタセット

MAC は、レイヤ 3 およびレイヤ 4 ベースのパケットフィルタ向けに 2 セットのレジスタを実装しています。レジスタセットには、**L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R)** などパケットフィルタを制御する制御レジスタがあります。さらに、次のような一致するレイヤ 3 およびレイヤ 4 フィールドをプログラムする 5 つのアドレスレジスタがあります。

- **レイヤ 4 アドレスフィルタ 0 レジスタ (ETH_MACL4A0R)**
- **レイヤ 3 アドレス 0 フィルタ 0 レジスタ (ETH_MACL3A00R)**
- **レイヤ 3 アドレス 1 フィルタ 0 レジスタ (ETH_MACL3A10R)**
- **レイヤ 3 アドレス 2 フィルタ 0 レジスタ (ETH_MACL3A20R)**
- **レイヤ 3 アドレス 3 フィルタ 0 レジスタ (ETH_MACL3A30R)**

2 つ目および独立したレジスタのセットは次のとおりです。つまり、**L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R)**、**レイヤ 4 アドレスフィルタ 1 レジスタ (ETH_MACL4A1R)**、**レイヤ 3 アドレス 0 フィルタ 1 レジスタ (ETH_MACL3A01R)**、**レイヤ 3 アドレス 1 フィルタ 1 レジスタ (ETH_MACL3A11R)**、**レイヤ 3 アドレス 2 フィルタ 1 レジスタ (ETH_MACL3A21R)**、および**レイヤ 3 アドレス 3 フィルタ 1 レジスタ (ETH_MACL3A31R)** です。

57.5.4 IEEE 1588 タイムスタンプサポート

IEEE 1588 標準は、ネットワーク通信、ローカルコンピューティング、分散オブジェクトなどのテクノロジーで実装される計測および制御システムでの高精度なクロック同期を可能にする高精度時間プロトコル (PTP) を定義しています。この PTP は、Ethernet をはじめとして（ただし、これに限られず）、マルチキャストメッセージングをサポートするローカルエリアネットワークによって通信するシステムに適用されます。このプロトコルは、固有精度、解像度、および安定性が異なるクロックを含む異種混合システムを同期することができます。このプロトコルは、最小限のネットワークおよびローカルクロック計算リソースを使用して、サブマイクロ秒範囲のシステム全体の正確な同期をサポートします。

この章には、次のセクションが含まれます。

- **IEEE 1588 タイムスタンプサポート**
- **IEEE 1588 システム時間ソース**
- **IEEE 1588 補助スナップショット**
- **柔軟性の高い pulse-per-second 出力**
- **PTP タイムスタンプオフロード機能**
- **ワンステップタイムスタンプ**

IEEE 1588 タイムスタンプサポート

Ethernet ペリフェラルは、IEEE 1588-2002 (バージョン 1) および IEEE 1588-2008 (バージョン 2) をサポートします。IEEE 1588-2002 は、UDP/IP 経由で転送される PTP をサポートします。IEEE 1588 2008 は、Ethernet 経由で転送される PTP をサポートします。ペリフェラルは、両方の標準に対してプログラム可能なサポートを提供します。以下の機能をサポートしています。

- 両方のタイムスタンプフォーマットのサポート
- すべてのパケットまたは PTP タイプパケットのみのオプションのスナップショット
- イベントメッセージのみのオプションのスナップショット
- クロックタイプに基づいたオプションのスナップショット : 通常、境界、エンドツーエンドトランスペアレント、ピアツーピアトランスペアレント
- 通常および境界クロックにマスタまたはスレーブとして機能するノードのオプションの選択
- Ethernet 経由で直接送信されたパケットでの PTP、メッセージタイプ、バージョン、PTP ペイロードの識別およびステータスの送信
- デジタルまたはバイナリ形式でのオプションの測定のサブセカンドの時間

クロックタイプ

MAC は、IEEE 1588-2008 仕様で定義された次のクロックタイプをサポートします。

- 通常クロック

ドメインの通常クロックは、プロトコルのシングルコピーをサポートします。シングル PTP 状態とシングル物理ポートを備えています。標準的な工業自動化アプリケーションでは、通常クロックがセンサやアクチュエータなどのアプリケーションデバイスと関連付けられます。電気通信アプリケーションでは、通常クロックをタイミング境界設定デバイスと関連付けることができます。

通常クロックは、グランドマスタクロックまたはスレーブクロックとすることができます。以下の機能をサポートしています。

- PTP メッセージの送受信。タイムスタンプスナップショットは、[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) に示すように制御できます。
- タイムスタンプの値などデータセットのメンテナンス。

下の表に、マスタノードおよびスレーブノードの受信側でタイムスタンプスナップショットを取得できるメッセージを示します。

表 626. 通常クロック : スナップショット用 PTP メッセージ

マスタ	スレーブ
Delay_Req	SYNC

通常クロックでは、バージョン 1 またはバージョン 2 のいずれかの PTP メッセージタイプのスナップショットを取得できます。両方の PTP メッセージタイプのスナップショットは取得できません。[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) で TSVER2ENA ビットをセットしてスナップショットモードを選択することで、スナップショットを取得できます。

- 境界クロック

通常、境界クロックにはネットワークと通信する複数の物理ポートがあります。同期、マスタスレーブ階層、シグナリングに関連するメッセージは、境界クロックのプロトコルエンジンで終了します。これらのメッセージは転送されません。MAC によって示された PTP メッセージタイプステータスは、メッセージのタイプを識別して、適切な操作を行うのに役立ちます。

境界クロックは、次の特徴を除いて通常クロックに類似しています。

- クロックのデータセットは、境界クロックではすべてのポートに共通です。
- ローカルクロックは、境界クロックではすべてのポートに共通です。

- エンドツーエンドトランスペアレントクロック

エンドツーエンドトランスペアレントクロックは、スレーブクロックとマスタクロックでのエンドツーエンド遅延測定メカニズムをサポートします。エンドツーエンドトランスペアレントクロックは、通常のブリッジ、ルータ、リピータのように、すべてのメッセージを転送します。PTP パケットの滞留時間は、入力ポートから出力ポートまで PTP パケットがかかる時間です。

エンドツーエンドトランスペアレントクロック内の SYNC パケットの滞留時間は、関連付けられた Follow_Up PTP パケットの補正フィールドで更新されてから送信されます。同様に、エンドツーエンドトランスペアレントクロック内の Delay_Req パケットの滞留時間は、関連付けられた Delay_Resp PTP パケットの補正フィールドで更新されてから送信されます。そのため、表 627 で言及されるメッセージに対してのみ、入力ポートと出力ポートの両方でスナップショットを取得する必要があります。スナップショットは、タイムスタンプ制御レジスタ (ETH_MACTSCR) で SNAPTYPSEL ビットを 10 にセットすると取得できます。

表 627. エンドツーエンドトランスペアレントクロック : スナップショット用 PTP メッセージ

PTP メッセージ
SYNC
Delay_Req

- ピア・ツー・ピアトランスペアレントクロック

ピアツーピアトランスペアレントクロックでは、リンク遅延の計算はリンクピアとの Pdelay_Req、Pdelay_Resp、および Pdelay_Resp_Follow_Up メッセージの交換に基づきます。ピアツーピアトランスペアレントクロックは、PTP タイミングメッセージを補正して処理する方法で、エンドツーエンドトランスペアレントクロックとは異なります。その他すべての面で、エンドツーエンドトランスペアレントクロックと同一です。

Pdelay_Req および関連する Pdelay_Resp パケットの滞留時間が、関連付けられた Pdelay_Resp_Followup パケットの補正フィールドに追加され挿入されます。そのため、Pdelay に関連するイベントメッセージに対するスナップショット取得のサポートが、表 635 に示されるとおり追加されます。

表 628. ピアツーピアトランスペアレントクロック : スナップショット用 PTP メッセージ

PTP メッセージ
SYNC
Pdelay_Req
Pdelay_Resp

スナップショットは、タイムスタンプ制御レジスタ (ETH_MACTSCR) で SNAPTYPSEL ビットを 11 にセットすると取得できます。

遅延リクエストレスポンスメカニズム

システムまたはネットワークは、タイミングおよびクロック情報を分配するために、マスタノードとスレーブノードに分類されます。図 800 では、PTP メッセージを交換することによってスレーブノードをマスタノードに同期するために PTP が使用するプロセスを示します。

図 800. ネットワーク時間の同期

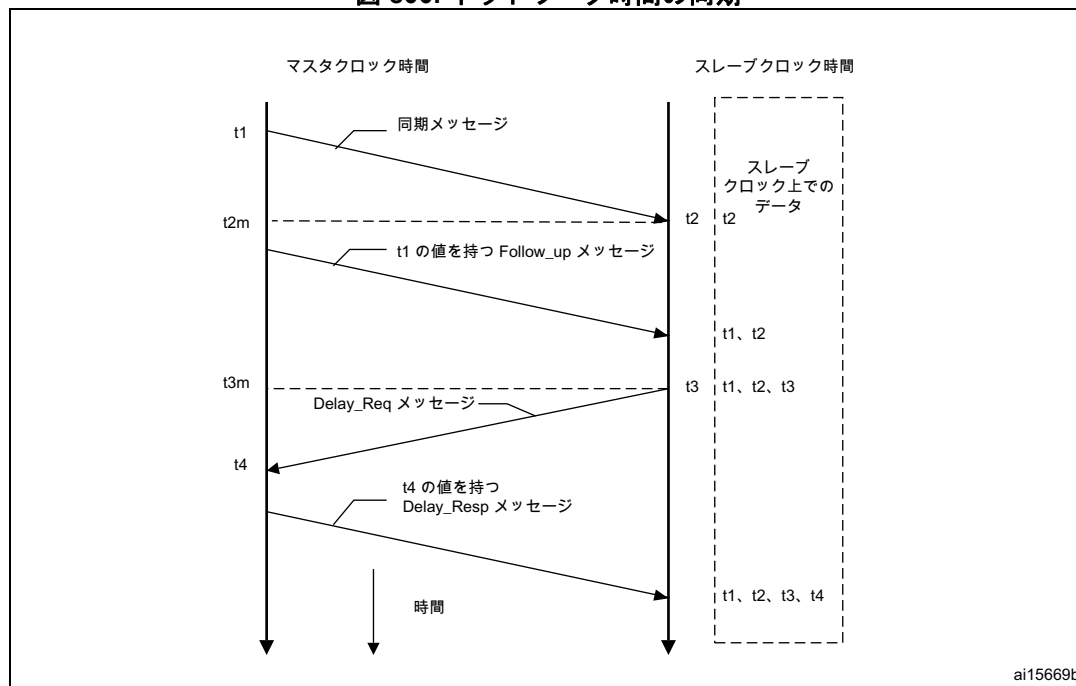


図 800 に示されるとおり、PTP は次のプロセスを使用します。

1. マスタは、PTP 同期メッセージをすべてのノードにブロードキャストします。同期メッセージは、マスタの基準時間情報を含んでいます。このメッセージは、 t_1 でマスタのシステムから送信されます。この時間は、MII で Ethernet ポートにおいてキャプチャされます。
2. スレーブは、SYNC メッセージを受信し、その基準タイミングを使用して、正確な時間 t_2 もキャプチャします。
3. マスタは、後で使用する t_1 情報を含んだ Follow_up メッセージをスレーブに送信します。
4. スレーブは、マスタに Delay_Req メッセージを送信して、このパッケージが MII インタフェースから送信された正確な時間 t_3 を知らせます。
5. マスタは、このメッセージを受信して、メッセージがシステムに着信した正確な時間 t_4 をキャプチャします。
6. マスタは、Delay_Resp メッセージで t_4 情報をスレーブに送信します。
7. スレーブは、 t_1 、 t_2 、 t_3 、および t_4 の 4 つの値を使用して、マスタのローカル基準タイミングをマスタの基準タイミングに同期します。

PTP 実装のほとんどは、UDP レイヤの上のソフトウェアで行われます。ただし、特定の PTP パッケージが MII インタフェースの Ethernet ポートで送受信された正確な時間をキャプチャするには、ハードウェアサポートが必要です。PTP の正しく高精度な実装のためには、このタイミング情報がキャプチャされて、ソフトウェアに返されなければなりません。

ピア・ツー・ピアトランスペアレントクロック (P2P TC) メッセージのサポート

IEEE 1588-2008 標準は、同期、遅延リクエスト、フォローアップ、遅延レスポンスのメッセージに加えて、ピアツーピア PTP (Pdelay) メッセージをサポートします。図 801 では、ピアツーピアパス補正をサポートするクロック単位での伝搬遅延を計算する方法を示しています。

図 801. ピアツーピアパス補正をサポートするクロック単位での伝搬遅延の計算

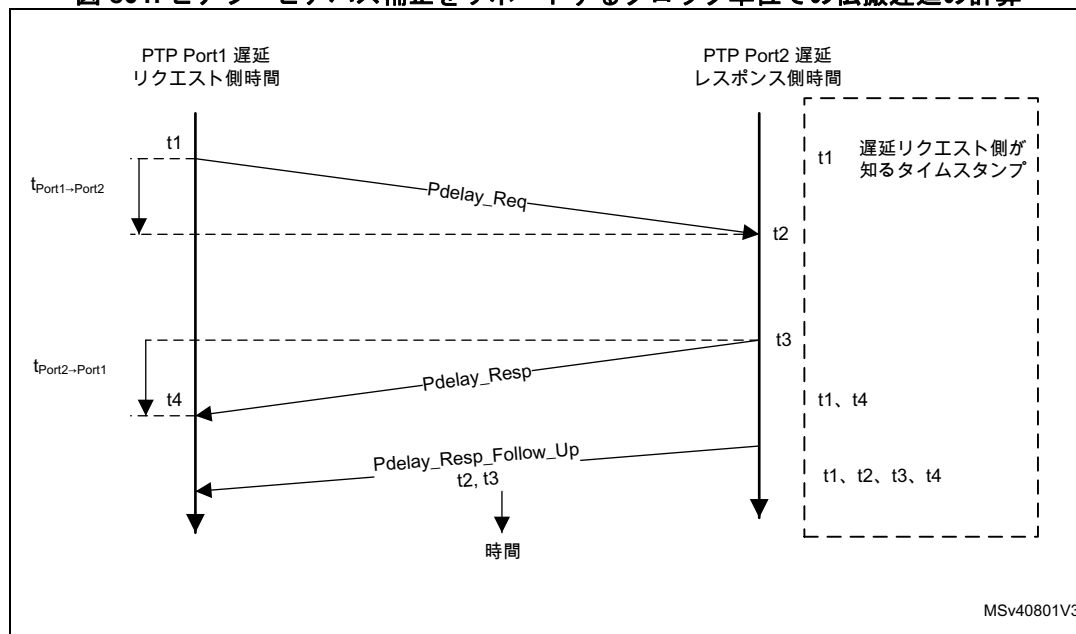


図 801 に示されるとおり、伝搬遅延は次のように計算されます。

1. ポート 1 は Pdelay_Req メッセージを発行し、Pdelay_Req メッセージにタイムスタンプ (t_1) を生成します。
2. ポート 2 は Pdelay_Req メッセージを受信し、このメッセージにタイムスタンプ (t_2) を生成します。
3. ポート 2 は Pdelay_Resp メッセージを返し、このメッセージにタイムスタンプ (t_3) を生成します。
2 つのポート間の周波数オフセットによって生じる誤差を最低限にするために、ポート 2 は Pdelay_Req メッセージの受信後、可及的速やかに Pdelay_Resp メッセージを返します。ポート 2 は、次のいずれかを返します。
 - Pdelay_Resp メッセージでのタイムスタンプ t_2 と t_3 の差
 - Pdelay_Resp_Follow_Up メッセージでのタイムスタンプ t_2 と t_3 の差
 - Pdelay_Resp および Pdelay_Resp_Follow_Up メッセージでのそれぞれのタイムスタンプ t_2 と t_3
4. ポート 1 は Pdelay_Resp メッセージを受信すると、タイムスタンプ (t_4) を生成します。
5. ポート 1 は 4 つすべてのタイムスタンプを使用して、平均リンク遅延を計算します。

タイムスタンプ補正

IEEE 1588 仕様に従って、メッセージタイムスタンプポイント（開始フレームデリミタオクテットの直後のオクテットの最初のビットの先端）がノードとネットワークの境界を越えるときに、タイムスタンプをキャプチャしなければなりません。

MAC は、ノードとネットワークの実際の境界から離れた内部ポイントでタイムスタンプを取得するため、キャプチャされたこのタイムスタンプは、入力/出力パス遅延（PHY レイヤの遅延を含む）のために補正/更新されます。キャプチャポイントにおいてクロック（MII Tx、Rx クロック）が、時刻の生成に使用される PTP クロック（clk_ptp_ref_i）とは異なるために生じる不正確さ/エラーのため、さらなる補正が実行されます。これによって生じる CDC（クロックドメイン横断）回路で、MII および PTP クロックのクロック周期依存のエラーが増加します。

入力補正

受信側では、内部スナップショットポイントでキャプチャされたタイムスタンプは、ポート境界でそのパケット SFD ビットが受信される時刻に比べて、遅延します（時刻が後になる）。そのため、キャプチャされたタイムスタンプから入力遅延と CDC サンプリングでのエラーを差し引く必要があります。この補正値は、ソフトウェアによって特定/算出し、[タイムスタンプ入力補正ナノ秒レジスタ \(ETH_MACTSICNR\)](#) に書き込む必要があります。

補正値は次の 3 つの要素からなります。

1. 境界点とコアの入力との間の PHY レイヤの外部遅延
IEEE 802.3 Clause 45 MMD レジスタに準拠した PHY は、最大入力遅延と最小入力遅延を示すレジスタを備えています。ソフトウェアでこれらのレジスタを読み出し、PHY の平均入力遅延を特定することができます。または（PHY でこれらのレジスタがサポートされていない場合）、入力遅延は PHY データシートまたはタイミング特性から特定する必要があります。
2. コアの入力から内部キャプチャポイントへの内部遅延
遅延は、[表 629](#) に示すように、アクティブな PHY インタフェース（MII または RMII など）と動作速度によって異なります。
3. CDC の同期
CDC 同期エラーは PTP クロック（clk_ptp_ref_i）のクロック周期の約 2 倍です。

これらの 3 つの要素から決まる値は、ソフトウェアによって追加される必要があり、[タイムスタンプ入力補正ナノ秒レジスタ \(ETH_MACTSICNR\)](#) の TSIC フィールドに書き込まれる必要があります。

注： レジスタに書き込まれる値は、キャプチャされたタイムスタンプから差し引く必要があるため、マイナス（2 の補数）でなければなりません。MAC レシーバは、キャプチャされたタイムスタンプにこのレジスタの値を付加し、生成された値を受信パケットのタイムスタンプとして与えます。

[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の TSCTRLSSR ビットがセットされている場合、キャプチャされたタイムスタンプのナノ秒フィールドは 1 ns 単位の 10 進数形式となります。そのため、TSIC のビット 31 は 1（負の値）にセットする必要があり、ビット [30:0] は「 $10^9 - \text{total ingress_correction_value}[\text{ナノ秒の部分}]$ 」によりバイナリで表現してプログラムする必要があります。たとえば、必要な補正値が -5 ns の場合、値は 0xBB9A C9FB となります。

[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の TSCTRLSSR ビットがリセットされている場合、キャプチャされたタイムスタンプのナノ秒フィールドは ~ 0.466 ns 単位のバイナリ数形式となります。そのため、ビット [30:0] は「 $2^{31} - \text{total ingress_correction_value}$ 」で書込み、ビット [31] = 1 としてバイナリで表現する必要があります。

出力補正

受信側では、内部スナップショットポイントでキャプチャされたタイムスタンプは、ポート境界でそのパケット SFD ビットが出力される時刻に比べて、早くなります（時刻が先になる）。そのため、キャプチャされたタイムスタンプを出力遅延と CDC サンプリングでのエラーにより補正する必要があります。

ます。この補正值は、ソフトウェアによって特定/算出し、[タイムスタンプ出力補正ナノ秒レジスタ \(ETH_MACTSECNR\)](#) に書き込む必要があります。

補正值は次の 3 つの要素からなります。

1. コアの出力とポートおよびネットワークの境界の間の PHY レイヤでの外部遅延
IEEE 802.3 Clause 45 MMD レジスタに準拠した PHY は、最大出力遅延と最小出力遅延を示すレジスタを備えています。ソフトウェアでこれらのレジスタを読み出し、PHY の平均出力遅延を特定することができます。または (PHY でこれらのレジスタがサポートされていない場合)、出力遅延は PHY データシートまたはタイミング特性から特定する必要があります。
2. 内部キャプチャポイントとコアの出力からの内部遅延
遅延は、[表 629](#) に示すように、アクティブな PHY インタフェース (RMII、MII など) と動作速度によって異なります。
3. CDC 同期エラー
同期によるタイムスタンプ補正は、次の式に基づいて行われます。

$$\text{EGRESS_SYNC_CORR} = (1 \times \text{PTP_CLK_PER} + 4 \times \text{TX_CLK_PER})$$

[表 629](#) に、各種 PHY インタフェースの出力遅延値と入力遅延値を示します。

表 629. PHY インタフェース用の出力遅延と入力遅延

PHY インタフェース		出力遅延	入力遅延
RGMII	1 Gbps	12	12
RGMII	100 Mbps	40	40
RGMII	10 Mbps	400	400
RMII	100 Mbps	40	120
RMII	10 Mbps	400	800

基準タイミングクロックの周波数範囲

タイムスタンプ情報は、MAC クロックドメインからアプリケーションクロックドメインへと、非同期クロックドメインをまたいで転送されます。そのため、連続する 2 回のタイムスタンプキャプチャの間に最小遅延が必要です。この遅延は、MII の 4 クロックサイクル、PTP クロックの 3 クロックサイクルです。2 回のタイムスタンプキャプチャ間の遅延がこの遅延より短い場合、MAC では 2 つめのパケットのタイムスタンプスナップショットが取得されません。

PTP クロック周波数制限は次のとおりです。

- 最大 PTP クロック周波数
最大 PTP クロック周波数は、基準時間の最大分解能によって制限されます (100 MHz で 10 ns)。また、同期の精度は、基準時間ソースの分解能または粒度によって決まります。そのため、PTP クロック周波数を大きくすると、システムのパフォーマンスは高まります。
- 最小 PTP クロック周波数
最小 PTP クロック周波数は、連続する 2 つの SFD バイトの間で必要な時間と、MII クロックドメインでの時間の同期のために取得される時間によって決まります。この関係は次の式で表されます。
$$3 \times \text{PTP クロック周期} + 4 \times \text{MII クロック周期} \leq 2 \text{ つの SFD 間の最小ギャップ}$$

MII クロック周波数は IEEE 仕様によって固定されます。そのため、適切な動作のために必要な最小 PTP クロック周波数は、[表 630](#) に示すように MAC の動作モードと動作速度によって決まります。

表 630. 最小 PTP クロック周波数例

モード	2 つの SFD 間の最小ギャップ	内部タイムスタンプ付きの最少 PTP 周波数
10 Mbps 全二重	168 MII クロック (64 バイトパケット用の 128 クロック + 最少 IFG の 24 クロック + プリアンブルの 16 クロック)	5 MHz
10 Mbps 半二重	48 MII クロック (衝突のために SFD の直後に送信された JAM パターン用の 8 クロック + IFG の 24 + プリアンブルの 16)	5 MHz
100 Mbps 全二重	168 MII クロック (64 バイトパケット用の 128 クロック + 最少 IFG の 24 クロック + プリアンブルの 16 クロック)	5 MHz
100 Mbps 半二重	48 MII クロック (衝突のために SFD の直後に送信された JAM パターン用の 8 クロック + IFG の 24 + プリアンブルの 16)	5 MHz

PTP 処理および制御

表 631 に、PTP メッセージ用の共通メッセージヘッダを示します。この形式は、IEEE 1588-2008 仕様から取得したものです。

表 631. IEEE 1588-2008 で定義されたメッセージ形式

ビット								オクテット	オフセット
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
予約済みです。				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
予約済みです。								1	5
flagField								2	6
correctionField								8	8
予約済みです。								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField ⁽¹⁾								1	32
logMessageInterval								1	33

1. controlField はバージョン 1 で使用されます。バージョン 2 では、messageType フィールドは別のメッセージタイプの検出用に使用されます。

Ethernet ペイロードの一部のフィールドは、PTP パケットタイプの検出と、取得するスナップショットの制御に使用できます。以下の PTP パケットについては、これらのフィールドは異なります。

- IPv4 経由の PTP パケット
- IPv6 経由の PTP フレーム
- Ethernet 経由の PTP パケット

IPv4 経由の PTP パケット

表 632 に、IEEE 1588 バージョン 1 および 2 に適合し、IPv4 の UDP 経由で送信された PTP パケットのスナップショットを制御するフィールドに関する情報を示します。タグ付きパケットのオクテット位置は、4 でオフセットされます。これは、IEEE 1588-2008 の付録 D と、表 631 に定義されたメッセージ形式に基づきます。

表 632. IEEE 1588-2008 で定義されたメッセージ形式

適合フィールド	オクテット位置	適合値	説明
MAC パケットタイプ	12, 13	0x0800	IPv4 データグラム
IP バージョンおよびヘッダ長	14	0x45	IP バージョンは IPv4 です。
レイヤ 4 プロトコル	23	0x11	UDP
IP マルチキャストアドレス (IEEE 1588 バージョン 1)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (または 0x82, 0x83, 0x84)	使用可能なマルチキャスト IPv4 アドレス : 224.0.1.129 224.0.1.130 224.0.1.131 224.0.1.132
IP マルチキャストアドレス (IEEE 1588 バージョン 2)	30, 31, 32, 33	0xE0, 0x00, 0x01, 0x81 (16 進数) 0xE0, 0x00, 0x00, 0x6B (16 進数)	PTP プライマリマルチキャストアドレス : 224.0.1.129 PTP Pdelay マルチキャストアドレス : 224.0.0.107
UDP デスティネーションポート	36, 37	0x013F, 0x0140	0x013F : PTP イベントメッセージ ⁽¹⁾ 0x0140 : PTP 全般メッセージ
PTP 制御フィールド (IEEE 1588 バージョン 1)	74	0x00, 0x01, 0x02, 0x03, 0x04、	0x00 : SYNC 0x01 : Delay_Req 0x02 : Follow_Up 0x03 : Delay_Resp 0x04 : 管理
PTP メッセージタイプフィールド (IEEE 1588 バージョン 2)	42 (ニブル)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, 0xD	0x0 : SYNC 0x1 : Delay_Req 0x2 : Pdelay_Req 0x3 : Pdelay_Resp 0x8 : Follow_Up 0x9 : Delay_Resp 0xA : Pdelay_Resp_Follow_Up 0xB : アナウンス 0xC : シグナリング 0xD : 管理
PTP バージョン	43 (ニブル)	0x1 または 0x2	0x1 : PTP バージョン 1 をサポート 0x2 : PTP バージョン 2 をサポート

1. PTP イベントメッセージは、SYNC、Delay_Req (IEEE 1588 バージョン 1 および 2) または Pdelay_Req、Pdelay_Resp (IEEE 1588 バージョン 2 のみ) です。

IPv6 経由の PTP フレーム

表 633 に、IEEE 1588 バージョン 1 および 2 に適合し、IPv4 の UDP 経由で送信された PTP パケットのスナップショットを制御するフィールドに関する情報を示します。タグ付きパケットのオクテット位置は、4 でオフセットされます。これは、IEEE 1588-2008 の付録 D と、表 631 に定義されたメッセージ形式に基づきます。

表 633. 制御とステータスに必要な IPv6-UDP PTP パケットフィールド

適合フィールド	オクテット位置	適合値	説明
MAC パケットタイプ	12, 13	0x86DD	IP データグラム
IP バージョン	14 (ビット [7:4])	0x6	IP バージョンは IPv6 です。
レイヤ 4 プロトコル	20 ⁽¹⁾	0x11	UDP
PTP マルチキャストアドレス	38 ~ 53	FF0x:0:0:0:0:0:181 (16 進数) FF02:0:0:0:0:0:6B (16 進数)	PTP プライマリマルチキャストアドレス : FF0x:0:0:0:0:0:0:181 (16 進数) PTP Pdelay マルチキャストアドレス : FF02:0:0:0:0:0:0:6B (16 進数)
UDP デスティネーションポート	56, 57a	0x013F, 0x140	0x013F : PTP イベントメッセージ 0x0140 : PTP 全般メッセージ
PTP 制御フィールド (IEEE 1588 バージョン 1)	94a	0x00, 0x01, 0x02, 0x03、 または 0x04	0x00 : SYNC 0x01 : Delay_Req 0x02 : Follow_Up 0x03 : Delay_Resp 0x04 : 管理 (バージョン 1)
PTP メッセージタイプフィールド (IEEE 1588 バージョン 2)	62a (ニブル)	0x0, 0x1, 0x2, 0x3, 0x8、 0x9, 0xB, 0xC, または 0xD	0x0 : SYNC 0x1 : Delay_Req 0x2 : Pdelay_Req 0x3 : Pdelay_Resp 0x8 : Follow_Up 0x9 : Delay_Resp 0xA : Pdelay_Resp_Follow_Up 0xB : アナウンス 0xC : シグナリング 0xD : 管理
PTP バージョン	63 (ニブル)	0x1 または 0x2	0x1 : PTP バージョン 1 をサポート 0x2 : PTP バージョン 2 をサポート

1. PTP パケット用に拡張ヘッダは定義されていません。

Ethernet 経由の PTP パケット

表 634 に、IEEE 1588 バージョン 1 および 2 に適合し、Ethernet 経由で送信された PTP パケットのスナップショットを制御するフィールドに関する情報を示します。タグ付きパケットのオクテット位置は、4 でオフセットされます。これは、IEEE 1588-2008 の付録 D と、メッセージ形式に基づきます。

表 634. 制御とステータスに必要な Ethernet PTP パケットフィールド

適合フィールド	オクテット位置	適合値	説明
MAC デスティネーション マルチキャストアドレス ⁽¹⁾	0 ~ 5	01-1B-19-00-00-00 01-80-C2-00-00-0E	すべての PTP メッセージで、次の任意のマルチキャストアドレスを使用できます ⁽²⁾ 。 01-1B-19-00-00-00 01-80-C2-00-00-0E ⁽³⁾
MAC パケットタイプ	12, 13	0x88F7	PTP Ethernet パケット
PTP 制御フィールド (IEEE 1588 バージョン 1)	46	0x00, 0x01, 0x02, 0x03, または 0x04	0x00 : SYNC 0x01 : Delay_Req 0x02 : Follow_Up 0x03 : Delay_Resp 0x04 : 管理
PTP メッセージタイプ フィールド (IEEE 1588 バージョン 2)	14 (ニブル)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, または 0xD	0x0 : SYNC 0x1 : Delay_Req 0x2 : Pdelay_Req 0x3 : Pdelay_Resp 0x8 : Follow_Up 0x9 : Delay_Resp 0xA : Pdelay_Resp_Follow_Up 0xB : アナウンス 0xC : シグナリング 0xD : 管理
PTP バージョン	15 (ニブル)	0x1 または 0x2	0x1 : PTP バージョン 1 をサポート 0x2 : PTP バージョン 2 をサポート

1. MAC アドレス 0 ~ 31 にプログラムされたデスティネーションアドレス (DA) のユニキャストアドレス一致は、[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の TSENMACADDR ビットがセットされている場合に使用されます。
2. IEEE 1588-2008 の付録 F
3. MAC では、ピア遅延マルチキャストアドレス (01-80-C2-00-00-0E) を持つ PTP バージョン 1 メッセージを有効な PTP とみなしません。

送信パス機能

MAC は、パケットの開始パケットデリミタ (SFD) が MII インタフェースで送信されたときにタイムスタンプをキャプチャします。タイムスタンプをキャプチャする必要のあるパケットは、パケットごとに制御できます。各送信パケットをマークして、タイムスタンプをキャプチャする必要があるかどうかを示すことができます。

MAC は、PTP パケットを識別するために送信されたパケットを処理しません。タイムスタンプをキャプチャする必要のあるパケットを指定する必要があります。パケットを定義するには、送信ディスクリプタの制御ビットを使用します ([セクション 57.10.3 : 送信ディスクリプタ](#)を参照)。MAC は、対応する送信ディスクリプタ内でソフトウェアにタイムスタンプを返して、特定の PTP パケットにタイムスタンプを自動的に結びつけます。

64 ビットタイムスタンプ情報は、TDES0 および TDES1 のフィールドに書き込まれます。TDES0 フィールドは、タイムスタンプの最下位 32 ビットを保持します。

受信パス機能

MAC は、MII インタフェースで受信したすべてのパケットのタイムスタンプをキャプチャしたり、パケットを処理して有効な PTP メッセージを特定したりするためにプログラムできます。アプリケーションに送信する時間のスナップショットは、[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の次のオプションを使用して制御できます。

- すべてのパケットに対するスナップショットの有効化
 - IEEE 1588 バージョン 1 またはバージョン 2 のタイムスタンプに対するスナップショットの有効化
 - Ethernet または UDP-IP-Ethernet 経由で直接送信される PTP パケットに対するスナップショットの有効化
 - IPv4 または IPv6 の受信パケットに対するタイムスタンプスナップショットの有効化
 - イベントメッセージに対するタイムスタンプスナップショットの有効化 (SYNC、DELAY_REQ、PDELAY_REQ、または PDELAY_RESP)
 - マスタまたはスレーブになるノードの有効化とスナップショットタイプの選択
- この機能は、スナップショットを取得するメッセージのタイプを制御します。

注： ペリフェラルは、VLAN パケット上の PTP メッセージもサポートしています。

表 635 では、[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の SNAPTYPSEL フィールドに応じてスナップショットが取られる PTP メッセージを示しています。

表 635. ETH_MACTSCR ビットへのタイムスタンプスナップショットの依存

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP メッセージ
00	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up
01	0	1	SYNC、Pdelay_Req、Pdelay_Resp
01	1	1	Delay_Req、Pdelay_Req、Pdelay_Resp
10	X	X	SYNC、Delay_Req
11	X	X	Pdelay_Req、Pdelay_Resp

DMA は、対応する受信ディスクリプタ内でソフトウェアアプリケーションにタイムスタンプを返します。タイムスタンプメッセージステータスおよび IPC ステータスを含む拡張ステータスは、RDES1 通常ディスクリプタに書き込まれ、タイムスタンプのスナップショットはコンテキストディスクリプタの RDES0 および RDES1 のフィールドに書き込みます。RDES0 フィールドは、タイムスタンプの最下位 32 ビットを保持します。

IEEE 1588 タイムスタンプに関するプログラミングガイドライン (システム時間補正)

参照： [セクション : システム時間の補正](#)、[セクション 57.9.9 : 2735 ページの IEEE 1588 タイムスタンプに関するプログラミングガイドライン](#)

IEEE 1588 システム時間ソース

時刻のスナップショットを取得するために、MAC では IEEE 1588-2002 (IEEE 1588-2008 で定義された 80 ビット形式) で定義された 64 ビット形式の基準時間が必要です。

IEEE 1588 システム時間ソースの説明

ペリフェラルではリファレンスクロック入力を使用し、これに基づいて基準時間（システム時間とも呼ばれます）を内部生成し、ペリフェラルのに使用リファレンス クロック入力は、タイムスタンプをキャプチャします。

タイムスタンプには、次のフィールドがあります。

- UInteger48 secondsField

秒フィールドは、秒単位でのタイムスタンプの整数部分です。このレジスタは 48 ビット幅です。たとえば、2.000000001 秒は secondsField = 0x0000 0000 0002 として表されます。

- UInteger32 nanosecondsField

ナノ秒フィールドは、ナノ秒単位でのタイムスタンプの分数部分です。たとえば、2.000000001 秒は nanoSeconds = 0x0000 0001 で表されます。

ナノ秒フィールドは次の 2 つのモードをサポートします。

- **デジタルロールオーバーモード**：このモードでは、ナノ秒フィールドの最大値が 0x3B9A C9FF で、 $(10e9-1)$ ナノ秒となります。
- **バイナリロールオーバーモード**：このモードでは、ナノ秒フィールドがロールオーバーし、値 0x7FFF FFFF の後に秒フィールドをインクリメントします。精度はビット当たり～0.466 ns です。

これらのモードは、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCTRLSSR ビットを使用してセットできます。

システム時間レジスタモジュール

64 ビットの PTP 時間は、PTP 入力リファレンスクロック clk_ptp_ref_i を使用して更新されます。この PTP 時間は、MII で送受信される Ethernet フレームのスナップショット（タイムスタンプ）をとるためのソースとして使用されます。

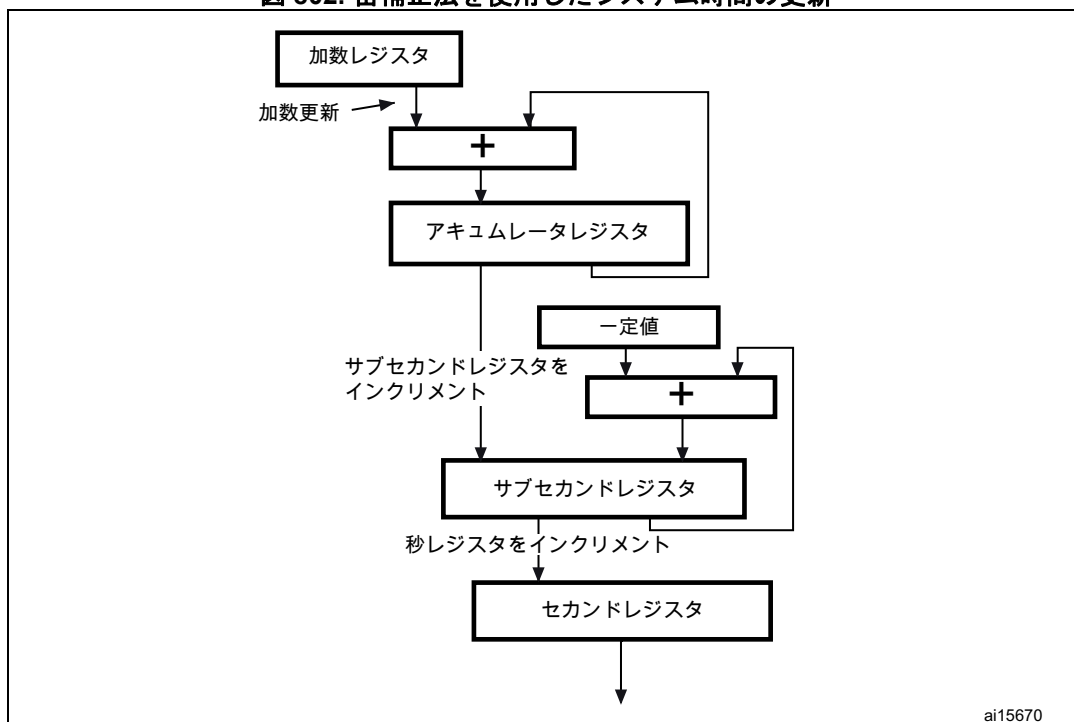
システム時間カウンタは、粗密補正法を使用して初期化または補正できます。

粗補正法では、初期値またはオフセット値がタイムスタンプ更新レジスタに書き込まれます。初期化の場合は、システムタイムカウンタにタイムスタンプ更新レジスタの値がプログラムされますが、システム時間の補正の場合は、システム時間に対してオフセット値（タイムスタンプ更新レジスタ）が加算または減算されます。

密補正法では、マスタクロックに対するスレーブクロック（基準クロック）の周波数ドリフト（IEEE 1588-2002 仕様で定義）が、一定時間にわたって補正されますが、粗補正法では、補正は 1 クロックサイクルで行われます。補正時間が長いほど、リニア時間を維持しやすく、PTP 同期メッセージインターバル間の基準時間に大きな変化（または大きなジッタ）が生じません。この方法では、[図 802](#) に示されるように、加算器が加数レジスタの内容を累積します。加算器が生成する桁上げは、システム時間カウンタをインクリメントするためのパルスとして使用されます。加算器と加数レジスタは、32 ビットのレジスタです。加算器は、高精度な周波数逓倍器または分周器として動作します。

このシステム時間更新アルゴリズムは、[図 802](#) に示しています。

図 802. 密補正法を使用したシステム時間の更新



ai15670

システム時間更新ロジックは、20 ns の精度を達成するために 50 MHz のクロック周波数を必要とします。周波数分周比は、必要なクロック周波数に対する基準周波数の比です。たとえば、リファレンスクロック (clk_ptp_ref_i) が 66 MHz の場合、この分周比は $66 \text{ MHz} / 50 \text{ MHz} = 1.32$ になります。したがって、レジスタで設定するデフォルトの加数値は、 $2^{32} / 1.32$ 、0xC1F07C1F です。

リファレンスクロックが低周波数側にドリフトして、たとえば 65 MHz になった場合、分周比は $65 / 50$ 、すなわち 1.3 となり、加数レジスタで設定する値は $2^{32} / 1.30$ 、すなわち 0xC4EC 4EC4 です。

基準クロックが高周波数側にドリフトして、たとえば 67 MHz になった場合は、加数レジスタを 0xBF0B 7672 に設定する必要があります。クロックのドリフトがないときには、デフォルトの加数値として 0xC1F0 7C1F ($2^{32} / 1.32$) をプログラムする必要があります。

図 802 では、サブセカンドレジスタを累積するために使用される一定値は 0.43 で、20 ns のシステム時間精度を達成します (言い換えると、20 ns 単位でインクリメントされます)。

ソフトウェアは、SYNC メッセージに基づいて周波数のドリフトを計算し、それに応じて加数レジスタを更新する必要があります。

最初に、スレーブクロックは加数レジスタで FreqCompensationValue0 で設定されます。この値は、次のとおりです。

$$\text{FreqCompensationValue}_0 = 2^{32} / \text{FreqDivisionRatio}$$

MasterToSlaveDelay が連続した同期メッセージで同じであると仮定した場合、このセクションで示されるアルゴリズムが適用されなければなりません。数サイクルの同期サイクル後、周波数ロックが発生します。スレーブクロックは、正確な MasterToSlaveDelay 値を決定して、新しい値を使用してマスタと再同期します。

アルゴリズムは、次のとおりです。

1. MasterSyncTime_n の時点で、マスタはスレーブクロックに SYNC メッセージを送信します。スレーブは、ローカルクロックが SlaveClockTime_n のときに、このメッセージを受信して、MasterClockTime_n を次のように計算します。

$$\text{MasterClockTime}_n = \text{MasterSyncTime}_n + \text{MasterToSlaveDelay}_n$$

2. 現在の同期サイクルのマスタクロックカウント、MasterClockCount_n は、次の式で求められます。

$$\text{MasterClockCount}_n = \text{MasterClockTime}_n - \text{MasterClockTime}_{n-1}$$

(MasterToSlaveDelay が同期サイクル n と n-1 で同じであると仮定)

3. 現在の同期サイクルのスレーブクロックカウント、SlaveClockCount_n は、次の式で求められます。

$$\text{SlaveClockCount}_n = \text{SlaveClockTime}_n - \text{SlaveClockTime}_{n-1}$$

4. 現在の同期サイクルのマスタとスレーブのクロックカウントの差、ClockDiffCount_n は、次の式で求められます。

$$\text{ClockDiffCount}_n = \text{MasterClockTime}_n - \text{SlaveClockTime}_n$$

5. スレーブクロックの周波数スケールファクタ、FreqScaleFactor_n は、次の式で求められます。

$$\text{FreqScaleFactor}_n = (\text{MasterClockCount}_n + \text{ClockDiffCount}_n) / \text{SlaveClockCount}_n$$

6. 加数レジスタの周波数補正值、FreqCompensationValue_n は、次の式で求められます。

$$\text{FreqCompensationValue}_n = \text{FreqScaleFactor}_n \times \text{FreqCompensationValue}_{n-1}$$

理論的には、このアルゴリズムは 1 同期サイクルでロックを達成します。ただし、ネットワークの伝搬遅延と動作条件の変動のために、数サイクルかかる場合もあります。このアルゴリズムは自己補正型です。最初にスレーブクロックがマスタにより誤った値にセットされた場合、アルゴリズムは追加の同期サイクルを費やして、それを補正します。

詳細なプログラミング手順については、[セクション 57.9.9: IEEE 1588 タイムスタンプに関するプログラミングガイドライン](#)を参照してください。

IEEE 1588 補助スナップショット

補助スナップショット機能によって、外部イベントに基づいたシステム時間のスナップショットを保存できます。イベントは、eth_ptp_trgx (ここで、x = 1 ~ 4) サイドバンド信号の立上りエッジとみなされます。

最大 4 つの補助スナップショット入力を設定し、最大 4 つのスナップショットを保存することができます。FIFO は、次のレジスタでアクセス可能です。[補助タイムスタンプ秒レジスタ \(ETH_MACATSSR\)](#) および [補助タイムスタンプナノ秒レジスタ \(ETH_MACATSNR\)](#)。

入力に対して取得されたスナップショットは、共通の FIFO に保存されます。64 ビットのみ確保されます。アプリケーションは、**タイムスタンプステータスレジスタ (ETH_MACTSSR)** を読み出して、この FIFO の最上部での読出しに入力を使用できるタイムスタンプを把握することができます。

スナップショットが保存されると、MAC は割り込みでアプリケーションに通知します。スナップショットの値は、FIFO レジスタアクセスから読み出されます。FIFO がフルになり、スナップショットを取得する外部トリガがアサートされた場合、**タイムスタンプステータスレジスタ (ETH_MACTSSR)** でスナップショットトリガミステータス (ATSSTM) がセットされます。これは、タイムスタンプの最新の補助スナップショットが FIFO に保存されないことを示します。FIFO がフルである場合、最新のスナップショットは書き込まれません。

アプリケーションが FIFO から 64 ビットタイムスタンプを読み出すとき、次のスナップショットを保存するために空きが使用できるようになります。FIFO は、**補助制御レジスタ (ETH_MACACR)** で ATSFC ビットをセットするとクリアできます。複数のスナップショットが FIFO にある場合、カウンタが **タイムスタンプステータスレジスタ (ETH_MACTSSR)** のビット [27:25] に示されます。

柔軟性の高い pulse-per-second 出力

MAC は、ETH_PPS_OUT および eth_ptp_pps_out 出力用に、固定の pulse-per-second 出力モード (固定モードとも呼ぶ) または柔軟性の高い pulse-per-second 出力モードをサポートしています。

- 固定の pulse-per-second 出力
このモードでは、**PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCTRL0 ビットをセットすることによって、PPS 出力の周波数だけを変更することができます。
- 柔軟性の高い pulse-per-second 出力
このモードでは、ソフトウェアには eth_ptp_pps_out 出力で生成されるパルスの開始時間、停止時間、幅、間隔をプログラムできる柔軟性があります。
開始時間と停止時間は、**PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)** でプログラムされます。
PPS の幅と間隔は、それぞれ **PPS 幅レジスタ (ETH_MACPPSWR)** および **PPS 間隔レジスタ (ETH_MACPPSIR)** でシステム時間の粒度 (サブセカンドのインクリメント値の単位の数) でプログラムされます。

- 注 :
- デフォルトでは、ペリフェラルは固定モードで、1 秒の間隔を示します。**PPS 制御レジスタ (ETH_MACPPSCR)** で PPSSEN0 を 0 にクリアして固定モードを選択した場合、次のようになります。
- すべての PPS 出力は、PPSCTRL_PPSCMD フィールドでプログラムされた値によって制御されます。固定モードでは、個別の PPS 出力の独立した制御はサポートされていません。
 - **PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** と **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)** は目標時間到達の割り込みを生成するためにのみ使用され、PPS 出力の生成には使用されません。
 - TRGTMODESEL0/1/2/3 は 0 にプログラムする必要があります。
 - **PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCTRL0 フィールドをセットすることによって、PPS 出力の周波数を変更することができます。

柔軟性の高い pulse-per-second (PPS) 出力の説明

ペリフェラルは柔軟性の高い PPS 出力により、次の機能をサポートします。

- システム時間に基づく開始時間または停止時間のプログラミング。
- 64 ビットのシステム時間に基づく、シングルパルスの開始位置およびパルス列の開始位置および終了位置のプログラミング。目標時間レジスタを使用した開始時間と終了時間のプログラミング。
- 終了時間の事前のプログラミング。つまり、実際に開始時間が経過する前に終了時間をプログラムすることができます。
- サブセカンドインクリメントレジスタ (ETH_MACSSIR) でプログラムされたサブセカンドインクリメント値の単位数に基づいた、PPS 信号出力の立上りエッジと対応する立下りエッジの間の幅のプログラミング。1 ~ 232-1 個の単位のサブセカンドインクリメント値によるパルス幅のプログラミング。
- サブセカンドインクリメント値の単位数に基づく、PPS 信号の立上りエッジ間の間隔のプログラミング。1 ~ 232-1 個の単位のサブセカンドインクリメント値により、パルス間隔をプログラムできます。
- プログラムされた PPS 開始または終了リクエストをキャンセルするオプション。
- プログラムされた開始または終了時間がすでに経過している場合のエラー。

注：以下のセクションで説明する PTP リファレンスクロックは、システム時間の更新に使用されるクロックです。タイムスタンプ制御レジスタ (ETH_MACTSCR) の TSCFUPDT ビットが 0 にセットされている場合、このクロックは clk_ptp_ref_i クロックと同様です。密補正モードでは、このクロック数でシステム時間が更新されます (サブセカンドインクリメントレジスタ (ETH_MACSSIR) を使用、図 802 を参照)。

柔軟性の高いパルス出力の設定方法の詳細については、セクション 57.9.12: 柔軟性の高い Pulse-Per-Second (PPS) 出力に関するプログラミングガイドラインを参照してください。

PPS の開始時間および終了時間

初期開始時間は、目標時間レジスタでプログラムできます。

必要に応じて、開始時間または終了時間を再度プログラムできます。ただし、これが可能になるのは、前にプログラムされている値が PTP クロックドメインと同期された後です。PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR) のビット 31 は同期が完了していることを示します。これにより、終了時間または開始時間が経過する前でも、事前に開始時間または終了時間をプログラムできます。

適切な PPS 信号出力であることを確認するために、開始時間や停止時間に高度なシステム時間をプログラムすることを推奨します。アプリケーションがすでに経過した開始時間や停止時間をプログラムした場合、MAC はプログラミングエラーを示すエラーステータスビットをセットします。有効になっている場合、MAC は目標時間到達の割込みイベントもセットします。対応する開始時間または停止時間が経過していない場合のみ、アプリケーションは開始または停止リクエストをキャンセルできます。時間が経過した場合、キャンセルコマンドは有効ではありません。

PPS の幅と間隔

PPS の幅と間隔は、システム時間の粒度 (サブセカンドのインクリメント値の単位数) に基づいてプログラムされます。たとえば、50 MHz の PTP リファレンスクロックにより 40 ns の PPS パルス幅と 100 ns の間隔を取得するには、幅と間隔をそれぞれ値 2 と 5 にプログラムします。粒度を小さくするには、PTP リファレンスクロックを速くします。

PPS 出力でパルスまたはパルス列をトリガするためのコマンドを実行する前に、PPS 信号出力の間隔と幅をプログラムするか更新します。

PTP タイムスタンプオフロード機能

この機能は、MAC が PTP ネットワークで特定のノードとして動作しているときに、実行する特定の PTP パケットの自動生成を可能にします。

これらのパケットは、周期的に生成したり、ホストソフトウェアでトリガしたりできます。その他のモードでは、この機能によってレシーバの着信 PTP パケットを解析でき、必要な PTP パケットを自動的に生成して応答できます。これは、特定の PTP ノードの機能をさらに高精度かつ低レスポンスレイテンシでオフロードするのに役立ちます。

PTP オフロード機能は [PTP オフロード制御レジスタ \(ETH_MACPOCR\)](#) で選択します。80 ビット PTP ノードアイデンティティを設定するのは、3 つのレジスタ、[PTP ソースポートアイデンティティ 0 レジスタ \(ETH_MACSPI0R\)](#)、[PTP ソースポートアイデンティティ 1 レジスタ \(ETH_MACSPI1R\)](#) および [PTP ソースポートアイデンティティ 2 レジスタ \(ETH_MACSPI2R\)](#) です。

PTP オフロード機能の説明

プログラムされたモードに応じて、MAC は周期的、またはアプリケーションから、または特定の PTP メッセージの受信に基づいて、PTP Ethernet メッセージを生成します。[表 636](#) は、PTP メッセージ生成の基準を示します。

表 636. PTP メッセージ生成の基準

プログラミング			モード	PTP メッセージの生成の基準	生成される PTP メッセージタイプ
SNAPTYPSEL	TSMSTRENA	TSEVNTENA			
00	0	1	通常スレーブまたは境界スレーブ	SYNC メッセージの受信	Delay_Req
00	1	1	通常マスタまたは境界マスタ	周期的またはアプリケーションからのトリガ時	SYNC
				Delay_Req メッセージの受信	Delay_Resp
01	0	1	トランスペアレントスレーブ	周期的またはアプリケーションからのトリガ時	Pdelay_Req
				Pdelay_Req メッセージの受信	Pdelay_Resp
				SYNC メッセージの受信	Delay_Req
01	1	1	トランスペアレントマスタ	周期的またはアプリケーションからのトリガ時	Pdelay_Req
				Pdelay_Req メッセージの受信	Pdelay_Resp
				周期的またはアプリケーションからのトリガ時	SYNC
				Delay_Req メッセージの受信	Delay_Resp

表 636. PTP メッセージ生成の基準 (続き)

プログラミング			モード	PTP メッセージの生成の基準	生成される PTP メッセージタイプ
SNAPTYPSEL	TSMSTRENA	TSEVNTENA			
11	X	X	ピアツーピア アトランス ペアレント	周期的または アプリケーションからの トリガ時	Pdelay_Req
				Pdelay_Req メッセージの受信	Pdelay_Resp
他のすべてのプログラミングの組み合わせは、PTP オフロード機能では無効です。					

注 : IEEE 1588-2008 仕様に従い、ピア遅延メカニズムに対応しているクロックで遅延リクエスト/遅延応答メッセージを生成しないようにする必要があります。ただし、ペリフェラルコントローラは、**PTP オフロード制御レジスタ (ETH_MACPOCR)** のプログラム可能な制御ビット (DRRDIS) を使用して、柔軟性を高めるためにこれをサポートしています。

DRRDIS ビットを使用して、遅延リクエスト/遅延応答メッセージの応答生成を制御することができます。たとえば、トランスペアレントスレーブモードでは、このビットがリセットされている場合のみ、受信した SYNC への応答として、遅延リクエストが生成されます。

MAC が PTP ネットワーク内の通常スレーブまたは境界スレーブクロックとしてセットされている場合、SYNC メッセージを受信すると、対応する Delay_Req メッセージを自動生成して送信することにより応答します。同様に、それ以外の各種動作モードを表 636 に示します。

MAC では、SYNC および Pdelay_Req PTP メッセージの生成について、マルチキャスト通信モデルをサポートしています。たとえば、Ethernet パケット上に生成された PTP のデスティネーションアドレスフィールドは、定義された特別なマルチキャストアドレスです (ピア遅延メカニズムメッセージ以外はすべて 0x011B 1900 0000、ピア遅延メカニズムメッセージ用は 0x0180 C200 000E)。

MAC が特別なマルチキャストデスティネーションアドレスを持つ受信済み SYNC、Delay_Req、および Pdelay_Req PTP メッセージに応答する場合、自動的に生成される Delay_Req、Delay_Resp、Pdelay_Resp PTP メッセージのそれぞれの DA フィールド内の対応する特別なマルチキャストアドレスも使用します。

MAC がユニキャストデスティネーションアドレスを持つ受信済み SYNC、Delay_Req、および Pdelay_Req PTP メッセージに応答する場合、受信したパケットの SA フィールドを取得し、自動的に生成される Delay_Req、Delay_Resp、Pdelay_Resp PTP メッセージのそれぞれの DA フィールドとして利用します。

また、MAC レシーバのパケットフィルタリングロジックを満たしている場合、受信したすべての PTP メッセージが、MAC によって応答が生成されたかどうかを示す Rx ステータスと一緒にアプリケーションに転送されます。

MAC で自動的に PdelayReq が生成される場合、または Delay_Req により応答される場合、これらの 2 つの PTP メッセージの出力タイムスタンプは Tx TS ステータスで示されます (Tx タイムスタンプステータスレジスタと生成された割込み)。

Ethernet 経由 PTP の基本メッセージ検出で messageType および versionPTP フィールドが照合されるだけでなく、受信した PTP メッセージのタイプを適格とするために次の追加フィールドが照合されます。

1. domainNumber フィールドがチェックされ、CSR でプログラムされた値と照合されます。
2. flagField フィールドの twoStepFlag のワンステップ指示 (0b0) がチェックされます。
3. デフォルトの Ethernet 経由 PTP (0b0000) または 802.1AS モード (0b1111) が有効になっている場合、これについて transportSpecific フィールドがチェックされます。

PTP パケット生成

このセクションでは、このモードが有効になっている場合に MAC によって自動生成される PTP パケットの形式と内容を説明します。共通の PTP メッセージヘッダのテンプレートと、生成される特定の PTP パケットのフィールドの詳細も示します。

表 637. 共通の PTP メッセージヘッダフィールド

ビット								オクテット	オフセット
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
予約済みです。				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
予約済みです。								1	5
flagField								2	6
correctionField								8	8
予約済みです。								4	16
sourcePortIdentity								10	20
sequenceId								2	30
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

PTP メッセージヘッダフィールド

- **messageType**
PTP メッセージタイプには、次のエンコード済みの値が使用されます。
 - SYNC : 0000
 - Delay_Req : 0001
 - Pdelay_Req : 0010
 - Pdelay_Resp : 0011
 - Delay_Resp : 1001
- **transportSpecific**
次の転送プロトコルエンコーディングが使用されます。
 - Ethernet 経由 デフォルト PTP : 0000
 - 802.1AS モード : 0001
- **versionPTP**
PTP バージョン 2 がサポートされているため、常に 2 にセットされます。
- **domainNumber**
このフィールドには、[PTP オフロード制御レジスタ \(ETH_MACPOCR\)](#) の値が格納されます。

- **flagField**

次の値が使用されます。

- alternateMasterFlag (オクテット 0 ビット 0) : SYNC および Delay_Resp の場合は 0
- twoStepFlag (オクテット 0 ビット 1) : SYNC および Pdelay_Resp の場合は 0
- unicastFlag (オクテット 0 ビット 2) : マルチキャストアドレスの場合は 0、ユニキャストアドレスの場合は 1

- **correctionField**

詳細については、表 638 を参照してください。

- **sourcePortIdentity**

このフィールドでは、PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)、PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)、PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R) でプログラムされた値を使用します。

- **sequenceId**

Pdelay_Resp および Delay_Resp では、受信した Pdelay_Req および Delay_Req PTP メッセージの同じ sequenceId フィールドを使用します。SYNC/Delay_Req、Pdelay_Req の場合、別の sequenceId カウンタが維持されます。これらの sequenceId カウンタは、対応するメッセージが生成され、送信されるたびに 1 ずつインクリメントされます。

- **controlField**

controlField には、次のエンコード済みの値が使用されます。

- SYNC : 0000 0000
- Delay_Req : 0000 0001
- Pdelay_Req : 0000 0010
- Pdelay_Resp : 0000 0101
- Delay_Resp : 0000 0011

- **logMessageInterval**

- SYNC :
このフィールドには、対応する MAC_Log_Message_Interval レジスタの logSyncInterval が格納されます。
- Delay_Resp :
このフィールドには、マルチキャスト PTP メッセージの場合はログメッセージ間隔レジスタ (ETH_MACLMIR) から取得した DRSYNCR 値と logSyncInterval 値の合計値、ユニキャスト PTP メッセージの場合は 0111 1111 が格納されます。
- Delay_Req、Pdelay_Req、Pdelay_Resp : 0111 1111
ここで、logSyncInterval = log2 (秒単位の間隔の平均値)

MAC は logSyncInterval フィールドについて -15 ~ 15 の値をサポートしています。これは、32.768 マイクロ秒 (2⁻¹⁵) から 215 秒の範囲に変換されます。ログ同期間隔 (N) の所定の値について、SYNC パケット間の間隔は次の式で表されます。

- $2^{(30+N)}$ ns : N が負 (-1 ~ -15) の場合
- 2^N 秒 : N が正 (0 ~ 15) の場合

例 :

- logSyncInterval が 1 にプログラムされている場合、間隔は 2^1 となるため、SYNC メッセージは 2 秒ごとに送信されます。
- logSyncInterval が -1 にプログラムされている場合、間隔は $2^{-1} = 0.536$ 秒となるため、SYNC メッセージは 536 ミリ秒ごとに送信されます。値が 0.536 秒となるのは、 $2^{-30} = 1$ ns であるためです。
- logSyncInterval が -5 にプログラムされている場合、間隔は $2^{-5} = 33.55$ ms となるため、SYNC メッセージは 33.55 ms ごとに送信されます。

注 : MAC では、周期的パケット送信の間隔を生成するため、PTP システム時間を使用します。ログメッセージの間隔に負の値がプログラムされている場合、システム時間のナノ秒フィールドはバイナリでないため、生成される周期は式 $2^{(30+N)}$ で得られる値と異なることがあります。

PTP メッセージ固有フィールド

メッセージ固有フィールドは次のとおりです。

- **messageLength**
接尾文字はサポートされていないため、このフィールドには、34 バイトの PTP 共通ヘッダとメッセージタイプに固有の本体を含む PTP メッセージの長さが格納されます。
SYNC および Delay_Req パケットの場合、このフィールドには 44 が格納されますが、Delay_Resp、Pdelay_Req、Pdelay_Resp の場合、54 が格納されます。
- **originTimestamp**
このフィールドは、SYNC、Delay_Req、Pdelay_Req PTP メッセージ用にキャプチャされた出カタイムスタンプです。
- **receiveTimestamp**
Delay_Resp PTP メッセージの場合、これは対応する受信済み Delay_Req PTP メッセージの入カタイムスタンプです。
- **requestingPortIdentity**
Delay_Resp および Pdelay_Resp PTP メッセージの場合、これは対応する受信済み Delay_Req および Pdelay_Req PTP メッセージから取得した sourcePortIdentity フィールドです。
- **requestReceiptTimestamp**
Pdelay_Resp PTP メッセージの場合、このフィールドは 0 にセットされます。

ワンステップタイムスタンプ

MAC はワンステップタイムスタンプ機能をサポートしていて、パケット内でオフセットを識別し、アプリケーションから受信したタイムスタンプをそのオフセットに挿入することができます。

ワンステップタイムスタンプ用の MAC 送信 PTP モード

メッセージのタイプとそのモードに応じて、MAC は送信 PTP パケットの次のフィールドを更新します。

- メッセージの PTP ヘッダ内の correctionField
- SYNC、Delay_Req、および Pdelay_Req メッセージ内の originTimestamp

表 638 に、[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の SNAPTYPSEL、TSMSTRENA、TSEVNTENA ビットの設定に基づいて PTP モードが選択される方法と、ワンステップタイムスタンプ動作中にそのモードのメッセージタイプに基づいて着信 PTP パケット用に更新されるフィールドを示します。

表 638. MAC 送信 PTP モードとワンステップタイムスタンプ動作

プログラミング			モード	パケットごとの制御 ⁽¹⁾			送信時に処理されるメッセージ
SNAPTYP SEL	TSMSTR ENA	TSEVNT ENA		TTSE (2)	OSTC (3)	TTS ⁽⁴⁾	
X	X	X	N/A	1	X	X	タイムスタンプがキャプチャされ、アプリケーションに返されます
X	X	X	N/A	X	0	X	OST 操作は実行されません (PTP パケットは変更されません)
00	X	0	エンドツー エンドトランス ペアレント	0	1	入力 TS	Sync (滞留時間および入力非対称補正用の補正フィールド) Delay_Req (滞留時間および出力非対称補正用の補正フィールド)
00	0	1	通常スレーブ または 境界スレーブ	1	1	X	Delay_Req (originTimestamp フィールド) Delay_Req (出力非対称補正用の補正フィールド)
00	1	1	通常マスタ または 境界マスタ	0	1	X	Sync (originTimestamp フィールド) Sync (サブナノ秒補正用の補正フィールド)
01	X	0	ピア遅延 メカニズム用の サポート付きの エンドツー エンドトランス ペアレント	0	1	入力 TS	Sync (滞留時間および入力非対称補正用の補正フィールド)
						入力 TS	Pdelay_Req (滞留時間および出力非対称補正用の補正フィールド)
						入力 TS	Pdelay_Resp (滞留時間および入力非対称補正用の補正フィールド)
01	0	1	ピア遅延 メカニズムの サポート付きの 通常スレーブ または境界 スレーブまたは ピアツーピア トランス ペアレント	0	1	入力 TS	Sync (滞留時間および入力非対称補正用の補正フィールド) (ピアツーピアトランスペアレントクロック動作の場合のみ適用可能)
				1	1	X	Delay_Req (originTimestamp フィールド) Delay_Req (出力非対称補正用の補正フィールド)
				1	1	X	Pdelay_Req (originTimestamp フィールド) Pdelay_Req (出力非対称補正用の補正フィールド)
				0	1	Pdelay_Req 用の 入力 TS	Pdelay_Resp (ターンアラウンド時間および入力非対称補正用の補正フィールド)
01	1	1	ピア遅延 メカニズムの サポート付きの 通常マスタ または 境界マスタ	0	1	X	Sync (originTimestamp フィールド) Sync (サブナノ秒補正用の補正フィールド)
				1	1	X	Pdelay_Req (originTimestamp フィールド) Pdelay_Req (出力非対称補正用の補正フィールド)
				0	1	Pdelay_Req 用の 入力 TS	Pdelay_Resp (ターンアラウンド時間および入力非対称補正用の補正フィールド)

表 638. MAC 送信 PTP モードとワンステップタイムスタンプ動作 (続き)

プログラミング			モード	パケットごとの制御 ⁽¹⁾			送信時に処理されるメッセージ
SNAPTYP SEL	TSMSTR ENA	TSEVNT ENA		TTSE (2)	OSTC (3)	TTS ⁽⁴⁾	
10	X	X	エンドツー エンドトランス ペアレント	0	1	入力 TS	Sync (滞留時間および入力非対称補正用の補正フィールド)
						入力 TS	Delay_Req (滞留時間および出力非対称補正用の補正フィールド)
11	X	X	ピアツーピア トランス ペアレント	0	1	入力 TS	Sync (滞留時間および入力非対称補正用の補正フィールド)
				1	1	X	Pdelay_Req (originTimestamp フィールド) Pdelay_Req (出力非対称補正用の補正フィールド)
				0	1	Pdelay_Req 用の 入力 TS	Pdelay_Resp (ターンアラウンド時間および入力非対称補正用の補正フィールド)

- ここに示すパケットごとの制御値は、通常の PTP 動作で、プログラムされたモード用にデバイスによって使用される推奨設定です。
- TTSE は TDES2 送信通常ディスクリプタの TTSE ビットを示します。TTSE 機能は OST 機能や、OST 用にプログラムされた動作モードから独立しています。TTSE ビットがセットされている場合、MAC はタイムスタンプをキャプチャして返します。
- OSTC は TDES3 送信コンテキストディスクリプタの OSTC ビットを示します
- TTS は、TDES0 および TDES1 送信通常ディスクリプタの TTSH、TTSL フィールドで提示されるタイムスタンプ値を示します (書き戻しフォーマット)。

注 : 滞留時間/ターンアラウンド時間は、キャプチャされたタイムスタンプ (出力タイムスタンプ) と入力タイムスタンプの差として計算されます。ピア遅延メカニズムをサポートするクロックでは遅延リクエストまたはレスポンスを使用しませんが、柔軟性を高めるために OST に組み込まれています。

ワンステップタイムスタンプの有効化

特定の packets に対してワンステップタイムスタンプ機能を有効にするには、TDES3 コンテキストディスクリプタでビット 20 (OSTC) をセットします。特定の PTP パケット内の補正フィールドを更新するには、TSSL および TSSH フィールドで入力タイムスタンプを割り当てる必要があります。

ワンステップタイムスタンプ機能は、Ethernet 経由 PTP パケットでのみサポートされます。IPv4/IPv6 経由 PTP パケットではサポートされません。

57.5.5 チェックサムオフロードエンジン

TCP や UDP などの通信プロトコルは、ネットワーク経由で送信されたデータの完全性を確認できるチェックサムフィールドを実装しています。最も広く利用されている Ethernet の用途は、IP データグラムによる TCP や UDP のカプセル化です。MAC は、チェックサム計算と送信パスへの挿入、および受信パスでのエラー検出をサポートするチェックサムオフロードエンジン (COE) を備えています。

送信チェックサムオフロードエンジン

送信パスで MAC はチェックサムを計算し、Tx パケットに挿入します。この機能を使用すると、ソフトウェアの負荷を軽減し、システムの全体的なスループットを向上させることができます。

COE モジュールは、2 種類のチェックサム計算および挿入をサポートします。チェックサムエンジンは、CIC ビット (TDES3 ビット [17:16]) をセットすることによって、パケットごとに制御できます。

注 : TCP、UDP、または ICMP のチェックサムは、完全なパケットに対して計算された後、対応するヘッダフィールドに挿入されます。この要件のため、MAC が閾値 (カットスルー) モードに設定されている場合でも、Tx FIFO は自動的にストアアンドフォワードモードで動作します。

パケットを MAC トランスミッタに送信する前に、Tx FIFO にパケット全体を格納できるだけの深さがあることを確認してください。プログラム済みのバーストデータ長に対応できるだけのスペースがないと、デッドロックを回避するために MTL Tx FIFO で読出しが開始されるためです。その場合、ペイロードチェックサムの計算と挿入が行なえるようになる前にパケットヘッダの開始が読み出されるため、COE がエラーとなります。そのため、パケット数が次の式で示されるバイト数より小さい場合のみ、チェックサムの挿入を有効化する必要があります。

$$\text{Packet size} < \text{TxQSize} - (\text{PBL} + 7) \times 4$$

ここで、

TxQSize はTx キュー動作モードレジスタ (ETH_MTLTXQOMR) の TQS ビットフィールドに対応します。

PBL はチャネル送信制御レジスタ (ETH_DMACTXCR) の TxPBL ビットフィールドに対応します。

IPv4、TCP、UDP、ICMP、IPv6、および ICMPv6 パケットヘッダの仕様については、IETF 仕様の RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460、および RFC 4443 をそれぞれ参照してください。

IP ヘッダチェックサムエンジン

IPv4 データグラムでは、ヘッダフィールドの完全性は、16 ビットのヘッダチェックサムフィールド (IPv4 データグラムの 11 番目と 12 番目のバイト) によって示されます。COE は、Ethernet パケットのタイプフィールドの値が 0x0800 であり、IP データグラムのバージョンフィールドの値が 0x4 のときに、IPv4 データグラムを検出します。入力パケットのチェックサムフィールドは、計算時には無視され、計算された値に置き換えられます。

注 : IPv6 ヘッダには、チェックサムフィールドがありません。そのため、COE は IPv6 ヘッダフィールドを変更しません。

この IP ヘッダチェックサムの計算結果は、送信ステータス (表 660 : TDES3 通常ディスクリプタ (書き戻しフォーマット) のビット 0) の IP ヘッダエラーステータスビットによって示されます。

このステータスビットは、Ethernet タイプフィールドと IP ヘッダのバージョンフィールドの値が一致しないとき、または Ethernet パケットに十分なデータがない (IP ヘッダの長さフィールドによって示されます) ときにセットされます。言い換えると、このビットは、IP ヘッダエラーが以下のような状況でアサートされたときにセットされます。

- IPv4 データグラムの場合 :
 - 受信した Ethernet タイプは 0x0800 ですが、IP ヘッダのバージョンフィールドは 0x4 ではないとき。
 - IPv4 ヘッダ長フィールドが 0x5 (20 バイト) 未満の値を示しているとき。
 - 合計パケット長が IPv4 ヘッダ長フィールドで指定された値より小さいとき。
- IPv6 データグラムの場合 :
 - Ethernet のタイプは 0x86DD だが、IP ヘッダのバージョンフィールドが 0x6 ではないとき。
 - IPv6 ヘッダ (40 バイト) または拡張ヘッダ (拡張ヘッダ内の対応するヘッダ長フィールドで指定) が完全に受信される前にパケットが終了したとき。

TCP/UDP/ICMP チェックサムエンジン

TCP/UDP/ICMP チェックサムエンジンは、IPv4 または IPv6 ヘッダ（拡張ヘッダを含む）を処理して、カプセル化されたペイロードが TCP、UDP、ICMP のいずれかを判断します。TCP、UDP、または ICMP ペイロードのチェックサムが計算され、ヘッダ内の対応するフィールドに挿入されます。Tx COE は次の 2 つのモードで動作します。

- TCP、UDP、または ICMPv6 擬似ヘッダは、チェックサムの計算に含まれず、入力パケットのチェックサムフィールドに存在するとみなされます。このエンジンは、チェックサムの計算にチェックサムフィールドを含め、このチェックサムフィールドを最終的に計算されたチェックサムに置き換えます。
- エンジンがチェックサムフィールドを無視し、TCP、UDP、または ICMPv6 擬似ヘッダデータをチェックサムの計算に含め、このチェックサムフィールドを最終的な計算値で上書きします。

注： ICMP-over-IPv4 パケットの場合、擬似ヘッダは定義されていないので、ICMP パケット内のチェックサムフィールドは、どちらのモードでも常に 0x0000 でなければなりません。0x0000 に等しくない場合、正しくないチェックサムがパケットに挿入される可能性があります。

この操作の結果は、送信ステータスペクタのペイロードチェックサムエラーステータスビット（表 660 : TDES3 通常ディスクリプタ（書き戻しフォーマット）のビット 12）によって示されます。パケットが FIFO に書き込まれているパケットの終了（EOP）のなしでストアアンドフォワードモードで MAC トランスミッタエンジンに転送されたことを検出した場合、または IP ヘッダのペイロード長フィールドで示されたバイト数を受信する前にパケットが終了した場合に、このエンジンはペイロードチェックサムエラーステータスビットをセットします。パケットが、示されているペイロード長よりも長いときには、COE はそれらをスタッフバイトとして無視し、エラーは報告されません。このエンジンが最初のタイプのエラーを検出したとき、TCP、UDP、または ICMP ヘッダを変更しません。2 番目のタイプのエラーの場合、計算されたチェックサムを対応するヘッダフィールドに挿入します。

表 639 では、パケットタイプに基づいて送信チェックサムオフロードエンジンによってサポートされる機能を説明します。MAC がチェックサムを挿入しないときは、表に「いいえ」と示されます。

注： IPv4 または IPv6 パケットが[セクション : 送信チェックサムオフロードエンジン](#)で指定されたフレームサイズ制限より大きい場合、チェックサムの挿入を有効化しないでください。誤ってチェックサムが挿入されたり、予想外の動作につながったりすることがあります。

表 639. 異なるパケットタイプの送信チェックサムオフロードエンジンの機能

パケットタイプ	ハードウェア IP ヘッダ チェックサム挿入	ハードウェア TCP/UDP チェックサム挿入
非 IPv4 または非 IPv6 パケット	不可	いいえ
IPv4 パケットが 1522 バイトより大きい（MAC で 2K パケット用の IEEE 802.3a d サポートが有効になっている場合は 2000 バイト）、 セクション : 送信チェックサムオフロードエンジン で指定されたフレームサイズの制限以下。	可能	はい
IPv6 パケットが 1522 バイトより大きい（MAC で 2K パケット用の IEEE 802.3a d サポートが有効になっている場合は 2000 バイト）、 セクション : 送信チェックサムオフロードエンジン で指定されたフレームサイズの制限以下。	適用外	可能

表 639. 異なるパケットタイプの送信チェックサムオフロードエンジンの機能 (続き)

パケットタイプ	ハードウェア IP ヘッダ チェックサム挿入	ハードウェア TCP/UDP チェックサム挿入
TCP、UDP、または ICMP での IPv4	可能	はい
IP オプションがある IPv4 パケット (IP ヘッダが 20 バイトを超える)	可能	はい
IPv4 の断片のパケット	可能	不可
メインまたは拡張ヘッダで次のヘッダフィールドを持つ IPv6 パケット : – ホップバイホップオプション (IPv6 メインヘッダ) – ホップバイホップオプション (IPv6 拡張ヘッダ) – 転送先オプション – ルーティング (残りセグメント 0) – ルーティング (残りセグメント 1 以上) – TCP、UDP、または ICMP – 認証 – メインまたは拡張ヘッダでのその他の次のヘッダフィールド	– 適用外 – 適用外 – 適用外 – 適用外 – 適用外 – 適用外 – 適用外 – 適用外	– 可能 – 不可 – 可能 – 不可 – 不可 – 不可 – 可能 – 不可 – 不可
オプションフィールドのある TCP ヘッダを持つ IPv4 パケット	可能	はい
IPv4 トンネル : – IPv4 トンネルでの IPv4 パケット – IPv4 トンネルでの IPv6 パケット	– はい (IPv4 トンネルヘッダ) – はい (IPv4 トンネルヘッダ)	– 不可 – いいえ
IPv6 トンネル : – IPv6 トンネルでの IPv4 パケット – IPv6 トンネルでの IPv6 パケット	– 適用外 – 適用外	– 不可 – いいえ
802.3ac タグのある IPv4 パケット (有効時 C-VLAN タグまたは S-VLAN タグ付き)	可能	はい
802.3ac タグのある IPv6 パケット (有効時 C-VLAN タグまたは S-VLAN タグ付き)	適用外	可能
セキュリティ機能のある IPv4 フレーム (カプセル化されたセキュリティペイロードなど)	可能	不可
セキュリティ機能のある IPv6 フレーム (カプセル化されたセキュリティペイロードなど)	適用外	不可

受信チェックサムオフロードエンジン

受信チェックサムオフロードエンジンを使用して IP パケット内のエラーを検出するには、ヘッダのチェックサムを計算して、受信したヘッダのチェックサムと照合します。このエンジンでは、受信した IP パケット内の TCP、UDP、または ICMP ペイロードも識別され、そのペイロードに応じてチェックサムが計算されます。

受信チェックサムオフロードエンジン (Rx COE) を有効化するには、[動作モード設定レジスタ \(ETH_MACCCR\)](#) の IPC ビットをセットします。このビットをセットすると、データの完全性チェックのために、受信 Ethernet パケットの IPv4 または IPv6 パケットが検出され、処理されます。MAC レシーバは、受信した Ethernet パケットのタイプフィールドの値が 0x0800 か 0x86DD かをチェックすることによって、そのパケットが IPv4 か IPv6 かを識別します。この識別は、シングル VLAN タグ付きパケットに適用できます。[VLAN タグレジスタ \(ETH_MACVTR\)](#) の EDVLP ビットがセットされている場合、ダブル VLAN タグ付きパケットにも適用されます。

Rx COE は、IPv4 ヘッダのチェックサムを計算して、受信した IPv4 ヘッダのチェックサムと一致するかどうかを確認します。この操作の結果（成功または失敗）は、受信ステータスワードへの挿入のための RFC モジュールに付与されます。示されているペイロードタイプ（Ethernet タイプフィールド）と IP ヘッダバージョン間に不一致があった場合、または、受信パケットのバイト数が、IPv4 ヘッダの長さフィールドで示されているバイト数より少ないとき（または、IPv4 または IPv6 ヘッダで使用可能なスペースが 20 バイト未満のとき）には、IP ヘッダエラービットがセットされます。

[Rx キュー動作モードレジスタ \(ETH_MTLRXQOMR\)](#) の DIS_TCP_EF ビットがリセットされ、FEP ビットがセットされている場合、TCP/IP エラーの発生したパケット（ヘッダまたはペイロード）は MTL でドロップされます。

このエンジンは、受信した IP データグラム（IPv4 または IPv6）のペイロードが TCP、UDP、または ICMP のいずれであるかも識別して、TCP、UDP、または ICMP 仕様での定義に従って、それぞれのペイロードのチェックサムを計算します。このエンジンでは、チェックサムの計算に、TCP、UDP または ICMPv6 擬似ヘッダバイトも含まれ、受信したチェックサムフィールドの値が計算値に一致するかどうかをチェックします。この操作の結果は、受信ステータスワードのペイロードチェックサムエラービットとして与えられます。このステータスビットは、TCP、UDP、または ICMP ペイロードが IP ヘッダで示された期待されたペイロード長に一致しない場合にもセットされます。

[表 640: 異なるパケットタイプの受信チェックサムオフロードエンジンの機能](#) では、パケットタイプに基づいて Rx COE によってサポートされる機能を説明します。IP パケットのペイロードが処理されない（表で「いいえ」と示されている）場合、その情報（チェックサムエンジンがバイパスされているかにかかわらず）は受信ステータスで付与されます。

注 : MAC は、受信した Ethernet パケットにペイロードチェックサムバイトを付加しません。

表 640. 異なるパケットタイプの受信チェックサムオフロードエンジンの機能

パケットタイプ	ハードウェア IP ヘッダ チェックサムチェック	ハードウェア TCP/UDP/ICMP チェックサムチェック
非 IPv4 または非 IPv6	不可	いいえ
IPv4 パケットが 1522 バイトより大きい (MAC で 2K パケット用の IEEE 802.3ad サポートが有効になっている場合は 2000 バイト)	可能	はい
IPv6 パケットが 1522 バイトより大きい (MAC で 2K パケット用の IEEE 802.3ad サポートが有効になっている場合は 2000 バイト)	適用外	可能
TCP、UDP、または ICMP での IPv4	可能	はい
TCP、UDP、または ICMP 以外のプロトコルを含む IPv4 ヘッダのプロトコルフィールド	可能	不可
IP オプションがある IPv4 パケット (IP ヘッダが 20 バイトを超える)	可能	はい
IPv4 の断片のパケット	可能	不可
メインまたは拡張ヘッダで次のヘッダフィールドを持つ IPv6 パケット : – ホップバイホップオプション (IPv6 メインヘッダ) – ホップバイホップオプション (IPv6 拡張ヘッダ) – 転送先オプション – ルーティング (残りセグメント 0) – ルーティング (残りセグメント 1 以上) – TCP、UDP、または ICMP – メインまたは拡張ヘッダでのその他の次のヘッダフィールド	– 適用外 – 適用外 – 適用外 – 適用外 – 適用外 – 適用外 – 適用外	– 可能 – 不可 – 可能 – 可能 – 不可 – 可能 – 不可
オプションフィールドのある TCP ヘッダを持つ IPv4 パケット	可能	はい
IPv4 トンネル : – IPv4 トンネルでの IPv4 パケット – IPv4 トンネルでの IPv6 パケット	– はい (IPv4 トンネルヘッダ) – はい (IPv4 トンネルヘッダ)	– 不可 – いいえ
IPv6 トンネル : – IPv6 トンネルでの IPv4 パケット – IPv6 トンネルでの IPv6 パケット	– 適用外 – 適用外	– 不可 – いいえ
802.3ac タグのある IPv4 パケット (有効時 C-VLAN タグまたは S-VLAN タグ付き)	可能	はい
802.3ac タグのある IPv6 パケット (有効時 C-VLAN タグまたは S-VLAN タグ付き)	適用外	可能
セキュリティ機能のある IPv4 フレーム (カプセル化されたセキュリティペイロードなど)	可能	不可
セキュリティ機能のある IPv6 フレーム (カプセル化されたセキュリティペイロードなど)	適用外	不可

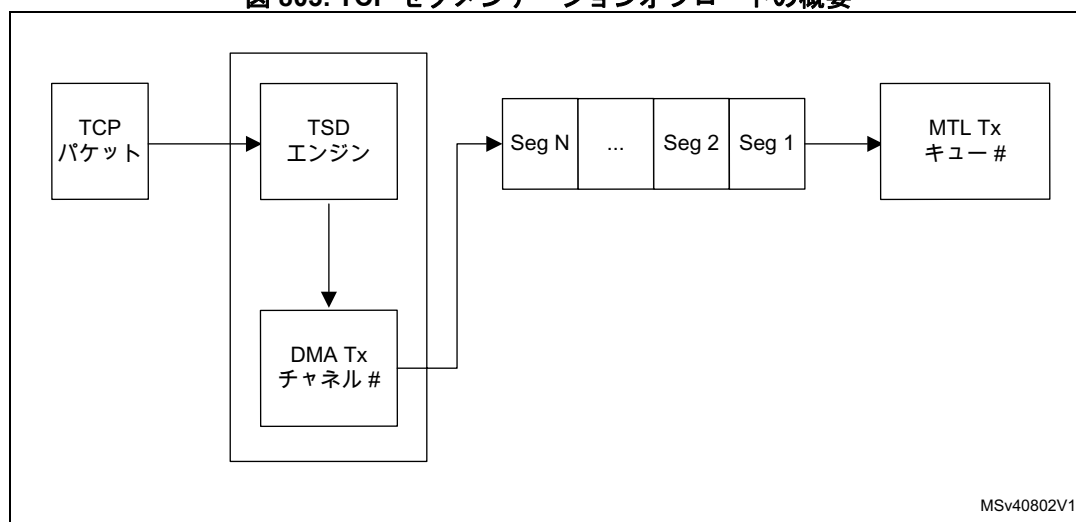
57.5.6 TCP セグメンテーションオフロード

MAC は、DMA が大きな TCP パケットを複数の小さいパケットに分割して、これらのパケットを [図 803](#)に示された MTL に渡す TCP セグメンテーションオフロード (TSO) 機能をサポートします。

この機能は、対応する ETH_DMCCR レジスタの TSE ビットをプログラムすることで有効になります ([チャンネル送信制御レジスタ \(ETH_DMACTXCR\)](#) を参照)。これは、MAC が全二重モードで動作している場合のみサポートされます。

詳細なプログラミング手順については、 [セクション 57.9.13 : TSO に関するプログラミングガイドライン](#) を参照してください。

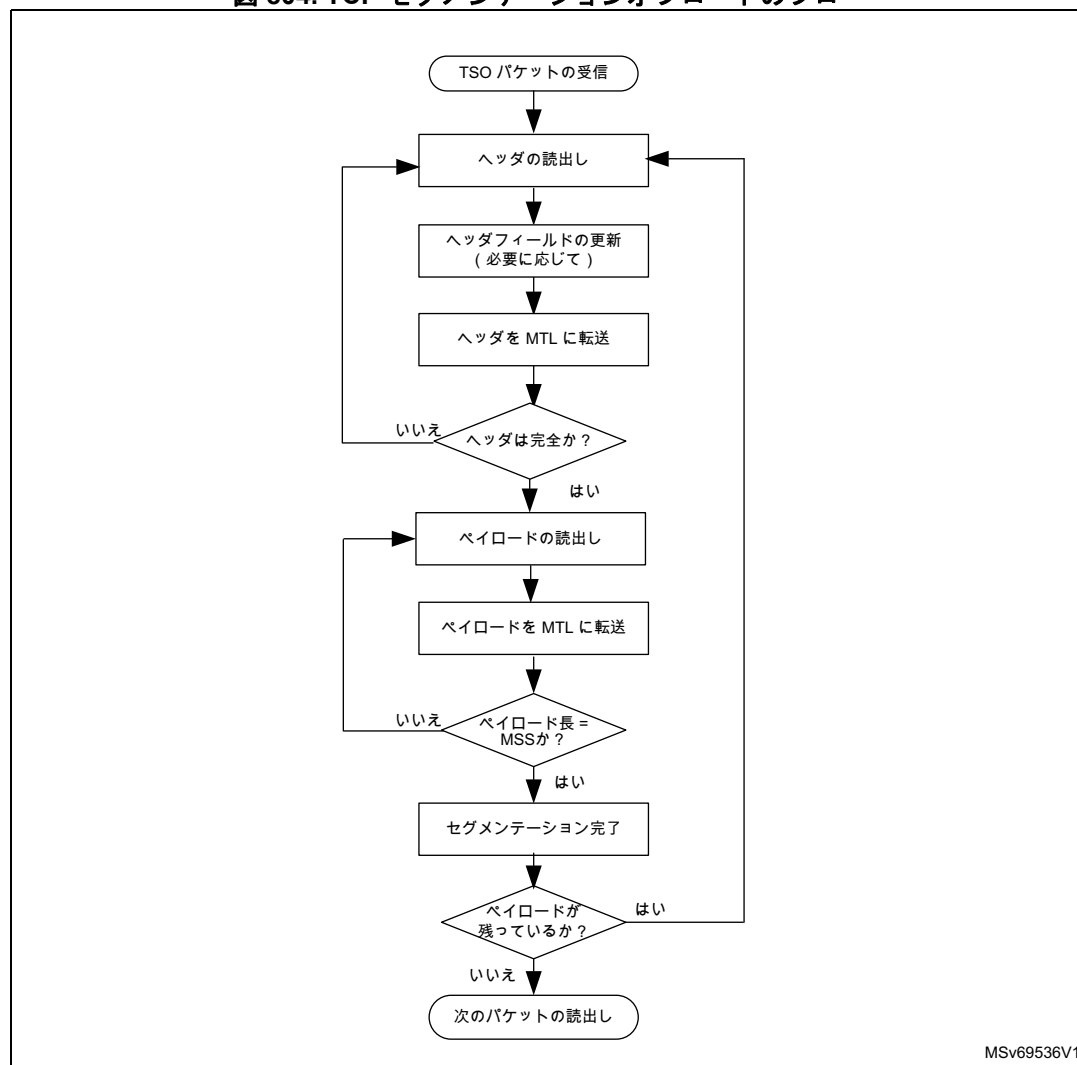
図 803. TCP セグメンテーションオフロードの概要



TSO 機能による DMA 動作

図 804 に TSO のフローを示します。

図 804. TCP セグメンテーションオフロードのフロー



TSO 機能の場合、Tx DMA 動作は次のとおりです。

1. アプリケーションは、送信ディスクリプタ (TDES0 ~ TDES3) をセットアップし、対応するデータバッファを Ethernet パケットデータでセットアップした後、Own ビット (TDES3[31]) をセットします。
2. アプリケーションにより、DMA Tx チャンネルのディスクリプタテイルポインタのオフセット値が加算されます。
3. 実行状態では、DMA は次に使用可能なディスクリプタをフェッチし、次のいずれかのアクションを実行します。
 - ディスクリプタがコンテキストディスクリプタで、コンテキストがパケットの最初のディスクリプタと最後のディスクリプタの間でない場合、DMA はコンテキスト値を保存します。

- ディスクリプタがコンテキストディスクリプタで、コンテキストがパケットの最初のディスクリプタと最後のディスクリプタの間にある場合、DMA はコンテキストディスクリプタエラー (TDES3[23]) を示してコンテキストディスクリプタをクローズし、次のディスクリプタをフェッチします。
 - ディスクリプタが通常ディスクリプタの場合、DMA は TSE ビットをチェックします。TSE ビットがセットされていない場合、DMA はデフォルトモードの動作を続行するか、(有効化されている場合は) OSF 動作を続行します。
4. DMA は TCP ペイロード長 (TDES3[17:0]) と MSS 値からセグメント数を算出します。
 5. DMA はチャネルアービトレーションを通してデータバッファをフェッチします。DMA はヘッダとペイロードを個別にフェッチします。
 6. 最初のセグメントの場合、DMA はシステムメモリからヘッダをフェッチし、TSO メモリに保存します (存在し、ヘッダの長さが TSO のメモリサイズ以下である場合)。分割された現在のパケットが最初のセグメントでない場合、最初のセグメントの場合と同様に、システムメモリ内でヘッダバッファを再度フェッチします。そのような場合、最後のセグメントのヘッダがフェッチされるまで、DMA はヘッダバッファが含まれる最初のディスクリプタをクローズしません。
 7. セグメンテーション要件に従ってヘッダバイト内の必要なフィールドが修正/更新され、対応する MTL Tx キューに書き込まれます。
 8. そして、DMA はペイロードバッファポインタを取得し、システムメモリから MSS のペイロードバイト数をフェッチし、MTL Tx キューに直接プッシュします。ディスクリプタ内のバッファに MSS ペイロード用の十分なデータがない場合 (最後のセグメントを除いて)、DMA はディスクリプタをクローズします。
 9. DMA はステップ 3 にジャンプして、最後のセグメントが Tx キューに書き込まれるまで、プロセスを繰り返します。
 10. DMA は最後のディスクリプタと最初のディスクリプタ (TSO メモリに保存されていない場合はヘッダバッファが格納されている) をクローズして、次のパケットの転送に進みます。

別のディスクリプタもある場合、DMA はこれらのすべてのステップを繰り返します。別のディスクリプタがない場合、DMA はサスペンド状態に移行します。

注 : TSO エンジンでは、パケット用の最初の通常 Tx ディスクリプタの TDES3 内の THL フィールド (TCP ヘッダ長) に基づいて、TSO または USO 動作を実行するかどうかを決定します。2 の値は USO を示し、5 以上の値は TSO を示します。

TCP/IP ヘッダフィールド

TCP パケットを分割しつつ、DMA は TCP/IP ヘッダフィールドを自動的に更新します。表 641 では、TCP ヘッダおよび IP ヘッダがどのように更新されるかを示しています。

表 641. TSO : TCP ヘッダおよび IP ヘッダフィールド

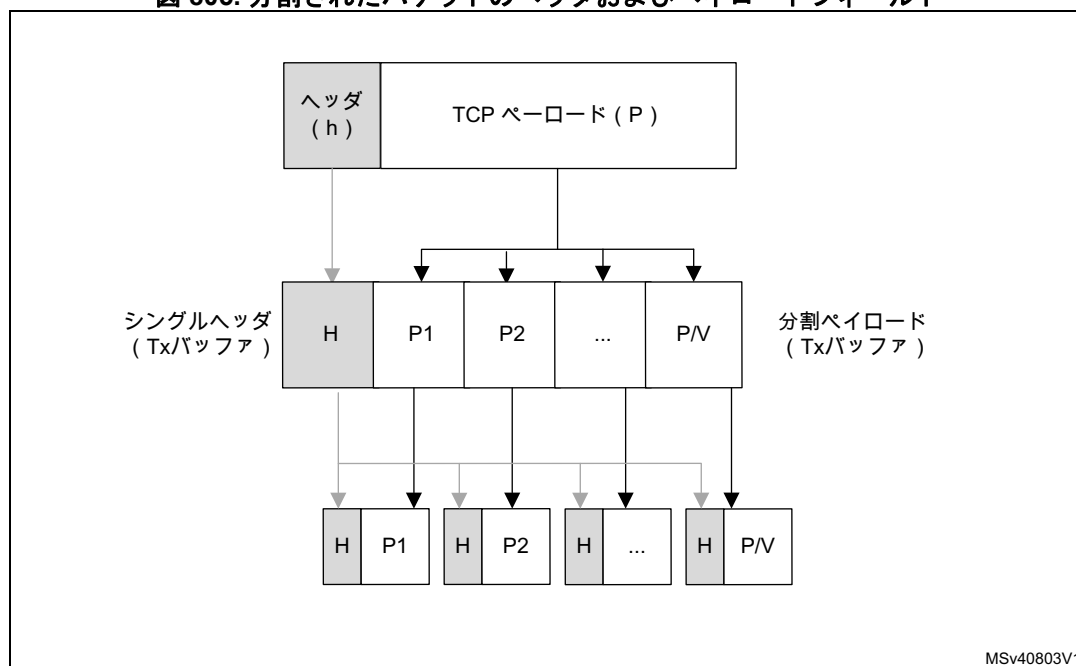
パケットシーケンス	TCP ヘッダ	IP ヘッダ
最初のパケット	<ol style="list-style-type: none"> 1. シーケンス数は更新されません。ヘッダで提供された値が使用されます。 2. セットしている場合、FIN フラグおよび PSH フラグがクリアされます。 3. TCP チェックサムが再計算されます。 	<p>IPv4 ヘッダ</p> <ul style="list-style-type: none"> - 合計の長さ = MSS + TCP ヘッダ長 + IP ヘッダ長です。 - 識別フィールドは変更されません。ソフトウェアによって提供されるヘッダに応じて送信されます。 - IPv4 ヘッダチェックサムが再計算されます。 <p>IPv6 ヘッダ</p> <ul style="list-style-type: none"> - ペイロード長 = MSS + TCP ヘッダ長 + IP 拡張ヘッダ長です。
後続のパケット	<ol style="list-style-type: none"> 1. シーケンス数は更新されます。MSS 値が、前のセグメントのシーケンス数値に追加されます。 2. セットしている場合、FIN フラグおよび PSH フラグがクリアされます。 3. TCP チェックサムが再計算されます。 	<p>IPv4 ヘッダ</p> <ul style="list-style-type: none"> - 合計の長さ = MSS + TCP ヘッダ長 + IP ヘッダ長です。 - 識別フィールド = 前の識別フィールド + 1 - IPv4 ヘッダチェックサムが再計算されます。 <p>IPv6 ヘッダ</p> <ul style="list-style-type: none"> - ペイロード長 = MSS + TCP ヘッダ長 + IP 拡張ヘッダ長です。
最後のパケット	<ol style="list-style-type: none"> 1. シーケンス数は更新されます。MSS 値が、前のセグメントのシーケンス数値に追加されます。 2. FIN フラグおよび PSH フラグが元のヘッダでセットされた場合、これらのフラグがセットされます。 3. TCP チェックサムが再計算されます。 	<p>IPv4 ヘッダ</p> <ul style="list-style-type: none"> - 合計の長さ = 残りのペイロード + TCP ヘッダ長 + IP ヘッダ長です。 - 識別フィールド = 前の識別フィールド + 1 - IPv4 ヘッダチェックサムが再計算されます。 <p>IPv6 ヘッダ</p> <ul style="list-style-type: none"> - ペイロード長 = 残りのペイロード長 + TCP ヘッダ長 + IP ヘッダ長です。

分割されたパケットのヘッダおよびペイロードフィールド

分割後、分割されたパケットは [表 641 : TSO : TCP ヘッダおよび IP ヘッダフィールド](#) に示されたものの以外のヘッダフィールドに親 TCP パケットと同じヘッダを使用します。[図 805: 分割されたパケットのヘッダおよびペイロードフィールド](#) では、分割されたパケットのヘッダフィールドに同じヘッダがどのように使用されるかを示しています。

アプリケーションは、分割されるパケットの先頭ディスクリプタのバッファ 1 でヘッダを作成し、先頭ディスクリプタ (FD = 1) の TDES2 にヘッダの長さを提供します。FD ビットがセットされている場合、DMA は TDES0 が指定しているヘッダバッファからヘッダを読み出します。先頭ディスクリプタのバッファ 2 は、後続のディスクリプタのペイロードおよび TDES0 および TDES1 に使用できます。後続のディスクリプタ (FD = 0) では、TDES0 および TDES1 が指定しているアドレスは同じパケットのペイロードバッファアドレスとして扱われます。

図 805. 分割されたパケットのヘッダおよびペイロードフィールド



コンテキストディスクリプタシーケンス

コンテキストディスクリプタは、セグメンテーションの最大セグメントサイズ (MSS) を提供できます。アプリケーションは、対応する TCP パケットに使用する通常ディスクリプタの前に、コンテキストディスクリプタを提供しなければなりません。アプリケーションが新しい MSS を提供する場合、新しい MSS 値で分割するパケットの最初の通常ディスクリプタの前に、ディスクリプタリストでコンテキストディスクリプタを作成する必要があります。コンテキストディスクリプタの MSS 値は、コンテキストディスクリプタの TDES3 の TCMSSV ビットがセットされ、OSTC ビットがリセットされている場合 ([セクション 57.10.3 : 送信ディスクリプタ](#) を参照) のみ有効です。

アプリケーションが有効な MSS 値を持つコンテキストディスクリプタを提供すると、DMA は MSS 値を内部で保存し、TDES3 通常ディスクリプタの TSE ビットを通じて TSO が有効になっているすべての後続のパケットにこの値を使用します。

アプリケーションがパケットの中間（パケットの先頭と最終ディスクリプタの間）にコンテキストディスクリプタを配置する場合、DMA は次の内容を実行します。

1. DMA がコンテキストを無視して、ディスクリプタをクローズします。
2. DMA がディスクリプタステータスでエラーを示します。
3. CDEE ビットが DMA チャンネルに対応する割込み有効レジスタでセットされている場合（[チャンネル割込み有効レジスタ \(ETH_DMACIER\)](#) を参照）、DMA は割込みを生成します。

アプリケーションは、DMA チャンネルに対応するステータスレジスタの CDE ビットを通じて割込みステータスを読み出せます（[チャンネルステータスレジスタ \(ETH_DMACSR\)](#) を参照）。

TSO 機能のためのディスクリプタとパケットの構築

パケットに対してセグメンテーションを有効にするために、アプリケーションは最初の通常ディスクリプタの TDES3 の TSE ビットをセットする必要があります（[セクション 57.10.3: 送信ディスクリプタ](#) を参照）。非 TCP-IP パケットに対して TDES3 で TSE ビットがセットされている場合、DMA の動作は予想できません。

アプリケーションは、TDES3 [17 : 0] に TCP パケットペイロードの長さを、TDES3 [22:19] に TCP ヘッダーをプログラムする必要があります。分割できる TCP パケットペイロードの最大長は 256 KB です。

Ethernet ヘッダ、L3 ヘッダ、L4 ヘッダを含むパケットのヘッダは、TSO パケットの最初の通常ディスクリプタのバッファ 1 で提供される必要があります。TSO の有効化されるパケットの最初の通常ディスクリプタのバッファ 1 のみ、ヘッダを含むバッファとして取得されます。

TCP ペイロードは、最初の通常ディスクリプタのバッファ 2 から開始し、2 番目の通常ディスクリプタのバッファ 1、バッファ 2、そして後続のディスクリプタへと続きます。

TCP ペイロードは、複数のバッファおよび複数のディスクリプタにまたがることができます。TCP ペイロードを含むバッファのサイズは、最初の通常ディスクリプタの TDES3[17:0] で提供される TCP ペイロード長と等しくなるまで追加する必要があります。

MAC は、DMA によって分割されたすべてのパケットに対して、必ず CRC を計算して付加し、パディング（必要な場合）を挿入します。TDES3 の TSE ビットが有効である場合、TDES3 の CRC PAD 制御（CPC）フィールドは予約済みになります。分割後の TCP パケットのサイズを決定するために、DMA はコンテキストディスクリプタを通じてアプリケーションが提供する最大セグメントサイズ（MSS）を使用します。DMA は、MSS を超えるペイロードサイズを持つパケットのみ分割します。アプリケーションは、ETH_DMCCR（[チャンネル制御レジスタ \(ETH_DMCCR\)](#) を参照）の MSS 値をプログラムするか、コンテキストディスクリプタを提供することによって MSS を提供する必要があります。DMA は、MSS の最後にプログラムされた値か、コンテキストを通じて提供される最後の MSS 値（後から提供されるどの値でも）を使用します。

ヘッダの長さ + MSS サイズ（各 TCP セグメントのサイズと同等）は 16383 バイトを超えてはなりません。そうしないと、MAC トランスミッタは CRC エラーを引き起こす 16383 バイト以降のパケットを切捨てます。

ヘッダの長さ + MSS サイズ + ETH_DMACTXCR レジスタ（[チャンネル送信制御レジスタ \(ETH_DMACTXCR\)](#) を参照）でプログラムされた PBL 値は、ETH_MTLTXQOMR レジスタ（[Tx キュー動作モードレジスタ \(ETH_MTLTXQOMR\)](#) を参照）の TQS フィールドでプログラムされた Tx キューサイズ未満にする必要があります。MSS とプログラムされた Tx キューサイズの半分に相当するヘッダが推奨されます。

DMA は、VLAN タグ付き TCP/IP フレームのセグメンテーションもサポートしています。TCP パケットに VLAN タグがある場合、提供される VLAN タグタイプ（C-VLAN または S-VLAN）に関係なく、すべてのセグメントに対して同じタグが使用されます。VLAN タグ挿入／書き換え制御ビットは、すべてのセグメントに対して使用されます。

ダブル VLAN 機能を選択している場合、提供される VLAN タグタイプ (C-VLAN または S-VLAN) に関係なく、すべてのセグメントに対して DMA が両方のタグを通過します。両方のタグの VLAN タグ挿入／書き換え制御ビットは、すべてのセグメントに対して適用できます。ダブル VLAN 機能を選択していない場合、アプリケーションは 2 つのタグを持つ TCP/IP パケットに対して TDES3 で TSE ビットをセットしてはなりません。このシナリオでの DMA の動作は予想できません。

TDES3 でパケット用に TSE ビットがセットされていて、提供されている TCP のヘッダ長が 5 未満の場合 (20 バイト未満であるため、無効な TCP ヘッダであることを意味します)、DMA はセグメンテーションを実行せず、代わりにパケット全体を単一パケットとして送信します。このシナリオでは、CRC パッド制御ビットは DMA によって強制的に 00 がセットされ (MAC が CRC およびパディングを実行)、チェックサム挿入制御ビットには強制的に 11 がセットされます (ヘッダとペイロードの両方に対して、ハードウェアでチェックサムの計算を実行)。

57.5.7 IPv4 ARP オフロード

MAC は、IPv4 パケットのアドレス認識プロトコル (ARP) オフロードをサポートします。この機能は、受信パスで IPv4 ARP リクエストパケットを処理して、送信パスで対応する ARP レスポンスパケットを生成できます。

MAC は、適切な ARP リクエストパケットに対して ARP 応答パケットを生成します。IPv4 用 ARP パケットは、0x0806 の長さ／タイプを持つ L2 レイヤパケットです。

ARP オフロードシーケンスは、次のとおりです。

1. リクエストターゲットプロトコルアドレスが MAC L3 レジスタでプログラムされた IPv4 アドレスと一致する場合、MAC レシーバが ARP リクエストを取得します。
2. MAC は、ARP 応答パケットを生成します。
3. MAC は、ARP リクエストの送信側ハードウェアアドレスフィールドを次のフィールドにコピーします。
 - Ethernet パケットヘッダの DA フィールド
 - ARP 応答パケットのターゲットハードウェアアドレスフィールド
4. MAC は、ARP リクエストの送信側プロトコルアドレスフィールドを ARP 応答パケットのターゲットプロトコルアドレスフィールドにコピーします。
5. MAC は、次のフィールドに MAC アドレスを配置します。
 - Ethernet パケットヘッダの SA フィールド
 - ARP 応答パケットの送信側ハードウェアアドレスフィールド
6. MAC は、ARP リクエストのターゲットプロトコルアドレスフィールドを ARP 応答パケットの送信側プロトコルアドレスフィールドにコピーします。
7. MAC は、ARP 応答パケットの OP コードフィールドを ARP 応答を示す 2 にセットします。
8. MAC は、生成された ARP 応答パケットに対して CRC を再計算し、パディングを実行します。
9. MAC トランスミッタは ARP 応答を送信します。

MAC で処理できるのは、一度に 1 つの ARP リクエストのみです。複数の ARP リクエストのフィールドは保存されません。ARP リクエストの処理中に別の ARP リクエストを受信した場合、MAC では新しい ARP リクエストに対する ARP 応答を生成しません。MAC は、「ARP 応答の生成なし」ステータスビット (ビット 34) がセットされた状態で、新しい ARP リクエストパケットをアプリケーションに転送します。ただし、パワーダウンモードでは、前の ARP リクエストの処理中に新しい ARP リクエストを受信すると、MAC では新しいリクエストをドロップします。[拡張動作モード設定レジスタ \(ETH_MACECR\)](#) の CRC チェックの無効化 (DCRCC) ビットがセットされている場合、MAC では ARP リクエストパケットの有効な CRC をチェックしません。別の条件が有効であれば、ARP レスポンスパケットを生成できます。ARP リクエストパケットは常に有効な CRC を備えていなければなりません。

注 : 受信した ARP リクエストが 64 バイトパケット長未満である場合、MAC では ARP レスポンスを送信しません。通常パケットとして処理され、MAC フィルタ設定に基づいてアプリケーションに転送されます。

57.5.8 ループバック

MAC は、送信したパケットのレシーバへのループバックをサポートします。

ループバックモードの使用に関するガイドライン

ループバックモードを使用するためのガイドラインは次のとおりです。

- ループバックは全二重モードでのみ有効化できます。半二重モードでは、パケットドロップなどの問題が発生する可能性のあるキャリアセンス信号または衝突信号入力のサンプルが取得されます。
- PHY チップを接続しないでループバックモードが有効化されている場合、Tx および Rx クロックを外部で生成し、これらのクロックを MAC に供給します。
- 大きなパケットはループバックしないでください。ループバック FIFO で破損する可能性があります。

送信クロックと受信クロックは、非同期のタイミングの関係とすることができます。そのため、非同期 FIFO を使用して、受信パスへの送信データのループバックパスが作成されます。FIFO はフリーランニングにより、書き込みクロック (eth_mii_tx_clk) で書き込み、すべての読み出しクロック (eth_mii_rx_clk) で読み出します。FIFO からの各パケット読み出しの開始時に、(10/100 Mbps モードで) オフセットが 4 になるように、書き込みおよび読み出しポインタが再初期化されます。これにより、パケット転送中のオーバーフローまたはアンダーフローが回避され、オーバーフローまたはアンダーフローがパケット間の IPG 周期中のみ発生するようになります。MII 送信クロックと受信クロックの周波数の差が 200 ppm、最大パケットサイズが 9022 バイトの場合、データの破損を回避するには FIFO の深度 5 または 9 で十分です。

そのため、大きなパケットは、このループバック FIFO で破損する可能性があるため、ループバックしないようにする必要があります。

各受信パケットの最後に、受信プロトコルエンジンモジュールにより、受信パケットステータスが生成され、受信パケットコントローラモジュールに送信されます。受信パケットコントローラモジュールで、制御、欠落パケット、フィルタ障害のステータスが受信ステータスに付加されます。MAC では、ループバックされる ARP または PMT パケットは処理しません。

ループバックモードの有効化

この機能を有効にするには、**動作モード設定レジスタ (ETH_MACCCR)** の LM ビットをプログラムします。ループバックはすべての PHY インタフェースに対して有効化できます。どの PHY インタフェースが選択されているかにかかわらず、データは常に 内部非同期 FIFO を通じて内部受信 MII インタフェースにループバックされます。

ループバックデータはまた、対応する送信 PHY インタフェースブロックを通じて Ethernet 回線に渡されます。

注 : ループバック中、データ/パケットは **mii_txd** 信号に反映されます。
横取りは、ループバックモードではサポートされていません。

57.5.9 フロー制御

送信フロー制御は、全二重モードでは送信しているポーズパケット、半二重モードではバックプレッシャを含め、リモートエンドからパケットのフローを制御します。ここでは、送信パスと受信パスのフロー制御について説明します。

全二重モードでフロー制御

全二重モードでは、MAC はフロー制御に IEEE 802.3x ポーズパケットを使用します。表 642 に、ポーズパケットのフィールドを示します。

表 642. ポーズパケットのフィールド

フィールド	説明
DA	特別なマルチキャストアドレスを含みます。
SA	MAC アドレス 0 を含みます。
タイプ	8808 を含みます。
MAC 制御 OP コード	IEEE 802.3x ポーズ制御パケット用に 0001 を含みます。
PT	Tx キューフロー制御レジスタ (ETH_MACQTXFCR) の PT フィールドで指定されたポーズ時間を含みます。

FCB ビットがセットされている場合、MAC は 1 つのポーズパケットを生成して送信します。ポーズパケット送信が完了した後に FCB が再びセットされた場合、ポーズ時間が完了しているかどうかにかかわらず、MAC は別のポーズパケットを送信します。ポーズ時間を延長したり、前に送信されたポーズパケットで指定された時間の前にポーズを終了するには、アプリケーションは、ポーズ時間レジスタを適切な値でプログラムしてから、FCB ビットを再びセットする必要があります。

半二重モードでのフロー制御

半二重モードでは、MAC はフロー制御に遅延メカニズム（バックプレッシャ）を使用します。アプリケーションがパケットの受信を停止するように要求すると、MAC は、パケットの受信を検出したときに 32 バイトのジャムパターンを送信します。ただし、送信フロー制御が有効な場合に限りです。これにより衝突が発生し、リモートステーションがバックオフします。アプリケーションがパケットの送信を要求した場合、バックプレッシャが有効なときでも、送信がスケジュールされ、実行されます。バックプレッシャが長時間有効なままの場合（および 16 を超える衝突イベントが連続して発生した場合）、リモートステーションは過剰衝突のために送信を中止します。

表 643 表 643 では、次のビットの設定に基づいた Tx パスでのフロー制御について説明します。

- Tx キューフロー制御レジスタ (ETH_MACQTXFCR) の TFE ビット
- 動作モード設定レジスタ (ETH_MACCCR) の DM ビット

フロー制御はすべてのキューで同様です。

表 643. Tx MAC フロー制御

TFE	DM	説明
0	X	MAC トランスミッタはフロー制御またはバックプレッシャの動作を実行しません。
1	0	Tx キューフロー制御レジスタ (ETH_MACQTXFCR) のビット 0 がセットされているとき、MAC トランスミッタはバックプレッシャを実行します。
1	1	Tx キューフロー制御レジスタ (ETH_MACQTXFCR) のビット 0 がセットされているとき、MAC トランスミッタはポーズパケットを送信します。

送信フロー制御

送信フロー制御は、Tx キューフロー制御レジスタ (ETH_MACQTXFCR) で TFE ビットがセットされているとき有効になります。

フロー制御トリガ

アプリケーションは、MAC にポーズパケットを送信するように要求するか、対応する Tx キューフロー制御レジスタ (ETH_MACQTXFCR) で FCB ビットをセットすることでバックプレッシャを開始できます。

受信フロー制御

受信パスでは、フロー制御は全二重モードでのみ機能します。ポーズパケットが半二重モードで受信される場合、パケットは通常の制御パケットとみなされます。

受信フロー制御の説明

表 644 では、次のビットの設定に基づいた Rx パスでのフロー制御について説明します。

- Rx フロー制御レジスタ (ETH_MACRXFCR) の RFE ビット
- 動作モード設定レジスタ (ETH_MACCR) の DM ビット

表 644. Rx MAC フロー制御

RFE	DM	説明
0	x	MAC レシーバは受信したポーズパケットを検出しません。
1	0	MAC レシーバは受信したポーズパケットを検出しますが、これらのパケットを制御パケットとして認識します。
1	1	MAC レシーバはポーズパケットを検出した処理したりして、MAC トランスミッタを停止することでこれらのパケットに応答します。

次のシーケンスでは、Rx フロー制御について説明します。

1. MAC は、以下のいずれかに対して、受信したポーズパケットのデスティネーションアドレス (DA) をチェックします。
 - マルチキャストデスティネーションアドレス : DA は 制御パケット用に指定された一意のマルチキャストアドレス (0x0180 C200 0001) に一致します。
 - ユニキャストデスティネーションアドレス : DA は MAC アドレス 0 レジスタ (MACアドレス 0 上位レジスタ (ETH_MACA0HR) とMACアドレス x 下位レジスタ (ETH_MACAxLR)) の内容に一致し、Rx フロー制御レジスタ (ETH_MACRXFCR) の UP ビットがセットされます。
UP ビットがセットされている場合、MAC は一意のマルチキャストアドレスのほか、ユニキャストデスティネーションアドレスを使用してポーズパケットを処理します。
2. MAC は受信したパケットの次のフィールドをデコードします。
 - タイプフィールド : このフィールドが 0x8808 かどうかチェックされます。
 - OP コードフィールド : このフィールドが 0x0001 かどうかチェックされます (ポーズパケット)。
 - ポーズ時間 : (ポーズパケットの) ポーズ時間をキャプチャして、トランスミッタをブロックする必要がある時間を決定します。
3. ステータスのバイトカウントが 64 バイトを示している場合と、CRC エラーがない場合、MAC トランスミッタは、デコードしたポーズ時間の値にスロット時間 (64 バイト時間) を掛けた時間の間、あらゆるデータパケットの送信を一時停止します。

前のポーズ時間が経過する前に次のポーズパケットが受信される場合、MAC は新しい値でポーズ時間を更新します。

受信フロー制御の有効化

Rx フロー制御レジスタ (ETH_MACRXFCR) の RFE ビットをセットして、ポーズフロー制御を有効化します。

57.5.10 MAC 管理カウンタ

ペリフェラルでは、アプリケーションを通じてアクセス可能なレジスタ内の送受信パケットについて、統計を保存することができます。

MAC 管理カウンタ (MMC) モジュールのカウンタは、CSR モジュールのレジスタアドレス空間の拡張とみなすことができます。MMC モジュールは、受信および送信パケットの統計を集めるために、一連のレジスタを維持します。このレジスタセットには、レジスタの動作を制御する制御レジスタ、生成された割込み (受信および送信) を含む 2 つの 32 ビットレジスタ、および割込みレジスタ (受信および送信) のマスクを含む 2 つの 32 ビットレジスタがあります。これらのレジスタは、CSR レジスタのアクセスと同じ方法で AHB スレーブインタフェースを通じてアプリケーションからアクセス可能です。これらのレジスタの構造を[セクション 57.11.4 : Ethernet MAC および MMC レジスタ](#)に示します。

MMC カウンタはフリーランニングです。カウンタを開始するための個別の有効化はありません。対応するパケットが受信または送信されると、特定の MMC カウンタがカウントを開始します。

受信 MMC カウンタは、アドレスフィルタ (AFM) ブロックによって渡されるパケット用に更新されます。AFM モジュールによってドロップされるパケットの統計は、6 バイト未満のラントパケットである (DA バイトが完全に受信されていない) 場合以外は更新されません。すべてのパケットの統計を取得するには、[パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) でビット 0 をセットします。MMC モジュールは、受信した Ethernet パケット内でカプセル化された IPv4、IPv6、TCP、UDP、ICMP ペイロードに関する統計を収集します。

制御レジスタに加えて、2 セットのレジスタが実装されています。

- 衝突、エラーおよび良好なパケットカウンタに対応して使用される 6 つのレジスタ :
 - Tx 単一衝突良好パケットレジスタ (Tx 単一衝突良好パケットレジスタ [\(ETH_TX_SINGLE_COLLISION_GOOD_PACKETS\)](#))
 - Tx 複数衝突良好パケットレジスタ (Tx 複数衝突良好パケットレジスタ [\(ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS\)](#))
 - Tx パケットカウント良好レジスタ (Tx パケットカウント良好レジスタ [\(ETH_TX_PACKET_COUNT_GOOD\)](#))
 - Rx CRC エラーパケットレジスタ (Rx CRC エラーパケットレジスタ [\(ETH_RX_CRC_ERROR_PACKETS\)](#))
 - Rx 整列エラーパケットレジスタ (Rx 整列エラーパケットレジスタ [\(ETH_RX_ALIGNMENT_ERROR_PACKETS\)](#))
 - Rx ユニキャストパケット良好レジスタ (Rx ユニキャストパケット良好レジスタ [\(ETH_RX_UNICAST_PACKETS_GOOD\)](#))
- LPI モード遷移を記録する 4 つのレジスタ :
 - Tx LPI マイクロ秒タイマレジスタ (Tx LPI マイクロ秒タイマレジスタ [\(ETH_TX_LPI_USEC_CNTR\)](#))
 - Tx LPI 遷移カウンタレジスタ (Tx LPI 遷移カウンタレジスタ [\(ETH_TX_LPI_TRAN_CNTR\)](#))
 - Rx LPI マイクロ秒カウンタレジスタ (Rx LPI マイクロ秒カウンタレジスタ [\(ETH_RX_LPI_USEC_CNTR\)](#))
 - Rx LPI 遷移カウンタレジスタ (Rx LPI 遷移カウンタレジスタ [\(ETH_RX_LPI_TRAN_CNTR\)](#))

定義

MMC レジスタの説明では、次の用語が使用されます。

- 送信されたパケットは、正常に送信された場合、「良好」とみなされます。言い換えると、送信されたパケットは、パケット送信が以下のいずれかのエラーによって中止されなければ、良好なパケットです。
 - ジャバタイムアウト
 - キャリアなしまたはキャリア喪失
 - レイトコリジョン
 - パケットのアンダーフロー
 - 過剰遅延
 - 過剰コリジョン
- 受信パケットは、以下のエラーが1つもなかった場合に「良好」とみなされます。
 - CRC エラー
 - ラントパケット (64 バイト未満)
 - アライメントエラー (10/100 Mbps のみ)
 - 長さエラー (非型パケットのみ)
 - 範囲外 (非型パケットのみ、1518 バイトより長い)
- 最大送信フレームサイズは、次のように、フレームのタイプに依存します。
 - タグなしフレームの最大サイズ = 1518
 - VLAN フレームの最大サイズ = 1522
 - ジャンボフレームの最大サイズ = 9018
 - JumboVLANフレームの最大サイズ = 9022
- 最大受信パケットサイズは、表 645 に示すように、パケットタイプと制御ビット (JE、S2KP、GPSLCE、および EDVLP) に依存します。

表 645. 最大受信パケットのサイズ

JE	S2KP	GPSLCE	EDVLP	タグなし フレームの 最大サイズ (バイト単位)	シングル VLAN フレームの 最大サイズ (バイト単位)	ダブル VLAN フレームの 最大サイズ (バイト単位)
1	X	X	1	9018	9022	9026
0	1	X	X	2000	2000	2000
0	0	1	1	GPSL	GPSL+4	GPSL+8
0	0	0	1	1518	1522	1526
1	X	X	0	9018	9022	9022
0	0	1	0	GPSL	GPSL+4	GPSL+4
0	0	0	0	1518	1522	1522

57.5.11 MAC によって生成される割込み

さまざまなイベントの結果として、MAC から割込みが生成されることがあります。これらの割込みイベントは、eth_sbd_intr_it 信号で DMA でのイベントと組み合わせられます。MAC 割込みはレベルタイプで、アプリケーションやソフトウェアによってクリアされるまで、割込みはアサート（ハイ）されたままになります。

[割込みステータスレジスタ \(ETH_MACISR\)](#) は、MAC からの割込みの原因となるイベントを記述します。MAC 割込みはデフォルトで有効にされます。[割込み有効レジスタ \(ETH_MACIER\)](#) の対応するマスクビットをセットすることによって、各イベントが eth_sbd_intr_it 信号で割込みをアサートしないようにすることができます。

割込みレジスタビットは、イベントが報告されるブロックを示すだけです。割込みをクリアするには、対応するステータスレジスタと他のレジスタを読み出す必要があります。

57.5.12 MAC および MMC レジスタの説明

[セクション 57.11.4 : Ethernet MAC および MMC レジスタ](#) を参照してください。

57.6 Ethernet機能の説明 : PHY インタフェース

Ethernet ペリフェラルは、複数の PHY インタフェースをサポートします。ルートインタフェースは MII インタフェースです。その他すべてのインタフェースは、図 806 に示すように、これから生成されます。

図 806. サポートされる PHY インタフェース

このセクションでは、PHY 制御インタフェースおよび異なる PHY インタフェースに使用される SMA モジュールについて説明します。以下のセクションが含まれています。

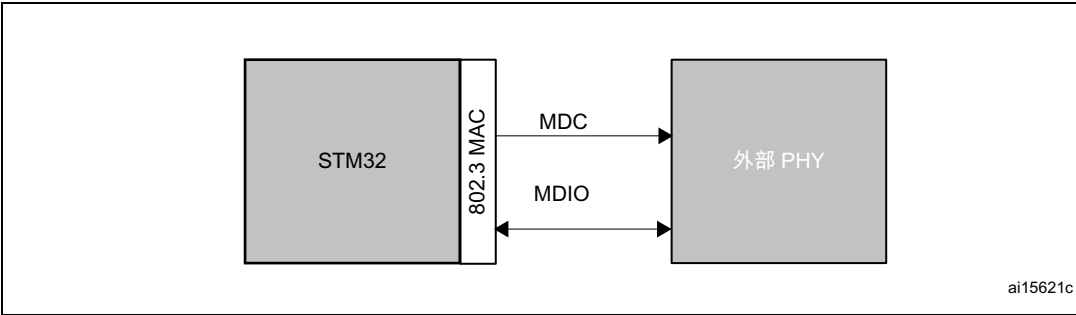
- ステーション管理エージェント (SMA)
- メディア独立インタフェース (MII)
- 減少メディア独立インタフェース (RMII)

57.6.1 ステーション管理エージェント (SMA)

アプリケーションは、ステーション管理エージェント (SMA) モジュールを通じて PHY レジスタにアクセスできます。SMA には、2 線ステーション管理インタフェース (MIM) が搭載されています。

SMA モジュールは、最大 32 の PHY へのアクセスをサポートします。アプリケーションは 32 の PHY の 32 個のレジスタのうち 1 個に対応できます。一度にアドレス指定できるのは、1 つの PHY 内の 1 つのレジスタだけです。図 807 に示すように、アプリケーションは PHY に制御データを送信して、SMA モジュールを通じて PHY からステータス情報を受信します。

図 807. SMA インタフェースブロック



SMA 機能概要

MAC は、MDC クロックに関する書き込みまたは読出し動作の管理を開始します。MDC クロックは、CSR クロックから生成されます (eth_hclk)。ETH_MDC ピンの最大動作周波数は、IEEE 802.3 仕様の規定に従い、2.5 MHz です。ただし、システムでこれより高いクロック周波数がサポートされている場合、別のドライバを選択することもできます。分周比は、MDIO アドレスレジスタ (ETH_MACMDIOAR) レジスタの CR[3:0] を通じて、クロック範囲設定に依存します。MDC クロックは次のように選択されます。

表 646. MCD クロックの選択

選択	eth_hclk	MDC クロック
0000	60~100 MHz	CSR クロック/42
0001	100~150 MHz	CSR クロック/62
0010	20~35 MHz	CSR クロック/16

表 646. MCD クロックの選択 (続き)

選択	eth_hclk	MDC クロック
0011	35~60 MHz	CSR クロック/26
0100	150~250 MHz	CSR クロック/102
0101	250~300 MHz	CSR クロック/124
0110, 0111	予約済みです。	-

MAC と PHY 間のデータ交換は、トライステートバッファを通じて行われ、PHY に接続された ETH_MDIO ラインとして出力されます。

図 808 は、Clause 45 MDIO パケットの構造を示し、表 647 ではパケットフィールドの詳細な説明を提示します。

図 808. MDIO のパケット構造 (Clause 45)

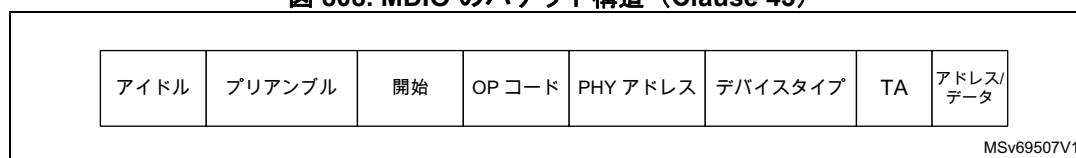


表 647. MDIO Clause 45 のフレーム構造

フィールド	説明
IDLE	ETH_MDIO ラインはトライステートです。ETH_MDC にクロックはありません。
PREAMBLE	値 1 の 32 個の連続ビット
START	パケットの開始は 0b00 です。
OPCODE	<ul style="list-style-type: none"> 0b00 : アドレス 0b01 : 書込み 0b10 : 読出し + アドレス 0b11 : 読出し
PHY ADDR	32 の PHY の 1 つに対して 5 ビットアドレス選択です。
DEV TYPE	5 ビットデバイスタイプ
TA	ターンアラウンド <ul style="list-style-type: none"> 0bZ0 : 読出しおよび事後読出しインクリメントアドレス 0b10 : 書込みおよびアドレス MDIO アクセス ここで、Z はトライステートレベルです
DATA	16 ビット値 : アドレスサイクル (OPCODE = 0b00) の場合、このフレームには、次のサイクルでアクセスされるレジスタのアドレスが格納されています。書込みフレームのデータサイクルの場合、このフィールドにはレジスタに書き込まれるデータが格納されています。読出しおよび事後読出しインクリメントアドレスフレームの場合、このフィールドには PHY から読み出されたレジスタの内容が格納されています。 <ul style="list-style-type: none"> アドレスおよびデータ書込みサイクルでは、これらの 16 ビットの転送中に、ペリフェラルにより ETH_MDIO ラインが駆動されます。 読出しおよび事後読出しインクリメントアドレスサイクルでは、これらの 16 ビットの転送中に、PHY により ETH_MDIO ラインが駆動されます。

Clause 22 フレームのフレーム構造もサポートされています。MDIO アドレスレジスタ (ETH_MACMDIOAR) の C45E ビットをプログラムし、Clause 22 または Clause 45 の動作モードを有効にできます。図 809 は、Clause 22 MDIO パケットの構造を示し、表 648 ではパケットフィールドの詳細な説明を提示します。

図 809. MDIO パケットの構造 (Clause 22)

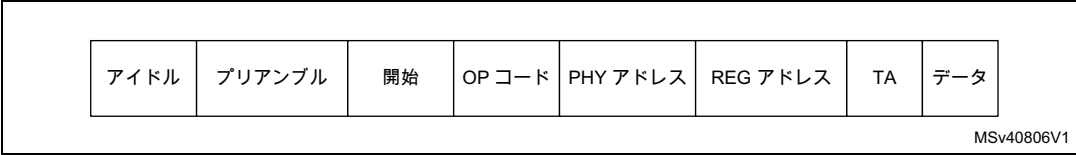


表 648. MDIO Clause 22 のフレーム構造

フィールド	説明
IDLE	ETH_MDIO ラインはトライステートです。ETH_MDC にクロックはありません。
PREAMBLE	値 1 の 32 個の連続ビット
START	パケットの開始は 0b01 です。
OPCODE	読出しの場合は 0b10、書込みの場合は 0b01 です。
PHY ADDR	32 の PHY の 1 つに対して 5 ビットアドレス選択です。
REG ADDR	選択された PHY でのレジスタアドレスです。
TA	ターンアラウンド – 0bZ0 : 読出しおよび事後読出しインクリメントアドレス – 0b10 : 書込みおよびアドレス MDIO アクセス ここで、Z はトライステートレベルです
DATA	任意の 16 ビット値です。書込み動作では、MAC が ETH_MDIO を駆動します。読出し動作では、PHY が駆動します。

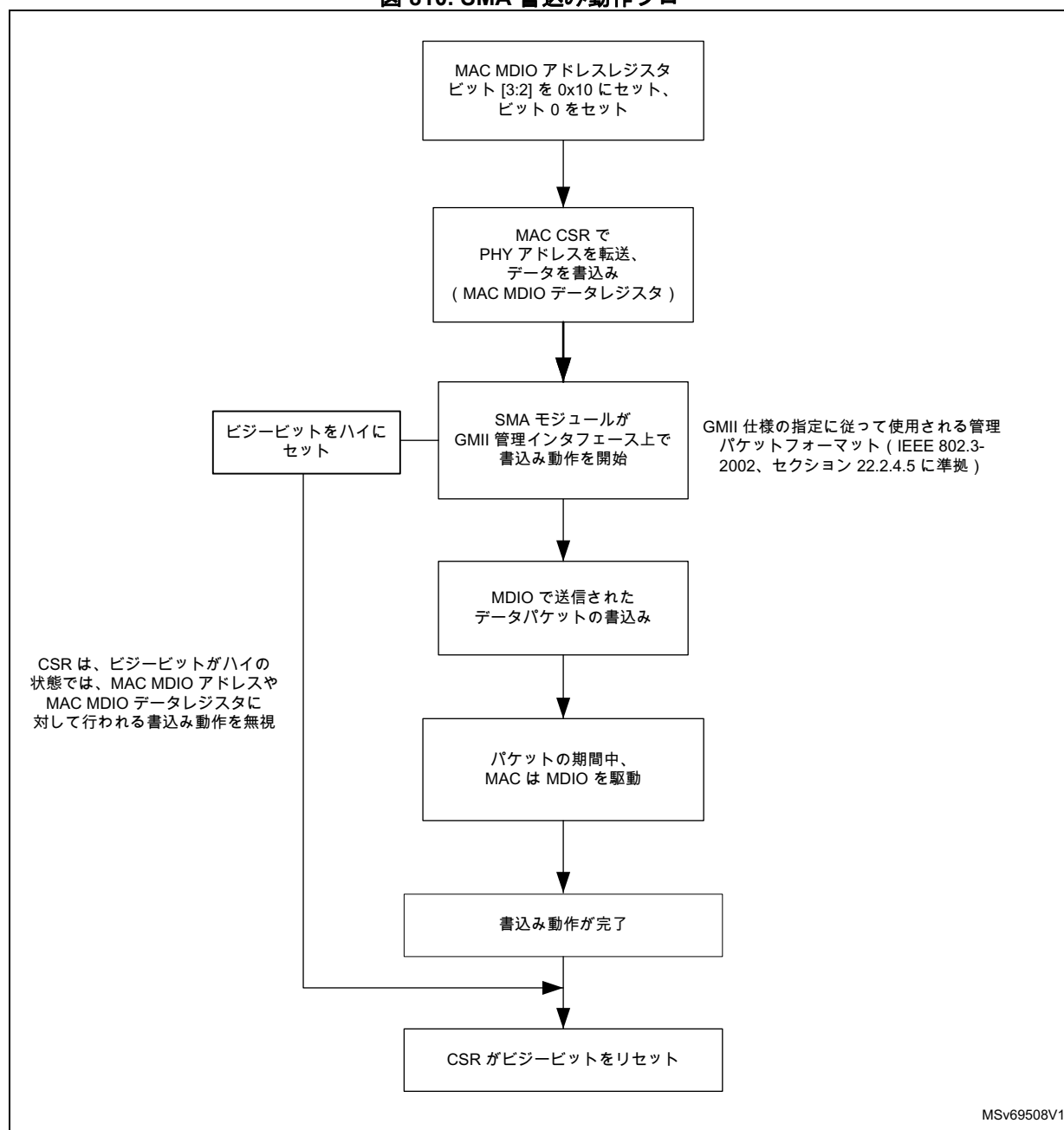
SMA では、通常の読出しおよび書込み動作のほか、Clause 45 モードでの動作中の事後読出しインクリメントアドレスもサポートしています。

MII 管理書込み動作

ステーション管理エージェントが PHY アドレスと書き込みデータを MAC CSR モジュールから受信した後、SMA は PHY レジスタへの書込み動作を開始します。

図 810 に SMA モジュールから PHY レジスタへの書込み動作のフローを示します。

図 810. SMA 書き込み動作フロー



MDIO アドレスレジスタ (ETH_MACMDIOAR) で、ビット [3:2] が 01 に、ビット 0 が 1 にセットされているとき、MAC CSR モジュールは PHY アドレス、PHY でのレジスタアドレス、書き込みデータ (MDIO データレジスタ (ETH_MACMDIODR)) を SMA に転送して、PHY レジスタへの書き込み動作を開始します。この時点で、SMA モジュールは MII 仕様 (IEEE 802.3-2002 仕様、セクション 22.2.4.5 に準拠) で指定された管理パケットフォーマットを使用して、MII 管理インタフェースで書き込み動作を開始します。

SMA モジュールが書き込み動作を開始したとき、書き込みデータパケットが MDIO ラインに送信されます。MAC はパケットの全送信時間中、MDIO ラインを駆動します。ビジービットは、書き込み動作が完了するまでハイにセットされます。この時間 (ビジービットがハイになっている時間) 中、CSR は、MDIO アドレスレジスタ (ETH_MACMDIOAR) または MDIO データレジスタ (ETH_MACMDIODR)

に実行される書き込み動作を無視します。書き込み動作が完了したとき、SMA モジュールは CSR に通知し、CSR はビジービットをリセットします。書き込み動作のパケットフォーマットは、以下のとおりです。

図 811. 書き込みデータパケット

アイドル	プリアンプル	開始	OP コード	PHY アドレス	REG アドレス	TA	データ	アイドル
Z	1111..11	01	01	AAAAA	RRRRR	10	DDD...DDD	Z

MSv40807V1

MII 管理読出し動作

[MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) で、ビット [3:2] が 11 に、ビット 0 が 1 にセットされているとき、MAC CSR モジュールは PHY アドレスおよび PHY でのレジスタアドレスを SMA に転送して、PHY レジスタでの読出し動作を開始します。この時点で、SMA モジュールは MII 仕様 (IEEE 802.3-2002 仕様、セクション 22.2.4.5 に準拠) で指定された管理パケットフォーマットを使用して、MII 管理インタフェースで読出し動作を開始します。

SMA モジュールが MDIO で読出し動作を開始すると、CSR は、この時間 (ビジービットがハイになっている時間) 中、[MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) または [MDIO データレジスタ \(ETH_MACMDIODR\)](#) レジスタへの書き込み動作を無視し、MCI インタフェースでエラーがない状態でトランザクションを完了します。読出し動作が完了したとき、SMA は CSR に通知します。CSR は、ビジービットをリセットし、PHY から読み出したデータで [MDIO データレジスタ \(ETH_MACMDIODR\)](#) を更新します。MAC は、PHY が MDIO ラインを駆動しているデータフィールド中を除くフレームの全時間中、MDIO ラインを駆動します。アプリケーションから PHY への通信に関する詳細については、IEEE 802.3z、1000BASE Ethernet の調整サブレイヤおよびメディア独立インタフェースの仕様のセクションを参照してください。

読出し動作のパケットフォーマットは、以下のとおりです。

図 812. 読出しデータパケット

アイドル	プリアンプル	開始	OP コード	PHY アドレス	REG アドレス	TA	データ	アイドル
Z	1111..11	01	10	AAAAA	RRRRR	Z0	DDD...DDD	Z

MSv40808V1

プリアンブル抑制

IEEE 標準では、MDIO フレーム用の 32 ビットのプリアンブル (すべて 1) を規定しています。ペリフェラルは、プリアンブル抑制をサポートするための制御を備えています。1 プリアンブルビットのみの MDIO フレームを送信します。プリアンブル抑制を有効化するには、[MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) で PSE ビットをセットします。

立下りクロックと連続トランザクション

ペリフェラルは、MDIO フレームの期間中、ETH_MDC クロックを駆動します。アイドル期間中に駆動されるクロックはありません。立下りクロック機能は、MDIO フレーム後の数サイクルの間、PHY で ETH_MDC クロックをアクティブにする必要がある場合に使用できます。[MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) の NTC[2:0] ビットフィールドでは、立下りクロックを 0 ~ 7 でプログラミングできます。

ペリフェラルでは、前の MDIO フレームに対して立下りクロックが完了する前でも次の MDIO フレームを開始できる連続トランザクションがサポートされています。この機能を有効にするには、立下りクロック機能も有効化された状態で、[MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) で BTB ビットをセットします。連続トランザクションが有効化された状態では、MDIO フレームが完了するとすぐに、GMII ビジービット (GB) がクリアされます。これによってソフトウェアで次のコマンドが発行されます。このコマンドは、前の MDIO フレームに対して立下りクロックがまだオンの間に、ペリフェラルによって実行されます。(GB) トランザクションが有効化されていない状態では、MDIO フレーム用の立下りクロックが完了した後に、GMII ビジービットがクリアされます。

MDIO トランザクション完了のための割込み

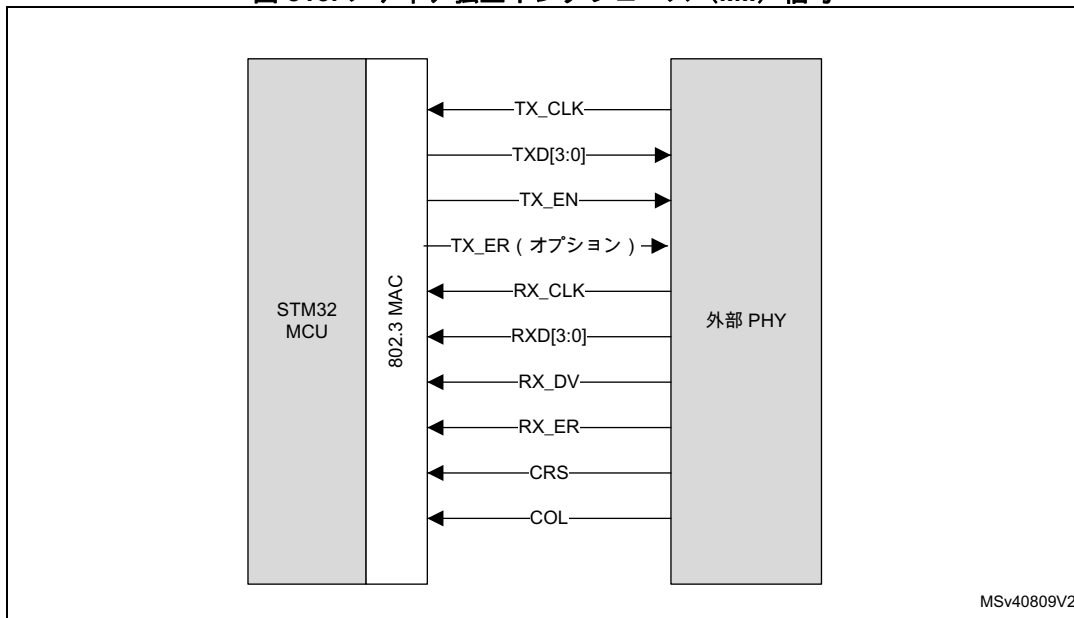
ペリフェラルでは、MDIO 読みまたは書き込みトランザクションの完了時に割込みを生成できます。そのため、MDIO コマンドの完了を知るために、アプリケーションで [MDIO アドレスレジスタ \(ETH_MACMDIOAR\)](#) の GMII ビジービットをポーリングする必要はありません。

57.6.2 メディア独立インタフェース (MII)

メディア独立インタフェース (MII) は、10 Mbit/s および 100 Mbit/s でのデータ転送における MAC サブレイヤと PHY 間の相互接続を定義します。

MII 信号を図 813 : メディア独立インタフェース (MII) 信号 に示します。

図 813. メディア独立インタフェース (MII) 信号



- **TX_CLK** : Tx データ転送の基準タイミングを供給する連続クロックです。公称周波数は、10 Mbit/s で 2.5 MHz で、100 Mbit/s で 25 MHz です。
- **TXD[3:0]** : 送信データです。
TXD は、MAC サブレイヤによって同期して駆動される 4 つのデータ信号の束であり、TX_EN 信号のアサーションによって有効なデータと認められます。TXD[0] は最下位ビットであり、TXD[3] が最上位ビットです。TX_EN がネゲートされている場合、送信データは PHY に何の影響も与えません。
- **TX_EN** : MAC が送信のために MII 上にニブルを提示していることを示す送信イネーブル信号です。プリアンプルの最初のニブル (TX_CLK) に同期してアサートされなければならず、送信されるすべてのニブルが MII に提示されるまで、アサート状態が維持されなければなりません。
- **TX_ER (オプション)** : 省電力型 Ethernet (EEE) でのみ必要です。送信エラーは CRC の反転により示されます。リモートステーションでは、不正 CRC を通して送信エラーを検出できます。
- **RX_CLK** : Rx データ転送の基準タイミングを供給する連続クロックです。公称周波数は、10 Mbit/s で 2.5 MHz で、100 Mbit/s で 25 MHz です。
- **RXD[3:0]** : 受信データ
RXD は、PHY によって同期して駆動される 4 つのデータ信号の束であり、RX_DV 信号のアサーションによって有効なデータと認められます。RXD[0] は最下位ビットであり、RXD[3] が最上位ビットです。RX_EN がネゲートされ、RX_ER がアサートされている間、PHY から特定の情報を転送するために特定の RXD[3:0] 値が使用されます。
- **RX_DV** : 受信データ有効です。
この信号は、PHY が MII 上に受信のために回収され、デコードされたニブルを提示していることを示します。フレームの最初の回収ニブルと同期してアサートされなければならず

(RX_CLK)、最後の回収ニブルまでアサート状態を維持しなければなりません。最後のニブル後の最初のクロックの前にネゲートされなければなりません。フレームを正しく受信するためには、RX_DV 信号は、SFD フィールドよりも遅れることなく開始して、フレームを取り込む必要があります。

- RX_ER : 受信エラーです。

この信号は、フレーム内のどこかでエラーが検出されたことを MAC サブレイヤに示すために、1 つ以上のクロック周期 (RX_CLK) の間、アサートされなければなりません。このエラー条件は、RX_DV ビットのアサートにより有効とされなければなりません。

- CRS : キャリアセンスです。

この信号は、送信または受信メディアがアイドルでないとき、PHY によってアサートされます。送信メディアと受信メディアの両方ともアイドルのときには、PHY によってネゲートされます。PHY は、衝突条件の全期間を通じて、CS 信号がアサートされたままであることを保証する必要があります。この信号は、Tx および Rx クロックに同期して遷移する必要はありません。全二重モードでは、この信号の状態は MAC サブレイヤに影響を与えません。

- COL : 衝突検出信号です。

メディア上で衝突が検出された場合、PHY によってこの信号がアサートされなければならず、衝突条件が続いている間、アサート状態が保たれなければなりません。この信号は、Tx および Rx クロックに同期して遷移する必要はありません。全二重モードでは、この信号の状態は MAC サブレイヤに影響を与えません。

57.6.3 減少メディア独立インタフェース (RMII)

減少メディア独立インタフェース (RMII) 仕様は、Ethernet PHY と STM32 マイクロコントローラのピン数を削減します。IEEE 802.3u に従って、MII はデータおよび制御のために 16 本のピンを備えています。RMII 仕様では、ピン数を 7 本に削減します。

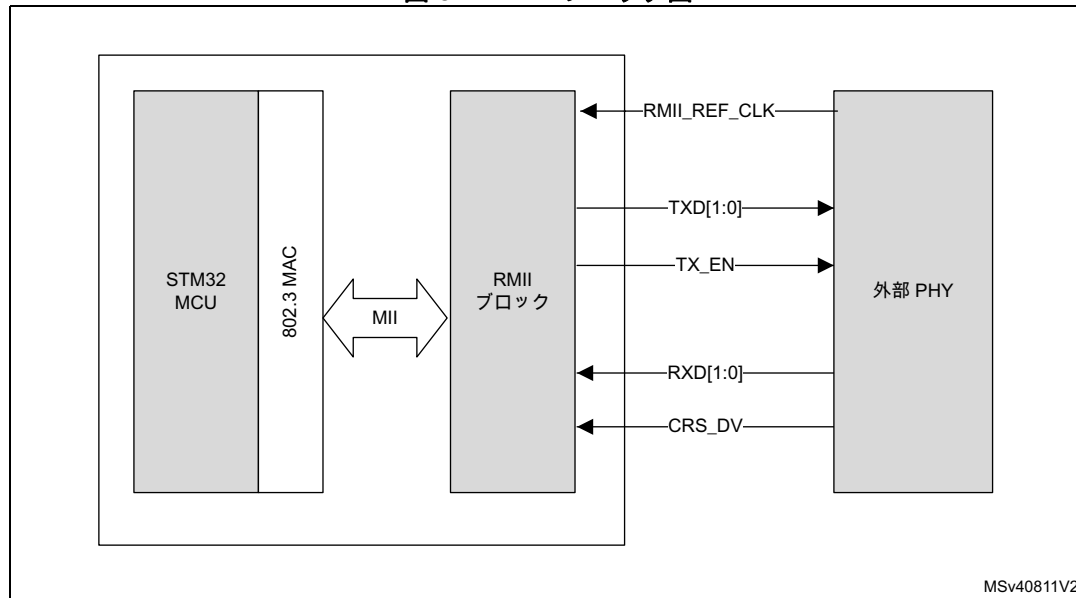
Ethernet ペリフェラルの一部、RMII モジュールは MAC 出力でインスタンス化されます。これは、MAC の MII を RMII に変換するのに役立ちます。RMII ブロックは、次のような特性を備えています。

- 10 Mbps および 100 Mbps の動作速度をサポートします。1000 Mbps 動作はサポートしていません。
- 2 つのクロック基準を外部から供給することで、独立した 2 ビット幅の送信および受信パスを提供します。

RMII ブロック図

図 814 : RMII ブロック図では、MAC と RMII PHY に関連する RMII ブロックの位置を示しています。RMII ブロックは MAC の前に配置され、MII 信号を RMII 信号に変換します。

図 814. RMII ブロック図

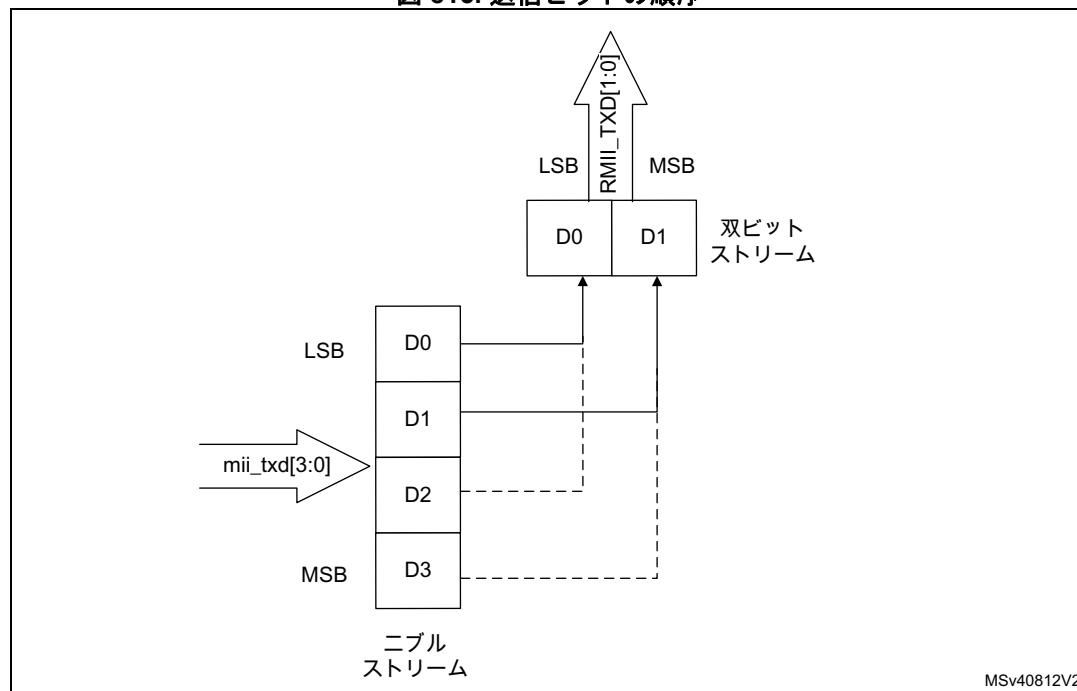


- RMII_REF_CLK : 50 MHz のリファレンスクロック連続入力
- TXD[1:0] : 送信データです
- TX_EN : 送信データが有効。
ハイのとき、このビットは TXD[1:0] で有効なデータが送信されていることを示します。
- RXD[1:0] : 受信データ
- CRS_DV : キャリアセンス (CRS) と RX データ有効 (RX DV) をオルタネートクロックサイクルで多重化。10 Mbit/s モードでは、10 クロックサイクルごとに切り替わります。

送信ビットの順序

MII インタフェースからの各ニブルは、RMII インタフェースで一度に 2 ビット（双ビット）ずつ、[図 815 : 送信ビットの順序](#)に示されている双ビット送信の順序で送信する必要があります。下位ビット（D1 と D0）が最初に送信され、次に上位ビット（D2 と D3）が送信されます。

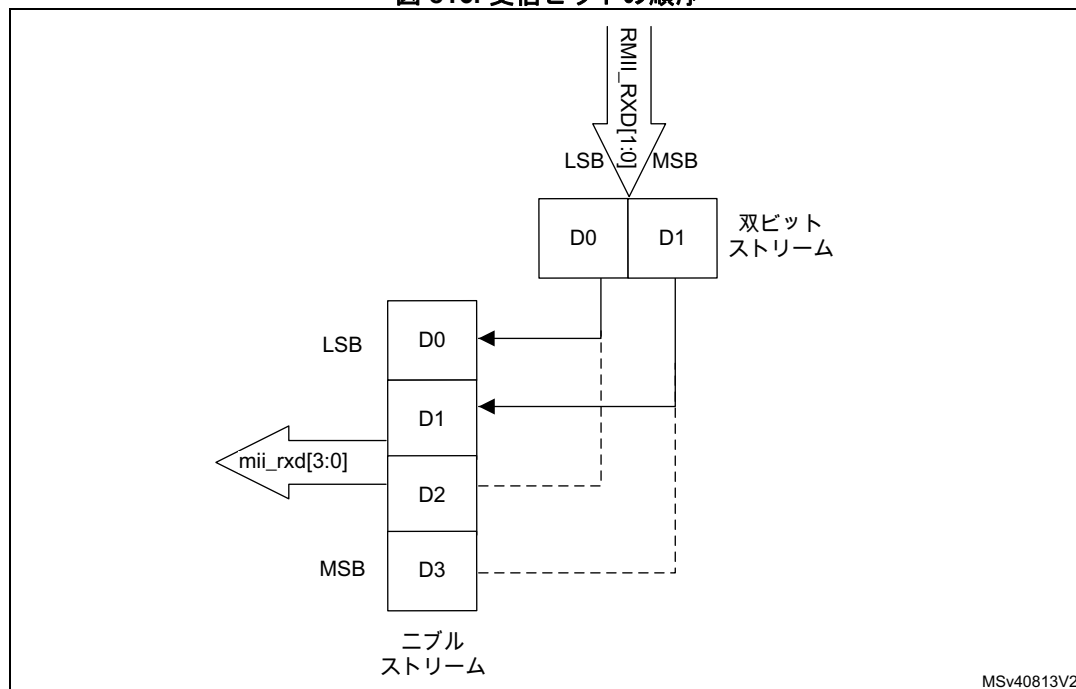
図 815. 送信ビットの順序



受信ビットの順序

各ニブルは、RMII インタフェースから受信した双ビットから MII インタフェースに、[図 816 : 受信ビットの順序](#)に示されているニブル送信順序で送信されます。下位ビット (D1 と D0) が最初に受信され、次に上位ビット (D2 と D3) が受信されます。

図 816. 受信ビットの順序



57.7 Ethernet 低電力モード

57.7.1 低電力管理

Ethernet ペリフェラルは、次の節電の方法に対応しています。

- マジックパケット
- リモートウェイクアップ

マジックパケットとリモートウェイクアップは、アイドル状態 (SLEEP モード) で、Ethernet ネットワークから特定の packets を受信したときのみウェイクアップする必要がある場合に、ホストシステムを節電するために使用されます。SLEEP モードでは、大部分のペリフェラルと一緒に、ホストロジックへの電力を停止することができます (MAC レシーバロジックを除く)。ネットワークからの特定の packets の受信時に、MAC はホストシステムへの電力を復旧し、通常状態に戻すためのトリガを供給します。

省電力型 Ethernet (EEE) モードは、IEEE 802.3az-2010 標準に準拠しています。主な目的は、ラインにトラフィックがない状態のときに、Ethernet ポートで節電することです。このモードでは、ホストは近い将来送信すべき packets がなく、トランスミッタポート (MAC コントローラ、PCS および PHY レイヤ) を低電力モードに切り替えることを遠端に通知します。同様に、レシーバポートでも、転送すべきトラフィックがないことを遠端のホストから通知された場合に、低電力モードに切り替えることができます。これにより、トラフィックプロファイルの中断やバーストのある Ethernet ポート (主に PHY) で大幅に電力を節約することができます。EEE モードの開始や終了のトリガは、MAC で制御され、ペリフェラル内でサポートされています。

EEE モードと、他の省電力モードの一方または両方の併用もサポートされています。

マジックパケットモードの説明

このセクションでは、マジックパケット検出を通じて節電する方法を説明します。

注 : マジックパケット機能は、Advanced Micro Device (AMD) のマジックパケットテクノロジーホワイトペーパーに基づいています。

マジックパケットのウォッチドッグタイムアウトの制限は、**動作モード設定レジスタ (ETH_MACCCR)** の WD ビットと **ウォッチドッグタイムアウトレジスタ (ETH_MACWTR)** の PWE ビットでプログラムされた値に関係なく、2048 バイトです。

マジックパケットベースの省電力モードでは、MAC レシーバで有効なマジックパケットを受信すると、低電力モードの終了がトリガされます。MAC は、**PMT 制御ステータスレジスタ (ETH_MACPCSR)** の PWRDWN ビットが 1 にプログラムされると、省電力モードに移行します。マジックパケットベースの省電力モードを終了するには、**PMT 制御ステータスレジスタ (ETH_MACPCSR)** の MGKPKTEN ビットを 1 にセットします。

マジックパケットは、デスティネーションアドレス、ソースアドレス、長さ/タイプフィールドの後の任意のオフセットに一意的なパターンが格納されています。独自のパターンマッチングのほか、MAC レシーバは受信パケットを有効なマジックパケットとして検出するために、以下についてもチェックします。

- パケットがこれにアドレス指定されている (受信パケットのデスティネーションアドレスは **MAC アドレス 0 上位レジスタ (ETH_MACA0HR)** および **MAC アドレス 0 下位レジスタ (ETH_MACA0LR)** に完全一致する必要があります) か、マルチキャスト/ブロードキャストアドレスによりアドレス指定されていること。
- パケットに 長さエラー、FCS エラー、ドリブルビットエラー、GMII エラー、衝突がないこと。
- ラントパケットでないこと (Ethernet ヘッダと FCS を含む長さは少なくとも 64 バイトです)。

マジックパケットのデータフォーマット

マジックパケットの独自のパターンの内容を以下に示します。

- 同期ストリームと呼ばれるすべてが 1 (0xFF FF FF FF FF FF) の 6 バイト。6 バイトを超える 0xFF が存在する場合がありますが、最後の 6 バイトのみが考慮されます。
- 同期ストリームの直後に、16 回反復されるパケットのデスティネーションアドレスフィールド (MAC アドレス 0 上位レジスタ (ETH_MACA0HR) および MAC アドレス 0 下位レジスタ (ETH_MACA0LR)) またはマルチキャスト/ブロードキャストアドレスが続きます。
- 同期ストリームとデスティネーションアドレスフィールドの最初の反復の間、または 16 回の反復内に中断や途切れはありません。

ノードの MAC アドレスが 0x00 11 22 33 44 55 である場合は、MAC は、次のデータシーケンスをスキャンします。

```

デスティネーションアドレス ソースアドレス 長さ/タイプ..... FF FF FF FF FF FF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
...CRC

```

リモートウェイクアップパケットモードの説明

このセクションでは、リモートウェイクアップパケットに基づく省電力モードについて説明します。

注 :

リモートウェイクアップパケット機能の実装は、デバイスクラスパワーマネージメントリファレンス仕様と実装固有の各種ホワイトペーパーに基づいて行われます。

マジックパケットのウォッチドッグタイムアウトの制限は、動作モード設定レジスタ (ETH_MACCR) の WD ビットとウォッチドッグタイムアウトレジスタ (ETH_MACWTR) の PWE ビットでプログラムされた値に関係なく、2048 バイトです。

リモートウェイクアップのマジックパケットベースの省電力モードでは、MAC レシーバで予測されるリモートウェイクアップパケットを受信すると、低電力モードの終了がトリガされます。MAC は、PMT 制御ステータスレジスタ (ETH_MACPCSR) の PWRDWN ビットが 1 にプログラムされると、省電力モードに移行します。リモートウェイクアップのマジックパケットベースの省電力モードの終了を有効にするには、PMT 制御ステータスレジスタ (ETH_MACPCSR) の RWKPKTEN ビットを 1 にプログラムします。

MAC はフィルタルックアップテーブル (リモートウェイクアップパケットフィルタレジスタ (ETH_MACRWKPFRR) を通じてプログラム) を実装します。ここで CRC、オフセット、パターンのバイトマスクがリモートウェイクアップパケットに組み込まれ、フィルタ動作コマンドがプログラムされます。

リモートウェイクアップパケットに組み込まれたパターンは、デスティネーションアドレスおよびソースアドレスフィールドの後の任意のオフセットに配置されます。パターンの CRC マッチングのほか、MAC レシーバは受信パケットを有効なリモートウェイクアップパケットとして検出するために、以下についてもチェックします。

- パケットがこれにアドレス指定されている (受信パケットのデスティネーションアドレスは MAC アドレス 0 上位レジスタ (ETH_MACA0HR) および MAC アドレス 0 下位レジスタ (ETH_MACA0LR) に完全一致する必要があります) か、マルチキャスト/ブロードキャストアドレスによりアドレス指定されていること。
- パケットに 長さエラー、FCS エラー、ドリブルビットエラー、GMII エラー、衝突がないこと。
- ラントパケットでないこと (Ethernet ヘッダと FCS を含む長さは少なくとも 64 バイトです)。

有効なリモートウェイクアップパケットを受信すると、MAC レシーバは [PMT 制御ステータスレジスタ \(ETH_MACPCSR\)](#) に RWKPRCVD ビットをセットし、pmt_intr_o 出力ポートで割込みをトリガします。[割込みステータスレジスタ \(ETH_MACISR\)](#) の PMTIS ビットは、低電力モードでパワーゲーティングが有効になっていない場合にセットされます。低電力モードで割込みが有効化されていて ([割込み有効レジスタ \(ETH_MACIER\)](#) で PMTIE ビットがセットされている)、CSR クロックがゲートオフされていない場合、sbd_intr_o 出力ポートでアプリケーションに対して割込みがトリガされます。

リモートウェイクアップパケットフィルタ

リモートウェイクアップベースの省電力モードが有効化されている場合、4 つのリモートウェイクアップフィルタを選択できます。リモートウェイクアップフィルタの構造を [表 649: リモートウェイクアップパケットフィルタレジスタ](#) に示します。

表 649. リモートウェイクアップパケットフィルタレジスタ

ETH_MACRWKPCR 値	フィールド							
0	フィルタ 0 バイトマスク							
1	フィルタ 1 バイトマスク							
2	フィルタ 2 バイトマスク							
3	フィルタ 3 バイトマスク							
4	予約済み です。	フィルタ 3 コマンド	予約済み です。	フィルタ 2 コマンド	予約済み です。	フィルタ 1 コマンド	予約済み です。	フィルタ 0 コマンド
5	フィルタ 3 オフセット		フィルタ 2 オフセット		フィルタ 1 オフセット		フィルタ 0 オフセット	
6	フィルタ 1 CRC - 16				フィルタ 0 CRC - 16			
7	フィルタ 3 CRC - 16				フィルタ 2 CRC - 16			

リモートウェイクアップフィルタフィールドについて、[表 650: リモートウェイクアップフィルタフィールドの説明](#)で説明します。

表 650. リモートウェイクアップフィルタフィールドの説明

レジスタ	説明
フィルタ <i>i</i> バイトマスク	<p>フィルタ <i>i</i> バイトマスクレジスタは、パケットがウェイクアップパケットであるかどうかを判定するために、フィルタ <i>i</i> (0、1、2、3) によって調べられるパケットのバイトを定義します。</p> <ul style="list-style-type: none"> MSB (31 番目のビット) は 0 でなければなりません。 ビット <i>j</i>[30:0] はバイトマスクです。 バイトマスクのビット <i>j</i> (バイト数) がセットされた場合、CRC ブロックが着信パケットのフィルタ <i>i</i> オフセット + <i>j</i> を処理します。そうでない場合、フィルタ <i>i</i> オフセット + <i>j</i> は無視されます。
フィルタ <i>i</i> コマンド	<p>この 4 ビットのフィルタ <i>i</i> コマンドは、フィルタ <i>i</i> 動作を制御します。</p> <ul style="list-style-type: none"> ビット 3 は、アドレスタイプを指定し、パターンへのデスティネーションアドレスタイプを定義します。このビットがセットされた場合、パターンがマルチキャストパケットにのみ適用され、このビットがリセットされた場合、パターンはユニキャストフレームにのみ適用されます。 ビット 2 (反転モード) がセットされると、CRC16 ハッシュ機能信号のロジックが反転し、一致する CRC_16 値を持つパケットを拒否します。 ビット 2 は、ビット 1 とともに、「パターン 1 AND NOT パターン 2」などのフィルタロジックを作成することで、MAC がリモートウェイクアップパケットのサブセットを拒否できます。 ビット 1 (And_Previous) は、ブール値ロジックを実装します。⁽¹⁾ セットされると、現在の入力の結果が、前のフィルタの結果と論理積されます。AND ロジックは、2、3、または 4 のフィルタの中のマスクを分割することで、32 バイトを超えるフィルタパターンを許可します。これは、And_Previous ビットをセットさせるフィルタ数に依存します。 ビット 0 はフィルタ <i>i</i> のイネーブルビットです。ビット 0 がセットされていない場合、フィルタ <i>i</i> は無効です。
フィルタ <i>i</i> オフセット	<p>このフィルタ <i>i</i> オフセットレジスタは、フィルタ <i>i</i> がパケットを調べる (パケット内の) オフセットを定義します。</p> <ul style="list-style-type: none"> この 8 ビットパターンのオフセットは、フィルタ <i>i</i> によって調べられる最初のバイトのオフセットです。 許される最小オフセット値は 12 であり、パケットの 13 番目のバイトを指します。 オフセット値 0 は、パケットの最初のバイトを指します。
フィルタ <i>i</i> CRC-16	<p>このフィルタ <i>i</i> CRC-16 レジスタは、パターンから計算された CRC_16 値と、ウェイクアップフィルタレジスタブロックに対してプログラムされたバイトマスクも含みます。</p> <ul style="list-style-type: none"> 16 ビット CRC 計算には、次の多項式が使用されます。 $G(x) = x^{16} + x^{15} + x^2 + 1$ ハッシュ機能計算で使用する各マスクは、そのマスクに関連付けられた 16 ビット値で比較されます。各フィルタには以下の機能があります。 32 ビットマスク: このマスクの各ビットは、検出されたパケットの 1 バイトに対応します。ビットが '1' である場合、対応するバイトが CRC16 の計算に取り入れられます。 8 ビットオフセットポインタ: CRC16 の計算を開始するバイトを指定します。 ポインタおよびマスクは、ともに CRC16 の計算で使用されるバイトの配置に使用されます。

1. And_Previous ビットの設定は、4 フィルタのセット内で適用できます。

有効化されていないフィルタの And_Previous ビットをセットしても効果はありません。つまり、4 フィルタのセットで番号が最少のフィルタの And_Previous ビットをセットしても効果はありません。たとえば、フィルタ 0 の And_Previous ビットをセットしても効果はありません。

所定のフィルタ用に And_Previous ビットをセットして、論理積で結ばれるフィルタを形成した場合、無効なフィルタが検出されると AND チェーンは切れます。例: フィルタ 2 の And_Previous ビットがセット (フィルタ 2 コマンドのビット 1 がセット) されているが、フィルタ 1 が有効ではない (フィルタ 1 コマンドのビット 0 がリセットされている) 場合、フィルタ 2 の結果だけが考慮されます。フィルタ 2 の And_Previous ビットがセット (フィルタ 2 コマンドのビット 1 がセット) されているが、フィルタ 1 が有効ではない (フィルタ 1 コマンドのビット 0 がリセットされている) 場合、フィルタ 2 の And_Previous ビットがセット (フィルタ 2 コマンドのビット 1 がセット) されているが、フィルタ 3 が有効ではない (フィルタ 3 コマンドのビット 0 がリセットされている) 場合、フィルタ 2 の And_Previous ビットのセットによる効果がないため、フィルタ 3 の結果と論理積されたフィルタ 1 の結果だけが考慮されます。

And_Previous ビットのセットによって結ばれたフィルタに、相補プログラミングが施されている場合、フレームが論理積で結ばれたフィルタを通過することはありません。たとえば、フィルタ 2 の And_Previous ビットがセット (フィルタ 2 コマンドのビット 1 がセット) され、マルチキャスト検出を示すフィルタ 1 の Address_Type ビットがセット (フィルタ 1 コマンドのビット 3 がセット) され、ユニキャスト検出を示すフィルタ 2 の Address_Type ビットがリセットされている (フィルタ 2 コマンドのビット 3 がリセットされている) 場合、またはその逆の場合、リモートウェイクアップフレームがユニキャストおよびマルチキャストアドレスタイプの両方になりえないため、リモートウェイクアップフレームは論理積で結ばれたフィルタを通過しません。

リモートウェイクアップフィルタレジスタは、4つのリモートウェイクアップフィルタ用の8つの間接アクセスレジスタ (wkuppktfilter_reg#i) として実装され、[リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) を通じてアプリケーションによってアクセスされます。wkuppktfilter_reg レジスタのセット全体に書き込むことで、リモートウェイクアップフィルタをプログラムする必要があります。wkuppktfilter_reg レジスタをプログラムするには、wkuppktfilter_reg0 から wkuppktfilter_reg3 までについて、[リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) で8つのレジスタ値を順に書き込みます。wkuppktfilter_reg レジスタは、類似した方法で読み出されます。MAC は [PMT 制御ステータスレジスタ \(ETH_MACPCSR\)](#) の RWKPTR フィールドの wkuppktfilter_reg レジスタカウンタポインタ値を更新します。

注 : [リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) にバイトまたはハーフワードモードでアクセスした場合、CPU がレーン 3 にアクセスすると、適切な wkuppktfilter_reg にアクセスするための内部カウンタがインクリメントされます。

[リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) が書き込まれると、書き込み操作の後に、その内容が CSR クロックドメインから PHY 受信クロックドメインに転送されます。PHY 受信クロックドメインで最初の書き込みが更新されるまで、[リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) に次の書き込みが行われないようにする必要があります。そうしないと、PHY 受信クロックドメインに2番目の書き込み操作での更新が行われません。そのため、[リモートウェイクアップパケットフィルタレジスタ \(ETH_MACRWKPFRR\)](#) への2回の書き込み操作の間の遅延を PHY 受信クロックの少なくとも4サイクル分とする必要があります。

PMT 割込み

PMT 割込み信号は、有効なリモートウェイクアップパケットが受信されたときにアサートされます。

表 651 は、PMT 割込みが生成されるリモートウェイクアップのシナリオを示しています。

表 651. リモートウェイクアップパケットおよび PMT 割込みの生成⁽¹⁾

フィルタ i コマンド			フレームタイプおよび CRC ステータス		割込みの生成
CAST	INV	EN	受信フレームのキャストタイプ	CRC ステータス	RWK INTR
0	0	1	ユニキャスト	MATCH	リモートウェイクアップパケットが検出され、PMT 割込みが生成されます。
0	1	1	ユニキャスト	MISMATCH	リモートウェイクアップパケットが検出され、PMT 割込みが生成されます。
1	0	1	マルチキャスト	MATCH	リモートウェイクアップパケットが検出され、PMT 割込みが生成されます。
1	1	1	マルチキャスト	MISMATCH	リモートウェイクアップパケットが検出され、PMT 割込みが生成されます。

1. その他の組み合わせでは、リモートウェイクアップパケットは検出されず、PMT 割込みは生成されません。

sbd_intr_o 信号のほか、pmt_intr_o (Rx クロックに同期) 信号がアサートされます。pmt_intr_o 信号 (Rx クロックドメインに同期) は、MAC がパワーダウンモードの場合にソフトウェアによってアプリケーションクロックを停止できるように提供されます。これは、lpi_intr_o 信号 ([セクション : LPI 割込み](#)を参照) と論理和がとられ、EXTI ペリフェラル (ライン 86) に紐づけられます。

pmt_intr_o 信号は PHY Rx クロックドメインで生成されるため、[PMT 制御ステータスレジスタ \(ETH_MACPCSR\)](#) の読出し時にすぐにはクリアされません。これは、生成されるクリア信号が PHY

Rx クロックドメインに渡されてから、割込みソースがクリアされなければならないためです。この遅延は少なくとも Rx クロックの少なくとも 4 クロックサイクル分で、ペリフェラルが 10 Mbps モードで動作している場合に大きくなる可能性があります。アプリケーションによりリモートウェイクアップパケットフィルタレジスタ (ETH_MACRWKPFRR) の PWRDWN ビットがクリアされると、MAC はパワーダウンモードから復帰しますが、このイベントでは PMT 割込みは生成されません。

パワーダウンシーケンス

ソフトウェアでは、パワーダウンシーケンスを開始するために次のタスクを実行する必要があります。

- **チャンネル送信制御レジスタ (ETH_DMACTXCR)** の ST ビットをクリアすることで送信 DMA (適用できる場合) を無効にします。
- 前のフレーム送信の完了を待ちます。これをチェックするには、**デバッグレジスタ (ETH_MACDR)** で TFCSTS[1:0] および TPESTS ビットを読み出し、すべての MTL Tx キューの Tx キューデバッグレジスタ (ETH_MTLTXQDR) で TXQSTS ビットを読み出します。
- **動作モード設定レジスタ (ETH_MACCCR)** の TE および RE ビットをクリアすることによって、MAC トランスミッタと MAC レシーバを無効にします。
- 受信 DMA が Rx FIFO のすべてのフレームをエンプティにするのを待ちます。これをチェックするには、すべての Rx キューの **Rx キューデバッグレジスタ (ETH_MTLRXQDR)** で PRXQ[13:0] を読み出します。これらのビットが 0 の場合、Rx FIFO がエンプティであることを示します。
- **PMT 制御ステータスレジスタ (ETH_MACPCSR)** でマジックパケット (MGKPKTEN) とリモートウェイクアップ (RWKPKTEN) 検出を設定します。
- **動作モード設定レジスタ (ETH_MACCCR)** でビット 31 (ARPEN) をセットします。
- RE ビットをセットして MAC レシーバを有効化してから、**PMT 制御ステータスレジスタ (ETH_MACPCSR)** で PWRDWN ビットをセットして MAC でパワーダウンシーケンスを開始します。

注 : この機能が有効化されていて、MAC トランスミッタが LPI モードの状態からパワーダウンモードに切り替えられると、MII インタフェースがクランプされ、LPI パターンがアサートされます。MAC トランスミッタが LPI モードでない状態でパワーダウンモードに切り替えられると、GMII 又は MII インタフェースがすべて 0 にクランプされます。

パワーアップシーケンス

MAC はマジックパケット、またはリモートウェイクアップフレームの受信時にウェイクアップします。パワーアップシーケンスは、次のとおりです。

- MAC が pmt_intr_o をアサートします。低電力モードでクロックゲーティングのみが使用されている場合、pmt_intr_o 信号を使用して、低電力モードに移行した後にゲートオフされたクロックを開始することができます。
- ソフトウェアは次のタスクを実行します。
 - **PMT 制御ステータスレジスタ (ETH_MACPCSR)** を読み出して、pmt_intr_o をネゲートします。
 - **PMT 制御ステータスレジスタ (ETH_MACPCSR)** とリモートウェイクアップパケットフィルタレジスタ (ETH_MACRWKPFRR) に (リセット値により) 書込み操作を実行して、常時オンブロックの対応する値が同期されるようにします。そうしないと、これらのレジスタの値が一致しなくなります。
 - **動作モード設定レジスタ (ETH_MACCCR)**、**MAC アドレス 0 上位レジスタ (ETH_MACA0HR)**、および **MAC アドレス 0 下位レジスタ (ETH_MACA0LR)** に書込み動作を実行し、CSR モジュールの値と常時オンブロックの各ビットを同期させます。そうしないと、受信有効化ビットが 0 にセットされている場合でも、MAC レシーバがオンになります。

これらのステップを完了した後で、ソフトウェアによりすべてのレジスタを初期化し、トランスミッタを有効化し、(DMA 設定で) DMA をプログラムして通常動作を再開する必要があります。

57.7.2 省電力型 Ethernet(EEE)

EEE は、IEEE 802.3 メディアアクセス制御 (MAC) のサブレイヤと物理レイヤのファミリを低電力アイドル (LPI) モードと一緒に動作させるための動作モードです。EEE 動作モードは、100 Mbps での IEEE 802.3 MAC 動作をサポートしています。ペリフェラルは EEE 用の IEEE 802.3az-2010 をサポートしています。

LPI モードでは、データが送受信されていないときに、通信デバイス機能の一部をオフに切り替えることで電力削減を実現できます。リンクの両側のシステムで一部の機能を無効にして、低リンクの使用中の電力を節約することができます。MAC は、システムの LPI モードへの移行または終了を制御し、これを PHY に伝えます。

EEE は、リンクパートナーが EEE をサポートしているか判定してから、両方のデバイスに共通するパラメータのセットを選択するために使用できる機能のネゴシエーション方式を指定します。

送信パス機能

送信パス機能には、PHY を LPI 状態に移行するために MAC で実行する必要のあるタスクが含まれます。

送信パスでは、ソフトウェアは **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の LPIEN ビットをセットして、MAC に送信を停止して LPI プロトコルを開始するように指示しなければなりません。MAC は実行中の送信を完了し、送信ステータスを生成し、**LPI タイマ制御レジスタ (ETH_MACLTCR)** の LPI LS TIMER LST[9:0] ビットフィールドに指定された期間、継続的にリンクステータスがアップ状態である場合、IDLE パターンの代わりに LPI パターンの送信を開始します。**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の PHY リンクステータス PLS ビットは PHY のリンクステータスを示します。

注 : **MAC が RMII を使用するように設定されている場合、EEE 機能はサポートされません。**

標準 (IEEE 802.3az-2010) に従って、MII (10 または 100) モードでは LPI 状態のときに PHY が TxCLK クロックを停止しないようにする必要があります。

PHY の LPI 状態に移行するために、MAC は以下のタスクを実行します。

1. TX_EN をネゲートします。
2. TX_ER をアサートします。
3. TXD[3:0] を 0x1 にセットします (100 Mbps の場合)。
4. ステータス (**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の TLPIEN ビット) を更新して、割込みを生成します。

注 : **MAC は、PHY が LPI 状態である間、TX_EN、TX_ER、TXD 信号の同じ状態を維持します。**

PHY を LPI 状態から復帰させるには、つまりソフトウェアで LPIEN ビットをリセットするときは、MAC は次のタスクを実行します。

1. LPI パターンの送信を停止し、IDLE パターンの送信を開始します。
2. LPI TW TIMER を開始します。

MAC は、PHY に指定されたウェイクアップ時間が終了するまで、送信を開始できません。自動ネゴシエーションウェイクアップクアップの間隔は、**LPI タイマ制御レジスタ (ETH_MACLTCR)** の TWT フィールドでプログラムされます。

3. LPI 終了ステータス (**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の TLPIEX ビット) を更新して、割込みを生成します。

図 817 に、LPI モードの遷移時の TX_EN、TX_ER、および TXD[3:0] 信号の動作を示します。

注 : MAC では TX_CLK クロックを停止しません。PHY で許可されている場合、また MAC で sbd_tx_clk_gating_ctrl_o 信号が 1 にセットされている場合は、アプリケーションでこのクロックを停止することができます (図 817 を参照)。sbd_tx_clk_gating_ctrl_o 信号は、9 つの Tx クロックサイクル、1 つのパルスシンクロナイザ遅延、1 つの CSR クロックサイクル後にアサートされます。sbd_tx_clk_gating_ctrl_o 信号のアサートは、LPI 制御およびステータスレジスタ (ETH_MACLCSR) の LPITCSE ビットに依存し、図 818 に示すように自動的に行うことができます。

RGMII インタフェースが選択されている場合、LPI パターンを送信するには Tx クロックが必要となるため、Tx クロックはゲーティングできません。

MAC が Tx LPI モードで、Tx クロックが停止している場合、Tx クロックドメインに同期されている CSR レジスタへの書き込みをアプリケーションで行わないようにする必要があります。

MAC が LPI モードで、アプリケーションからソフトリセットまたはハードリセットを発行すると、MAC トランスミッタは LPI モードから復帰します。

図 817. LPI 遷移 (送信、100 Mbps)

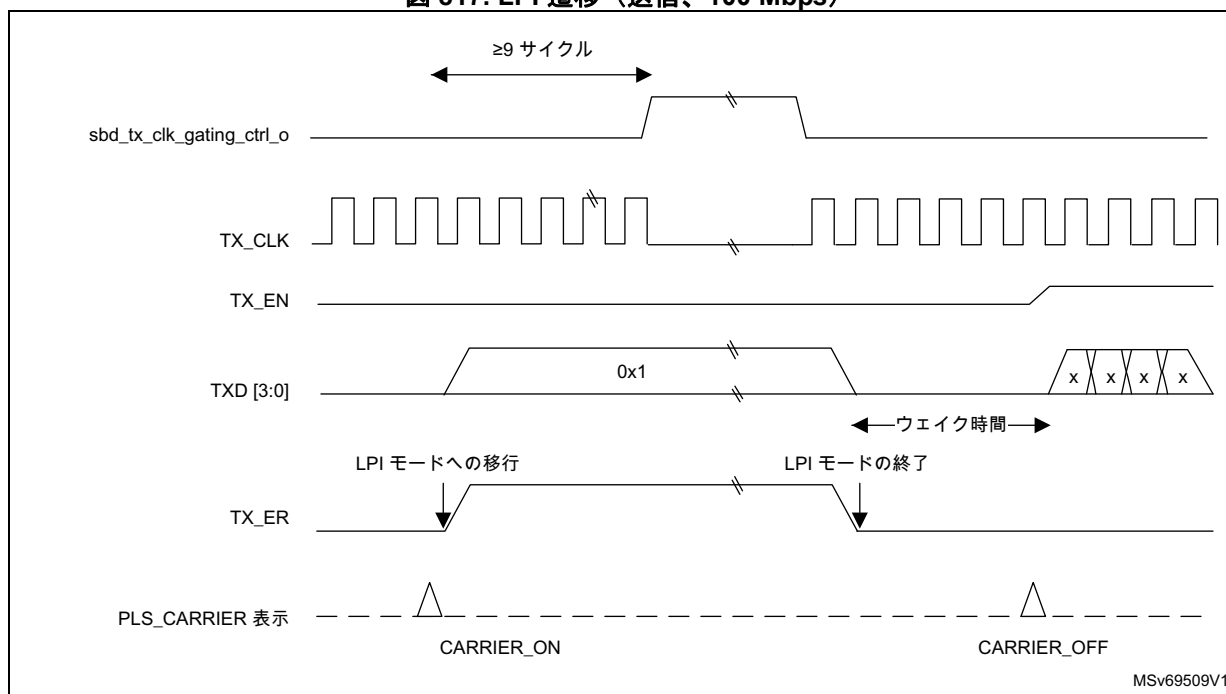
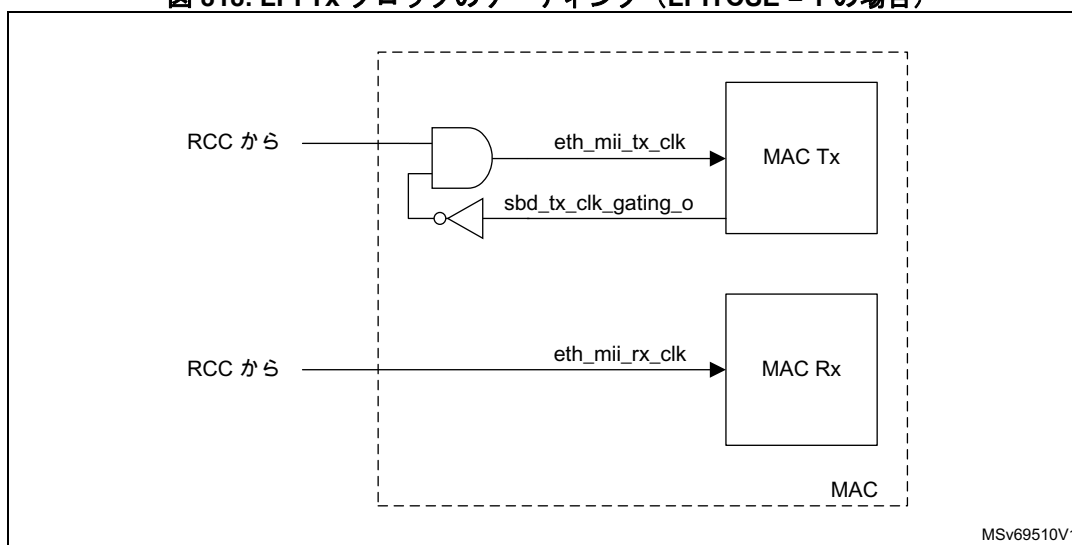


図 818. LPI Tx クロックのゲーティング (LPITCSE = 1 の場合)



MSv69510V1

送信パスでの LPI モードの自動開始/終了

MAC トランスミッタは、一定時間アイドル状態であるか、または転送するパケットがあるかどうかに基づいて、自動的に LPI アイドルモードの移行および終了を行うようにプログラムできます。これらのモードは、[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) で有効化して制御されます。

[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) の LPITXA および LPIEN がセットされているとき、MAC 送信パス (MTL レイヤおよび DMA レイヤを含む) がアイドル状態である場合、MAC トランスミッタは LPI アイドル状態に移行します。Tx パスで何らかの機能 (DMA、MTL または MAC) がパケット転送の開始によってアイドル状態ではなくなると、すぐに MAC トランスミッタは LPI アイドル状態を終了し、LPITXEN ビットをクリアします。

また、LPITE もセットされているとき、[LPI 移行タイマレジスタ \(ETH_MACLETR\)](#) の値で示された時間、送信パスがアイドル状態 (動作なし) のままである場合のみ、MAC トランスミッタは LPI アイドル状態に移行します。このモードでも、何らかの機能がアイドル状態ではなくなると、すぐに MAC トランスミッタは LPI アイドル状態を終了します。ただし、MAC が再びアイドルになったとき、ソフトウェアの介入がなくても LPI アイドル状態に再移行できるように、LPIEN ビットはクリアされずアクティブなままとなります。

LPITE および LPITXA ビットの両方がクリアされた場合、LPIEN ビットをプログラムすることで、LPI アイドル状態の移行と終了を直接制御できます。

受信パス機能

受信パス機能には、PHY が LPI 状態を終了するためにリンクパートナーから信号を受信した時に、PHY および MAC で実行すべきタスクが含まれます。

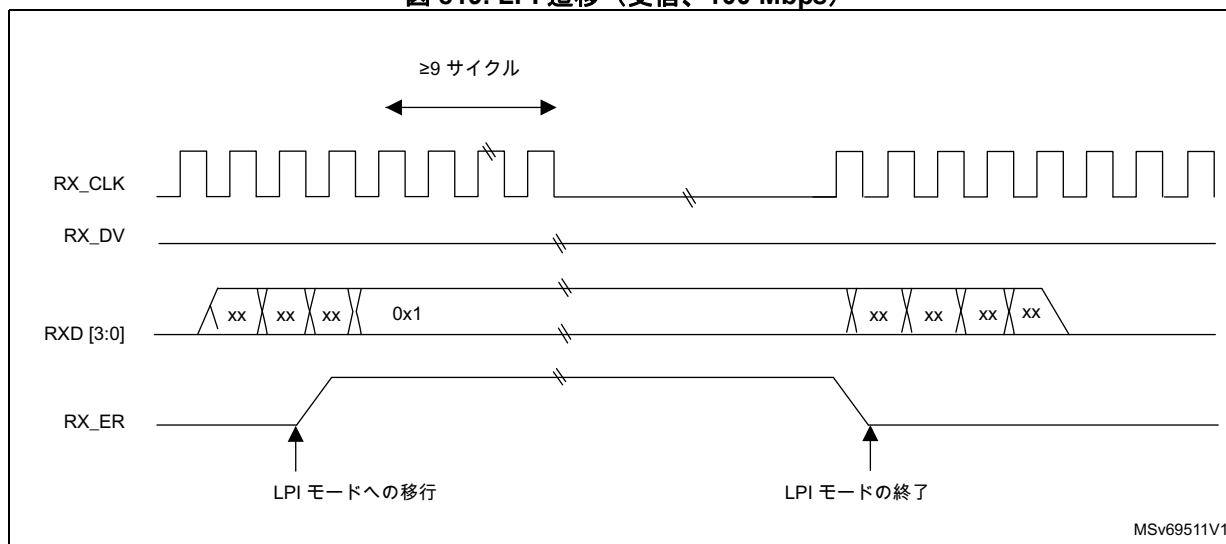
受信パスで、PHY が LPI 状態に移行するためのリンクパートナーからの信号を受信したときに、PHY および MAC では次のタスクを実行します。

1. PHY が RX_ER をアサートします。
2. PHY が RXD[3:0] を 0x1 にセットします (100 Mbps の場合)。
3. PHY が RX_DV をネゲートします。
4. MAC が [LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) の RLPIEN ビットを更新して、すぐに割込みを生成します。

- 注 : PHY は、PHY が LPI 状態である間、RX_ER、RXD、RX_DV 信号の同じ状態を維持します。
- LPI パターンが非常に短い期間だけ（つまり、Rx クロックの 2 サイクル未満）検出された場合、MAC は Rx LPI モードに移行しません。
- 現在の Rx LPI パターンの終了と次の Rx LPI パターンの開始の間の期間が非常に短い（つまり、Rx クロックの 2 サイクル未満）場合、MAC は Rx LPI モードを終了し、再度 Rx LPI モードに移行します。MAC は Rx LPI 終了およびエントリ割込みをトリガしません。
- PHY が LPI 状態を終了するためのリンクパートナーからの信号を受信したときに、PHY および MAC では次のタスクを実行します。
1. PHY が RX_ER をネゲートし、通常の packets 間状態に戻ります。
 2. MAC が **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の RLPIEX ビットを更新して、すぐに割込みを生成します。サイドバンド信号 lpi_intr_o (Rx クロックに同期) もアサートされます。

図 819 に、LPI モードの遷移時の RX_ER、RX_DV、および RXD[3:0] 信号の動作を示します。

図 819. LPI 遷移 (受信、100 Mbps)



- 注 : PHY が MAC に LPI を示しているときに RX_CLK_stoppable ビット (PHY レジスタで MDIO を通じて書き込まれる) がアサートされた場合、図 819 に示すように、LPI 状態が開始してから 9 クロックサイクル以上後の任意の時点で、PHY は RX_CLK を停止することがあります。
- MAC が LPI モードで、アプリケーションからソフトリセットまたはハードリセットが発行された場合、リセット中、MAC レシーバは LPI モードを終了します。リセットがネゲートされた後にも LPI パターンを受信する場合、MAC レシーバは再度 LPI 状態に移行します。
- RX LPI モードで RX クロックが停止する場合、アプリケーションは RX クロックドメインに同期されている CSR レジスタに書き込まないようにする必要があります。
- PHY が LPI パターンを送信するとき、EEE 機能が有効になっていれば、MAC は自動的に LPI 状態に移行します。MAC が LPI 状態に移行するのを防止するためのソフトウェア制御はありません。

LPI タイマ

トランスミッタでは、LPI LS TIMER、LPI TW TIMER、および LPI AUTO ENTRY TIMER タイマを管理しています。

次の LPI タイマには、[LPI タイマ制御レジスタ \(ETH_MACLTCR\)](#) および [LPI 移行タイマレジスタ \(ETH_MACLETR\)](#) から各値がロードされます。

- LPI LS TIMER

LPI LS TIMER は、リンクステータスがアップ状態になってから経過した時間をミリ秒単位でカウントします。

このタイマは、リンクがダウン状態になるたびにクリアされます。リンクが再びアップ状態になるとインクリメントを開始し、タイマの値がターミナルカウントと等しくなるまでインクリメントを続けます。ターミナルカウントに達すると、リンクがアップ状態であるかぎり、タイマは同じ値を維持します。ターミナルカウントは、[LPI タイマ制御レジスタ \(ETH_MACLTCR\)](#) の LST[9:0] ビットフィールドでプログラムされた値です。LPI LS TIMER は 10 ビット幅です。ソフトウェアでは最大 1023 ミリ秒までプログラムできます。

- LPI TW TIMER

LPI TW TIMER は、LPI のネゲート以降に経過した時間をミリ秒単位でカウントします。ターミナルカウントは [LPI タイマ制御レジスタ \(ETH_MACLTCR\)](#) の Bit[15:0] でプログラムする必要があります。タイマのターミナルカウントは、自動ネゴシエーション時間である解決済みの送信 TW の値で、この時間の経過後に MAC は通常の送信動作を再開できます。LPI モードが終了してから、TW タイマがターミナルカウントに達した後に、MAC は通常動作を再開します。

MAC は、マイクロ秒単位の LPI TW TIMER をサポートしています。LPI TW TIMER は 16 ビット幅です。そのため、ソフトウェアでは最大 65535 マイクロ秒までプログラムできます。

- LPI AUTO ENTRY TIMER

このタイマはMAC のパス送信がアイドル状態（動作なし）のままになる時間を 8 マイクロ秒刻みでカウントします。これは、MAC トランスミッタが LPI IDLE 状態に移行し、LPI パターンの送信を開始するまでの時間です。このタイマは、[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) で LPITE ビットがセットされると有効になります。

LPI 割込み

Tx 側または Rx 側が LPI 状態の移行時または終了時に、MAC は LPI 割込みを生成します。LPI 割込みステータスがセットされている場合、割込み sbd_intr_o がアサートされます。LPI 割込みは、[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) を読み出すことによってクリアできます。

MAC が Rx LPI 状態を終了すると、sbd_intr_o のほか、サイドバンド信号 lpi_intr_o (Rx クロックに同期) もアサートされます。lpi_intr_o 信号を使用すると、外部クロックゲーティング回路でのMAC へのアプリケーションクロックの復旧をトリガできます。lpi_intr_o 信号 (Rx クロックドメインに同期) は、MAC がLPI の状態の場合にソフトウェアによってアプリケーションクロックを停止できるように提供されます。これは、pmt_intr_o 信号 ([セクション : PMT 割込み](#)を参照) と論理和がとられ、EXTI ペリフェラル (ライン 86) に紐づけられます。

lpi_intr_o 信号は Rx クロックドメインで生成されます。[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) が読み出されてすぐにクリアされないこともあります。これは、CSR クロックドメインで生成されるクリア信号が Rx クロックドメインを横断する必要があり、割込みソースをクリアする必要があるためです。この遅延は Rx クロックの少なくとも 4 クロックサイクル分で、ペリフェラルが 10 Mbps モードで動作している場合に大きくなる可能性があります。

省電力型 Ethernet に関するプログラミングガイドライン

プログラミングガイドラインについての詳細は、[セクション 57.9.11 : 2740 ページの省電力型 Ethernet \(EEE\) に関するプログラミングガイドライン](#)を参照してください。

57.8 Ethernet 割込み

Ethernet ペリフェラルは、1 つの割込み信号 (eth_sbd_intr_it) を生成します。この信号は、さまざまなイベントの結果として発生することがあります。これらのイベントはステータスレジスタでキャプチャされ、対応する割込み有効がセットされたときのみ割込み信号がイベントにアサートされるように、割込み有効が割込みの各ソースに対して提供されます。

割込みステータスと対応する有効レジスタは階層形式で構成されているため、ソフトウェアが割込みイベントのソースを素早く横断して識別しやすくなっています。割込みがアサートされると、**割込みステータスレジスタ (ETH_DMAISR)** レジスタは割込みイベントソースの主なブロックを示す最初のレベルになります。このレジスタは読み出し専用で、各 DMA チャンネル (TX & RX ペア)、MTL、MAC に対応するビットを含みます。次に、ソフトウェアアプリケーションは、セットされているビットに対応する次のレジスタのうち 1 つ (または複数) を読み出す必要があります。

- ETH_DMCSR : チャンネルステータス (**チャンネルステータスレジスタ (ETH_DMCSR)** を参照)
- ETH_MTLISR : 割込みステータス (**割込みステータスレジスタ (ETH_MTLISR)** を参照)
- ETH_MACISR : 割込みステータス (**割込みステータスレジスタ (ETH_MACISR)** を参照)

57.8.1 DMA 割込み

割込みレジスタの説明

ETH_DMCSR : チャンネルステータスレジスタ (**チャンネルステータスレジスタ (ETH_DMCSR)** を参照) は、その TxDMA および RxDMA チャンネルのすべての割込みイベントをキャプチャします。ETH_DMCIER : チャンネル割込み有効レジスタ (**チャンネル割込み有効レジスタ (ETH_DMCIER)** を参照) には、割込みイベントごとに対応する有効ビットが含まれます。

DMA チャンネルには、通常割込みと異常割込みという 2 つのグループの割込みがあります。これらはそれぞれ ETH_DMCSR レジスタのビット [15:14] によって示されます。通常グループは、パケットの通常転送中に発生するイベント (TI : 送信割込み、RI : 受信割込み、TBU : 送信バッファ使用不可) に対応し、異常割込みイベントはエラーイベントに対応します。

割込みはキューされません。ドライバが前のイベントに対応する前に、同じ割込みイベントが再び発生した場合、追加の割込みは生成されません。割込みは、複数のイベントについて 1 つしか生成されません。ドライバは、**割込みステータスレジスタ (ETH_DMAISR)** で割込みの原因をスキャンして、それぞれのステータスレジスタのソースをクリアしなければなりません。**割込みステータスレジスタ (ETH_DMAISR)** のすべてのビットがクリアされたときのみ割込みがクリアされます。

送信および受信割込みの周期的スケジュール

システムスループットのパフォーマンスを理由に、DMA によって転送される各パケットに対して割込み (RI および TI) を生成することは推奨されません。Ethernet ペリフェラルは、2 つの方法を使用して一定間隔で割込みをスケジュールする柔軟性を備えています。

1. 送信されるパケットのそれぞれの「必要」数につき 1 つ、送信ディスクリプタに完了時割込みビット (表 655 : **TDES2 通常ディスクリプタ (読み出しフォーマット)**) の TDES2[31]) をセットします。
2. 同様に、受信ディスクリプタの指定した間隔でのみ IOC (表 668 : **RDES3 通常ディスクリプタ (読み出しフォーマット)**) の RDES3[30]) ビットをセットします。こうして、システムメモリへの受信パケットの転送が完了し、そのパケット転送に使用されるディスクリプタのいずれかに IOC ビットがセットされているときは、必ず RI イベントが生成されます。

上記のほかに、割込みタイマ (ETH_DMCCRXTWTR : チャンネル Rx 割込みウォッチドッグタイマ) が提供され、受信割込みの柔軟性の高い制御および周期的なスケジューリングを可能にします。この割込みタイマが 0 でない値でプログラムされると、対応する完了の割込み IOC ビット (表 668 : **RDES3 通常ディスクリプタ (読み出しフォーマット)**) の RDES3[30]) がセットされていないため、

Rx DMA が受信割り込みをアサートせずに、受信パケットのシステムメモリへの転送を完了すると、割り込みタイマが起動します。このタイマがプログラムされた値に従ってタイムアウトになると、RI ビットがセットされ、ETH_DMACIER レジスタ（[チャンネル割り込み有効レジスタ \(ETH_DMACIER\)](#)）を参照）で対応する RIE が有効な場合、割り込みがアサートされます。ディスクリプタの IOC がセットされたパケット転送に対して RI がセットされている場合、タイマは終了する前に停止されてクリアされます。RI イベントが生成されずに次のパケット転送が完了した後、タイマは自動的に再び起動します。

チャンネル転送完了割り込み

送信転送完了割り込み (TI) および受信転送完了割り込み (RI) は、チャンネルステータスレジスタ（[チャンネルステータスレジスタ \(ETH_DMACSR\)](#)）に反映されます。Tx DMA チャンネルが IOC ビット（完了時割り込み - TDES2[31]）がセットされたディスクリプタをクローズすると必ず TI ビットがセットされます。同様に、Rx DMA チャンネルが LD ビットがセットされたディスクリプタをクローズすると必ず RI ビットがセットされ、そのパケットの転送に使用されるディスクリプタのいずれかで、IOC ビット（完了時割り込み有効 - RDES3[30]）がセットされます。

割り込み信号は、チャンネル割り込み有効レジスタ（[チャンネル割り込み有効レジスタ \(ETH_DMACIER\)](#)）で対応する割り込みが有効なときのみ、転送完了割り込みに対してアサートされます。

RI/TI 割り込みの動作は、ETH_DMAMR レジスタ（[DMA モードレジスタ \(ETH_DMAMR\)](#)）での INTM フィールド（ビット [17:16]）の設定に応じて変化します。表 652 で、転送完了割り込みの動作について説明しています。

表 652. 転送完了割り込みの動作

割り込みモード	TI/RI および割り込み信号の動作
INTM=0	TI/RI ステータス信号は、転送完了イベントが検出されると必ずセットされます。これらのビットは、ソフトウェアドライバによって 1 を書き込まれると必ずクリアされます。割り込み信号は、対応する割り込みが ETH_DMACIER レジスタでも有効になっていると必ずアサートされます。
INTM=1	TI/RI は上記に説明されたようにセットされます。ただし、割り込みは RI/TI イベントに対してアサートされません。
INTM=2	RI/TI ステータスビットは、転送完了イベントが検出されると必ずセットされ、ソフトウェアドライバが 1 を書き込むことによってクリアすると必ずリセットされます。ただし、ソフトウェアによってクリア（機能）される前に、別の転送完了イベントが検出された場合、これらのステータスビットは自動的に再びセットされます。ただし、割り込みは TI/RI に基づいて生成されません。

57.8.2 MTL 割り込み

MTL 割り込みイベントは、DMA でのイベントと組み合わせて、割り込み信号を生成します。

[割り込みステータスレジスタ \(ETH_MTLISR\)](#) は、イベントを管理するキュー数をレポートします。ETH_MTLQICSR : キュー割り込み制御ステータスは、イベントの説明のために読み出されます。

MTL 割り込みはデフォルトで有効にされます。[割り込みステータスレジスタ \(ETH_MTLISR\)](#) レジスタの対応するマスクビットをセットすることによって、各イベントが割り込みをアサートしないようにすることができます。

MTL 割り込み信号は、次のいずれかのイベントによって駆動されます。

- 受信キューオーバーフロー割り込み
- 送信キューアンダーフロー

57.8.3 MAC 割込み

MAC 割込みイベントは、DMA でのイベントと組み合わせて、割込み信号を生成します。

MAC 割込みはレベルタイプで、アプリケーションやソフトウェアによってクリアされるまで、割込みはアサート（ハイ）されたままになります。

割込みステータスレジスタ (ETH_MACISR) は、MAC からの割込みの原因となるイベントを記述します。MAC 割込みはデフォルトで有効にされます。**割込みステータスレジスタ (ETH_MACISR)** の対応するマスクビットをセットすることによって、各イベントが割込みをアサートしないようにすることができます。

割込みレジスタビットは、イベントが報告されるブロックを示すだけです。割込みをクリアするには、対応するステータスレジスタと他のレジスタを読み出す必要があります。

MAC 割込み信号は、次のいずれかのイベントによって駆動されます。

- 受信ステータス割込み
- 送信ステータス割込み
- タイムスタンプ割込みステータス
- MMC 割込みステータス
 - MMC 受信チェックサムオフロード割込みステータス
 - MMC 送信割込みステータス
 - MMC 受信割込みステータス
- LPI 割込みステータス
- PMT 割込みステータス
- PHY 割込み

注 : 2 つのサイドバンド信号が、LPI および PMT 割込みとともに生成されます。lpi_intr_o および pmt_intr_o です。これらは EXTI レベルでのウェイクアップイベント検出に使用されます。

57.9 Ethernet のプログラミングモデル

この章では、適切なシーケンスでの DMA または MAC レジスタの初期化に関して説明します。以下のセクションが含まれています。

- DMA の初期化 ([セクション 57.9.1](#) を参照)
- MTL の初期化 ([セクション 57.9.2](#) を参照)
- MAC の初期化 ([セクション 57.9.3](#) を参照)
- 通常受信および送信動作の実行 ([セクション 57.9.4](#) を参照)
- 送信の停止および開始 ([セクション 57.9.5](#) を参照)
- MII リンク状態遷移に関するプログラミングガイドライン ([セクション 57.9.8](#) を参照)
- IEEE 1588 タイムスタンプに関するプログラミングガイドライン ([セクション 57.9.9](#) を参照)
- 省電力型 Ethernet に関するプログラミングガイドライン ([セクション 57.9.11](#) を参照)
- 柔軟性の高い Pulse-Per-Second (PPS) 出力に関するプログラミングガイドライン ([セクション 57.9.12](#) を参照)
- TSO に関するプログラミングガイドライン ([セクション 57.9.13](#) を参照)
- 受信時の VLAN フィルタに関するプログラミングガイドライン ([セクション 57.9.14](#) を参照)

57.9.1 DMA の初期化

DMA を初期化するには、次の手順を完了します。

1. すべての MAC 内部レジスタおよびロジック ([DMA モードレジスタ \(ETH_DMAMR\)](#) のビット 0) をリセットするためにソフトウェアリセットを提供します。
2. リセットプロセスの完了を待ちます (リセット動作が完了したときにクリアされる [DMA モードレジスタ \(ETH_DMAMR\)](#) のビット 0 にポーリングします)。
3. 次のフィールドをプログラムして、[システムバスモードレジスタ \(ETH_DMASBMR\)](#) を初期化します。
 - a) AAL
 - b) 固定バーストまたは未定義バースト
 - c) AHB バスインタフェースの場合のバーストモード値
4. 送信および受信のディスクリプタリストを作成します。さらに、受信ディスクリプタが DMA によって所有されていることを確認します (TDES3/RDES3 ディスクリプタのビット 31 をセットします)。ディスクリプタの詳細については、[セクション 57.10 : ディスクリプタ](#)を参照してください。

注 : リングの開始から終了までのディスクリプタアドレスは、4 GB の境界を越えないようにします。

5. ETH_DMACTXRLR および ETH_DMACRXRLR レジスタをプログラムします ([チャンネル Tx ディスクリプタリング長レジスタ \(ETH_DMACTXRLR\)](#) および [チャンネル Rx ディスクリプタリング長レジスタ \(ETH_DMACRXRLR\)](#) を参照)。プログラムされるリングの長さは、4 以上にセットする必要があります。
6. 受信および送信のディスクリプタリストのアドレスを、送信および受信のディスクリプタのベースアドレスで初期化します ([チャンネル Tx ディスクリプタリストアドレスレジスタ \(ETH_DMACTXDLAR\)](#)、[チャンネル Rx ディスクリプタリストアドレスレジスタ \(ETH_DMACRXDLAR\)](#))。さらに、DMA に使用できるディスクリプタを伝える送信および受信のテイルポインタレジスタをプログラムします ([チャンネル Tx ディスクリプタテイルポインタレジスタ \(ETH_DMACTXDTPR\)](#) および [チャンネル Rx ディスクリプタテイルポインタレジスタ \(ETH_DMACRXDTPR\)](#))。

7. ETH_DMCCR、ETH_DMACRXCRCR および ETH_DMACRXCRCR レジスタ (チャンネル制御レジスタ (ETH_DMCCR)、チャンネル送信制御レジスタ (ETH_DMACRXCRCR)、およびチャンネル受信制御レジスタ (ETH_DMACRXCRCR) を参照) をプログラムして、DMA によって開始される最大バースト長 (PBL)、ディスクリプタスキップ長、TxDMA の OSP、RxDMA の RBSZ[13:0] などのパラメータを設定します。
8. ETH_DMCIER レジスタをプログラムすることで割込みを有効にします (チャンネル割込み有効レジスタ (ETH_DMCIER) を参照)。
9. チャンネル受信制御レジスタ (ETH_DMACRXCRCR) の SR (ビット 0) および ETH_DMACRXCRCR の ST (ビット 0) をセットすることで受信および送信 DMA を開始します (チャンネル送信制御レジスタ (ETH_DMACRXCRCR) を参照)。

57.9.2 MTL の初期化

MTL レジスタを初期化するには、次の手順を完了します。

1. 次のフィールドをプログラムして、Tx キュー動作モードレジスタ (ETH_MTLTXQOMR) の動作モードを初期化します。
 - a) 閾値モードが使用されている場合、送信ストアアンドフォワード (TSF) または送信閾値制御 (TTC)
 - b) 送信キュー有効 (TXQEN) の値を 2'b10 にして送信キュー 0 を有効化
 - c) 送信キューサイズ (TQS)
2. 次のフィールドをプログラムして、ETH_MTLRXQOMR レジスタの動作モードを初期化します (Rx キュー動作モードレジスタ (ETH_MTLRXQOMR) を参照)。
 - a) 閾値モードが使用されている場合、受信ストアアンドフォワード (RSF) または RTC
 - b) MTL 受信 FIFO のフロー制御有効化および無効化閾値 (RFA および RFD)
 - c) 有効化に続くエラーパケットおよびアンダーサイズの良いパケット (FEP および FUP)
 - d) 受信キューサイズ (TQS)

57.9.3 MAC の初期化

MAC 設定レジスタは、MAC の動作モードを確立します。可能であれば、DMA を初期化する前にこれらのレジスタを初期化する必要があります。以下の MAC 初期化操作は、DMA 初期化後に実行することもできます。DMA が設定される前に MAC 初期化が完了した場合、DMA がアクティブになった後にのみ MAC レシーバを有効にします (以下のシーケンスの最後の手順)。そうしないと、受信フレームが Rx FIFO を満たしてオーバーフローします。

1. 次の MAC アドレスレジスタが提示されます。MAC アドレス x 下位レジスタ (ETH_MACAxLR)、MAC アドレス 0 上位レジスタ (ETH_MACA0HR) および MAC アドレス x 上位レジスタ (ETH_MACAxHR) です。
2. 次のフィールドをプログラムして、パケットフィルタ制御レジスタ (ETH_MACPFR) の着信フレームに適切なフィルタを設定します。
 - a) すべてを受信
 - b) プロミスキヤスモード
 - c) ハッシュまたは完全フィルタ
 - d) ユニキャスト、マルチキャスト、ブロードキャストおよび制御フレームフィルタ設定
3. 次のフィールドをプログラムして、Tx キューフロー制御レジスタ (ETH_MACQTXFCR) で適切なフロー制御を行います。
 - a) ポーズ時間およびその他のポーズフレーム制御ビット
 - b) 送信フロー制御ビット

- c) フロー制御ビジー
- 4. 設定に適用できる場合、**割込み有効レジスタ (ETH_MACIER)** を必須としてプログラムします。
- 5. **動作モード設定レジスタ (ETH_MACCCR)** レジスタの適切なフィールドをプログラムします。たとえば、送信およびジャバ―無効化中のパケット間隔。
- 6. **動作モード設定レジスタ (ETH_MACCCR)** レジスタのビット 0 および 1 をセットして、MAC トランスミッタおよびレシーバを開始します。

ジャンボ送信/受信パケットをサポートするには、以下の手順に従ってください。

- **動作モード設定レジスタ (ETH_MACCCR)** で
 - a) JE ビットを 1 にセットします。
 - b) ジャイアントパケットエラーレポートを回避するため、JD および WD ビットを 0 にセットします。
 - c) GPSLCE ビットを 1 にセットします。
 - d) **拡張動作モード設定レジスタ (ETH_MACECR)** の GPSL ビットフィールドに 9026 を超える値をセットします。

最大 16K の送信/受信パケットをサポートするには、以下の手順に従ってください。

- **動作モード設定レジスタ (ETH_MACCCR)** で
 - a) ジャイアントパケットエラーレポートを回避するため、JD および WD ビットを 1 にセットします。
 - b) GPSLCE ビットを 1 にセットします。
 - c) **拡張動作モード設定レジスタ (ETH_MACECR)** の GPSL ビットフィールドを 16383 にセットします。

57.9.4 通常受信および送信動作の実行

通常動作では、次の手順を完了します。

1. 通常送信および受信割込みでは、割込みステータスを読み出します。次に、ホスト（送信または受信）によって所有されるディスクリプタのステータスを読み出し、ディスクリプタにポーリングします。
2. ディスクリプタに適切な値をセットします。データの送受信を再開するために、送信および受信ディスクリプタが DMA によって所有されていることを確認します。
3. ディスクリプタが DMA によって所有されていない場合（またはディスクリプタが使用できない場合）、DMA はサスペンド状態に入ります。ディスクリプタを開放して、**ETH_DMACTXDTPR (チャンネル Tx ディスクリプタテイルポインタレジスタ (ETH_DMACTXDTPR))** および **ETH_DMACRXDTPR (チャンネル Rx ディスクリプタテイルポインタレジスタ (ETH_DMACRXDTPR))** を参照) を書き込むことによって送受信を再開できます。
4. デバッグモードでは、現在のホストのトランスミッタおよびレシーバのディスクリプタアドレスポインタの値は、**ETH_DMACCATXDR** および **ETH_DMACCARXDR** レジスタで読み出せます (**チャンネル現在のアプリケーション送信ディスクリプタレジスタ (ETH_DMACCATXDR)** および **チャンネル現在のアプリケーション受信ディスクリプタレジスタ (ETH_DMACCARXDR)**)。
5. デバッグモードでは、現在のホストの送信バッファアドレスポインタおよび受信バッファアドレスポインタの値は、**ETH_DMACCATXDR** および **ETH_DMACCARXDR** レジスタで読み出せます (**チャンネル現在のアプリケーション送信ディスクリプタレジスタ (ETH_DMACCATXDR)** および **チャンネル現在のアプリケーション受信ディスクリプタレジスタ (ETH_DMACCARXDR)**)。

57.9.5 送信の停止および開始

次の手順を完了して、しばらく送信を一時停止します。

1. ETH_DMACTXCR レジスタのビット 0 (ST) をクリアすることで送信 DMA (適用できる場合) を無効にします (チャネル送信制御レジスタ (ETH_DMACTXCR) を参照)。
2. 前のフレーム送信の完了を待ちます。Tx キューデバッグレジスタ (ETH_MTLTXQDR) の適切なビットを読み出すことでこれをチェックできます (TRCSTS[1:0] が 01 ではなく、TXQSTS=0)。
3. 動作モード設定レジスタ (ETH_MACCR) レジスタの RE および TE ビットをクリアすることによって、MAC トランスミッタと MAC レシーバを無効にします。
4. Rx FIFO のデータがシステムメモリに転送されたことを (Tx キューデバッグレジスタ (ETH_MTLTXQDR) の適切なビット、PRXQ=0 および RXQSTS[1:0] = 00 を読み出すことによって) 確認した後、受信 DMA (適用できる場合) を無効にします。
5. Tx キューおよび Rx キュー両方がエンプティであること (Tx キューデバッグレジスタ (ETH_MTLTXQDR) で TXQSTS が 0、RXQSTS[1:0] が 00 にセットされていること) を確認します。
6. 動作を再開するには、まず DMA を開始してから、MAC トランスミッタおよびレシーバを有効にします。

注 : MAC での送受信中は設定 (二重モード、速度、ポート、ループバックなど) を変更しないでください。これらのパラメータは、MAC トランスミッタおよびレシーバが動作中でないときにのみ、ソフトウェアによって変更されます。

同様に、DMA 関連の設定の変更も、送信 DMA および受信 DMA の動作中に行わないでください。

57.9.6 RxDMA での新しいディスクリプタリストへの切り替えのためのプログラミングガイドライン

新しいディスクリプタリストへの切り替えは、Rx DMA と Tx DMA では異なります。新しいディスクリプタリストへの切り替えは、以下のように、RxDMA がサスペンド状態のときに行われます。

- 一般的に、RxDMA で事前にディスクリプタが用意されます。
- ディスクリプタが利用できないために RxDMA がサスペンド状態に移行した場合は、重大な障害が発生します (いっぱいになったディスクリプタ/バッファがソフトウェアで解放できない)。問題がすぐに解決しない場合、Rx FIFO のオーバーフローによりフレームが失われます。そのため、ソフトウェアにより新しいディスクリプタリストを作成して RxDMA をプログラムすることにより、STOP 状態に移行することなく、すぐにそれを使い始めることができます。

57.9.7 AHB クロック周波数を切り替えるためのプログラミングガイドライン

(ソフトリセットまたはハードリセットを適用しないで) AHB クロック周波数を動的に変更するには、次の手順に従ってください。

1. 送信 DMA を無効にして (妥当な場合)、前のフレーム送信の完了を待ちます。フレーム送信が完了すると、Tx FIFO はエンプティになり、Tx DMA は STOP 状態に移行します。Tx FIFO ステータスは Tx キューデバッグレジスタ (ETH_MTLTXQDR) で規定され、DMA のステータスは デバッグステータスレジスタ (ETH_DMADSR) で規定されます。
2. 動作モード設定レジスタ (ETH_MACCR) で適切なビットをクリアすることによって、MAC トランスミッタと MAC レシーバを無効にします。
3. Rx FIFO のデータがシステムメモリに転送されたことを確認した後、受信 DMA を無効にします (妥当な場合)。Rx FIFO エンプティ状態は Rx キューデバッグレジスタ (ETH_MTLRXQDR) で規定されます。

4. アプリケーションでレジスタの読出しまたは書込み操作が行われていないことを確認します。
5. AHB クロックの周波数を変更します。
6. MAC トランスミッタまたは MAC レシーバと送信 DMA または受信 DMA を有効にします。これらの手順で、クロック周波数の切り替え時に Tx FIFO または Rx FIFO に有効なデータがないことを確認し、データの破損を防止します。

57.9.8 MII リンク状態遷移に関するプログラミングガイドライン

リンクがダウンしているときに送信および受信クロックが動作している場合

送信および受信クロックが動作しているときにリンクがダウンした場合、次の手順を完了します。

1. **チャネル制御レジスタ (ETH_DMCCR)** のビット 0 (ST) をクリアすることで送信 DMA (適用できる場合) を無効にします。
2. **動作モード設定レジスタ (ETH_MACCR)** の RE ビットをクリアすることで MAC レシーバを無効にします。
3. 前のフレーム送信の完了を待ちます。Tx キューデバッグレジスタ (**ETH_MTLTXQDR**) の適切なビットを読み出すことでこれをチェックできます (TRCSTS[1:0] が 01 ではない)。または
より速いエンプティ操作を行うために Tx FIFO を一掃します。
4. **動作モード設定レジスタ (ETH_MACCR)** レジスタの TE ビットをクリアすることで MAC トランスミッタを無効にします。
5. Tx および Rx キュー両方がエンプティであること (Tx キューデバッグレジスタ (**ETH_MTLTXQDR**) で TXQSTS が 0、Rx キューデバッグレジスタ (**ETH_MTLRXQDR**) で RXQSTS[1:0] が 00 にセットされていること) を確認します。
6. リンクがアップになってから、PHY レジスタを読み出して最新の設定を特定し、それに応じて MAC レジスタをプログラムします。
7. Tx DMA を開始することで動作を再開します。続いて、MAC トランスミッタとレシーバを有効にします。
Rx DMA は有効にする必要はありません。レシーバが無効になっているため、Rx FIFO にデータはありません。

リンクがダウンしているときに送信および受信クロックが停止している場合

リンクがダウンして、送信および受信クロックが停止している場合、次の手順を完了します。

1. **動作モード設定レジスタ (ETH_MACCR)** の RE および TE ビットをクリアすることによって、MAC トランスミッタとレシーバを無効にします。クロックがないため、これによる影響はたちにはありません。
2. リンクがアップになり、クロックが復旧するまで待ちます。
3. 送信／受信クロックが停止したときに進行中だった操作がある場合、部分的なフレームの転送が完了するまで待ちます。これは、**デバッグレジスタ (ETH_MACDR)** を読み出すことでチェックできます (すべてのビットを 0 にセットする必要があります)。一部の古いパケットは MAC トランスミッタが停止した状態のまま TXFIFO に残る場合があります。
4. PHY レジスタを読み出して最新の動作モードを特定し、それに応じて MAC レジスタをプログラムします。
5. RE および TE ビットをセットすることによって、MAC トランスミッタとレシーバを再起動します。

57.9.9 IEEE 1588 タイムスタンプに関するプログラミングガイドライン

システム時間生成の初期化

タイムスタンプ機能は、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 0 をセットすることによって有効にできます。ただし、このビットをセットした後に、タイムスタンプカウンタを初期化する必要があります。ペリフェラルの初期化を実行するには、次の手順を完了します。

1. **割込み有効レジスタ (ETH_MACIER)** のビット 12 をクリアすることでタイムスタンプトリガ割込みをマスクします。
2. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 0 をセットしてタイムスタンプを有効にします。
3. PTP クロック周波数に基づいて、**サブセカンドインクリメントレジスタ (ETH_MACSSIR)** をプログラムします。
4. 密補正法を使用する場合、**タイムスタンプ加数レジスタ (ETH_MACTSAR)** をプログラムして**タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 5 をセットします。
5. ビット 5 がクリアされるまで、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** にポーリングします。
6. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 1 をプログラムして、密更新法（必要な場合）を選択します。
7. **システム時間秒更新レジスタ (ETH_MACSTSUR)** および**システム時間ナノ秒更新レジスタ (ETH_MACSTNUR)** を適切な時間値でプログラムします。
8. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 2 をセットします。
タイムスタンプカウンタは、タイムスタンプ更新レジスタに書き込まれた値で初期化されると開始します。ワンステップタイムスタンプが必要な場合：
 - a) TDES3 コンテキストディスクリプタのビット 27 をプログラムすることでワンステップタイムスタンプを有効にします。
 - b) PDelay_Req PTP メッセージの補正フィールドを更新するために**タイムスタンプ入力非対称補正レジスタ (ETH_MACTSIACR)** をプログラムします。
9. 適切なタイムスタンプングのためには、MAC レシーバおよびトランスミッタを有効にします。

注： **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 0 をクリアすることによってタイムスタンプ動作が無効にされた場合、タイムスタンプ動作を再開するには、これらすべての手順を繰り返します。

システム時間の補正

システム時間を一気に同期または更新するには（粗補正法）、次の手順を完了します。

1. タイムスタンプ更新レジスタ（**システム時間秒更新レジスタ (ETH_MACSTSUR)** および**システム時間ナノ秒更新レジスタ (ETH_MACSTNUR)**）でオフセットをセットします（正または負）。
2. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 3 (TSUPDT) をセットします。
TSUPDT ビットがクリアされると、タイムスタンプ更新レジスタの値がシステム時間に加えられるか、差し引かれます。

システム時間を同期または更新して、システム時間のジッタを減らすには（密補正法）、次の手順を完了します。

1. **セクション : システム時間レジスタモジュール**で説明されているアルゴリズムを使用して、システム時間の～インクリメントまたはデクリメントの割合を計算します。
2. **タイムスタンプ加数レジスタ (ETH_MACTSAR)** を新しい値で更新して、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** レジスタのビット 5 をセットします。
3. 加数レジスタの新しい値として有効にしたい時間を待ちます。この操作は、システム時間が目標値に達した後、タイムスタンプトリガ割込みを有効にして実行できます。
4. **PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および**PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)** で、必要な目標時間をプログラムします。
5. **割込み有効レジスタ (ETH_MACIER)** のビット 12 でタイムスタンプ割込みを有効にします。
6. レジスタ **タイムスタンプ制御レジスタ (ETH_MACTSCR)** のビット 4 をセットします。
7. このトリガによって割込みが生成されたときには、**割込みステータスレジスタ (ETH_MACISR)** を読み出します。
8. **タイムスタンプ加数レジスタ (ETH_MACTSAR)** を古い値で再プログラムして、再びビット 5 をセットします。

57.9.10 PTP オフロード機能に関するプログラミングガイドライン

PTP 同期メッセージの自動周期的生成を有効にするためのプログラミングガイドライン

PTP 同期メッセージの自動周期的生成を有効にするには、以下の手順に従ってください。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 0、1、1 にプログラムし、ノードを通常マスタまたは境界マスタとして設定します（トランスペアレントマスタの場合は 1、1、1）。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、出力 PTP 同期メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の ASYNCEN ビットをプログラムし、PTP 同期メッセージの周期的生成を有効にします。
4. 出力 PTP 同期メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** の 80 ビットのソースポートアイデンティティをプログラムします。
5. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** の LSI フィールドをプログラムし、PTP 同期メッセージの周期性をプログラムします。
たとえば、1 の値は 2^1 に対応し、2 秒ごとの PTP 同期メッセージに変換されます。0xFF (-1 の 2 の補数) は 2^{-1} に対応し、0.536 秒ごとの PTP 同期メッセージに変換されます。
6. **割込み有効レジスタ (ETH_MACIER)** の TSIE ビットをプログラムし、タイムスタンプ割込みの生成を有効にします。
7. **タイムスタンプステータスレジスタ (ETH_MACTSSR)** の TXTSSIS ビットをセットして生成される sbd_intr_o 割込みを待ちます。これは、PTP 同期メッセージのタイムスタンプが **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** および **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** にキャプチャされることを示します。

PTP Pdelay_Req メッセージの周期的生成を有効にするためのプログラミングガイドライン

PTP Pdelay_Req メッセージの自動周期的生成を有効にするには、以下の手順に従ってください。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 1、0、1 にプログラムし、ノードをトランスペアレントスレーブとして設定します (トランスペアレントマスタの場合は 1、1、1、またはピアツーピアトランスペアレントの場合は 3、X、X)。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、出力 PTP Pdelay_Req メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の APDREQEN ビットをプログラムし、PTP Pdelay_Req メッセージの周期的生成を有効にします。
4. 出力 PTP Pdelay_Req メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** の 80 ビットのソースポートアイデンティティをプログラムします。
5. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** の LMPDRI フィールドをプログラムし、PTP Pdelay_Req メッセージの周期性をプログラムします。
たとえば、1 の値は 2^1 に対応し、2 秒ごとの PTP Pdelay_Req メッセージに変換されます。0xFF (-1 の 2 の補数) は 2^{-1} に対応し、0.536 秒ごとの PTP Pdelay_Req メッセージに変換されます。
6. **割込み有効レジスタ (ETH_MACIER)** の TSIE ビットをプログラムし、タイムスタンプ割込みの生成を有効にします。
7. **タイムスタンプステータスレジスタ (ETH_MACTSSR)** の TXTSSIS ビットをセットして生成される sbd_intr_o 割込みを待ちます。これは、PTP 同期メッセージのタイムスタンプが **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** および **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** にキャプチャされることを示します。

通常マスタまたは境界マスタモード用に PTP 応答メッセージの生成を有効にするためのプログラミングガイドライン

通常マスタまたは境界マスタモード用に PTP 応答メッセージの生成を有効にするには、以下の手順に従ってください (PTP Delay_Req メッセージに回答して周期的 PTP 同期メッセージと PTP Delay_Resp メッセージが生成されます)。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 0、1、1 にプログラムします。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、入力 PTP Delay_Req メッセージと照合して出力 PTP Delay_Resp メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. 入力 PTP Delay_Req メッセージと照会して出力 PTP Delay_Resp メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** で 80 ビットのソースポートアイデンティティをプログラムします。
4. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** で DRSYNCR および LSI フィールドをプログラムします。両方のフィールドの合計が、PTP Delay_Resp メッセージの logMinDelayReqInterval フィールドで更新されます。

通常スレーブまたは境界スレーブモード用に PTP 応答メッセージの生成を有効にするためのプログラミングガイドライン

通常スレーブまたは境界スレーブモード用に PTP 応答メッセージの生成を有効にするには、以下の手順に従ってください (PTP 同期メッセージに回答して PTP Delay_Req メッセージが生成されます)。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 0、0、1 にプログラムします。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、入力 PTP 同期メッセージと照合して出力 PTP Delay_Req メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. 入力 PTP 同期メッセージと照合して出力 PTP Delay_Req メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** で 80 ビットのソースポートアイデンティティをプログラムします。
4. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** で DRSYNCR フィールドをプログラムして、いくつかの受信 PTP 同期メッセージに回答して 1 つの PTP Delay_Req メッセージを生成するかを示します。

トランスペアレントスレーブモード用に PTP 応答メッセージの生成を有効にするためのプログラミングガイドライン

トランスペアレントスレーブモード用に PTP 応答メッセージの生成を有効にするには、以下の手順に従ってください (PTP 同期メッセージに回答して PTP Delay_Req メッセージが生成され、PTP Pdelay_Req メッセージに回答して PTP Pdelay_Resp メッセージが生成され、周期的 PTP Pdelay_Req メッセージが生成されます)。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 1、0、1 にプログラムします。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、入力 PTP 同期メッセージまたは Pdelay_Req メッセージと照合して出力 PTP Delay_Req または Pdelay_Resp または Pdelay_Req メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. 入力 PTP 同期メッセージまたは PdelayReq メッセージまたは Pdelay_Req メッセージと照合して出力 PTP Delay_Req または Pdelay_Resp または Pdelay_Req メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** で 80 ビットのソースポートアイデンティティをプログラムします。
4. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** で DRSYNCR と LMPDRI フィールドをプログラムして、いくつかの受信 PTP 同期メッセージに回答して 1 つの PTP Delay_Req メッセージを生成するかと、PTP Pdelay_Req メッセージの周期性を示します。
5. **割込み有効レジスタ (ETH_MACIER)** の TSIE ビットをプログラムし、タイムスタンプ割込みの生成を有効にします。
6. **タイムスタンプステータスレジスタ (ETH_MACTSSR)** の TXTSSIS ビットをセットして生成される sbd_intr_o 割込みを待ちます。これは、出力 PTP Pdelay_Req および Pdelay_Resp メッセージ用に、PTP 同期メッセージのタイムスタンプが **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** および **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** にキャプチャされることを示します。

トランスペアレントマスタモード用に PTP 応答メッセージの生成を有効にするためのプログラミングガイドライン

トランスペアレントマスタモード用に PTP 応答メッセージの生成を有効にするには、以下の手順に従ってください (PTP Delay_Req メッセージに응答して PTP Delay_Resp メッセージが生成され、PTP Pdelay_Req メッセージに응答して PTP Pdelay_Resp メッセージが生成され、周期的 PTP Pdelay_Req メッセージまたは同期メッセージが生成されます)。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 1、1、1 にプログラムします。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、入力 PTP Delay_Req メッセージまたは Pdelay_Req メッセージと照会して出力 PTP Delay_Resp または Pdelay_Resp または Pdelay_Req メッセージまたは同期メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. 入力 PTP Delay_Req メッセージまたは Pdelay_Req メッセージと照会して出力 PTP Delay_Resp または Pdelay_Resp または Pdelay_Req メッセージまたは同期メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** で 80 ビットのソースポートアイデンティティをプログラムします。
4. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** で DRSYNCR、LSI、LMPDRI ビットフィールドをプログラムします。DRSYNCR と LSI の合計は PTP Delay_Resp メッセージの logMinDelayReqInterval フィールドで更新されます。また、PTP 同期メッセージまたは Pdelay_Req メッセージの周期性を示します。
5. **割込み有効レジスタ (ETH_MACIER)** の TSIE ビットをプログラムし、タイムスタンプ割込みの生成を有効にします。
6. **タイムスタンプステータスレジスタ (ETH_MACTSSR)** の TXTSSIS ビットをセットして生成される sbd_intr_o 割込みを待ちます。これは、出力 PTP 同期メッセージ、Pdelay_Req および Pdelay_Resp メッセージ用に、PTP 同期メッセージのタイムスタンプが **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** および **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** にキャプチャされることを示します。

ピアツーピアトランスペアレントモード用に PTP 応答メッセージの生成を有効にするためのプログラミングガイドライン

ピアツーピアトランスペアレントモード用に PTP 応答メッセージの生成を有効にするには、以下の手順に従ってください (PTP Pdelay_Req メッセージに응答して PTP Pdelay_Resp メッセージが生成され、周期的 PTP Pdelay_Req メッセージが生成されます)。

1. **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の SNAPTYPSEL、TSMSTRENA、TSEVNTENA フィールドをそれぞれ 3、X、X にプログラムします。
2. **PTP オフロード制御レジスタ (ETH_MACPOCR)** の PTOEN ビットおよび DN フィールドをプログラムし、入力 PTP Pdelay_Req メッセージと照会して出力 PTP Pdelay_Resp メッセージで送信するために PTP オフロード機能とドメイン番号を有効にします。
3. 入力 PTP Pdelay_Req メッセージと照会して出力 PTP Pdelay_Resp メッセージで送信するために、**PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)**、**PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)**、**PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)** で 80 ビットのソースポートアイデンティティをプログラムします。
4. **ログメッセージ間隔レジスタ (ETH_MACLMIR)** の LMPDRI フィールドをプログラムし、PTP Pdelay_Req メッセージの周期性を示します。
5. **割込み有効レジスタ (ETH_MACIER)** の TSIE ビットをプログラムし、タイムスタンプ割込みの生成を有効にします。

6. **タイムスタンプステータスレジスタ (ETH_MACTSSR)** の TXTSSIS ビットをセットして生成される `sbd_intr_o` 割込みを待ちます。これは、出力 PTP `Pdelay_Req` および `Pdelay_Resp` メッセージ用に、PTP 同期メッセージのタイムスタンプが **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** および **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** にキャプチャされることを示します。

57.9.11 省電力型 Ethernet (EEE) に関するプログラミングガイドライン

Tx LPI モードの移行と終了

EEE を使用すると、IEEE 802.3 メディアアクセス制御 (MAC) のサブレイヤと物理レイヤのファミリを低電力アイドル (LPI) モードで一緒に動作させることができます。送信パスでは、ソフトウェアは **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の LPIEN ビットをセットして、MAC に送信を停止して LPI プロトコルを開始するように示すしなければなりません。

MAC 初期化中に次の手順を完了します。

1. MDIO インタフェースを通じて PHY レジスタを読み出し、リモートエンドに EEE 機能があるかどうかチェックします。次に、タイマの値をネゴシエイトします。
2. MDIO インタフェースを通じて PHY レジスタをプログラムします (LPI モードで Rx クロックを停止するかどうかを PHY に通知する `RX_CLK_stoppable` ビットを含む)。
3. **LPI タイマ制御レジスタ (ETH_MACLTR)** でビット 25 から 16 と、ビット 15 を 0 にプログラムします。
4. MDIO インタフェースを使用して PHY リンクステータスを読み出し、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** のビット 17 を更新します。
それに応じて、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** を更新します。この更新は、PHY チップのリンクステータスに変化するとき、必ず実行してください。
5. CSR スレーブポートにアクセスするために使用されるクロックの周波数に従って **1 マイクロ秒ティックカウンタレジスタ (ETH_MAC1USTCR)** をプログラムします。
6. MAC が LPI 状態に移行する前に待つ必要があるアイドル時間を **LPI 移行タイマレジスタ (ETH_MACLETR)** の LPIET ビットにプログラムします。
7. **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の LPITE および LPITXA (ビット 20 から 19) をセットして、LPI 状態から LPI 自動移行および MAC 自動終了を有効にします。
8. **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** のビット 16 をセットして、MAC トランスミッタを LPI 状態にします。
すべてのスケジュールされたパケットが完了すると、MAC は LPI 状態に移行します。LPIET ビットで示された時間はアイドル状態のままとなります。LPI 状態に移行した後、TLPIEN (ビット 0) をセットします。
9. パケット送信がスケジュールされているとき (TxDMA がアイドル状態を終了したとき、または ATI や MTI インタフェースにパケットがあるとき)、MAC トランスミッタは自動的に LPI 状態を終了します。TLPIEX 割込みステータスビットがセットされるまで TWT の時間を待ってから、パケット送信を再開します。
10. LPIET の時間、MAC トランスミッタがアイドル状態のままだった場合、LPI 状態に再び移行します。そして、TLPIEN ビットがセットされ、移行と終了のサイクルが続きます。
11. アプリケーションが自動移行/終了モードを上書きして、MAC トランスミッタの LPI 状態を直接終了させる必要がある場合、LPIEN ビットをリセットします。

注 : Tx FIFO にキューイングされたすべてのフレームの送信が完了してから MAC が LPI 状態に移行するには、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** で LPITXA ビットをセットします。

LIP 状態でのシステムの残りの部分への CSR クロック供給または電力供給をオフにするには、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の TLPIEN 割込みが生成されるのを待ちます。LIP 状態を終了するには、ステップ 6 を実行する前にクロックを復旧してください。

LPI モードでの CSR クロックのゲートオフ

MAC が低電力アイドル (LPI) モードのとき、節電のために CSR クロックをゲートオフすることができます。

Rx LPI モードでの CSR クロックのゲートオフ

MAC が PHY から LPI パターンを受信すると、次の処理が行われます。

1. MAC RX が LPI モードに移行し、Rx LPI エントリ割込みステータス (**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の RLPIEN 割込み) がセットされます。
2. 割込みピン (sbd_intr_o) がアサートされます。ホストが **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** を読み出すと、sbd_intr_o 割込みがクリアされます。

sbd_intr_o 割込みがアサートされて MAC Tx が LPI モードに移行すると、CSR クロックのゲートオフが可能になります。CSR クロックのゲートオフ時に MAC TX が LPI モードに移行していない場合、CSR で MAC トランスミッタのイベントがレポートされないか、更新されません。CSR クロックを復旧するには、PHY からの LPI 終了の通知を待つてください。この後に、MAC は lpi_intr_o に LPI 終了割込みをアサートします (clk_rx_i に同期)。lpi_intr_o 割込みは、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** が読み出されたときにクリアされます。

Tx LPI モードでの CSR クロックのゲートオフ

LPI 制御およびステータスレジスタ (ETH_MACLCSR) のビット 16 (LPIEN) がセットされている場合、次の処理が行われます。

1. 送信 LPI エントリ割込み (**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の TLPIEN ビット) がセットされます。
2. 割込みピン (sbd_intr_o) がアサートされます。ホストが **LPI 制御およびステータスレジスタ (ETH_MACLCSR)** を読み出すと、sbd_intr_o 割込みがクリアされます。

sbd_intr_o 割込みがアサートされて MAC RX が LPI モードに移行すると、CSR クロックのゲートオフが可能になります。CSR クロックのゲートオフ時に MAC RX が LPI モードに移行していない場合、CSR で MAC レシーバのイベントがレポートされないか、更新されません。CSR クロックを復旧するには、MAC が TX LPI モードを終了したときに CSR クロックをオンにします。CSR クロックが再開してから、**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** のビット 16 (LPIEN) をリセットして MAC の LPI モードを終了します。

57.9.12 柔軟性の高い Pulse-Per-Second (PPS) 出力に関するプログラミングガイドライン

PPS でのシングルパルスの生成

PPS でのシングルパルスの生成方法 :

1. **PPS 制御レジスタ (ETH_MACPPSCR)** で TRGTMODSEL[1:0] ビットを 11 または 10 (割込み) にプログラムします。これで、MAC に目標時間レジスタ (**PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)**) を PPS 信号出力の開始時間として使用するよう命令します。
2. 目標時間レジスタ (レジスタ **PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)**) で開始時間値をプログラムします。
3. **PPS 幅レジスタ (ETH_MACPPSWR)** レジスタで PPS 信号出力の幅をプログラムします。
4. **PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCMD[3:0] を 0001 にプログラムします。これで、MAC に目標時間レジスタでプログラムされた時間に PPS 信号出力でシングルパルスを生成するよう命令します。

PPS での次のパルスの生成

PPSCMD が実行されると (PPSCMD ビット = 0)、プログラムされた開始時間が経過する前に、開始キャンセルコマンド (PPSCMD=0011) を示すことでパルス生成をキャンセルできます。予め次のパルスの動作をプログラムすることもできます。次のパルスをプログラムする方法 :

1. 目標時間レジスタで次のパルスの開始時間をプログラムします。この時間は、立下りエッジが前のパルスに発生する時間より大きくする必要があります。
 2. **PPS 幅レジスタ (ETH_MACPPSWR)** で次の PPS 信号出力の幅をプログラムします。
 3. **PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCMD[3:0] ビットをプログラムして、前のパルスがネゲートされた後にシングルパルスを生成します。これで、MAC に目標時間レジスタでプログラムされた時間に PPS 信号出力でシングルパルスを生成するよう命令します。
- 前のパルスがローになる前に、このコマンドが示された場合、新しいコマンドは前のコマンドを上書きして、ペリフェラルによって 1 つだけ拡張されたパルスが生成されます。

PPS でのパルス列の生成

PPS でのパルス列の生成方法 :

1. **PPS 制御レジスタ (ETH_MACPPSCR)** で TRGTMODSEL[1:0] ビットを 11 または 10 (割込み) にプログラムします。これで、MAC に目標時間レジスタ (**PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)**) を PPS 信号出力の開始時間として使用するよう命令します。
2. 目標時間レジスタ (レジスタ **PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)**) で開始時間値をプログラムします。
3. **PPS 間隔レジスタ (ETH_MACPPSIR)** で PPS 信号出力でのパルス列の間隔をプログラムします。
4. **PPS 幅レジスタ (ETH_MACPPSWR)** で PPS 信号出力の幅をプログラムします。
5. **PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCMD[3:0] ビットを 0010 にプログラムします。これで、MAC に目標時間レジスタでプログラムされた開始時間に PPS 信号出力でパルス列を生成するよう命令します。

デフォルトでは、PPS パルス列は「パルス列の同時停止」または「パルス列の即時停止」コマンドを発行して停止されない限り、フリーランニングです。

6. 目標時間レジスタで停止値をプログラムします。目標時間レジスタを再びプログラムする前に、PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR) の TSTRBUSY ビットがリセットされていることを確認します。
7. PPS 制御レジスタ (ETH_MACPPSCR) の PPSCMD[3:0] ビットを 0100 にプログラムして、手順 6 で指定されプログラムされた停止時間が経過した後に PPS 信号出力のパルス列を停止します。

パルス列は、PPSCMD[3:0] フィールドで 0101 をプログラムすることで、いつでも停止できます。

同様に、(手順 7 に示された) パルス列停止コマンドは、(手順 6 でプログラムされた) 時間が経過する前に PPSCMD[3:0] ビットを 0110 にプログラムすることでキャンセルできます。

パルス列の生成は、(手順 2 でプログラムされた) 開始時間が経過する前に PPSCMD[3:0] を 0011 にプログラムすることで停止できます。

PPS に影響しない割込みの生成

PPS 制御レジスタ (ETH_MACPPSCR) の TRGTMODSEL[1:0] ビットで、目標時間レジスタ (PPS 目標時間秒レジスタ (ETH_MACPPSTTSR) および PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)) をプログラムして、次のいずれかの操作が可能です。

- 割込みのみの生成
- 割込みおよび PPS の開始時間と停止時間の生成
- PPS の開始時間と停止時間のみの生成

目標時間レジスタをプログラムして割込みイベントのみを生成する方法 :

1. PPS 制御レジスタ (ETH_MACPPSCR) の TRGTMODSEL[1:0] ビットを 00 (割込み) にプログラムします。これで、MAC に目標時間レジスタを目標時間の割込みに使用するように命令します。
2. 目標時間レジスタで目標時間値をプログラムします。これで、MAC に目標時間が経過したときに割込みを生成するように命令します。

(たとえば PPS を制御するために) TRGTMODSEL[1:0] ビットが変更された場合、割込み生成は新しいモードと新しくプログラムされた目標時間レジスタ値で上書きされます。

注 : プログラムされた目標時間が システム時間秒レジスタ (ETH_MACSTSR) および システム時間ナノ秒レジスタ (ETH_MACSTNR) のシステム時間と比べて小さい (つまり、過去の時間に相当する) 場合、タイムスタンプステータスレジスタ (ETH_MACTSSR) の TSTRGTERR0 ビットがセットされます。

割込み有効レジスタ (ETH_MACIER) で TSIE ビットがセットされている場合、割込みが生成されず (sbd_intr_o)。

そのため、望ましくない割込みを避けるための正しい書き込み順序は次のようになります。

1. PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR) を参照してください。
2. PPS 目標時間秒レジスタ (ETH_MACPPSTTSR) を参照してください。
3. PPS 間隔レジスタ (ETH_MACPPSIR) を参照してください。
4. PPS 幅レジスタ (ETH_MACPPSWR) を参照してください。
5. PPS 制御レジスタ (ETH_MACPPSCR) の PPSCMD[3:0] および PPSCMD[3:0] および PPSEN0 ビットフィールド

57.9.13 TSO に関するプログラミングガイドライン

TCP セグメンテーションオフロード (TSO) エンジン、ハードウェアへの TCP セグメンテーション機能をオフロードするために使用されます。TSO をプログラムするには、TSE ビットをセットして TCP パケットのセグメンテーションを有効にし、ディスクリプタフィールドをプログラムして現在のパケット用に TSO を有効化します。

TSO をプログラムするには、次の手順に従います。

1. 対応する **チャネル送信制御レジスタ (ETH_DMACTXCR)** の TSE ビットをプログラムして、その DMA で TCP パケットのセグメンテーションを有効にします。
2. 通常の転送ディスクリプタ設定に加えて、次のディスクリプタフィールドをプログラムして現在のパケットに TSO を有効にしなければなりません。
 - a) TDES3 の TSE (ビット 18) を有効化します。
 - b) TDES3 のビット 17~0 で分割されていない TCP/IP パケットペイロードの長さと、TDES3 のビット 22~19 で TCP ヘッダをプログラムします。
 - c) 以下でセグメントの最大サイズをプログラムします。
 - **チャネル制御レジスタ (ETH_DMCCR)** の MSS[13:0]
 - またはコンテキストディスクリプタの MSSMSS[13:0] フィールドが **チャネル制御レジスタ (ETH_DMCCR)** とコンテキストディスクリプタの両方でプログラムされている場合、最新のソフトウェアによってプログラムされたシーケンスが考慮されます。
3. 分割されていない TCP/IP パケットヘッダは、先頭ディスクリプタのバッファ 1 に格納する必要があります。このバッファはペイロードのバイトを保持してはなりません。ペイロードはバッファ 2 と後続のディスクリプタのバッファに割り当てられます。

注意 : 非 TCP-IP パケットに対して TDES3 で TSE が有効になっている場合、結果は予想できません。

57.9.14 受信時の VLAN フィルタに関するプログラミングガイドライン

受信時に VLAN フィルタを実行するには、以下のシーケンスに従います。

1. 次のビットに **VLAN タグレジスタ (ETH_MACVTR)** をプログラムして、フィルタ方法を選択します。
 - ETV : 12 ビット VLAN タグ比較または 16 ビット VLAN タグ比較を有効にします。
 - VTHM : VLAN タグハッシュテーブル一致を有効にします。
 - ERIVLT : 内部 VLAN タグまたは外部 VLAN タグを有効にします (内部または外部 VLAN タグフィルタを有効にするには、EDVLP をセットしてダブル VLAN 処理を有効にする必要があります)。
 - ERSVLM : 受信 S-VLAN 一致または C-VLAN 一致を有効にします (S-VLAN 処理を有効にするには、ESVL をセットします)。
 - DOVLTC : タグ一致の VLAN タイプを無視します。
 - VTIM : 通常の VLAN タグ一致ではなく、VLAN タグ反転一致を有効にします。
2. 12 ビットまたは 16 ビット VLAN タグについては **VLAN タグレジスタ (ETH_MACVTR)** で VL ビットをプログラムします。
3. VLAN タグハッシュフィルタが有効になっている場合、**VLAN ハッシュテーブルレジスタ (ETH_MACVHTR)** をプログラムします。
 - ETV ビットがリセットされている場合、VLAN タグの計算された CRC-32 の上位 4 ビットが反転して使用され、**VLAN ハッシュテーブルレジスタ (ETH_MACVHTR)** の内容をインデックスします。
 - ETV ビットがセットされている場合、VLAN タグの計算された CRC-32 の上位 4 ビットが使用され、**VLAN ハッシュテーブルレジスタ (ETH_MACVHTR)** の内容をインデックスします。

たとえば、ETV ビットがセットされている場合、0b1000 のハッシュ値は、VLAN ハッシュテーブルのビット 8 を選択します。ETV ビットがリセットされている場合、0b1000 のハッシュ値は、VLAN ハッシュテーブルのビット 7 を選択します。

57.10 ディスクリプタ

57.10.1 ディスクリプタの概要

Ethernet ペリフェラルでは、DMA はリンクされたディスクリプタのリストに基づいてデータを転送します。アプリケーションは、システムメモリ (SRAM) 内にディスクリプタを作成します。次の 2 種類のディスクリプタをサポートしています。

- **通常ディスクリプタ**

通常ディスクリプタは、パケットデータに使用され、送信するパケットに適用できる制御情報を提供します。

- **コンテキストディスクリプタ**

コンテキストディスクリプタは、送信するパケットに適用できる制御情報を提供するために使用されます。

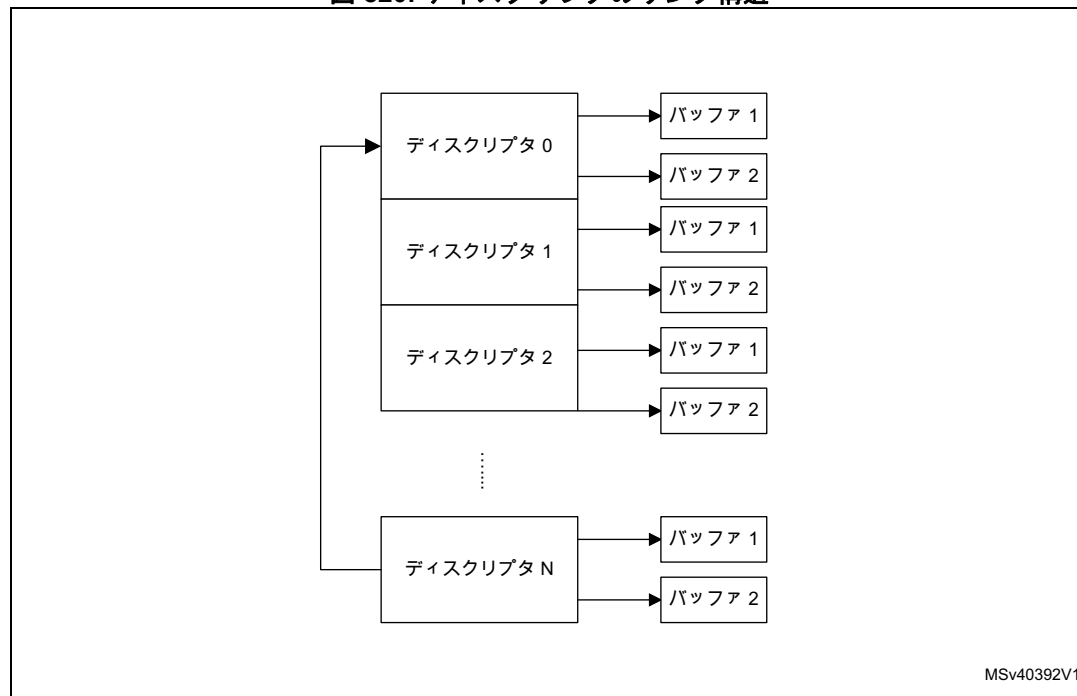
各通常ディスクリプタには、2 つのバッファと 2 つのアドレスポイントが含まれます。これらのバッファは、アダプタポートにさまざまなタイプのメモリ管理構成との互換性を持たせます。

1 パケットに使用できるディスクリプタの数に制限はありません。

57.10.2 ディスクリプタの構造

Ethernet ペリフェラルは、DMA ディスクリプタのリング構造をサポートします。

図 820. ディスクリプタのリング構造



リング構造では、ディスクリプタが**チャンネル制御レジスタ (ETH_DMCCR)** の DSL フィールドでプログラムされた 32 ビットワード数で分けられます。アプリケーションは、合計のリングの長さをプログラムする必要があり、これは DMA チャンネルの次のレジスタでのリング範囲内のディスクリプタの合計数になります。

- **チャンネル Tx ディスクリプタリング長レジスタ (ETH_DMACTXRLR)**)
- **チャンネル Rx ディスクリプタリング長レジスタ (ETH_DMACRXRLR)**)

チャンネル Tx ディスクリプタテイルポインタレジスタ (ETH_DMACTXDTPR) や**チャンネル Rx ディスクリプタテイルポインタレジスタ (ETH_DMACRXDTPR)** には、ディスクリプタアドレスのポインタ (N) が含まれます。ベースアドレスと現在のディスクリプタポインタによって、DMA が処理できる現在のディスクリプタのアドレスが決定されます。ディスクリプタのテイルポインタによって示されたものより最大 1 か所少ないディスクリプタ (N-1) が DMA によって所有されます。DMA は、次のいずれかの条件が発生するまでディスクリプタの処理を続けます。

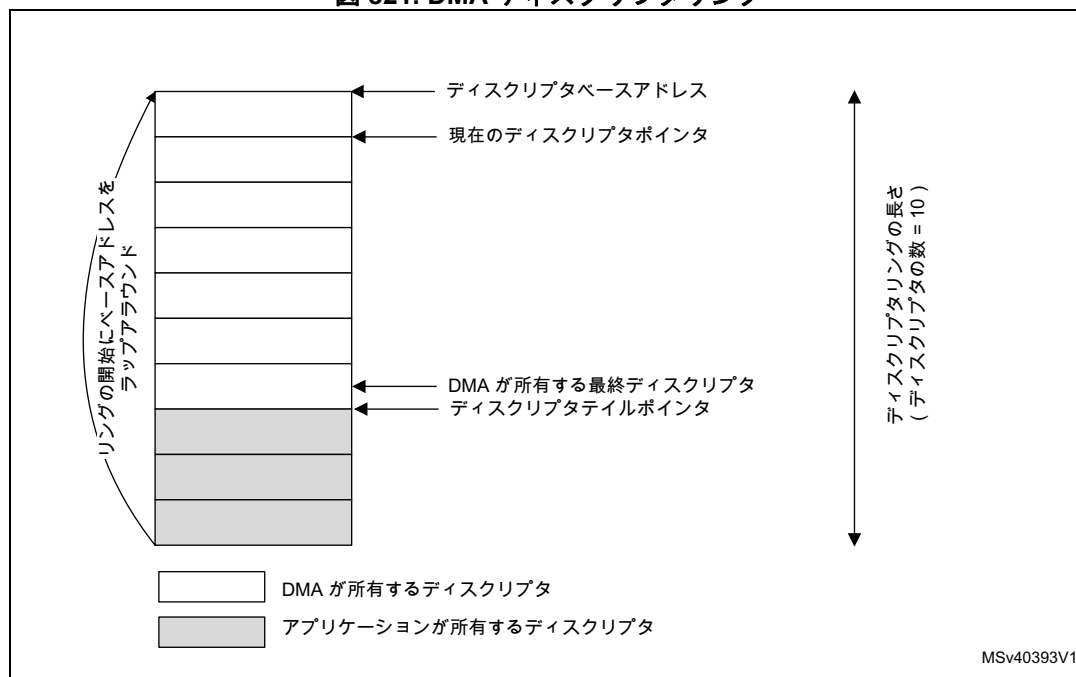
Current Descriptor Pointer == Descriptor Tail Pointer;

この条件が発生すると、DMA はサスペンド状態に移行します。アプリケーションは、ディスクリプタのテイルポインタレジスタに書き込み動作を実行し、次の条件を満たすようにテイルポインタを更新する必要があります。

Current Descriptor Pointer < Descriptor Tail Pointer;

図 821 : DMA ディスクリプタリングに示すように、リングの終わりに達すると、DMA は自動的にベースアドレスを巻き戻します。

図 821. DMA ディスクリプタリング



アプリケーションによって所有されるディスクリプタの場合、DES3 の OWN ビットが 0 にリセットされます。

DMA によって所有されるディスクリプタの場合、OWN ビットが 1 にセットされます。

最初に、アプリケーションがディスクリプタを 1 つだけ所有している場合、最終ディスクリプタアドレス (テイルポインタ) はディスクリプタベースアドレス + 1 にセットされます。次に、DMA は先頭ディスクリプタを処理して、アプリケーションがテイルポインタをインクリメントするのを待ちます。

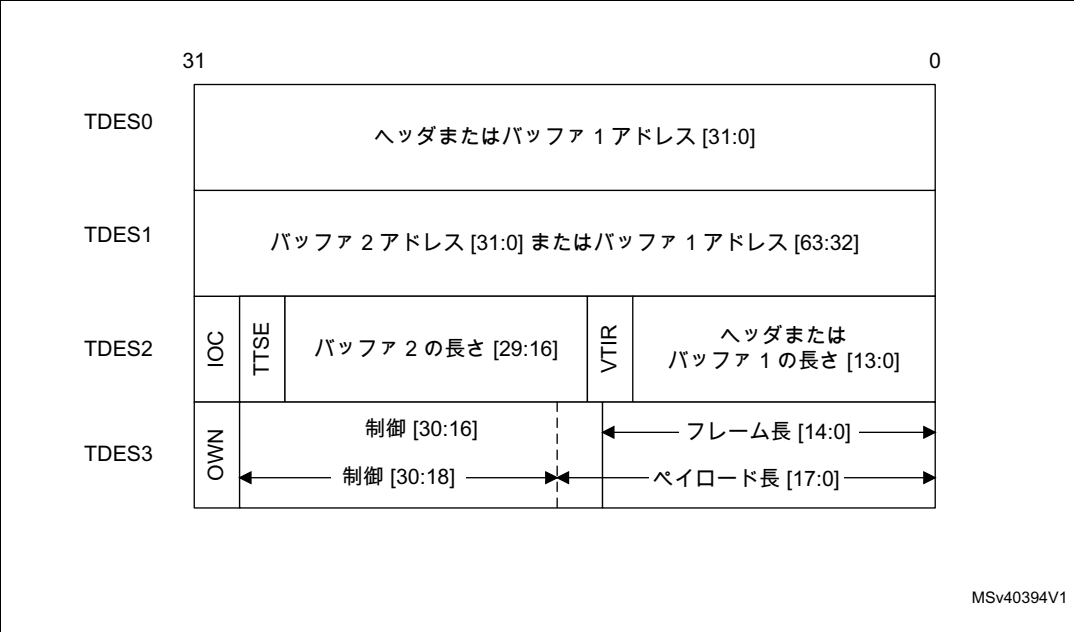
57.10.3 送信ディスクリプタ

Ethernet ペリフェラルの DMA は、送信パケットに最低 1 つのディスクリプタを必要とします。2 つのバッファ、2 つのバイトカウントバッファ、2 つのアドレスポインタに加えて、送信ディスクリプタ機能は、送信パケット単位で MAC 動作を管理するために使用できるフィールドを制御します。送信通常ディスクリプタには、次の 2 つのフォーマットがあります。読み出しフォーマットと書き戻しフォーマットです。

送信通常ディスクリプタ（読み出しフォーマット）

図 822 に、送信通常ディスクリプタの読み出しフォーマットを示します。表 653～表 656 では、すべての送信通常ディスクリプタ（読み出しフォーマット）について詳細に説明しています。

図 822. 送信ディスクリプタ（読み出しフォーマット）



- TDES0 通常ディスクリプタ（読み出しフォーマット）

表 653. TDES0 通常ディスクリプタ（読み出しフォーマット）

ビット	名前	説明
31:0	BUF1AP	バッファ 1 アドレスポインタまたは TSO ヘッダアドレスポインタ 次のビットがセットされているとき、これらのビットはバッファ 1 または TSO ヘッダのアドレスポインタを示します。 – TDES3 の TSE ビット – TDES3 の FD ビット

- TDES1 通常ディスクリプタ (読み出しフォーマット)

表 654. TDES1 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31:0	BUF2AP	バッファ 2 またはバッファ 1 アドレスポインタ : ディスクリプタのリング構造が使用されているとき、これらのビットはバッファ 2 の物理アドレスを示します。バッファアドレスの整列に制約はありません。

- TDES2 通常ディスクリプタ (読み出しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOC	TTSE	B2L													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTIR		HL または B1L													

表 655. TDES2 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31	IOC	完了割込み : このビットは、現在のパケット送信が完了したときに チャンネルステータスレジスタ (ETH_DMCSR) で TI ビットをセットします。
30	TTSE	送信タイムスタンプ有効化 このビットは、ディスクリプタで参照される送信パケット用の IEEE1588 タイムスタンプを有効にします。
29:16	B2L	バッファ 2 の長さ ドライバがこのフィールドをセットします。セット時、このフィールドはバッファ 2 の長さを示します。
15:14	VTIR	VLAN タグの挿入または書き換え : これらのビットは、パケットを送信する前に VLAN タグ付けまたはタグ削除を実行するよう MAC に要求します。VLAN タグの挿入、書き換え、または削除がパケットに有効である場合、アプリケーションは CRC パッド制御ビットを適切にセットする必要があります。これらのビットの値は次のとおりです。 00 : VLAN タグを追加しません。 01 : 送信前にパケットから VLAN タグを削除します。このオプションは、VLAN パケットでのみ使用する必要があります。 10 : VLAN 内包レジスタ (ETH_MACVIR) でプログラムされたタグ値またはコンテキストディスクリプタで VLAN タグを挿入します。 11 : VLAN 内包レジスタ (ETH_MACVIR) でプログラムされたタグ値またはコンテキストディスクリプタでパケットの VLAN タグを書き換えます。このオプションは、VLAN パケットでのみ使用する必要があります。

表 655. TDES2 通常ディスクリプタ (読出しフォーマット)

ビット	名前	説明
13:0	HL または B1L	<p>ヘッダの長さまたはバッファ 1 の長さ</p> <p>ヘッダの長さの場合、ビット [9:0] のみ考慮されます。ビット 13~0 はバッファ 1 の長さにのみ適用できます。</p> <p>TCP セグメンテーションオフロード機能が TDES3 の TSE ビットを通じて有効になっている場合、このフィールドはヘッダの長さと同しくなります。TSE ビットが TDES3 でセットされている場合、ヘッダの長さには Ethernet ソースアドレスから TCP ヘッダの終了までの長さ (バイト単位) が含まれます。TSO 機能でサポートされる最大ヘッダ長は 1023 バイトです。</p> <p>TCP セグメンテーションオフロード機能が有効ではない場合、このフィールドはバッファ 1 の長さと同しくなります。</p>

• TDES3 通常ディスクリプタ (読み出しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	CPC	SAIC	THL	TSE	TPL							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT/L															

表 656. TDES3 通常ディスクリプタ (読出しフォーマット)

ビット	名前	説明
31	OWN	<p>オウンビット</p> <p>1 : DMA がディスクリプタを所有しています。</p> <p>0 : アプリケーションがディスクリプタを所有しています。</p> <p>DMA は、関連付けられたバッファで示されたデータの転送を完了した後、このビットをクリアします。</p>
30	CTXT	<p>コンテキストタイプ</p> <p>このビットは通常ディスクリプタでは 0 にセットする必要があります。</p>
29	FD	<p>先頭ディスクリプタ</p> <p>このビットがセットされた場合、バッファがパケットの最初のセグメントを含んでいることを示します。</p>
28	LD	<p>最終ディスクリプタ</p> <p>このビットがセットされた場合、バッファがパケットの最後のセグメントを含んでいることを示します。B1L または B2L フィールドは 0 以外の値でなければなりません。</p>

表 656. TDES3 通常ディスクリプタ (読出しフォーマット) (続き)

ビット	名前	説明
27:26	CPC	<p>CRC パッド制御</p> <p>このフィールドは、Tx パケットの CRC とパッド挿入を制御します。先頭ディスクリプタビット (TDES3[29]) がセットされた場合にのみ有効です。ビット [27:26] の値は次のとおりです。</p> <p>00 : CRC およびパッド挿入</p> <p>MAC は、60 バイト以上の長さの送信パケットの最後に巡回冗長検査 (CRC) を付加します。MAC は、60 バイト未満の長さのパケットには自動的にパディングと CRC を付加します。</p> <p>01 : CRC 挿入 (パッド挿入無効化)</p> <p>MAC は、送信パケットの最後に CRC を付加しますが、パディングを付加しません。アプリケーションは、パディングバイトが送信バッファから送信されているパケットに存在している、つまり送信バッファから送信されているパケットが 60 バイト以上の長さであることを確認する必要があります。</p> <p>10 : CRC 挿入の無効化</p> <p>MAC は、送信パケットの最後に CRC を付加しません。アプリケーションは、パディングおよび CRC バイトが送信バッファから送信されているパケットに存在していることを確認する必要があります。</p> <p>11 : CRC 書き換え</p> <p>MAC は、送信パケットの最後の 4 バイトを、再計算された CRC バイトで書き換えます。アプリケーションは、パディングおよび CRC バイトが送信バッファから送信されているパケットに存在していることを確認する必要があります。</p> <p>TSE ビットがセットされると、CRC およびパッド挿入がセグメンテーションに対して必ず行われるため、MAC はこのフィールドを無視します。</p>
25:23	SAIC	<p>SA 挿入制御</p> <p>これらのビットは、Ethernet パケットのソースアドレスフィールドに対して、MAC アドレス 0 レジスタに示された値の追加または書き換えを行うよう MAC に要求します。SA 挿入制御がパケットに対して有効である場合、アプリケーションは適切に CRC パッド制御ビットをセットする必要があります。</p> <p>ビット 25 は、ソースアドレスの挿入または書き換えに使用される MAC アドレスレジスタ (1 または 0) の値を指定します。</p> <p>次のリストではビット [24:23] の値を示しています。</p> <p>00 : ソースアドレスを含みません。</p> <p>01 : ソースアドレスを含む、または挿入します。信頼できる送信では、アプリケーションはソースアドレスなしでフレームを提供する必要があります。</p> <p>10 : ソースアドレスを書き換えます。信頼できる送信では、アプリケーションはソースアドレスありでフレームを提供する必要があります。</p> <p>11 : 予約済みです。</p> <p>これらのフィールドは、最初のセグメント制御ビット (TDES3 [29]) がセットされているとき有効です。</p>
22:19	THL	<p>THL : TCP ヘッダの長さ</p> <p>TSE ビットがセットされている場合、このフィールドには TCP/UDP ヘッダの長さが入ります。このフィールドの最小値は、TCP ヘッダの場合は 5 でなければなりません。THL 値は、UDP ヘッダの場合は 2 でなければなりません。このフィールドは、最初のディスクリプタについてのみ有効です。</p>
18	TSE	<p>TCP セグメンテーション有効化</p> <p>このビットがセットされた場合、DMA がパケットに TCP/UDP セグメンテーションを実行します。このビットは、FD ビットがセットされたときにのみ有効です。</p>

表 656. TDES3 通常ディスクリプタ (読出しフォーマット) (続き)

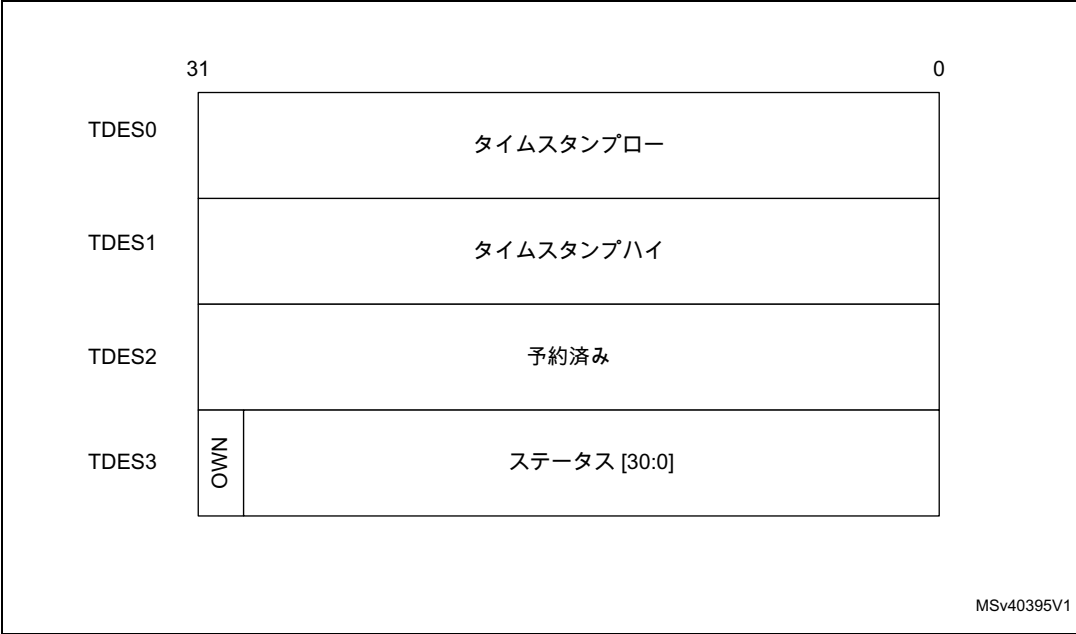
ビット	名前	説明
17:16	CIC/TPL	<p>チェックサム挿入制御または TCP ペイロード長 これらのビットは、チェックサムの計算と挿入を制御します。以下の値を取得できます。 00 : チェックサム挿入が無効化されます。 01 : IP ヘッダチェックサムの計算と挿入のみ有効化されます。 10 : IP ヘッダチェックサムとペイロードチェックサムの計算と挿入が有効ですが、擬似ヘッダチェックサムはハードウェアでは計算されません。 11 : IP ヘッダチェックサムとペイロードチェックサムの計算と挿入が有効であり、擬似ヘッダチェックサムはハードウェアで計算されます。 TSE ビットがリセットされているとき、このフィールドは有効です。TSE ビットがセットされているとき、TCP ペイロード長の上位ビット [17:16] を含みます。これにより、TCP パケット長フィールドが TDES3[17:0] にわたって拡大され、256 KB のパケット長をサポートします。</p>
15	TPL	<p>予約済みまたは TCP ペイロード長 TSE ビットがリセットされると、このビットは予約済みとなります。TSE ビットがセットされると、TCP ペイロード長 [17:0] のビット 15 となります。</p>
14:0	FL/TPL	<p>予約済みまたは TCP ペイロード長 TSE ビットがセットされている場合、このフィールドは TCP ペイロード長の下位 15 ビットと等しくなります。この長さに、Ethernet ヘッダや TCP/UDP/IP ヘッダの長さは含まれません。 TSE ビットがリセットされると、このビットは予約済みとなります。</p>

送信通常ディスクリプタ (書き戻しフォーマット)

書き戻しフォーマットは、対応するパケットの最終ディスクリプタにのみ適用できます。LD ビット (TDES3[28]) は、DMA が対応する送信パケットのステータスおよびタイムスタンプの情報を書き戻すディスクリプタにセットされます。

図 823 に、送信通常ディスクリプタの書き戻しフォーマットを示します。表 657～表 660 では、すべての送信通常ディスクリプタ (書き戻しフォーマット) について詳細に説明しています。

図 823. 送信ディスクリプタの書き戻しフォーマット



- TDES0 通常ディスクリプタ (書き戻しフォーマット)

表 657. TDES0 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾

ビット	名前	説明
31:0	TTSL	送信パケットタイムスタンプロー DMA は、このフィールドを、対応する送信パケットについてキャプチャされたタイムスタンプの下位 32 ビットで更新します。TDES2 の TTSE ビットがパケットの先頭ディスクリプタにセットされている場合のみ、DMA はタイムスタンプを書き込みます。このフィールドは、ディスクリプタの最後のセグメントビット (LS) がセットされ、タイムスタンプステータス (TTSS) ビットがセットされている場合にのみ、タイムスタンプを保持します。

1. このフォーマットは、パケットの最終ディスクリプタにのみ適用できます。

- TDES1 通常ディスクリプタ (書き戻しフォーマット)

表 658. TDES1 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾

ビット	名前	説明
31:0	TTSH	送信パケットタイムスタンプハイ DMA は、このフィールドを、対応する受信パケットについてキャプチャされたタイムスタンプの上位 32 ビットで更新します。TDES2 の TTSE ビットがパケットの先頭ディスクリプタにセットされている場合のみ、DMA はタイムスタンプを書き込みます。このフィールドは、ディスクリプタの最後のセグメントビット (LS) がセットされ、タイムスタンプステータス (TTSS) ビットがセットされている場合にのみ、タイムスタンプを保持します。

1. このフォーマットは、パケットの最終ディスクリプタにのみ適用できます。

- TDES2 通常ディスクリプタ (書き戻しフォーマット)

表 659. TDES2 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾

ビット	説明
31:0	予約済みです。

1. このフォーマットは、パケットの最終ディスクリプタにのみ適用できます。

- TDES3 通常ディスクリプタ (書き戻しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	予約済みです。										TTSS	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	PCE	LoC	NC	LC	EC	CC				ED	UF	DB	IHE

表 660. TDES3 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾

ビット	名前	説明
31	OWN	オウンビット このビットがセットされた場合、DMA がディスクリプタを所有していることを示します。DMA はパケット送信を完了すると、このビットをクリアします。書き戻しが完了すると、このビットは 0 にセットされます。
30	CTXT	コンテキストタイプ このビットは通常ディスクリプタでは 0 にセットする必要があります。
29	FD	先頭ディスクリプタ このビット、バッファがパケットの最初のセグメントを含んでいることを示します。
28	LD	最終ディスクリプタ このビットは、パケットの最終ディスクリプタに対して 1 にセットされます。DMA はパケットの最終ディスクリプタでのみ、ステータスフィールドを書き込みます。
27:18		予約済みです。
17	TTSS	Tx タイムスタンプステータス このステータスビットは、対応する送信パケットに対してタイムスタンプがキャプチャされたことを示します。このビットがセットされると、TDES0 と TDES1 は、送信パケットに対してキャプチャされたタイムスタンプの値を持ちます。このフィールドは、ディスクリプタの最後のセグメント制御ビット (TDES3 [28]) がセットされたときのみ有効です。
16		予約済みです。

表 660. TDES3 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾ (続き)

ビット	名前	説明
15	ES	エラー概要 このビットは、以下のビットの論理 OR を示します。 TDES3[0] : IP ヘッダエラー TDES3[14] : ジャバertimeアウト TDES3[13] : パケットー掃 TDES3[12] : ペイロードチェックサムエラー TDES3[11] : キャリア喪失 TDES3[10] : キャリアなし TDES3[9] : 遅れ衝突 TDES3[8] : 過剰衝突 TDES3[3] : 過剰遅延 TDES3[2] : アンダーフローエラー
14	JT	ジャバertimeアウト このビットは、MAC トランスミッタがジャバertimeアウトを検出しことを示します。動作モード設定レジスタ (ETH_MACCR) の JD ビットがセットされていないときにのみ、このビットがセットされます。
13	FF	パケットー掃 このビットは、DMA/MTL が、CPU によって与えられたソフトウェア掃コマンドにより、パケットを一掃したことを示します。
12	PCE	ペイロードチェックサムエラー このビットは、チェックサムオフロードエンジンに不具合があり、カプセル化された TCP、UDP、または ICMP ペイロードにチェックサムが挿入されなかったことを示します。この不具合は、IP ヘッダのペイロード長フィールドによって示されるバイト不足、または MTL がまだチェックサムが計算されていない状態でストアアンドフォワードモードでパケットを MAC トランスミッタに転送し始めることによって発生する可能性があります。この 2 つ目のエラー条件は、送信 FIFO の深さがデッドロックを回避するために送信される Ethernet パケットの長さより小さい場合にのみ発生します。ストアアンドフォワードモードでも FIFO がフルである場合、MTL はパケットの転送を開始します。このエラーは、パケット転送中にバスエラーが検出された場合も発生する可能性があります。
11	LoC	キャリア喪失 このビットは、パケット送信中にキャリアの喪失が発生した (すなわち、パケット送信中に 1 つ以上の送信クロック周期の間、ETH_CRS 信号が無効であった) ことを示します。このビットは、MAC が半二重モードで動作しているとき、衝突なしで送信されたパケットについてののみ有効です。
10	NC	キャリアなし このビットは、送信中に PHY からのキャリアセンス信号がアサートされなかったことを示します。
9	LC	遅れ衝突 このビットは、衝突ウィンドウ (MII モードではプリアンプルを含めて 64 バイト時間) の後で衝突が発生したためにパケット送信が中止されたことを示します。このビットは、アンダーフローエラーがセットされた場合は無効です。

表 660. TDES3 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾ (続き)

ビット	名前	説明
8	EC	過剰衝突 このビットは、現在のパケットの送信を試みている間に 16 の連続した衝突が発生した後、送信が中止されたことを示します。動作モード設定レジスタ (ETH_MACCCR) で DR ビットがセットされている場合、このビットが最初の衝突後にセットされて、パケットの送信が中止されます。
7:4	CC	衝突カウント この 4 ビットのカウンタの値は、パケットが送信される前に発生した衝突の回数を示します。このカウンタは、EC ビットがセットされているときには無効です。
3	ED	過剰遅延 このビットは、動作モード設定レジスタ (ETH_MACCCR) の DC ビットがセットされている場合、24,288 ビット時間を超える過剰遅延があったために送信が終了したことを示します。
2	UF	アンダーフローエラー このビットは、システムメモリからのデータが遅れて到着したために、MAC がパケット送信を中止したことを示します。アンダーフローエラーは、以下の条件のいずれかによって発生する可能性があります。 DMA がパケット送信中に空の送信バッファを検出した場合 アプリケーションが MAC の送信速度より遅く MTL Tx FIFO を満たす場合 送信プロセスはサスペンド状態に移行し、ETH_MTLISR レジスタでキューに対応するアンダーフロービットをセットします。
1	DB	遅延ビット このビットは、キャリアの存在のために MAC がパケット送信を延期したことを示します。このビットは、半二重モードでのみ有効です。
0	IHE	IP ヘッダエラー IP ヘッダエラーがセットされている場合、このビットはチェックサムオフロードエンジンが IP ヘッダエラーを検出したことを示します。COE が IP ヘッダエラーを検出したときに、Ethernet タイプフィールドが IPv4 ペイロードを示している場合は、IPv4 ヘッダチェックサムを挿入します。

1. このフォーマットは、パケットの最終ディスクリプタにのみ適用できます。

送信コンテキストディスクリプタ

送信コンテキストディスクリプタは、パケットディスクリプタの前にいつでも提供できます。このコンテキストは、現在のパケットと後続のパケットに有効です。コンテキストディスクリプタは、ワンステップタイムスタンプ補正ではタイムスタンプ、VLAN 挿入機能では VLAN タグ ID を提供するために使用されます。書き戻しは、OWN ビットをリセットするためにコンテキストディスクリプタでのみ実行されます。

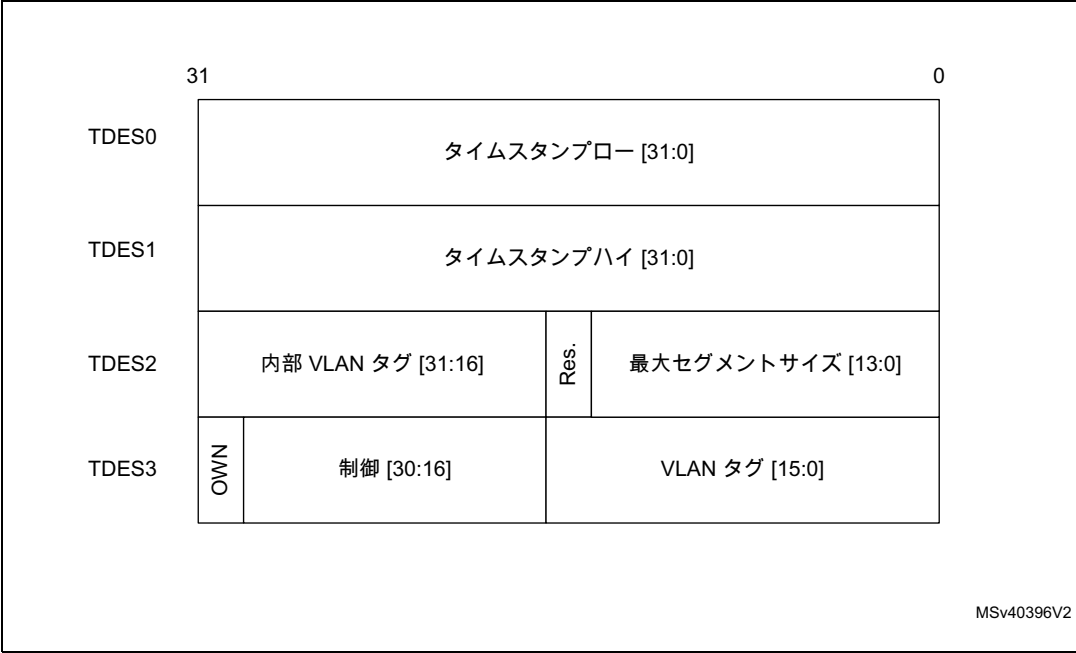
注 : VLAN タグ ID および MSS 値は、対応する有効ビットがセットされたコンテキストディスクリプタでアプリケーションによって提供され、DMA によって内部的に格納されます。

外部または内部 VLAN タグが有効ビットがセットされた状態で提供される場合、DMA は常に最後の有効な VLAN タグを MTL に渡します。アプリケーションは、DMA によって格納された有効な VLAN タグを無効にできません。VLAN タグは、パケットに提供される制御入力に基づいて挿入または書き換えされます。

内部 VLAN タグ制御入力は、コンテキストディスクリプタの直後に続くパケットにのみ使用されます。アプリケーションは、DMA が内部 VLAN タグ制御入力を使用する各パケットの通常ディスクリプタの前に、コンテキストディスクリプタを提供しなければなりません。

図 824 に、送信コンテキストディスクリプタのフォーマットを示します。表 661～表 664 では、すべての送信コンテキストディスクリプタについて詳細に説明しています。

図 824. 送信コンテキストディスクリプタのフォーマット



- TDES0 コンテキストディスクリプタ（読み出しフォーマット）

表 661. TDES0 コンテキストディスクリプタ

ビット	名前	説明
31:0	TTSL	送信パケットタイムスタンプロー ワンステップ補正では、ドライバはこのディスクリプタワードで下位 32 ビットのタイムスタンプを提供できます。DMA は、ワンステップタイムスタンプ補正を行うための下位ワードとしてこの値を使用します。このフィールドは、TDES3 コンテキストディスクリプタの OSTC および TCMSSV ビットがセットされた場合のみ有効です。

- TDES1 コンテキストディスクリプタ（読み出しフォーマット）

表 662. TDES1 コンテキストディスクリプタ

ビット	名前	説明
31:0	TTSH	送信パケットタイムスタンプハイ ワンステップ補正では、ドライバはこのディスクリプタで上位 32 ビットのタイムスタンプを提供できます。DMA は、ワンステップタイムスタンプ補正を行うための上位ワードとしてこの値を使用します。このフィールドは、TDES3 コンテキストディスクリプタの OSTC および TCMSSV ビットがセットされた場合のみ有効です。

- TDES2 コンテキストディスクリプタ (読み出しフォーマット)

表 663. TDES2 コンテキストディスクリプタ

ビット	名前	説明
31:16	IVT	内部 VLAN タグ TDES3 コンテキストディスクリプタの IVLTV ビットがセットされ、TDES3 コンテキストディスクリプタの TCMSSV および OSTC ビットがリセットされている場合、TDES2[31:16] には後続の送信バケットに挿入される内部 VLAN タグが含まれます。
15:14	予約済みです。	
13:0	MSS	最大セグメントサイズ このセグメントサイズは、TCP/IP ペイロードを分割する際に使用されます。このフィールドは、TDES3 コンテキストディスクリプタの TCMSSV ビットがセットされ、TDES3 コンテキストディスクリプタの OSTC ビットがリセットされている場合のみ有効です。

- TDES3 コンテキストディスクリプタ (読み出しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	予約済みです。	OSTC	TCMSSV	予約済みです。	CDE	予約済みです。						IVLTV	VLTV	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VT															

表 664. TDES3 コンテキストディスクリプタ

ビット	名前	説明
31	OWN	オウンビット 1 : DMA がディスクリプタを所有しています。 0 : アプリケーションがディスクリプタを所有しています。 DMA は読み出し操作の直後にこのビットをクリアします。
30	CTXT	コンテキストタイプ このビットはコンテキストディスクリプタでは 1 にセットする必要があります。
29:28	予約済みです。	
27	OSTC	ワンステップタイムスタンプ補正有効化 このビットがセットされた場合、DMA は TDES0 および TDES1 で提供されたタイムスタンプ値を参照して、ワンステップタイムスタンプ補正を実行します。
26	TCMSSV	ワンステップタイムスタンプ補正入力または MSS 有効 このビットおよび OSTC ビットがセットされた場合、TDES0 および TDES1 で提供されたタイムスタンプ補正入力が有効であることを示します。 OSTC ビットがリセットされ、このビットおよび TDES3 の TSE ビットが後続の通常ディスクリプタでセットされた場合、TDES2 の MSS 入力が有効であることを示します。
25:24	予約済みです。	

表 664. TDES3 コンテキストディスクリプタ (続き)

ビット	名前	説明
23	CDE	<p>コンテキストディスクリプタエラー</p> <p>このビットがセットされた場合、コンテキストのディスクリプタが適切でないことを示します。DMA は、コンテキストディスクリプタをクローズする際の書き戻し中に、このビットをセットします。</p> <p>コンテキストディスクリプタエラーは次の場合があります。</p> <ul style="list-style-type: none"> コンテキストディスクリプタの不正なシーケンス。たとえば、パケットの最初のディスクリプタより前の場所。 すべてが 1。 CD、LD、および FD ビットが 1 にセットされている。 <p>注： すべてが 1 であるため、または CTXT、LD、および FD ビットが 1 にセットされているためにコンテキストディスクリプタエラーが発生する場合、送信 DMA は DE および LD ビットを 1 にセットして送信ディスクリプタをクローズします。対応する最初のディスクリプタの TDES2 の IOC ビットが 1 にセットされている場合、送信 DMA により チャンネルステータスレジスタ (ETH_DMCSR) の TI ビットがセットされます。</p> <p>送信ディスクリプタの CTXT、LD、FD ビットに基づいて、(FD ビットがセットされていない場合でも) その後のディスクリプタが最初のディスクリプタとみなされ、パケットの一部が送信される場合があります。</p> <p>このエラーは異常イベントとして分類され、復旧するにはソフトウェアリセットが必要です (DMA の停止/再設定/再起動復旧メカニズムはサポートされていません)。</p>
22:20		予約済みです。
19:18	IVTIR	<p>内部 VLAN タグの挿入または書き換え</p> <p>これらのビットがセットされた場合、パケットを送信する前に VLAN タグ付けまたはタグ削除を実行するよう MAC に要求します。パケットが VLAN タグについて変更された場合、MAC は CRC バイトを自動的に再計算して書き換えます。</p> <p>このビットフィールドには以下の値があります。</p> <p>00 : 内部 VLAN タグを追加しません。</p> <p>01 : 送信前にパケットから内部 VLAN タグを削除します。このオプションは、VLAN フレームでのみ使用する必要があります。</p> <p>10 : 内部 VLAN 内包レジスタ (ETH_MACIVIR) でプログラムされたタグ値またはコンテキストディスクリプタで内部 VLAN タグを挿入します。</p> <p>11 : 内部 VLAN 内包レジスタ (ETH_MACIVIR) でプログラムされたタグ値またはコンテキストディスクリプタでパケットの内部 VLAN タグを書き換えます。このオプションは、VLAN フレームでのみ使用する必要があります。</p>
17	IVLTV	<p>内部 VLAN タグ有効</p> <p>このビットがセットされた場合、TDES2 の IVT フィールドが有効であることを示します。</p>
16	VLTV	<p>VLAN タグ有効</p> <p>このビットがセットされた場合、TDES3 の VT フィールドが有効であることを示します。</p>
15:0	VT	<p>VLAN タグ</p> <p>このフィールドは、パケットで挿入または書き換えられる VLAN タグを含みます。このフィールドは、VLAN 内包レジスタ (ETH_MACVIR) の VLTi ビットがリセットされたときのみ、VLAN として使用されます。</p>

57.10.4 受信ディスクリプタ

Ethernet ペリフェラルの DAM は、テイルポインタがベースポインタや現在のポインタと異なる場合のみディスクリプタの読出しを試みます。MAC によって受信される完全なパケットを最低 2 つ収容できる長さを持ったディスクリプタリングを準備することを推奨します。そうしないと、ディスクリプタの使用不能のために DMA のパフォーマンスが大きな影響を受けます。このような状況では、MTL RxFIFO がフルになり、パケットのドロップが開始します。

次の受信ディスクリプタが存在します。

- 読み出しフォーマットと書き戻しフォーマットを持つ通常ディスクリプタ
- コンテキストディスクリプタ

すべての受信したディスクリプタはソフトウェアによって準備され、DMA に「通常」ディスクリプタとして付与されます（それらの内容の詳細については、[図 825 : 受信通常ディスクリプタ（読み出しフォーマット）](#)を参照してください）。DMA は、このディスクリプタを読み出し、受信パケット（またはその一部）をディスクリプタによって示されたバッファに転送した後に、Rx DMA は対応するパケットステータスのディスクリプタをクローズします。ステータスフォーマットを[図 826 : 受信通常ディスクリプタ（書き戻しフォーマット）](#)に示します。

一部のパケットでは、通常ディスクリプタビットが足らず、完全なステータスを書き込めません。このようなパケットでは、Rx DMA は拡張ステータスを次のディスクリプタに（そのディスクリプタに埋め込まれたバッファポインタを処理したり使用したりせずに）書き込みます。この書き戻しディスクリプタのフォーマットおよび内容を[図 827 : 受信コンテキストディスクリプタ](#)に示します。

受信通常ディスクリプタ（読み出しフォーマット）

[図 825](#) に、受信通常ディスクリプタの読み出しフォーマットを示します。[表 665](#)～[表 668](#) では、すべての受信通常ディスクリプタ（読み出しフォーマット）について詳細に説明しています。

図 825. 受信通常ディスクリプタ（読み出しフォーマット）



注： 受信ディスクリプタ（読み出しフォーマット）で、バッファアドレスフィールドに 0 だけが含まれている場合、MAC はこのバッファにデータを転送せず、次のバッファまたは次のディスクリプタにスキップします。

- RDES0 通常ディスクリプタ (読み出しフォーマット)

表 665. RDES0 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31:0	BUF1AP	<p>バッファ 1 アドレスポインタ</p> <p>これらのビットは、バッファ 1 の物理アドレスを示します。</p> <p>アプリケーションが、このバッファのバイト整列アドレスをプログラムでき、このフィールドの LS ビットを 0 以外にセットできることを示します。ただし、パケットの開始の転送中、DMA は 64/128 ビット設定の場合は RDES2[1:0]=0 での書込み動作を 0 で実行します。ただし、パケットデータは、バッファアドレスポインタで示される実際のオフセットによってシフトされます。</p> <p>パケットの中央または最後の部分が格納されているバッファをアドレスポインタが指定する場合、DMA はオフセットアドレスを無視して、データ幅によって示された完全な位置に書き込みます。</p>

- RDES1 通常ディスクリプタ (読み出しフォーマット)

表 666. RDES1 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31:0	予約済みです。	フィールドが予約済みです。

- RDES2 通常ディスクリプタ (読み出しフォーマット)

表 667. RDES2 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31:0	BUF2AP	<p>バッファ 2 アドレスポインタ</p> <p>これらのビットはバッファ 2 物理アドレスを示します。</p> <p>RxDMA は、パケットの開始バイトを転送する間だけ、ポインタアドレスの LS ビットを使用します。BUF2AP がパケットの中央または最後の部分が格納されているバッファのアドレスを提供している場合、DMA は RDES2[1:0]=0 を無視して完全な位置に書き込みます。</p>

• RDES3 通常ディスクリプタ (読み出しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	IOC	予約済みです。				BUF2V	BUF1V	予約済みです。							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済みです。															

表 668. RDES3 通常ディスクリプタ (読み出しフォーマット)

ビット	名前	説明
31	OWN	オウンビット このビットがセットされた場合、DMA がディスクリプタを所有していることを示します。このビットがリセットされた場合、アプリケーションがディスクリプタを所有していることを示します。DMA は、以下の条件のいずれかが当てはまる場合に、このビットをクリアします。 <ul style="list-style-type: none"> – DMA がパケット受信を完了した場合 – ディスクリプタに関連付けられたバッファがフルである場合
30	IOC	完了時割込み有効化 このビットがセットされた場合、DMA がディスクリプタをクローズするときに割込みがアプリケーションに対して発行されます。
29:26	予約済みです。	
25	BUF2V	バッファ 2 アドレス有効 このビットがセットされた場合、RDES2 で指定されたバッファ 2 アドレスが有効であることを DMA に示します。アプリケーションは、RDES0 のバッファ 2 アドレスが指定しているアドレスを DMA が使用できるように、このビットをセットして受信パケットデータを書き込む必要があります。
24	BUF1V	バッファ 1 アドレス有効 セット時、RDES0 で指定されたバッファ 1 アドレスが有効であることを DMA に示します。アプリケーションは、RDES0 でバッファ 1 アドレスが指定するアドレスが DMA で使用できる場合に、この値をセットして受信パケットデータを書き込む必要があります。
23:0	予約済みです。	

受信通常ディスクリプタ (書き戻しフォーマット)

図 826 に、受信通常ディスクリプタの書き戻しフォーマットを示します。表 669～表 672 では、すべての受信通常ディスクリプタ (書き戻しフォーマット) について詳細に説明しています。

図 826. 受信通常ディスクリプタ (書き戻しフォーマット)

RDES0	内部 VLAN タグ [31:16]				外部 VLAN タグ [15:0]			
RDES1	OAM コードまたは MAC 制御 OP コード [31:16]				拡張ステータス			
RDES2	MAC フィルタステータス [31:16]				VF[15]	Res. [14:11]	ARPNR [10]	Res[9:0]
RDES3	OWN	INT	CT	LD	LD	ステータス [27:15]		ES
						パケット長 [14:0]		

MSv40398V2

- RDES0 通常ディスクリプタ (書き戻しフォーマット)

表 669. RDES0 通常ディスクリプタ (書き戻しフォーマット)

ビット	名前	説明
31:16	IVT	内部 VLAN タグ このフィールドには、RDES3 の RS0V ビットがセットされた場合に受信パケットの内部 VLAN タグが含まれます。これが有効なのは、ダブル VLAN タグ処理および VLAN タグストリッピングが有効になっている場合のみです。
15:0	OVT	外部 VLAN タグ このフィールドには、RDES3 の RS0V ビットがセットされた場合に受信パケットの外部 VLAN タグが含まれます。

- RDES1 通常ディスクリプタ (書き戻しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPC															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD	TSA	PV	PFT	PMT				IPCE	IPCB	IPV6	IPV4	IPHE	PT		

表 670. RDES1 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾

ビット	名前	説明
31:16	OPC	OAM サブタイプコード、または MAC 制御パケット OP コード OAM サブタイプコード RDES3 のビット [18:16] が 111 にセットされている場合、このフィールドには OAM サブタイプおよびコードフィールドが入ります。 MAC 制御パケット OP コード RDES3 のビット [18:16] が 110 にセットされている場合、このフィールドには MAC 制御パケット OP コードフィールドが入ります。
15	TD	タイムスタンプドロップ このビットは、タイムスタンプがこのパケットに対してキャプチャされたが、オーバーフローのために MTL Rx FIFO でドロップしたことを示します。
14	TSA	タイムスタンプ使用可能 タイムスタンプがある場合、このビットはタイムスタンプ値がコンテキストディスクリプタワード 2 (RDES2) およびワード 1 (RDES1) で使用可能なことを示します。これは、最終ディスクリプタビット (RDES3 [28]) がセットされた場合にのみ有効です。 コンテキストディスクリプタは、パケットの最終通常ディスクリプタの直後に、次のディスクリプタに書き込まれます。
13	PV	PTP バージョン 1 : 受信 PTP メッセージは IEEE 1588 バージョン 2 フォーマットです。 0 : 受信 PTP メッセージは IEEE 1588 バージョン 1 フォーマットです。
12	PFT	PTP パケットタイプ このビットは PTP メッセージが Ethernet を介して直接送信されることを示します。
11:8	PMT	PTP メッセージタイプ これらのビットは、受信メッセージのタイプを表すようにエンコードされます。 0000 : 受信 PTP メッセージなし 0001 : SYNC (全クロックタイプ) 0010 : Follow_Up (全クロックタイプ) 0011 : Delay_Req (全クロックタイプ) 0100 : Delay_Resp (全クロックタイプ) 0101 : Pdelay_Req (ピアツーピアトランスペアレントクロック) 0110 : Pdelay_Resp (ピアツーピアトランスペアレントクロック) 0111 : Pdelay_Resp_Follow_Up (ピアツーピアトランスペアレントクロック) 1000 : アナウンス 1001 : 管理 1010 : シグナリング 1011~1110 : 予約済みです。 1111 : 予約済みのある PTP パケットメッセージタイプ

表 670. RDES1 通常ディスクリプタ (書き戻しフォーマット)⁽¹⁾ (続き)

ビット	名前	説明
7	IPCE	IP ペイロードエラー このビットがセットされた場合、以下のいずれかを示します。 <ul style="list-style-type: none"> MAC によって計算された 16 ビット IP ペイロードチェックサム (すなわち、TCP、UDP、または ICMP チェックサム) が、受信セグメント内の対応するチェックサムフィールドに一致しません。 TCP、UDP、または ICMP セグメント長が IP ヘッダフィールド内のペイロード長に一致しません。 TCP、UDP、または ICMP セグメント長が TCP、UDP、または ICMP の最低許容セグメント長より短くなっています。 このビットがセットされると、RDES3 のビット 15 (ES) はセットされません。
6	IPCB	IP チェックサムのバイパス このビットはチェックサムオフロードエンジンがバイパスされていることを示します。
5	IPV6	IPv6 ヘッダあり このビットは IPv6 ヘッダが検出されたを示します。
4	IPV4	IPv4 ヘッダあり このビットは IPv4 ヘッダが検出されたを示します。
3	IPHE	IP ヘッダエラー <ul style="list-style-type: none"> このビットがセットされた場合、以下のいずれかを示します。 MAC によって計算された 16 ビット IPv4 ヘッダチェックサムが、受信チェックサムバイトに一致しません。 IP データグラムバージョンが Ethernet タイプ値と一致していません。 Ethernet パケットに、期待される IP ヘッダバイトの数がありません。 このビットは、ビット 5 またはビット 4 がセットされたときに有効です。
2:0	PT	ペイロードタイプ これらのビットは、受信チェックサムオフロードエンジン (COE) によって処理された IP データグラムでカプセル化されたペイロードのタイプを示します。 <ul style="list-style-type: none"> 000 : 不明なタイプまたは処理されていない IP/AV ペイロード 001 : UDP 010 : TCP 011 : ICMP 100 : IPv4 ヘッダありビットがセットされている場合、IGMP その他 : 予約済み IP ヘッダエラーまたは断片化された IP があるため、COE が IP データグラムのペイロードを処理しない場合、これらのビットを 3'b000 にセットします。

1. 書き戻しフォーマットのステータスフィールドは、最終ディスクリプタ (RDES3[28] がセット) にのみ有効です。

• RDES2 通常ディスクリプタ (書き戻しフォーマット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3L4FM			L4FM	L3FM	MADRM								HF	DAF	SAF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VF	予約済みです。				ARPRN	予約済みです。									

表 671. RDES2 通常ディスクリプタ (書き戻しフォーマット)

ビット	名前	説明
31:29	L3L4FM	<p>レイヤ 3 およびレイヤ 4 フィルタ数の一致</p> <p>これらのビットは、受信パケットと一致したレイヤ 3 およびレイヤ 4 フィルタの数を示します。</p> <ul style="list-style-type: none"> – 000 : フィルタ 0 – 001 : フィルタ 1 – 010 : フィルタ 2 – 011 : フィルタ 3 – 100 : フィルタ 4 – 101 : フィルタ 5 – 110 : フィルタ 6 – 111 : フィルタ 7 <p>このフィールドは、ビット 28 またはビット 27 がハイにセットされたときのみ有効です。複数のフィルタが一致する場合、これらのビットは最小フィルタの数を提供します。</p>
28	L4FM	<p>レイヤ 4 フィルター一致</p> <p>このビットがセットされた場合、受信パケットが有効化されたレイヤ 4 ポート番号フィールドの 1 つと一致することを示します。以下の条件のいずれかが当てはまる場合のみ、このステータスが付与されます。</p> <ul style="list-style-type: none"> – レイヤ 3 フィールドが有効でなく、すべての有効化されたレイヤ 4 フィールドが一致する。 – すべての有効化されたレイヤ 3 およびレイヤ 4 フィルタフィールドが一致する。 <p>複数のフィルタが一致する場合、このビットはビット [31:29] によって示されたフィルタのレイヤ 4 フィルタステータスを付与します。</p>
27	L3FM	<p>レイヤ 3 フィルター一致</p> <p>このビットがセットされた場合、受信パケットが有効化されたレイヤ 3 IP アドレスフィールドの 1 つと一致することを示します。以下の条件のいずれかが当てはまる場合のみ、このステータスが付与されます。</p> <ul style="list-style-type: none"> – すべての有効化されたレイヤ 3 が一致し、レイヤ 4 フィールドがバイパスされる。 – すべての有効化されたフィルタフィールドが一致する。 <p>複数のフィルタが一致する場合、このビットはビット [31:29] によって示されたフィルタのレイヤ 3 フィルタステータスを付与します。</p>
26:19	MADRM	<p>MAC アドレス一致またはハッシュ値</p> <p>HF ビットがリセットされた場合、このフィールドに、受信パケットのデスティネーションアドレスと一致する MAC アドレスレジスタ番号が入ります。このフィールドは、DAF ビットがリセットされたときにのみ有効です。</p> <p>HF ビットがセットされている場合、このフィールドには MAC によって計算されたハッシュ値が入ります。ハッシュ値に対応するビットがハッシュフィルタレジスタでセットされている場合、パケットがハッシュフィルタを通過します。</p>
18	HF	<p>ハッシュフィルタステータス</p> <p>このビットがセットされた場合、パケットが MAC アドレスハッシュフィルタを通過したことを示します。ビット [26:19] はハッシュ値を示します。</p>

表 671. RDES2 通常ディスクリプタ（書き戻しフォーマット）（続き）

ビット	名前	説明
17	DAF	デスティネーションアドレスフィルタ失敗 このビットがセットされた場合、パケットが MAC での DA フィルタに失敗したことを示します。
16	SAF	SA アドレスフィルタ失敗 このビットがセットされた場合、パケットが MAC での SA フィルタに失敗したことを示します。
15	VF	VLAN フィルタステータス このビットがセットされた場合、受信パケットの VLAN タグが VLAN フィルタを通過したことを示します。
14:11	予約済みです。	
10	ARPNR	ARP 応答の生成なし このビットがセットされた場合、MAC が受信した ARP リクエストパケットに ARP 応答を生成しなかったことを示します。MAC が先の ARP リクエストへの ARP 応答の送信でビジー状態である場合、このビットがセットされます。
9:0	予約済みです。	

• RDES3 通常ディスクリプタ（書き戻しフォーマット）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	PL														

表 672. RDES3 通常ディスクリプタ (書き戻しフォーマット)

ビット	名前	説明
31	OWN	OWNビット 1 : DMA がディスクリプタを所有しています。 0 : アプリケーションがディスクリプタを所有しています。 DMA は、以下の条件のいずれかが当てはまる場合に、このビットをクリアします。 DMA がパケット受信を完了した場合 ディスクリプタに関連付けられたバッファがフルである場合
30	CTXT	受信コンテキストディスクリプタ このビットがセットされた場合、現在のディスクリプタがコンテキストタイプのディスクリプタであることを示します。通常受信ディスクリプタでは、DMA がこのビットに 0 を書き込みます。 CTXT および FD ビットと一緒に使用されている場合、{CTXT, FD} で表現できる値は次のようになります。 00 : 中間のディスクリプタ 01 : 先頭ディスクリプタ 10 : 予約済みです。 11 : ディスクリプタエラー (すべてが 1 のため) 注 : ディスクリプタエラーが発生すると、受信 DMA はディスクリプタエラーを示している受信ディスクリプタをクローズします。この受信ディスクリプタはスキップされ、そのバッファアドレスはパケットデータの書き込みで使用されません。受信 DMA は チャンネルステータスレジスタ (ETH_DMCSR) の CDE フィールドをセットしますが、IOC フィールドがセットされている場合でも、RI フィールドはそのパケットの最後の受信ディスクリプタとしてマークされていないため、セットしません。パケットデータを書き込むために、その後の有効な受信ディスクリプタが使用されます。
29	FD	先頭ディスクリプタ このビットがセットされた場合、このディスクリプタがパケットの最初のバッファを含んでいることを示します。最初のバッファのサイズが 0 の場合、2 番目のバッファがパケットの開始を含みます。2 番目のバッファのサイズも 0 の場合、次のディスクリプタがパケットの開始を含みます。 CTXT ビットと FD ビットと一緒に使用する方法については、CTXT ビットの説明を参照してください。
28	LD	最終ディスクリプタ このビットがセットされた場合、このディスクリプタが指定しているバッファが、パケットの最後のバッファであることを示します。
27	RS2V	受信ステータス RDES2 有効 このビットがセットされた場合、RDES2 のステータスが有効で、DMA によって書き込まれることを示します。このビットは、RDES3 の LD ビットがセットされたときにのみ有効です。
26	RS1V	受信ステータス RDES1 有効 このビットがセットされた場合、RDES1 のステータスが有効で、DMA によって書き込まれることを示します。このビットは、RDES3 の LD ビットがセットされたときにのみ有効です。
25	RS0V	受信ステータス RDES0 有効 このビットがセットされた場合、RDES0 のステータスが有効で、DMA によって書き込まれることを示します。このビットは、RDES3 の LD ビットがセットされたときにのみ有効です。

表 672. RDES3 通常ディスクリプタ (書き戻しフォーマット) (続き)

ビット	名前	説明
24	CE	CRC エラー このビットがセットされた場合、受信パケットで巡回冗長検査 (CRC) エラーが発生したことを示します。このフィールドは、RDES3 の LD ビットがセットされたときにのみ有効です。
23	GP	ジャイアントパケット このビットがセットされた場合、パケット長が 1518、1522、または 2000 バイト (ジャンボパケットが有効の場合は 9018 または 9022 バイト) の指定された最大 Ethernet サイズを超えることを示します。 ジャイアントパケットは、パケット長のみ示します。パケットの切詰めは発生しません。
22	RWT	受信ウォッチドッグタイムアウト このビットがセットされた場合、現在のパケットの受信中に受信ウォッチドッグタイマが時間切れになったことを示します。現在のパケットは、ウォッチドッグタイムアウト後に切捨てられます。
21	OE	オーバーフローエラー このビットがセットされた場合、受信パケットが Rx FIFO でのバッファオーバーフローのため破損したことを示します。 DMA が部分的なパケットをアプリケーションに転送した場合のみ、このビットがセットされます。これは、Rx FIFO が閾値モードで動作している場合のみ発生します。ストアアンドフォワードモードでは、すべての部分的なパケットは Rx FIFO で完全にドロップされます。
20	RE	受信エラー このビットがセットされた場合、ETH_RX_DV 信号がパケット受信中にアサートされている間、ETH_RX_ER 信号がアサートされることを示します。
19	DE	ドリブルビットエラー このビットがセットされた場合、受信パケットに非整数倍 (奇数ニブル) のバイトがあることを示します。このビットは、MII モードでのみ有効です。
18:16	LT	長さ/タイプフィールド このフィールドは、受信したパケットが長さパケットかタイプパケットかを示します。3 ビットのエンコーディングは次のとおりです。 000 : パケットは、長さパケットです。 001 : パケットは、タイプパケットです。 011 : パケットは、ARP リクエストパケットタイプです。 100 : パケットは、VLAN タグ付きタイプパケットです。 101 : パケットは、ダブル VLAN タグ付きタイプパケットです。 110 : パケットは、MAC 制御パケットタイプです。 111 : パケットは、OAM パケットタイプです。 010 : 予約済みです。

表 672. RDES3 通常ディスクリプタ (書き戻しフォーマット) (続き)

ビット	名前	説明
15	ES	<p>エラー概要</p> <p>このビットがセットされた場合、以下のビットの論理 OR を示します。</p> <p>RDES3[19] : ドリブルエラー</p> <p>RDES3[20] : 受信エラー</p> <p>RDES3[21] : オーバーフローエラー</p> <p>RDES3[22] : ウォッチドッグタイムアウト</p> <p>RDES3[23] : ジャイアントパケット</p> <p>RDES3[24] : CRC エラー</p> <p>このフィールドは、RDES3 の LD ビットがセットされたときにのみ有効です。</p>
14:0	PL	<p>パケット長</p> <p>これらのビットは、システムメモリに転送された受信パケットのバイト長 (CRC を含む) を示します。</p> <p>このフィールドは、RDES3 の LD ビットがセットされ、オーバーフローエラービットがリセットされたときに有効です。パケット長には、IP チェックサム計算が有効で、受信パケットが MAC 制御パケットではないときに Ethernet パケットに付加される 2 バイトも含まれます。</p> <p>RDES3 の LD ビットがリセットされた場合、このフィールドには、現在のパケット用に転送された (部分的な) 累積バイト数が格納されます。</p>

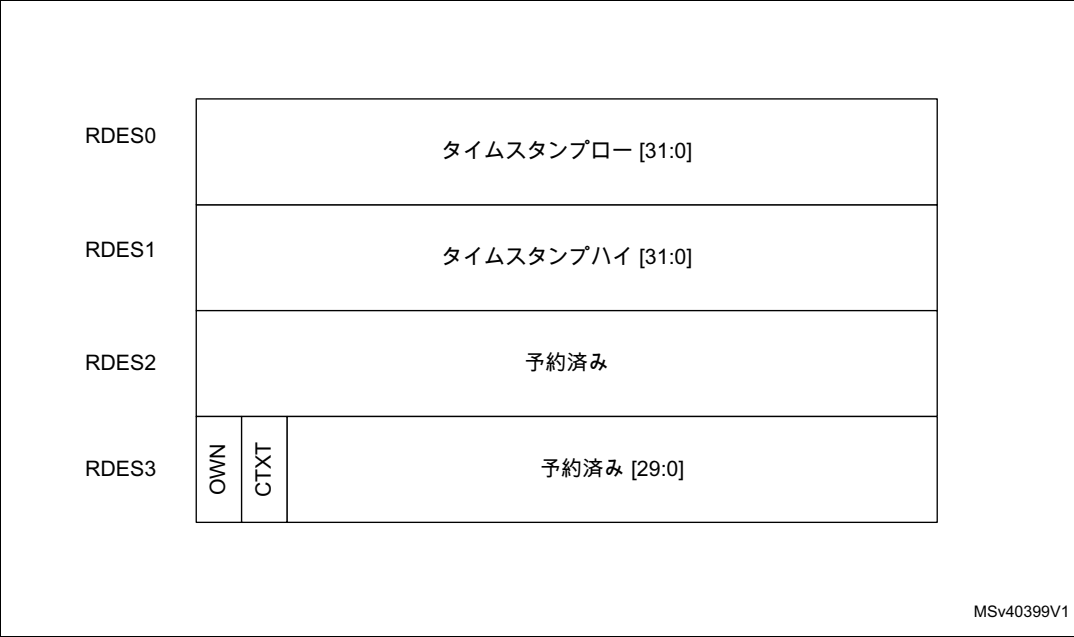
受信コンテキストディスクリプタ

このディスクリプタは、アプリケーションに対して読出し専用です。このディスクリプタは、DMA によってのみ書込みできます。

コンテキストディスクリプタは、最後に受信したパケットに関連する拡張ステータスについての情報を提供します。RDES3 のビット 30 は、コンテキストタイプのディスクリプタを示します。

図 827 に、受信コンテキストディスクリプタのフォーマットを示します。表 673～表 676 では、すべての受信コンテキストディスクリプタについて詳細に説明しています。

図 827. 受信コンテキストディスクリプタ



- RDES0 コンテキストディスクリプタ

表 673. RDES0 コンテキストディスクリプタ

ビット	名前	説明
31:0	RTSL	受信パケットタイムスタンプロー DMA は、このフィールドを、対応する受信パケットについてキャプチャされたタイムスタンプの下位 32 ビットで更新します。このフィールドと RDES1 の RTSH フィールドがすべて 1 を示している場合、タイムスタンプは破損しているものとみなす必要があります。

- RDES1 コンテキストディスクリプタ

表 674. RDES1 コンテキストディスクリプタ

ビット	フィールド	説明
31:0	RTSH	受信パケットタイムスタンプハイ DMA は、このフィールドを、対応する受信パケットについてキャプチャされたタイムスタンプの上位 32 ビットで更新します。このフィールドと RDES0 の RTSL フィールドがすべて 1 の値を示している場合、タイムスタンプは破損しているものとみなす必要があります。

- RDES2 コンテキストディスクリプタ

表 675. RDES2 コンテキストディスクリプタ

ビット	説明
31:0	予約済みです。

- RDES3 コンテキストディスクリプタ

表 676. RDES3 コンテキストディスクリプタ

ビット	名前	説明
31	OWN	オウンビット 1 : DMA がディスクリプタを所有しています。 0 : アプリケーションがディスクリプタを所有しています。 DMA は、以下の条件のいずれかが当てはまる場合に、このビットをクリアします。 DMA がパケット受信を完了した場合 ディスクリプタに関連付けられたバッファがフルである場合
30	CTXT	受信コンテキストディスクリプタ このビットがセットされた場合、現在のディスクリプタがコンテキストディスクリプタであることを示します。コンテキストディスクリプタでは、DMA がこのビットに 1'b1 を書き込みます。
29:0		予約済みです。

57.11 Ethernet レジスタ

57.11.1 Ethernet レジスタマップ

このセクションでは、以下のレジスタマップを提供します。

- DMA レジスタ ([セクション 57.11.2 : Ethernet DMA レジスタ](#)を参照)
- MTL レジスタ ([セクション 57.11.3 : Ethernet MTL レジスタ](#)を参照)
- MMC レジスタを含む MAC レジスタ ([セクション 57.11.4 : Ethernet MAC および MMC レジスタ](#)を参照)

57.11.2 Ethernet DMA レジスタ

DMA モードレジスタ (ETH_DMAMR)

アドレスオフセット : 0x1000

リセット値 : 0x0000 0000

DMA モードレジスタは、DMA のバス動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM[1:0]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PR[2:0]			TXPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DA	SWR
	rw	rw	rw	rw										rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **INTM[1:0]** : 割り込みモード

このフィールドは Ethernet ペリフェラルの割り込みモード を定義します。

割り込み信号および ETH_DMACSR レジスタの RI/TI ビットの動作は、INTM 値に依存して変化します ([表 652 : 転送完了割り込みの動作](#)を参照)。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **PR[2:0]** : 優先比率

これらのビットは、Rx DMA と Tx DMA の間の加重ラウンドロビン方式アービトレーションの優先比率を制御します。これらのビットは、DA ビットがリセットされた時のみ有効です。優先比率は、TXPR ビットがリセットされているかセットされているかに応じて、Rx:Tx または Tx:Rx となります。

000 : 優先比率は、1:1 です。

001 : 優先比率は、2:1 です。

010 : 優先比率は、3:1 です。

011 : 優先比率は、4:1 です。

100 : 優先比率は、5:1 です。

101 : 優先比率は、6:1 です。

110 : 優先比率は、7:1 です。

111 : 優先比率は、8:1 です。

ビット 11 **TXPR** : 送信優先順位

セットされると、このビットは、システム側バスのアービトレーションの間、Tx DMA が Rx DMA より高い優先順位であることを示します。

ビット 10:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DA** : DMA Tx または Rx アービトレーション構成

このビットは、すべてのチャネルの送信パスと受信パスのアービトレーション構成を指定します。

0 : Rx:Tx または Tx:Rx での加重ラウンドロビン

パス間の優先順位がビット [14:12] で指定された優先順位に従い、優先順位の重みが TXPR ビットで指定されます。

1 : 固定優先順位

TXPR ビットがセットされた場合、Tx パスは Rx パスより優先されます。それ以外は、Rx パスが Tx パスより優先されます。

ビット 0 **SWR** : ソフトウェアリセット

このビットがセットされると、MAC および DMA コントローラが DMA、MTL、および MAC のレジスタおよびすべての内部レジスタをリセットします。このビットは、リセット動作がすべてのクロックドメインで完了した後に自動でクリアされます。レジスタを再プログラムする前に、このビットで 0 の値を読み出す必要があります。

注 : すべてのアクティブなクロックドメインのすべてのリセットがネゲートされたときのみ、リセット動作が完了します。そのため、(選択された PHY インタフェースに適用可能な) すべての PHY 入力クロックがソフトウェアリセット完了に対して存在する必要があります。ソフトウェアリセット動作を完了するための時間は、最も遅いアクティブクロックの周波数に依存します。

システムバスモードレジスタ (ETH_DMASBMR)

アドレスオフセット : 0x1004

リセット値 : 0x0000 0000

システムバスモードレジスタは、AHB マスタの動作を制御します。未処理リクエストのバーストスプリットおよび数を主に制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB	MB	Res.	AAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FB
r	r		rw												rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **RB** : INCRx バーストの再構築

このビットがハイにセットされ、AHB マスタが SPLIT、RETRY、または早期バースト終了 (EBT) レスポンスを受信した場合、AHB マスタインタフェースは、開始されたバースト転送のペンディング中のビートを、INCRx および SINGLE 転送で再構築します。デフォルトでは、AHB マスタインタフェースは EBT のペンディング中のビートを、未指定 (INCR) バーストで再構築します。

ビット 14 **MB** : 混合バースト

このビットがハイにセットされ、FB ビットがローの場合、AHB マスタは 16 以上のバースト長に対して未定義バースト転送 (INCR) を実行します。16 未満のバースト長の場合、AHB マスタは固定バースト転送 (INCRx および SINGLE) を実行します。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **AAL** : アドレス整列ビート

このビットが 1 にセットされた場合、マスタは読み出しおよび書き込みチャネルでアドレス整列バースト転送を実行します。

ビット 11:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **FB** : 固定バースト長

このビットが 1 にセットされた場合、AHB マスタは指定された長さのバースト転送 (INCRx または SINGLE) を開始します。

このビットが 0 にセットされた場合、AHB マスタは未指定の長さ (INCR) の転送または SINGLE 転送を開始します。

割込みステータスレジスタ (ETH_DMAISR)

アドレスオフセット : 0x1008

リセット値 : 0x0000 0000

アプリケーションは、割込みサービスルーチン中またはポーリング中に、この割込みステータスを読み出して、DMA チャンネル、MTL キュー、MAC の割込みステータスを決定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC0IS
															r

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **MACIS** : MAC 割込みステータス

このビットは、MAC の割込みイベントを示します。このビットを 1'b0 にリセットするために、ソフトウェアは MAC で対応するレジスタを読み出して、割込みの正確な原因を取得し、そのソースをクリアしなければなりません。

ビット 16 **MTLIS** : MTL 割込みステータス

このビットは、MTL の割込みイベントを示します。このビットを 1'b0 にリセットするために、ソフトウェアは MTL で対応するレジスタを読み出して、割込みの正確な原因を取得し、そのソースをクリアしなければなりません。

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DC0IS** : DMA チャンネル割込みステータス

このビットは、DMA チャンネルの割込みイベントを示します。このビットを 0 にリセットするために、ソフトウェアは DMA チャンネルで対応するレジスタを読み出して、割込みの正確な原因を取得し、そのソースをクリアしなければなりません。

デバッグステータスレジスタ (ETH_DMADSR)

アドレスオフセット : 0x100C

リセット値 : 0x0000 0000

デバッグステータスレジスタは、デバッグの目的で DMA チャンネルの受信および送信プロセスステータスを提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0 [3:0]				RPS0 [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXWH STS
r	r	r	r	r	r	r	r								r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **TPS0[3:0]** : DMA チャンネル 送信プロセス状態

このフィールドは、チャンネルの Tx DMA FSM 状態を示します。

000 : 停止 (送信リセットまたは送信停止コマンドの発行)

001 : 実行中 (Tx 転送ディスクリプタのフェッチ)

010 : 実行中 (ステータス待ち)

011 : 実行中 (システムメモリバッファからデータの読出し、および Tx バッファ (TxFIFO) のキューへの挿入)

100 : タイムスタンプ書き込み状態

101 : 将来の使用のために予約済み。

110 : 中断 (Tx ディスクリプタ使用不可、または Tx バッファアンダーフロー)

111 : 実行中 (送信ディスクリプタのクローズ)

このフィールドの MSB は必ず 0 に戻ります。このフィールドは割込みを生成しません。

ビット 11:8 **RPS0[3:0]** : DMA チャンネル受信プロセス状態

このフィールドは、チャンネルの Rx DMA FSM 状態を示します。

000 : 停止 (受信リセットまたは受信停止コマンドの発行)

001 : 実行中 (Rx 転送ディスクリプタのフェッチ)

010 : 将来の使用のために予約済み。

011 : 実行中 (Rx パケット待ち)

100 : 中断 (Rx ディスクリプタ使用不可)

101 : 実行中 (Rx ディスクリプタのクローズ)

110 : タイムスタンプ書き込み状態

111 : 実行中 (Rx パケットデータの Rx バッファからシステムメモリへの転送)

このフィールドの MSB は必ず 0 に戻ります。このフィールドは割込みを生成しません。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **AXWHSTS** : AHB マスタ書き込みチャンネル

ハイの場合、このビットは AHB マスタ FMS の書き込みチャンネルがアイドル状態ではないことを示します。

チャンネル制御レジスタ (ETH_DMCCR)

アドレスオフセット : 0x1100

リセット値 : 0x0000 0000

DMA チャンネル制御レジスタは、セグメンテーションの MSS 値、2 つのディスクリプタ間でスキップする長さを指定し、ヘッダスプリットおよび 8xPBL モードなども備えています。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBLX8
											rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MSS[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:18 **DSL[2:0]** : ディスクリプタスキップ長

このビットは、2 つの連結されていないディスクリプタ間でスキップする 32 ビットワードを指定します。アドレススキッピングは、現在のディスクリプタの終わりから始まり、次のディスクリプタの始めまでです。

DSL 値が 0 である場合、DMA は連続したディスクリプタテーブルを取得します。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **PBLX8** : 8xPBL モード

このビットがセットされると、**チャンネル送信制御レジスタ (ETH_DMACTXCR)** のビット [21:16] にプログラムされた PBL 値は 8 倍になります。したがって、DMA は、PBL の値に応じて、8、16、32、64、128、および 256 ビートでデータを転送します。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **MSS[13:0]** : 最大セグメントサイズ

このフィールドは、パケットの分割時に使用される最大セグメントサイズを指定します。このフィールドは、**チャンネル送信制御レジスタ (ETH_DMACTXCR)** の TSE ビットがセットされたときにのみ有効です。

このフィールドにプログラムされた値は、バイトで設定されたデータ幅より大きくなければなりません。64 バイト以上の MSS 値を使用することを推奨します。

チャンネル送信制御レジスタ (ETH_DMACTXCR)

アドレスオフセット : 0x1104

リセット値 : 0x0000 0000

DMA チャンネル送信制御レジスタは、PBL、TCP セグメンテーション、Tx チャンネルの重みなど Tx 機能を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXPBL[5:0]					
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	Res.	Res.	Res.	ST
			rw								rw				rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:16 **TXPBL[5:0]** : 送信のプログラム可能なバースト長

これらのビットは、1 回の DMA データ転送で転送される最大ビット数を示します。これは、1 回のブロック読み出しまたは書き込みで使用される最大値です。DMA は、常に、アプリケーションパス上でバースト転送を開始するたびに、PBL での指定に従ってバーストを試みます。PBL は、次の値でプログラムできます。1、2、4、8、16 または 32 です。その他の値は、未定義の動作をもたらします。

32 ビットを超える転送では、以下の手順を実行します。

- ETH_DMCCR で PBLx8 モードをセットします。
- TXPBL[5:0] をセットします。

注 : TXPBL の最大値は、ビットについて、Tx キューサイズ (**Tx キュー動作モードレジスタ (ETH_MTLTXQOMR)** の TQS フィールド) の半分以下でなければなりません。これは、MTL Tx キューコントローラが MAC にデータを転送している間に、少なくとも別の Tx PBL のデータを格納できるだけのスペースを Tx キューが確保するために必要です。サイズ 2048 バイトの Tx キューの合計格納場所は 512 で、TXPBL および 8xPBL は 512/2 以下になるようにプログラムする必要があります。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TSE** : TCP セグメンテーション有効

このビットがセットされた場合、チャンネル x のパケットに TCP セグメンテーションを実行します。TCP セグメンテーションは、Tx 通常ディスクリプタで TSE ビット (TDES0[19]) がセットされたパケットに対してのみ行われます。このビットがセットされた場合、TxPBL 値は 4 以上でなければなりません。

ビット 11:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OSF** : 2 番目のパケットに対する動作

このビットがセットされた場合、最初のパケットのステータスが取得される前でも、送信データの 2 番目のパケットを処理するように DMA に指示します。

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **ST** : 送信開始または送信停止コマンド

このビットがセットされた場合、送信が実行中状態に置かれます。DMA は、送信パケットの現在位置での送信リストをチェックします。

DMA は、以下のいずれかの位置からディスクリプタの取得を試みます。

- リスト内の現在位置 : これは、ETH_DMACTXDLAR レジスタによってセットされた送信リストのベースアドレスです。
- 前に送信が停止された位置

DMA が現在のディスクリプタを所有していない場合、送信はサスペンド状態に入り、ETH_DMACSR の TBU ビットがセットされます。送信開始コマンドは、送信が停止されたときにのみ有効となります。ETH_DMACTXDLAR レジスタをセットする前にコマンドが発行された場合、DMA の動作は予想できません。

このビットがリセットされると、送信プロセスは現在のパケットの送信を終了した後、停止状態に置かれます。送信リスト内の次のディスクリプタの位置が保存され、送信がリスタートされるときに現在位置となります。リストアドレスを変更するには、このビットがリセットされたときに ETH_DMACTXDLAR レジスタを新しい値でプログラムする必要があります。新しい値は、このビットが再びセットされたときに考慮されます。送信停止コマンドは、現在のパケットの送信が完了したとき、または送信がサスペンド状態にあるときにのみ有効です。

チャンネル受信制御レジスタ (ETH_DMARXCR)

アドレスオフセット : 0x1108

リセット値 : 0x0000 0000

DMA チャンネル受信制御レジスタは、PBL、バッファサイズ、拡張ステータスなど Rx 機能を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RPF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXPBL[5:0]					
rw										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RBSZ[13:0]														SR
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 RPF : DMA Rx チャンネルパケット一掃

このビットが 1 にセットされている場合、DMA Rx チャンネルがシステムバスエラーの発生後に停止すると、Rx キューから DMA Rx チャンネルに送信される予定のパケットは、Ethernet ペリフェラルにより自動的に一掃されます。このビットがセットされたまま、DMA がソフトウェアドライバによって再起動された場合、この RxDMA が停止したときに受信していて、Rx キューに残っているパケットは一掃されます。RxDMA の再起動後に MAC によって受信されたパケットは、RxDMA にルーティングされます。この一掃は、Rx キューの読み出し側で発生します。

このビットが 0 にセットされている場合、DMA Rx チャンネルに送信される予定で、システムバスエラーにより DMA が停止した後に Rx キューに残っているパケットは、Ethernet ペリフェラルにより一掃されません。

これにより、対応する Rx キューで行頭ブロッキングが発生することがあります。

注 : RPF をセットして Rx DMA チャンネルからアプリケーションへのパケットの流れを停止することは、Rx キューと Rx DMA チャンネルの間に 1 対 1 の対応がある場合のみ有効です。動的マッピングモードでは、ETH_DMARXCR レジスタに RPF ビットをセットすると、停止した Rx DMA チャンネルに送信する予定の、予定外の Rx キューからパケットが一掃されることがあります。

ビット 30:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:16 RXPBL[5:0] : 受信のプログラム可能なバースト長

これらのビットは、1 回の DMA データ転送で転送される最大ビート数を示します。これは、1 回のブロック読み出しまたは書き込みで使用される最大値です。DMA は、常に、アプリケーションバス上でバースト転送を開始するたびに、PBL での指定に従ってバーストを試みます。PBL は、次の値でプログラムできます。1、2、4、8、16 または 32 です。その他の値は、未定義の動作をもたらします。

32 ビートを越える転送では、以下の手順を実行します。

- ETH_DMCCR で PBLx8 モードをセットします。
- RXPBL[5:0] をセットします。

注 : RXPBL の最大値は、ビートについて、Rx キューサイズ (Rx キュー動作モードレジスタ (ETH_MTLRXQOMR) の RQS フィールド) の半分以下でなければなりません。これは、MTL Rx キューコントローラが MAC にデータを転送している間に、少なくとも別の Rx PBL のデータを格納できるだけのスペースを Rx キューが確保するために必要です。サイズ 2048 バイトの Rx キューの合計格納場所は 512 で、RXPBL および 8xPBL は 512/2 以下になるようにプログラムする必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:1 RBSZ[13:0] : 受信バッファサイズ

このフィールドは、Rx バッファのサイズをバイト単位で示します。最大バッファサイズは 16 KB に制限されます。

注 : バッファサイズは 4 の倍数でなければなりません。バッファアドレスポインタの値がバス幅に整列されていない場合でも、これは必須となります。バッファサイズが 4 の倍数でない場合、未定義の動作が発生する可能性があります。

LSB ビット (1:0) は無視され、DMA は内部的に LSB ビットをすべて 0 とみなします。したがって、これらの LSB ビットは読み出し専用 (RO) です。

ビット 0 SR : 受信開始または停止

このビットがセットされた場合、DMA は受信リストからのディスクリプタの取得を試みて、着信パケットを処理します。

DMA は、以下のいずれかの位置からディスクリプタの取得を試みます。

- リスト内の現在位置 : これは、[チャンネル Rx ディスクリプタリストアドレスレジスタ \(ETH_DMCCRXLAR\)](#) によってセットされたアドレスです。
- 前に Rx プロセスが停止された位置

DMA が現在のディスクリプタを所有していない場合、受信はサスペンドされ、ETH_DMACSR の RBU ビットがセットされます。受信開始コマンドは、受信が停止されたときにのみ有効となります。[チャンネル Rx ディスクリプタリストアドレスレジスタ \(ETH_DMCCRXLAR\)](#) をセットする前にコマンドが発行された場合、DMA の動作は予想できません。

このビットがリセットされると、Rx DMA の動作は、現在のパケットの転送後に停止されます。受信リスト内の次のディスクリプタの位置が保存され、Rx プロセスがリスタートされた後に現在位置となります。受信停止コマンドは、Rx プロセスが実行中 (Rx パケットを待機中) またはサスペンド状態にあるときにのみ有効です。

チャンネル Tx ディスクリプタリストアドレスレジスタ (ETH_DMACTXDLAR)

アドレスオフセット : 0x1114

リセット値 : 0x0000 0000

チャンネル Tx ディスクリプタリストアドレスレジスタは、DMA に送信ディスクリプタリストの開始を示します。ディスクリプタリストは、アプリケーションの物理メモリ空間にあり、ワード整列されていなければなりません。DMA は、対応する LSB をローにセットすることによって、内部でバス幅で整列されたアドレスに変換します。

Tx DMA が停止したとき (ETH_DMACTXCR レジスタで ST ビットが 0 にセットされたとき) のみ、このレジスタに書き込むことができます。停止された場合、新しいディスクリプタリストアドレスをこのレジスタに書き込みできます。ST ビットが 1 にセットされた場合、DMA は新しくプログラムされたディスクリプタベースアドレスを取得します。ST ビットが 0 にセットされているときに、このレジスタが変更されない場合、DMA は前に停止した位置のディスクリプタアドレスを取得します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDES LA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES LA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

ビット 31:0 TDES LA[31:0] : 送信リストの開始

このフィールドは、送信ディスクリプタリストの先頭ディスクリプタのベースアドレスを含みます。DMA は、32 ビットバス幅に対して LSB ビット (1:0) を無視し、これらのビットを内部的にすべて 0 とみなします。したがって、これらの LSB ビットは読み出し専用 (RO) です。

チャンネル Rx ディスクリプタリストアドレスレジスタ (ETH_DMCRXDLAR)

アドレスオフセット : 0x111C

リセット値 : 0x0000 0000

チャンネル Rx ディスクリプタリストアドレスレジスタは、DMA に受信ディスクリプタリストの開始を示します。

このレジスタは、受信ディスクリプタリストの開始を示します。ディスクリプタリストは、アプリケーションの物理メモリ空間にあり、ワード整列されていなければなりません。DMA は、対応する LS ビットをローにセットすることによって、内部でバス幅で整列されたアドレスに変換します。このレジスタへの書き込みは、受信が停止されたときにのみ許されます。受信が停止された場合、受信開始コマンドが与えられる前に、このレジスタに書き込まなければなりません。Rx DMA が停止したとき (ETH_DMCRXCR レジスタで SR ビットが 0 にセットされたとき) のみ、このレジスタに書き込むことができます。停止された場合、新しいディスクリプタリストアドレスをこのレジスタに書き込みできます。

SR ビットが 1 にセットされた場合、DMA は新しくプログラムされたディスクリプタベースアドレスを取得します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDESLA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDESLA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **RDESLA[31:0]** : 受信リストの開始

このフィールドは、Rx ディスクリプタリストの先頭ディスクリプタのベースアドレスを含みます。DMA は、32 ビットバス幅に対して LSB ビット (1:0) を無視し、これらのビットを内部的にすべて 0 とみなします。したがって、これらの LSB ビットは読み出し専用 (RO) です。

チャンネル Tx ディスクリプタテイルポインタレジスタ (ETH_DMACTXDTPR)

アドレスオフセット : 0x1120

リセット値 : 0x0000 0000

チャンネル Tx ディスクリプタテイルポインタレジスタは、ベースからのオフセットを示し、最後の有効なディスクリプタの位置を示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **TDT[31:0]** : 送信ディスクリプタテイルポインタ

このフィールドは、Tx ディスクリプタリングのテイルポインタを含みます。このソフトウェアは、Tx チャンネルにディスクリプタを追加するためにテイルポインタを書き込みます。ハードウェアは、ヘッドとテイルポインタレジスタの間のディスクリプタによって参照されるすべてのパケットの送信を試みます。

チャンネル Rx ディスクリプタテイルポインタレジスタ (ETH_DMCRXDTPR)

アドレスオフセット : 0x1128

リセット値 : 0x0000 0000

チャンネル Rx ディスクリプタテイルポインタは、ベースからのオフセットを示し、最後の有効なディスクリプタの位置を示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **RDT[31:0]** : 受信ディスクリプタテイルポインタ

このフィールドは、Rx ディスクリプタリングのテイルポインタを含みます。このソフトウェアは、Rx チャンネルにディスクリプタを追加するためにテイルポインタを書き込みます。ハードウェアは、ヘッドとテイルポインタレジスタの間で参照されるディスクリプタにすべての受信パケットの書き込みを試みます。

チャンネル Tx ディスクリプタリング長レジスタ (ETH_DMACTXRLR)

アドレスオフセット : 0x112C

リセット値 : 0x0000 0000

Tx ディスクリプタリング長レジスタには、送信ディスクリプタリングの長さが含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **TDRL[9:0]** : 送信ディスクリプタリング長

このフィールドは、循環ディスクリプタリングでの Tx の最大数をセットします。ディスクリプタの最大数は 1000 個に制限されます。最小リングディスクリプタ長を 4 とすることを推奨します。

たとえば、このフィールドでは 0x3FF までの任意の値をプログラムできます。このフィールドは 10 ビット幅で、0x3FF をプログラムすると、1024 のディスクリプタを使用できます。10 のディスクリプタを使用する場合、0x9 の値にプログラムします。

チャンネル Rx ディスクリプタリング長レジスタ (ETH_DMCCRRLR)

アドレスオフセット : 0x1130

リセット値 : 0x0000 0000

チャンネル Rx ディスクリプタリング長レジスタには、受信ディスクリプタ循環リングの長さが含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARBS[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RDRL[9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **ARBS[7:0]** : オルタネート受信バッファサイズ

ARBS[7:0] が 0 以外の値にプログラムされている場合、バッファ 1 のサイズ (バイト単位) を示します。

ARBS[7:0] = 0 の場合、Rx Buffer1 および Rx Buffer2 のサイズは [チャンネル受信制御レジスタ \(ETH_DMCCRXC\)](#) の RBSZ[13:0] フィールドに基づいて決まります。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **RDRL[9:0]** : 受信ディスクリプタリング長

このレジスタは、循環ディスクリプタリングでの Rx の最大数をセットします。ディスクリプタの最大数は 1000 個に制限されます。

たとえば、このフィールドでは 0x3FF までの任意の値をプログラムできます。このフィールドは 10 ビット幅です。0x3FF をプログラムすると、1024 のディスクリプタを使用できます。10 のディスクリプタを使用する場合、0x9 の値にプログラムします。

チャンネル割込み有効レジスタ (ETH_DMCCR)

アドレスオフセット : 0x1134

リセット値 : 0x0000 0000

チャンネル割込み有効レジスタは、ステータスレジスタによって報告された割込みを有効にします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **NIE** : 通常割込み要約有効化このビットがセットされた場合、通常割込み要約が有効になります。このビットは、[チャンネルステータスレジスタ \(ETH_DMCCR\)](#) で次の割込みを有効にします。

ビット 0 : 送信割込み

ビット 2 : 送信バッファ使用不可

ビット 6 : 受信割込み

ビット 11 : アーリー受信割込み

このビットがリセットされた場合、通常割込み要約が無効になります。

ビット 14 AIE : 異常割込み要約有効化

このビットがセットされた場合、異常割込み要約が有効になります。このビットは、**チャンネルステータスレジスタ (ETH_DMCSR)** で次の割込みを有効にします。

ビット 1 : 送信プロセス停止

ビット 7 : Rx バッファ使用不可

ビット 8 : 受信プロセス停止

ビット 9 : 受信ウォッチドッグタイムアウト

ビット 10 : アーリー送信割込み

ビット 12 : 致命的バスエラー

このビットがリセットされた場合、異常割込み要約が無効になります。

ビット 13 CDEE : コンテキストディスクリプタエラー有効化

このビットが AIE ビットとともにセットされた場合、コンテキストディスクリプタエラー割込みが有効になります。このビットがリセットされると、コンテキストディスクリプタエラー割込みが無効になります。

ビット 12 FBEE : 致命的バスエラー有効化

このビットが AIE ビットとともにセットされた場合、致命的バスエラー割込みが有効になります。このビットがリセットされると、致命的バスエラー割込みが無効になります。

ビット 11 ERIE : アーリー受信割込み有効化

このビットが NIE ビットとともにセットされた場合、アーリー受信割込みが有効になります。このビットがリセットされると、アーリー受信割込みが無効になります。

ビット 10 ETIE : アーリー送信割込み有効化

このビットが AIE ビットとともにセットされた場合、アーリー送信割込みが有効になります。このビットがリセットされると、アーリー送信割込みが無効になります。

ビット 9 RWTE : 受信ウォッチドッグタイムアウト有効化

このビットが AIE ビットとともにセットされた場合、受信ウォッチドッグタイムアウト割込みが有効になります。このビットがリセットされると、受信ウォッチドッグタイムアウト割込みが無効になります。

ビット 8 RSE : 受信停止有効化

このビットが AIE ビットとともにセットされた場合、受信停止割込みが有効になります。このビットがリセットされると、受信停止割込みが無効になります。

ビット 7 RBUE : 受信バッファ使用不可有効化

このビットが AIE ビットとともにセットされた場合、受信バッファ使用不可割込みが有効になります。このビットがリセットされると、受信バッファ使用不可割込みが無効になります。

ビット 6 RIE : 受信割込み有効化

このビットが NIE ビットとともにセットされた場合、受信割込みが有効になります。このビットがリセットされると、受信割込みが無効になります。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 TBUE : 送信バッファ使用不可有効化

このビットが NIE ビットとともにセットされた場合、送信バッファ使用不可割込みが有効になります。このビットがリセットされると、送信バッファ使用不可割込みが無効になります。

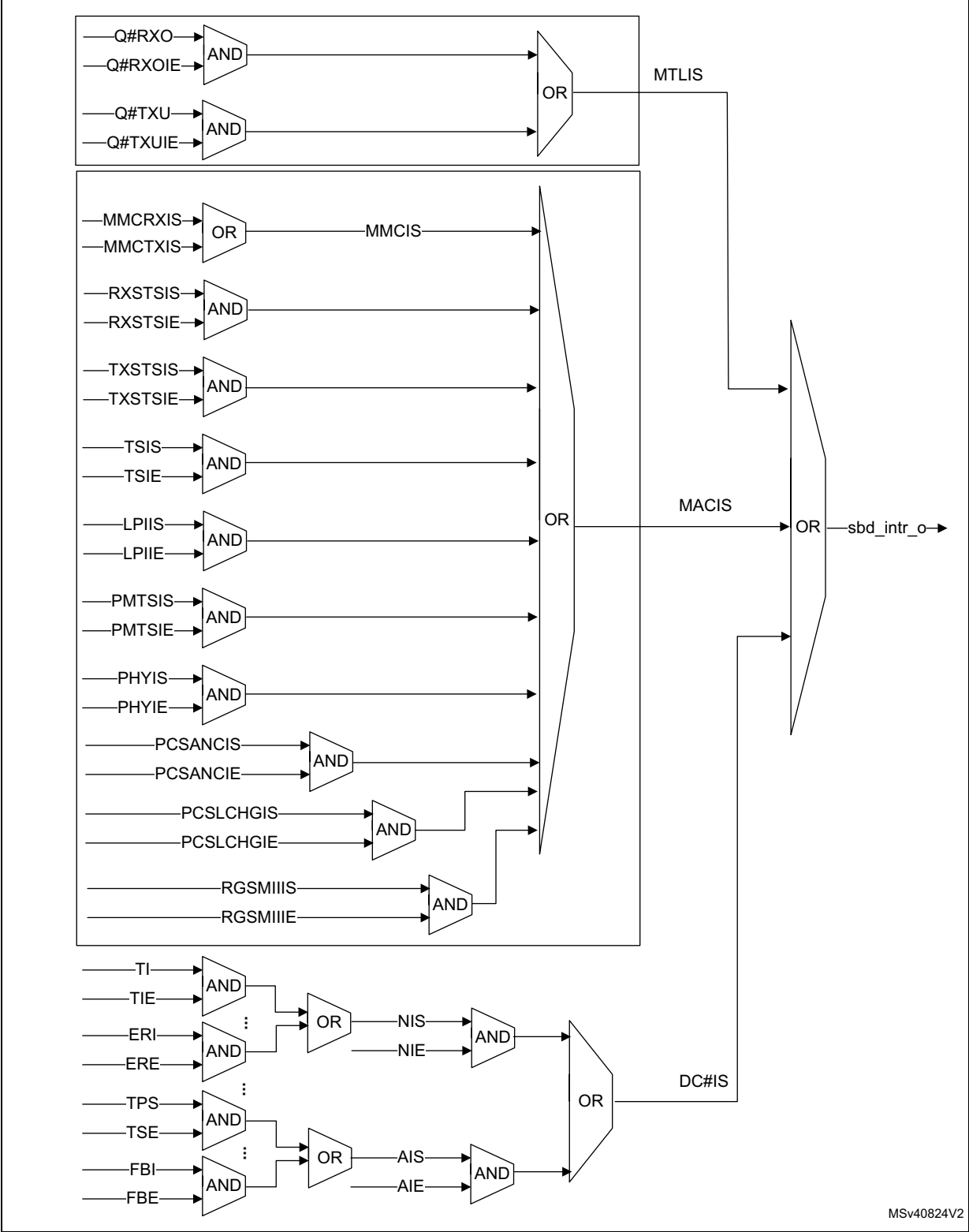
ビット 1 TXSE : 送信停止有効化

このビットが AIE ビットとともにセットされた場合、送信停止割込みが有効になります。このビットがリセットされると、送信停止割込みが無効になります。

ビット 0 TIE : 送信割込み有効化

このビットが NIE ビットとともにセットされた場合、送信割込みが有効になります。このビットがリセットされると、送信割込みが無効になります。

図 828. ETH_DMAISR フラグの生成



MSv40824V2

チャンネル Rx 割込みウォッチドッグタイマレジスタ (ETH_DMARXIWTR)

アドレスオフセット : 0x1138

リセット値 : 0x0000 0000

受信割込みウォッチドッグタイマレジスタは、DMA からの受信割込み (RI) のウォッチドッグタイムアウトを示します。このレジスタに 0 以外の値を書き込むと、[チャンネルステータスレジスタ \(ETH_DMCSR\)](#) の RI ビットに対してウォッチドッグタイマを有効にします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWTU[1:0]	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **RWTU[1:0]** : 受信割込みウォッチドッグタイマカウント単位

このフィールドは、RWT[7:0] フィールド内の 1 つの単位に対応するシステムクロックサイクル数を示します。

00 : 256

01 : 512

10 : 1024

11 : 2048

たとえば、RWT[7:0] = 2 で RWTU[1:0] = 1 の場合、 $2 * 512 = 1024$ のシステムクロックサイクルに対してウォッチドッグタイマがセットされます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RWT[7:0]** : 受信割込みウォッチドッグタイマカウント

このフィールドは、ウォッチドッグタイマがセットされる RWTU フィールドに示される係数によって掛けられた、システムクロックサイクル数を示します。

ウォッチドッグタイマは、対応するディスクリプタの割込み有効ビット RDES3[30] の設定のため、Rx DMA が ETH_DMCSR で RI ビットがセットされていないパケットの転送を完了した後に、プログラムされた値でトリガされます。

ウォッチドッグタイマがタイムアウトすると、RI ビットがセットされ、タイマが停止します。受信パケットの割込み有効ビット RDES3[30] に従う RI の自動設定のために RI ビットがハイに設定されたとき、ウォッチドッグタイマはリセットされます。

チャンネル現在のアプリケーション送信ディスクリプタレジスタ (ETH_DMACCATXDR)

アドレスオフセット : 0x1144

リセット値 : 0x0000 0000

チャンネル現在のアプリケーション送信ディスクリプタレジスタは、DMA によって読み出された現在の送信ディスクリプタを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTDESAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTDESAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CURTDESAPTR[31:0]** : アプリケーション送信ディスクリプタアドレスポインタ

DMA は Tx 動作中にこのポインタを更新します。このポインタはリセット時にクリアされます。

チャンネル現在のアプリケーション受信ディスクリプタレジスタ (ETH_DMACCARXDR)

アドレスオフセット : 0x114C

リセット値 : 0x0000 0000

チャンネル現在のアプリケーション受信ディスクリプタレジスタは、DMA によって読み出された現在の受信ディスクリプタを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRDESAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRDESAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CURRDESAPTR[31:0]** : アプリケーション受信ディスクリプタアドレスポインタ

DMA は Rx 動作中にこのポインタを更新します。このポインタはリセット時にクリアされます。

チャンネル現在のアプリケーション送信バッファレジスタ (ETH_DMACCATXBR)

アドレスオフセット : 0x1154

リセット値 : 0x0000 0000

チャンネル現在のアプリケーション送信バッファアドレスレジスタは、DMA によって読み出された現在の Tx バッファアドレスを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTBUFAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTBUFAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CURTBUFAPTR[31:0]** : アプリケーション送信バッファアドレスポインタ

DMA は Tx 動作中にこのポインタを更新します。このポインタはリセット時にクリアされます。

チャンネル現在のアプリケーション受信バッファレジスタ (ETH_DMACCARXBR)

アドレスオフセット : 0x115C

リセット値 : 0x0000 0000

チャンネル現在のアプリケーション受信バッファアドレスレジスタは、DMA によって読み出された現在の Rx バッファアドレスを示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRBUFAPTR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRBUFAPTR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CURRBUFAPTR[31:0]** : アプリケーション受信バッファアドレスポインタ

DMA は Rx 動作中にこのポインタを更新します。このポインタはリセット時にクリアされます。

チャンネルステータスレジスタ (ETH_DMCSR)

アドレスオフセット : 0x1160

リセット値 : 0x0000 0000

ソフトウェアドライバ (アプリケーション) は、割込みサービスルーチン中またはポーリング中に、ステータスレジスタを読み出して、DMA のステータスを決定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]		
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rw	rc_w1	rc_w1	rc_w1				rc_w1	rc_w1	rc_w1

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:19 **REB[2:0]** : Rx DMA エラービット

このフィールドは、バスエラーの原因となったエラーのタイプを示します。たとえば、AHB インタフェース上でのエラーレスポンスがあります。

ビット [2] : 1 の場合は Rx DMA によるデータ転送中にエラーあり、0 の場合は Rx DMA によるデータ転送中にエラーなし。

ビット [1] : 1 の場合はディスクリプタアクセス中のエラー、0 の場合はデータバッファアクセス中のエラー
ビット [0] : 1 の場合は読出し転送中のエラー、0 の場合は書込み転送中のエラー

このフィールドは、FBE ビットがセットされたときにのみ有効です。このフィールドは割込みを生成しません。

ビット 18:16 **TEB[2:0]** : Tx DMA エラービット

このフィールドは、バスエラーの原因となったエラーのタイプを示します。たとえば、AHB インタフェース上でのエラーレスポンスがあります。

ビット [2] : 1 の場合は Tx DMA によるデータ転送中にエラーあり、0 の場合は Tx DMA によるデータ転送中にエラーなし

ビット [1] : 1 の場合はディスクリプタアクセス中のエラー、0 の場合はデータバッファアクセス中のエラー
ビット [0] : 1 の場合は読出し転送中のエラー、0 の場合は書込み転送中のエラー

このフィールドは、FBE ビットがセットされたときにのみ有効です。このフィールドは割込みを生成しません。

ビット 15 **NIS** : 通常割込み要約

通常割込み要約ビットの値は、ETH_DMACIER レジスタで対応する割込みビットが有効なとき、以下のビットの論理 OR です。

ビット 0 : 送信割込み

ビット 2 : 送信バッファ使用不可

ビット 6 : 受信割込み

ビット 11 : アーリー受信割込み

マスクされていないビット (割込み有効化が ETH_DMACIER レジスタでセットされている割込み) のみが、通常割込み要約ビットに影響します。

これはスティッキービットです。NIS をセット状態にする対応ビットがクリアされるたびに、このビットを (このビットに 1 を書き込むことによって) クリアしなければなりません。

ビット 14 AIS : 異常割込み要約

異常割込み要約ビットの値は、ETH_DMACIER レジスタで対応する割込みビットが有効なとき、以下の論理 OR です。

ビット 1 : 送信プロセス停止

ビット 7 : 受信バッファ使用不可

ビット 8 : 受信プロセス停止

ビット 10 : アーリー送信割込み

ビット 12 : 致命的バスエラー

ビット 13 : コンテキストディスクリプタエラー

マスクされていないビットのみが異常割込み要約ビットに影響します。

これはスティッキービットです。AIS をセット状態にする対応ビットがクリアされるたびに、このビットを（このビットに 1 を書き込むことによって）クリアしなければなりません。

ビット 13 CDE : コンテキストディスクリプタエラー

このビットは、DMA Tx/Rx エンジンがディスクリプタエラーを受信したことを示します。パケットフローの途中で無効なコンテキストがある（中間のディスクリプタ）か、Tx の場合はすべてが 1 のディスクリプタであることを示します。また、Rx 側では DMA がいずれかのバッファアドレスで無効なものとしてディスクリプタを読み出したことを示します。

ビット 12 FBE : 致命的バスエラー

このビットは、(EB フィールドで示される) バスエラーが発生したことを示します。このビットがセットされると、対応する DMA チャンネルエンジンは、すべてのバスアクセスを無効にします。

ビット 11 ERI : アーリー受信割込み

このビットは、DMA がパケットの最初のデータバッファを一杯にしたことを示します。このレジスタの RI ビットが、このビットを自動的にクリアします。

ビット 10 ETI : アーリー送信割込み

このビットは、パケットが完全に MTL Tx FIFO に転送されたことを示します。

ビット 9 RWT : 受信ウォッチドッグタイムアウト

このビットは、2,048 バイト（ジャンボパケットモードが有効の場合は 10,240 バイト）より長いパケットが受信されたときにアサートされます。

ビット 8 RPS : 受信プロセス停止

このビットは、Rx プロセスが停止状態に入ったときにアサートされます。

ビット 7 RBU : 受信バッファ使用不可

このビットは、アプリケーションが受信リストに次のディスクリプタを所有しており、DMA が取得できないことを示します。Rx プロセスはサスペンドされます。Rx ディスクリプタの処理を再開するには、アプリケーションは、ディスクリプタの所有権を変更して、受信ポーリング要求コマンドを発行しなければなりません。このコマンドが発行されなかった場合、Rx プロセスは、認識可能な次の着信パケットが受信されたときに再開されます。リングモードでは、アプリケーションはチャンネルの受信ディスクリプタテイルポインタレジスタを上げる必要があります。このビットは、DMA が前の Rx ディスクリプタを所有している場合のみセットされます。

ビット 6 RI : 受信割込み

このビットはパケット受信が完了したことを示します。パケット受信が完了すると、RDES1 のビット 31 が最終ディスクリプタでリセットされ、特定のパケットステータス情報がそのディスクリプタで更新されます。

受信は実行中状態のままです。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TBU** : 送信バッファ使用不可

このビットは、アプリケーションが送信リストに次のディスクリプタを所有しており、DMA が取得できないことを示します。送信は中断されます。[デバッグステータスレジスタ \(ETH_DMADSR\)](#) レジスタの TPSi フィールドは、送信プロセス状態の遷移を表します。

送信ディスクリプタの処理を再開するには、アプリケーションは以下の手順を実行する必要があります。

1. TDES3 のビット 31 をセットすることによってディスクリプタの所有権を変更します。
2. 送信ポーリング要求コマンドを発行します。

リングモードでは、アプリケーションはチャンネルの送信ディスクリプタテイルポインタレジスタを上げる必要があります。

ビット 1 **TPS** : 送信プロセス停止

このビットは、送信が停止されたときにセットされます。

ビット 0 **TI** : 送信割込み

このビットはパケット送信が完了したことを示します。送信が完了すると、TDES3 のビット 31 が最終ディスクリプタでリセットされ、特定のパケットステータス情報がそのディスクリプタで更新されます。

チャンネル欠落フレームカウントレジスタ (ETH_DMAMFCR)

アドレスオフセット : 0x116C

リセット値 : 0x0000 0000

このレジスタには、バスエラーまたは[チャンネル受信制御レジスタ \(ETH_DMARXCR\)](#) レジスタの RPF フィールドのプログラムにより、DMA がドロップしたパケット数のカウンタがあります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFC0	Res.	Res.	Res.	Res.	MFC[10:0]										
rc_r					rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **MFC0** : MFC カウンタのオーバーフローステータス

このビットがセットされた場合、MFC カウンタはそれ以上インクリメントされません。このレジスタが読み出されると、ビットはクリアされます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MFC[10:0]** : ドロップされたパケットのカウンタ

このカウンタは、バスエラーまたは[チャンネル受信制御レジスタ \(ETH_DMARXCR\)](#) の RPF フィールドのプログラミングが原因で DMA でドロップしたパケットカウンタの数を示します。このレジスタが読み出されると、カウンタはクリアされます。

Ethernet DMA レジスタマップとリセット値

表 677. ETH_DMA 共通レジスタマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x1000	ETH_DMAMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM[1:0]		Res.	Res.	PR[2:0]		TXPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.			DA	SWP	
	リセット値															0	0		0	0	0	0										0	0	
0x1004	ETH_DMASBMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RB	MB	Res.	AAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FB	
	リセット値																0	0	0	0													0	0
0x1008	ETH_DMAISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC0IS	
	リセット値															0	0																0	0
0x100C	ETH_DMADSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TPS0 [3:0]			RPS0 [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXWHSTS
	リセット値																	0	0	0	0	0	0	0	0								0	0

表 678. ETH_DMA_CH レジスタマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x1100	ETH_DMCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBLX8	Res.	Res.	MSS[13:0]														
	リセット値												0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1104	ETH_DMACTXCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXPBL[5:0]					Res.	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	Res.	Res.	ST		
	リセット値											0	0	0	0	0	0			0									0			0		
0x1108	ETH_DMACRXCR	RPF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXPBL[5:0]					Res.	RBSZ[13:0]													Res.	Res.	SR	
	リセット値	0										0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0				0	
0x110C - 01110	予約済みです。																																	
0x1114	ETH_DMACTXDLAR	TDESLA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1118	予約済みです。																																	
0x111C	ETH_DMACRXDLAR	RDESLA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1120	ETH_DMACTXDTPR	TDT[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 678. ETH DMA CH レジスタマップとリセット値 (続き)

[illegible]

表 678. ETH_DMA_CH レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x116C	ETH DMACMFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MFCO	Res.	Res.	Res.	Res.	Res.	MFC[10:0]									
	リセット値																	0					0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

57.11.3 Ethernet MTL レジスタ

動作モードレジスタ (ETH_MTL0MR)

アドレスオフセット : 0x0C00

リセット値 : 0x0000 0000

動作モードレジスタは、送信および受信動作モードとコマンドを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CNT CLR	CNT PRST	Res.	Res.	Res.	Res.	Res.	Res.	DTX STS	Res.
						rw	rw							rw	

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CNTCLR** : カウンタリセット

このビットがセットされた場合、すべてのカウンタがリセットされます。このビットは、1 クロックサイクル後に自動的にクリアされます。

このビットが CNTPRST ビットとともにセットされた場合、CNTPRST が優先されます。

ビット 8 **CNTPRST** : カウンタプリセット

このビットがセットされる場合 :

- Tx キューアンダーフローレジスタ (ETH_MTLTXQUR) は 0x7F0 に初期化/プリセットされます。
- Rx キュー欠落パケットおよびオーバーフローカウンタレジスタ (ETH_MTLRXQMPOCR) の欠落パケットおよびオーバーフローパケットカウンタは、0x7F0 に初期化/プリセットされます。

このビットは自動的にクリアされます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DTXSTS** : ドロップ送信ステータス

このビットがセットされた場合、MAC から受信した Tx パケットステータスが MTL でドロップされます。このビットがリセットされた場合、MAC から受信した Tx パケットステータスがアプリケーションに転送されます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

割込みステータスレジスタ (ETH_MTLISR)

アドレスオフセット : 0x0C20

リセット値 : 0x0000 0000

ソフトウェアドライバ (アプリケーション) は、割込みサービスルーチン中またはポーリング中に、このレジスタを読み出して、MTL キュー、MAC の割込みステータスを決定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Q0IS
															r

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **Q0IS** : キュー 割込みステータス

このビットは、割込みがキューによって生成されたことを示します。このビットをリセットするために、ETH_MTLQICSR レジスタを読み出して、割込みの原因を特定し、そのソースをクリアします。

Tx キュー動作モードレジスタ (ETH_MTLTXQOMR)

アドレスオフセット : 0x0D00

リセット値 : 0x0007 0008

キュー送信動作モードレジスタは、送信キュー動作モードとコマンドを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
									rw	rw	rw	r	r	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **TQS[2:0]** : 送信キューサイズ

このフィールドは、割り当てられた送信キューのサイズを 256 バイトのブロック単位で示します。
キューサイズの範囲は 256 バイト (TQS=0b000) から 2048 バイト (TQS=0b111) までです。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **TTC[2:0]** : 送信閾値制御

これらのビットは、MTL Tx キューの閾値レベルを制御します。送信は、MTL Tx キュー内のパケットサイズが閾値より大きいときに開始されます。さらに、閾値より短い完全なパケットも送信されます。これらのビットは、TSF ビットがリセットされたときのみ使用されます。

000 : 32
001 : 64
010 : 96
011 : 128
100 : 192
101 : 256
110 : 384
111 : 512

ビット 3:2 **TXQEN[1:0]** : 送信キュー有効化

このフィールドは、送信キューの有効化/無効化に使用されます。

00 : 無効
10 : 有効

その他 : 予約済み。使用できません。

注 : 複数の Tx キュー設定では、すべてのキューがデフォルトで無効化されています。このフィールドをプログラムすることで Tx キューを有効にします。

ビット 1 **TSF** : 送信ストアアンドフォワード

このビットがセットされると、完全なパケットが MTL Tx キューに存在しているときに送信が開始されます。このビットがセットされると、このレジスタのビット [6:4] で指定された TTC の値は無視されます。このビットは、送信が停止されたときのみ変更されなければなりません。

ビット 0 **FTQ** : 送信キューの一掃

このビットがセットされた場合、Tx キューコントローラロジックはデフォルト値にリセットされます。そのため、Tx キューのすべてのデータが失われるか一掃されます。このビットは、一掃動作が完了したときに内部的にリセットされます。このビットがリセットされるまで、ETH_MTLTXQOMR レジスタに書き込まないでください。MAC トランスミッタによってすでに許可されたデータは一掃されません。送信にスケジュールされ、アンダーフローおよびラントパケット送信となります。

注 : 一掃動作は、Tx キューがエンプティで、アプリケーションがすべての送信パケットのペンディング中の Tx ステータスを許可したときのみ完了します。この一掃動作を完了するには、PHY Tx クロック (eth_mii_tx_clk) がアクティブでなければなりません。

Tx キューアンダーフローレジスタ (ETH_MTLTXQUR)

アドレスオフセット : 0x0D04

リセット値 : 0x0000 0000

キューアンダーフローカウンタレジスタには、送信キューアンダーフローのためにアボートされたパケットおよび受信キューパケット一掃のために欠落したパケットのカウントがあります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UFCNT OVF	UFFRMCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UFCNTOVF** : アンダーフローパケットカウンタのオーバーフロービット

このビットは、Tx キューアンダーフローパケットカウンタフィールドがオーバーフローし、最大数を超えるたびにセットされます。このようなシナリオでは、オーバーフローパケットカウンタがすべて 0 にリセットされ、このビットはロールオーバーが発生したことを示します。

ビット 10:0 **UFFRMCNT[10:0]** : アンダーフローパケットカウンタ

このフィールドは、Tx キューアンダーフローのためにコントローラによってアボートされたパケットの数を示します。このカウンタは、アンダーフローのために MAC が進行中のパケットをアボートするたびにインクリメントされます。このレジスタが読み出されると、カウンタはクリアされます。

Tx キューデバッグレジスタ (ETH_MTLTXQDR)

アドレスオフセット : 0x0D08

リセット値 : 0x0000 0000

キュー送信デバッグレジスタは、送信キューに関連するさまざまなブロックのデバッグステータスを提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTS[2:0]			Res.	PTXQ[2:0]		
									r	r	r		r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTS FSTS	TXQ STS	TWC STS	TRCSTS[1:0]	TXQPA USED	
										r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **STXSTS[2:0]** : キューの Tx ステータス FIFO のステータスワード数

このフィールドは、このキューの Tx ステータス FIFO の現在のステータス数を示します。

ETH_MTLQMR レジスタの DTXSTS ビットが 1 にセットされている場合、このフィールドは Tx ステータス FIFO のステータスワード数を反映しません。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **PTXQ[2:0]** : 送信キューの packets 数

このフィールドは、Tx キューの現在の packets 数を示します。

動作モードレジスタ (ETH_MTL0MR) レジスタの DTXSTS ビットが 1 にセットされている場合、このフィールドは送信キューの packets 数を反映しません。

ビット 15:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TXSTSFSTS** : MTL Tx ステータス FIFO フルステータス

ハイになっている場合、このビットは MTL Tx ステータス FIFO がフルになっていることを示します。そのため、MTL は送信のための追加の packets を許可できません。

ビット 4 **TXQSTS** : MTL Tx キューノットエンプティステータス

このビットがハイの場合、MTL Tx キューがエンプティではなく、送信するデータが残っていることを示します。

ビット 3 **TWCSTS** : MTL Tx キュー書き込みコントローラステータス

ハイの場合、このビットは MTL Tx キュー書き込みコントローラがアクティブで、Tx キューにデータを転送していることを示します。

ビット 2:1 **TRCSTS[1:0]** : MTL Tx キュー読み出しコントローラステータス

このフィールドは、Tx キュー読み出しコントローラの状態を示します。

00 : アイドル状態

01 : 読み出し状態 (データを MAC トランスミッタへ転送中)

10 : MAC トランスミッタからのペンディング中の Tx ステータス待ち

11 : MAC からの packets アポートリクエストによる Tx キューの一掃

ビット 0 **TXQPAUSED** : ポーズ中の送信キュー

このビットがハイで、Rx フロー制御が有効である場合、Tx キューは次の理由により、(全二重専用モード) ポーズ状態となります。

- PFC 有効時に Tx キューに割り当てられた優先順位のための PFC packets の受信
- PFC 有効時の 802.3x ポーズ packets の受信

キュー割込み制御ステータスレジスタ (ETH_MTLQICSR)

アドレスオフセット : 0x0D2C

リセット値 : 0x0000 0000

このレジスタは、キュー割込みの割込み有効化およびステータスビットを格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX0IE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX0FIS
							rw								rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX0IE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX0FIS
							rw								rc_w1

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **RX0IE** : 受信キューオーバーフロー割込み有効化

このビットがセットされた場合、受信キューオーバーフロー割込みが有効になります。このビットがリセットされた場合、受信キューオーバーフロー割込みが無効になります。

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **RXOVFIS** : 受信キューオーバーフロー割込みステータス

このビットは、パケット受信中に受信キューにオーバーフローがあったことを示します。部分的なパケットがアプリケーションに転送された場合、RDES3[21] でオーバーフローステータスがセットされます。このビットは、アプリケーションがこのビットに 1 を書き込むとクリアされます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TXUIE** : 送信キューアンダーフロー割込み有効化

このビットがセットされた場合、送信キューアンダーフロー割込みが有効になります。このビットがリセットされた場合、送信キューアンダーフロー割込みが無効になります。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **TXUNFIS** : 送信キューアンダーフロー割込みステータス

このビットは、パケット送信中に送信キューにアンダーフローがあったことを示します。送信は中断され、アンダーフローエラー TDES3[2] がセットされます。このビットは、アプリケーションがこのビットに 1 を書き込むとクリアされます。

Rx キュー動作モードレジスタ (ETH_MTLRXQOMR)

アドレスオフセット : 0x0D30

リセット値 : 0x0070 0000

キュー受信動作モードレジスタは、受信キュー動作モードとコマンドを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS[2:0]			Res.	Res.	Res.	Res.
									r	r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC[1:0]	
									rw	rw	rw	rw		rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **RQS[2:0]** : 受信キューサイズ

このフィールドは読み出し専用で、設定された 256 バイトのブロック単位の Rx FIFO サイズはリセット値に反映されます。キューのサイズは $(RQS + 1) * 256$ バイトです。

ビット 19:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DIS_TCP_EF** : TCP/IP チェックサムエラーパケットのドロップングディスエーブル

このビットがセットされた場合、MAC は受信チェックサムオフロードエンジンによって検出されたエラーだけがあるパケットをドロップしません。このようなパケットでは、カプセル化されたペイロードでのみエラーがあります。MAC から受信した Ethernet パケットにエラー (FCS エラーを含む) はありません。

このビットがリセットされると、FEF ビットがリセットされていた場合、すべてのエラーパケットがドロップされます。

ビット 5 **RSF** : 受信キューストアアンドフォワード

このビットがセットされた場合、Ethernet ペリフェラルは、完全なパケットが書き込まれた後のみ、Rx キューからパケットを読み出し、このレジスタの RTC フィールドは無視されます。このビットがリセットされると、Rx キューは、このレジスタの RTC フィールドで指定された閾値に従って、閾値 (カットスルー) モードで動作します。

ビット 4 FEP : エラーパケット転送

このビットがリセットされると、Rx キューは、エラーステータス (CRC エラー、受信エラー、ウォッチドッグタイムアウト、オーバーフロー) のパケットをドロップします。ただし、パケットの開始バイト (書き込み) ポインタが読み出しコントローラ側にすでに転送されている場合 (閾値モードで)、パケットはドロップされません。

このビットがセットされた場合、ラントエラーパケットを除くすべてのパケットがアプリケーションまたは DMA に転送されます。RSF ビットがセットされ、部分的なパケットが書き込まれたときに Rx キューがオーバーフローした場合、このビットの設定にかかわらずパケットがドロップされます。ただし、RSF ビットがリセットされ、部分的なパケットが書き込まれたときに Rx キューがオーバーフローした場合、部分的なパケットはアプリケーションまたは DMA に転送できます。

ビット 3 FUP : アンダーサイズの良いパケットの転送

このビットがセットされた場合、Rx キューは、パッドバイトと CRC を含めて、アンダーサイズの良いパケット (エラーがなく、長さが 64 バイト未満のパケット) を転送します。このビットがリセットされると、RTC = 01 など Rx 閾値の低い値であるために、パケットがすでに転送されていない限り、Rx キューは 64 バイト未満のすべてのパケットをドロップします。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 RTC[1:0] : 受信キュー閾値制御

これらのビットは、MTL Rx キューの閾値レベルを (バイト単位で) 制御します。

00 : 64

01 : 32

10 : 96

11 : 128

受信パケットは、MTL Rx キュー内のパケットサイズが閾値より大きいときに、アプリケーションまたは DMA に転送されます。さらに、閾値より短い完全なパケットは自動的に転送されます。

このフィールドは、RSF ビットが 0 である場合のみ有効です。このフィールドは、RSF ビットが 1 にセットされた場合は無視されます。

Rx キュー欠落パケットおよびオーバーフローカウンタレジスタ (ETH_MTLRXQMPOCR)

アドレスオフセット : 0x0D34

リセット値 : 0x0000 0000

キュー欠落パケットおよびオーバーフローカウンタレジスタには、受信キューパケット一掃のために欠落したパケットおよび受信キューオーバーフローのために破棄されたパケットのカウンタがあります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MISCN TOVF	MISPKTCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	OVFCN TOVF	OVFPKTCNT[10:0]										
				rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **MISCNTOVF** : 欠落パケットカウンタオーバーフロービット

セットされた場合、このビットは Rx キュー欠落パケットカウンタが最大制限を超えたことを示します。

ビット 26:16 **MISPKTCNT[10:0]** : 欠落パケットカウンタ

このフィールドは、アプリケーションがこのキューに対してパケットの一掃をリクエストしたため、Ethernet ペリフェラルによって見逃されたパケットの数を示します。このレジスタが読み出されると、このカウンタはリセットされます。

このカウンタは、バッファ使用不可のために DMA がパケットを破棄すると 1 つインクリメントされます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **OVFCNTOVF** : オーバーフローカウンタオーバーフロービット

セットされた場合、このビットは Rx キューオーバーフローパケットカウンタが最大制限を超えたことを示します。

ビット 10:0 **OVFPKTCNT[10:0]** : オーバーフローパケットカウンタ

このフィールドは、受信キューオーバーフローのために Ethernet ペリフェラルによって破棄されたパケットの数を示します。このカウンタは、Ethernet ペリフェラルがオーバーフローのために着信パケットを破棄するたびにインクリメントされます。このレジスタが読み出されると、このカウンタはリセットされます。

Rx キューデバッグレジスタ (ETH_MTLRXQDR)

アドレスオフセット : 0x0D38

リセット値 : 0x0000 0000

キュー受信デバッグレジスタは、受信キューに関連するさまざまなブロックのデバッグステータスを提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PRXQ[13:0]													
		r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
										r	r		r	r	r

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:16 **PRXQ[13:0]** : 受信キューの packets 数

このフィールドは、Rx キューの現在の packets 数を示します。このフィールドの理論上の最大値は 256Kbyte/16bytes = 16K packets で、Max_Queue_Size/Min_Packet_Size です。

ビット 15:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **RXQSTS[1:0]** : MTL Rx キューフィルレベルステータス

このフィールドは、Rx キューのフィルレベルのステータスを提供します。

00 : Rx キューは空です。

01 : Rx キューフィルレベルがフロー制御停止閾値より下です。

10 : Rx キューフィルレベルがフロー制御開始閾値より上です。

11 : Rx キューはフルです。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **RRCSTS[1:0]** : MTL Rx キュー読み出しコントローラ状態

このフィールドは、Rx キュー読み出しコントローラの状態を提供します。

00 : アイドル状態

01 : パケットデータの読み出し中

10 : パケットステータス (またはタイムスタンプ) 読み出し中

11 : パケットデータとステータスの一掃中

ビット 0 **RWCSTS** : MTL Rx キュー書き込みコントローラアクティブステータス

ハイの場合、このビットは MTL Rx キュー書き込みコントローラがアクティブで、Rx キューに受信パケットを転送していることを示します。

Ethernet MTL レジスタマップとリセット値

表 679. ETH_MTL レジスタマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0C00	ETH_MTLOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0
	リセット値																							0	0	0	0	0	0	0	0	0	0	
0x0C04 - 0x0C1C	予約済みです。																																	
0x0C20	ETH_MTLISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Q0IS	
	リセット値																																0	
0x0C24 - 0x0CFC	予約済みです。																																	
0x0C40	予約済みです。																																	
0x0D00	ETH_MTLTXQOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]		TXQEN[1:0]		Tsf	FTQ	
	リセット値														1	1	1										0	0	0	1	0	0	0	
0x0D04	ETH_MTLTXQUR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UFCNTOVF	UFFRMCNT[10:0]										
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0	
0x0D08	ETH_MTLTXQDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]		Res.	PTXQ[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]	TXQPAUSED	
	リセット値										0	0	0		0	0	0											0	0	0	0	0	0	
0x0D0C - 0x0D28	予約済みです。																																	
0x0D2C	ETH_MTLQICSR	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	TXUNFIS	
	リセット値							0								0									0							0		
0x0D30	ETH_MTLRXQOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIS	TCP	EF	RSF	FEP	FUP	Res.	RTC[1:0]
	リセット値										1	1	1														0	0	0	0			0	0

表 679. ETH_MTL レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D34	ETH_ MTLRXQMPO CR	Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]										Res.	Res.	Res.	Res.	OVCNTOVF	OVFPKTCNT[10:0]											
	リセット値					0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0
0x0D38	ETH_ MTLRXQDR	Res.	Res.	PRXQ[13:0]													Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXCSTS[1:0]	Res.	RRCSTS[1:0]	RWCSTS				
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0											0	0		0	0	0	0
0xD3C - 0xD58	予約済みです。																																

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。



57.11.4 Ethernet MAC および MMC レジスタ

動作モード設定レジスタ (ETH_MACCR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 0000

MAC 設定レジスタは、MAC の動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPEN :	SARC[2:0]			IPC	IPG[2:0]			GPSLCE	S2KP	CST	ACS	WD	Res.	JD	JE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res.	BL[1:0]		DC	PRELEN[1:0]		TE	RE
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **ARPEN** : ARP オフロード有効化

このビットがセットされた場合、MAC は着信 ARP リクエストパケットを認識でき、送信用の ARP パケットをスケジュールします。ARP パケットをアプリケーションに転送し、Rx ステータスでイベントも示します。

このビットがリセットされた場合、MAC は ARP パケットを認識できず、Rx ステータスでタイプフレームとして示します。

ビット 30:28 **SARC[2:0]** : ソースアドレスの挿入または書き換え制御

このフィールドは、すべての送信パケットのソースアドレスの挿入または書き換えを制御します。ビット 30 は、ビット [29:28] の値に基づいたソースアドレスの挿入または書き換えに使用される MAC アドレスレジスタ (0 または 1) を指定します。

010 : MAC はすべての送信パケットの SA フィールドに、MAC アドレス 0 レジスタ (**MACアドレス 0 上位レジスタ (ETH_MACA0HR)** および **MACアドレス x 下位レジスタ (ETH_MACAxLR)**) の内容を挿入します。

011 : MAC はすべての送信パケットの SA フィールドの MAC アドレス 0 レジスタ (**MACアドレス 0 上位レジスタ (ETH_MACA0HR)** および **MACアドレス x 下位レジスタ (ETH_MACAxLR)**) の内容を書き換えます。

110 : MAC はすべての送信パケットの SA フィールドに、MAC アドレス 1 レジスタ (**MACアドレス x 上位レジスタ (ETH_MACAxHR)** および **MACアドレス x 下位レジスタ (ETH_MACAxLR)**) の内容を挿入します。

111 : MAC はすべての送信パケットの SA フィールドの MAC アドレス 1 レジスタ (**MACアドレス x 上位レジスタ (ETH_MACAxHR)** および **MACアドレス x 下位レジスタ (ETH_MACAxLR)**) の内容を書き換えます。

その他 : 予約済み。使用できません。

注 : このフィールドへの変更は、パケットの開始時にのみ有効になります。パケットが送信されているときに、このレジスタフィールドに書き込む場合、後続のパケットのみが更新された値を使用でき、現在のパケットは更新された値を使用しません。

ビット 27 **IPC** : チェックサムオフロード

セットされた場合、このビットは、IPv4 ヘッダチェックサムのチェックおよび IPv4 または IPv6 の TCP、UDP、ICMP ペイロードチェックサムのチェックを有効にします。このビットがリセットされた場合、レシーバの COE 機能は無効になります。

レイヤ 3 およびレイヤ 4 パケットフィルタ機能は、受信側で IPC フルチェックサムオフロードエンジンを自動的に選択します。この機能が有効になっている場合、IPC ビットをセットしなければなりません。

ビット 26:24 **IPG[2:0]** : パケット間隔

これらのビットは、送信時のパケット間の最小 IPG を制御します。

000 : 96 ビット時間

001 : 88 ビット時間

010 : 80 ビット時間

.....

111 : 40 ビット時間

最小 IPG の範囲は、全二重モードで有効です。

半二重モードでは、最小 IPG は、64 ビット時間 (IPG = 100) についてのみ設定できます。それより小さな値は考慮されません。

JAM パターンがバックプレッシャ有効化のために送信されている場合、MAC は最小 IPG を考慮しません。

上記の機能 (96 ビット時間未満の IPG) は、ETH_MACECR レジスタの EIPGEN ビットがリセットされているときのみ有効となります。EIPGEN がセットされた場合、最小 IPG (96 ビット時間を超える) は、ETH_MACECR レジスタの EIPG フィールドに示される説明に従って制御されます。

ビット 23 **GPSLCE** : ジャイアントパケットのサイズ制限制御の有効化

このビットがセットされた場合、MAC は ETH_MACECR レジスタの GPSL フィールドの値を考慮して、受信パケットをジャイアントパケットとして宣言します。このフィールドは、1,518 バイト以上にプログラムする必要があります。そうしないと、MAC は 1,518 バイトをジャイアントパケットの制限とみなします。

このビットがリセットされると、MAC は受信パケットのサイズが 1,518 バイト (タグ付きパケットの場合は 1,522 バイト) を超えた場合に、ジャイアントパケットとみなします。

ウォッチドッグタイムアウト制限、ジャンボパケットの有効化および 2K パケットの有効化は、このビットより高い優先順位であり、ジャンボパケットが有効化された状態では受信パケットのサイズが 9,018 バイト (タグ付きパケットの場合は 9,022 バイト) を超えた場合、および 2K パケットが有効化された状態では 2,000 バイトを超えた場合に、MAC はジャイアントパケットとみなします。有効化された場合、ウォッチドッグ制限に達すると、ウォッチドッグタイムアウトは受信パケットを終了します。そのため、プログラムされたジャイアントパケットの制限をウォッチドッグ制限未満にして、ジャイアントパケットステータスを取得する必要があります。

ビット 22 **S2KP** : 2K パケットの IEEE 802.3as サポート

このビットがセットされた場合、MAC は 2,000 バイト長までのすべてのパケットを通常パケットとみなします。JE ビットがセットされていない場合、MAC は 2,000 バイトを超えるサイズのすべての受信パケットをジャイアントパケットとみなします。

このビットがリセットされ、JE ビットがセットされていない場合、MAC は 1,518 バイト (タグ付きの場合は 1,522 バイト) を超えるサイズのすべての受信パケットをジャイアントパケットとみなします。

このビットと JE ビットの設定のジャイアントパケットステータスへの影響の詳細については、表 680 : [S2KP および JE ビットに基づいたジャイアントパケットステータス](#)を参照してください。

注 : JE ビットがセットされた場合、このビットの設定によるジャイアントパケットステータスへの影響はありません。

ビット 21 **CST** : タイプパケット用 CRC ストリッピング

このビットがセットされた場合、パケットをアプリケーションに転送する前に Ether タイプ (1,536 より大きいタイプフィールド) の全パケットの最後の 4 バイト (FCS) が排除され、ドロップされます。この機能は、IP チェックサムエンジン (タイプ 1) が MAC レジスタで有効になっているとき有効ではありません。この機能は、タイプ 2 チェックサムオフロードエンジンが有効化されている場合に有効です。

注 : ACS ビットとこのビットのパケット長さへの影響の詳細については、表 681 : [CST および ACS ビットに基づいたパケット長](#)を参照してください。

ビット 20 **ACS** : 自動パッドまたは CRC ストリッピング

このビットがセットされた場合、MAC は、長さフィールドの値が 1,536 バイト未満の場合のみ、着信パケットのパッドおよび FCS フィールドを削除します。長さフィールドが 1,536 バイト以上の受信パケットは、パッドまたは FCS フィールドを削除せずに、そのままアプリケーションに渡されます。

このビットがリセットされると、MAC は、着信するすべてのパケットをそのままアプリケーションに渡します。

注 : CST ビットとこのビットの設定のパケット長さへの影響の詳細については、表 681 : CST および ACS ビットに基づいたパケット長を参照してください。

ビット 19 **WD** : ウォッチドッグディスエーブル

このビットがセットされた場合、MAC は、レシーバのウォッチドッグタイマを無効にします。MAC は、最大 16,383 バイトのパケットを受信できます。

このビットがリセットされると、MAC は 2,048 バイト (JE がハイにセットされている場合は 10,240) を超えるパケットは受信できません。MAC は、2,048 バイト以降に受信したバイトを切り捨てます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **JD** : ジャバードイスエーブル

このビットがセットされた場合、MAC はトランスミッタのジャバータイマを無効にします。MAC は、最大 16,383 バイトのパケットを転送できます。

このビットがリセットされると、アプリケーションが送信中に 2,048 バイト (JE がハイにセットされている場合は 10,240) を超えるデータを送信する場合、MAC はそのパケット内の残りのバイトを送信しません。

ビット 16 **JE** : ジャンボパケット有効化

このビットがセットされた場合、MAC は Rx パケットステータスでジャイアントパケットエラーを報告せずに 9,018 バイト (VLAN タグ付きパケットの場合は 9,022 バイト) のジャンボパケットを許可します。

このビットと JE ビットの設定のジャイアントパケットステータスへの影響の詳細については、表 680 : S2KP および JE ビットに基づいたジャイアントパケットステータスを参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **FES** : MAC 速度

このビットは、10/100 Mbps モードの速度を選択します。

0 : 10 Mbps

1 : 100 Mbps

ビット 13 **DM** : 全二重モード

このビットがセットされた場合、MAC は全二重モードで動作し、送信と受信を同時に行うことができます。

ビット 12 **LM** : ループバックモード

このビットがセットされた場合、MAC は、MII でループバックモードで動作します。ループバックが正しく機能するためには、MII Rx クロック入力 (eth_mii_rx_clk) が必要です。これは Tx クロックが内部的にループバックされないためです。

ビット 11 **ECRSFD** : 全二重モードでの送信前のキャリアセンスの有効化

このビットがセットされた場合、MAC トランスミッタは、全二重モードでのパケット送信の前に CRS 信号をチェックします。CRS 信号がローレベルのときのみ、MAC が送信を開始します。

このビットがリセットされた場合、MAC トランスミッタは CRS 信号のステータスを無視します。

ビット 10 **DO** : 受信オウンの無効化

このビットがセットされた場合、半二重モードで ETH_TX_EN がアサートされたときに、MAC はパケットの受信を無効にします。このビットがリセットされた場合、MAC は PHY によって与えられるすべてのパケットを受信します。

このビットは、全二重モードでは適用できません。このビットは、全二重専用の設定ではデフォルト値で予約済みかつ読み出し専用 (RO) となります。

ビット 9 DCRS : 送信中のキャリアセンスの無効化

このビットがセットされた場合、MAC トランスミッタは、半二重モードでのパケット送信中に MII CRS 信号を無視します。その結果、送信中のキャリアの喪失やキャリアなしによるエラーは生成されません。このビットがリセットされた場合、MAC トランスミッタはキャリアセンスのためにエラーを生成します。MAC は送信もアボートする可能性があります。

ビット 8 DR : 再試行の無効化

このビットがセットされると、MAC は送信を 1 回だけ試みます。MII インタフェースで衝突が発生すると、MAC は現在のパケット送信を無視して、Tx パケットステータスで過剰衝突エラーによるパケットアボートを報告します。

このビットがリセットされた場合、MAC は、BL フィールドの設定に基づいて再試行を行います。このビットは、半二重モードでのみ適用されます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:5 BL[1:0] : バックオフリミット

バックオフリミットは、衝突の後、再試行時に送信の試みを再スケジュールする前に、MAC が待機するスロット時間遅延 (10/100 Mbps の場合は 512 ビット時間) のランダム整数値 (r) を指定します:

00 : $k = \min(n, 10)$

01 : $k = \min(n, 8)$

10 : $k = \min(n, 4)$

11 : $k = \min(n, 1)$

ここで、 n = 再送信試行回数

ランダム整数値 r は、 $0 \leq r < 2^k$ の範囲の値を取ります。

このビットは、半二重モードでのみ適用されます。

ビット 4 DC : 遅延チェック

このビットがセットされた場合、MAC の遅延チェック機能が有効になります。MAC は、Tx ステートマシンが 10 または 100 Mbps モードで 24,288 ビット時間以上遅延した場合、パケットアボートステータスを発行して、Tx パケットステータスの過剰遅延エラービットをセットします。

遅延は、トランスミッタの送信準備ができたときに開始しますが、MII 上に有効なキャリアセンス信号 (CRS) があるときには開始されません。

遅延時間は累積されません。たとえば、アクティブである CRS 信号が非アクティブになったために、トランスミッタが 10,000 ビット時間遅延した場合、トランスミッタが送信して衝突が発生します。衝突のため、トランスミッタはバックオフしてから、完了後に再度遅延する必要があります。このようなシナリオでは、遅延タイマが 0 にリセットされ、再開されます。

このビットがリセットされた場合、遅延チェック機能は無効であり、MAC は CRS 信号が無効になるまで遅延します。

このビットは、半二重モードでのみ適用されます。

ビット 3:2 PRELEN[1:0] : 送信パケット用プリアンブル長

これらのビットは、各 Tx パケットの冒頭に追加されるプリアンブルバイトの数を制御します。MAC が全二重モードで動作している場合のみ、プリアンブルの削減が発生します。

00 : 7 バイトのプリアンブル

01 : 5 バイトのプリアンブル

10 : 3 バイトのプリアンブル

11 : 予約済み。使用できません。

ビット 1 TE : トランスミッタ有効化

このビットがセットされた場合、MAC の Tx ステートマシンは、MII インタフェース上で送信を行うために有効になります。このビットがリセットされると、MAC Tx ステートマシンは現在のパケットの送信完了後に無効になります。Tx ステートマシンはそれ以上パケットを送信しません。

ビット 0 RE : レシーバ有効化

このビットがセットされた場合、MAC の Rx ステートマシンは、MII インタフェースからパケットを受信するために有効になります。このビットがリセットされると、MAC Rx ステートマシンは現在のパケットの受信完了後に無効になります。Rx ステートマシンは、MII インタフェースからそれ以上パケットを受信しません。

表 680 に、ETH_MACCR レジスタの S2KP および JE ビットの設定によるジャイアントパケットステータスへの影響を示します。

表 680. S2KP および JE ビットに基づいたジャイアントパケットステータス⁽¹⁾

長さ/タイプ フィールド	受信パケット長	S2KP	JE	ジャイアント パケットステータス
タグなしパケット	> 1,518	0	0	1
	> 2,000	1	0	1
	> 9,018	x	1	1
VLAN タグ付き パケット	> 1,522	0	0	1
	> 2,000	1	0	1
	> 9,022	x	1	1

1. 他のすべての組み合わせでは、ジャイアントパケットステータスは 0 です。

表 681 に、ETH_MACCR レジスタの CST および ACS ビットの設定による CRC の長さがパケット長に含まれるかどうかへの影響を示します。

表 681. CST および ACS ビットに基づいたパケット長

受信パケット長	CST	ACS	FCS 削除終了
< 1,536	x	0	いいえ
	x	1	はい (Ethernet パケット)
≥ 1,536	0	x	いいえ
	1	x	はい (タイプパケット)

拡張動作モード設定レジスタ (ETH_MACECR)

アドレスオフセット : 0x0004

リセット値 : 0x0000 0000

MAC 拡張設定レジスタは、MAC の動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EIPG[4:0]				EIPGEN		Res.	Res.	Res.	Res.	Res.	USP	SPEN	DCRCC
		rW	rW	rW	rW	rW	rW						rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	GPSL[13:0]													
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:25 **EIPG[4:0]** : 拡張パケット間隔

このフィールドの値は、EIPGEN ビットがセットされた場合に適用できます。このフィールド（最上位ビットとして）は、**動作モード設定レジスタ (ETH_MACCCR)** の IPG フィールドとともに、8 ビット時間単位で 96 ビット時間を超える最小 IPG を提供します。例 :

- EIPG = 0 かつ IPG = 0 で 104 ビット時間
- EIPG = 0 かつ IPG = 1 で 112 ビット時間
- EIPG = 0 かつ IPG = 2 で 120 ビット時間
- ..
- EIPG = 7 かつ IPG = 31 で 2144 ビット時間

ビット 24 **EIPGEN** : 拡張パケット間隔の有効化

このビットがセットされた場合、MAC は**動作モード設定レジスタ (ETH_MACCCR)** レジスタの EIPG フィールドおよび IPG フィールドを合わせて 8 ビット時間単位で 96 ビット時間を超える最小 IPG として解釈します。

このビットがリセットされた場合、MAC は**動作モード設定レジスタ (ETH_MACCCR)** の EIPG フィールドを無視して、IPG フィールドを 8 ビット時間単位で 96 ビット時間以下の最小 IPG として解釈します。

注 : 拡張パケット間隔機能は、全二重モードだけで動作する場合に有効にする必要があります。半二重モードで有効にすると、バックプレッシャ機能およびフレーム送信で望ましくない効果をもたらす場合があります。

ビット 23:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **USP** : ユニキャストスロープロトコルパケット検出

このビットがセットされた場合、MAC は **MACアドレス 0 上位レジスタ (ETH_MACA0HR)** および MAC アドレス 0 下位レジスタ **MACアドレス x 下位レジスタ (ETH_MACAxLR)** で指定されたステーションのユニキャストアドレスを持つスロープロトコルパケットを検出します。MAC は、スロープロトコルマルチキャストアドレス (01-80-C2-00-00-02) を持つスロープロトコルパケットも検出します。このビットがリセットされた場合、IEEE 802.3-2008、セクション 5 で指定されたスロープロトコルマルチキャストアドレスを持つスロープロトコルパケットを検出します。

ビット 17 **SPEN** : スロープロトコル検出有効化

このビットがセットされた場合、MAC はスロープロトコルパケット (Ether タイプ 0x8809) を処理し、Rx ステータスを提供します。MAC は、無効なサブタイプを持つスロープロトコルパケットを破棄します。このビットがリセットされた場合、MAC は、エラーなしスロープロトコルパケットをすべてアプリケーションに転送します。MAC は、このパケットを通常タイプパケットとみなします。

ビット 16 **DCRCC** : 受信パケット用 CRC チェックの無効化

このビットがセットされた場合、MAC レシーバは受信パケットの CRC フィールドをチェックしません。このビットがリセットされた場合、MAC レシーバは受信パケットの CRC フィールドを必ずチェックします。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **GPSL[13:0]** : ジャイアントパケットのサイズ制限

受信パケットサイズがバイト単位でこのフィールドにプログラムされた値を超える場合、MAC は受信パケットをジャイアントパケットとして宣言します。このフィールドにプログラムされた値は、1,518 バイト以上でなければなりません。それ以外のプログラムされた値は、1,518 バイトとみなされます。VLAN タグ付きパケットでは、MAC はプログラムされた値に 4 バイトを追加します。ダブル VLAN タグ付きパケットでは、MAC はプログラムされた値に 8 バイトを追加します。このフィールドの値は、ETH_MACCCR レジスタで GPSLCE ビットがセットされた場合に適用できます。

パケットフィルタ制御レジスタ (ETH_MACPFR)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

MAC のパケットフィルタレジスタは、パケットを受信するためのフィルタ制御ビットを含みます。このレジスタの制御の一部は、MAC のアドレスチェックブロックに渡されて、アドレスフィルタリングの最初のレベルを実行します。フィルタの第 2 のレベルは、不良パケットのパスや制御パケットのパスなど、その他の制御に基づいて、着信パケットに対して実行されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNTU	IPFE	Res.	Res.	Res.	VTFE
rw										rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **RA** : すべてを受信

このビットがセットされた場合、MAC レシーバモジュールは、アドレスフィルタをパスしたかどうかに関係なく、すべての受信パケットをアプリケーションに渡します。SA または DA フィルタの結果は、Rx ステータスワードの対応するビットで更新（通過または失敗）されます。

このビットがリセットされた場合、レシーバモジュールは、SA または DA アドレスフィルタを通過するパケットだけをアプリケーションに渡します。

ビット 30:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **DNTU** : IP パケット経由の非 TCP/UDP のドロップ

このビットがセットされた場合、MAC が IP パケット経由の非 TCP または UDP をドロップします。MAC はレイヤ 4 フィルタによって処理されるこれらのパケットだけを転送します。このビットがリセットされた場合、MAC は IP パケット経由の非 TCP または UDP をすべて転送します。

ビット 20 **IPFE** : レイヤ 3 およびレイヤ 4 フィルタの有効化

このビットがセットされた場合、MAC は有効化されたレイヤ 3 およびレイヤ 4 フィルタと一致しないパケットをドロップします。レイヤ 3 およびレイヤ 4 フィルタが一致に対して有効になっていない場合、このビットに効果はありません。

このビットがリセットされた場合、MAC はレイヤ 3 およびレイヤ 4 フィールドの一致ステータスに関係なく、すべてのパケットを転送します。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **VTFE** : VLAN タグフィルタの有効化

このビットがセットされた場合、MAC は VLAN タグに一致しない VLAN タグ付きパケットをドロップします。このビットがリセットされた場合、MAC は VLAN タグの一致ステータスに関係なく、すべてのパケットを転送します。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **HPF** : ハッシュまたは完全フィルタ

このビットがセットされると、アドレスフィルタが HMC または HUC ビットによってセットされた完全フィルタまたはハッシュフィルタに一致する場合、パケットを通過させます。

このビットがリセットされ、HUC または HMC ビットがセットされると、ハッシュフィルタに一致する場合のみ、パケットを通過させます。

ビット 9 **SAF** : ソースアドレスフィルタの有効化

このビットがセットされた場合、MAC は受信パケットの SA フィールドと、有効な SA レジスタでプログラムされた値とを比較します。比較に失敗した場合、MAC はパケットをドロップします。

このビットがリセットされた場合、MAC は、SA アドレス比較に応じて更新された Rx ステータスの SAF ビットで受信パケットをアプリケーションに転送します。

注 : IEEE 仕様に従って、SA のビット 47 は予約済みとなります。ただし、MAC は全 48 ビットを比較します。ソフトウェアドライバは、SA のために MAC アドレスレジスタをプログラムしている間、そのことについて考慮する必要があります。

ビット 8 **SAIF** : SA 逆フィルタ

このビットがセットされた場合、アドレスチェックブロックは、SA アドレス比較の際に逆フィルタモードで動作します。パケットの SA が SA レジスタでプログラムされた値と一致する場合、SA アドレスフィルタで失敗したものとしてマークされます。

このビットがリセットされると、パケットの SA が SA レジスタでプログラムされた値と一致しない場合、SA アドレスフィルタで失敗したものとしてマークされます。

ビット 7:6 **PCF[1:0]** : 制御パケットの通過

これらのビットは、すべての制御パケット (ユニキャストおよびマルチキャストポーザパケットを含む) の転送を制御します。

00 : MAC は、すべての制御パケットがアプリケーションに届くのをフィルタします。

01 : MAC は、アドレスフィルタに失敗した場合もポーザパケットを除いたすべての制御パケットをアプリケーションへ転送します。

10 : MAC は、アドレスフィルタに失敗した場合もすべての制御パケットをアプリケーションへ転送します。

11 : MAC は、アドレスフィルタを通過する制御パケットを転送します。

ビット 5 **DBF** : ブロードキャストパケットの無効化

このビットがセットされた場合、AFM モジュールは、着信するすべてのブロードキャストパケットをブロックします。さらに、その他すべてのフィルタ設定を上書きします。

このビットがリセットされた場合、AFM モジュールは、受信したすべてのブロードキャストパケットを通過させます。

ビット 4 **PM** : すべてのマルチキャストを通過

このビットがセットされた場合、マルチキャストデスティネーションアドレス (デスティネーションアドレスフィールドの最初のビットが '1') を持つすべての受信パケットが通過されることを示します。このビットがリセットされた場合、マルチキャストパケットのフィルタは、HMC ビットに依存します。

ビット 3 **DAIF** : DA 逆フィルタ

このビットがセットされた場合、アドレスチェックブロックは、ユニキャストおよびマルチキャストパケットの両方について、DA アドレス比較の際に逆フィルタモードで動作します。このビットがリセットされた場合、通常のパケットフィルタが行われます。

ビット 2 HMC : ハッシュマルチキャスト

このビットがセットされた場合、MAC はハッシュテーブルに従って、受信したマルチキャストパケットのデスティネーションアドレスフィルタを実行します。

このビットがリセットされた場合、MAC はマルチキャストパケットについて完全デスティネーションアドレスフィルタリングを実行します。すなわち、DA フィールドを DA レジスタでプログラムされた値と比較します。

ビット 1 HUC : ハッシュユニキャスト

このビットがセットされた場合、MAC はハッシュテーブルに従って、ユニキャストパケットのデスティネーションアドレスフィルタを実行します。

このビットがリセットされた場合、MAC はユニキャストパケットについて完全デスティネーションアドレスフィルタリングを実行します。すなわち、DA フィールドを DA レジスタでプログラムされた値と比較します。

ビット 0 PR : プロミスキューモード

このビットがセットされると、アドレスフィルタモジュールは、デスティネーションまたはソースアドレスに関係なく、着信するすべてのパケットを通過させます。PR がセットされると、Rx ステータスワードの SA または DA フィルタ失敗ステータスビットが常にクリアされます。

ウォッチドッグタイムアウトレジスタ (ETH_MACWTR)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

ウォッチドッグタイムアウトレジスタは、受信パケットのウォッチドッグタイムアウトを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWE	Res.	Res.	Res.	Res.	WTO[3:0]			
							rw					rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 PWE : プログラム可能なウォッチドッグの有効化

このビットがセットされ、[動作モード設定レジスタ \(ETH_MACCR\)](#) レジスタの WD ビットがリセットされた場合、WTO フィールドは受信パケットのウォッチドッグタイムアウトとして使用されます。このビットがクリアされた場合、受信パケットのウォッチドッグタイムアウトは、[動作モード設定レジスタ \(ETH_MACCR\)](#) レジスタの WD および JE ビットの設定で制御されます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **WTO[3:0]** : ウォッチドッグタイムアウト

PWE ビットがセットされ、**動作モード設定レジスタ (ETH_MACCCR)** レジスタの WD ビットがリセットされた場合、このフィールドは受信パケットのウォッチドッグタイムアウトとして使用されます。受信パケットの長さがこのフィールドの値を超える場合、このパケットは終了し、エラーパケットと宣言されます。

エンコーディングは、次のとおりです。

0x0 : 2 KB

0x1 : 3 KB

0x2 : 4 KB

0x3 : 5 KB

..

0xC : 14 KB

0xD : 15 KB

0xE : 16383 バイト

0xF : 予約済み。使用できません。

注 : PWE ビットがセットされた場合、このフィールドの値は 1,522 (0x05F2) 以上の必要があります。そうしないと、IEEE 802.3 指定の有効なタグ付きパケットが、エラーパケットと宣言されてドロップされます。

ハッシュテーブル 0 レジスタ (ETH_MACHT0R)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

ハッシュテーブルレジスタ 0 は、ハッシュテーブルの最初の下位 32 ビット (64 ビット) を含みます。

ハッシュテーブルは、グループアドレスフィルタリングに使用されます。

ハッシュフィルタの場合、着信パケットのデスティネーションアドレスの内容は、CRC ロジックを通して渡され、CRC レジスタの上位 6 ビットがハッシュテーブルの内容のインデックスに使用されます。最上位ビットは、使用されるレジスタ (ハッシュテーブルレジスタ 0 または 1) を決め、下位 5 ビットはレジスタ内のビットを決めます。たとえば、0b100000 のハッシュ値は、ハッシュテーブルレジスタ 1 のビット 0 を選択します。

デスティネーションアドレスのハッシュ値は、次の方法で計算されます。

1. DA の 32 ビット CRC を計算します (CRC32 の計算手順については IEEE 802.3、セクション 3.2.8 を参照)。
2. 手順 1 で得られた値のビットの反転を実行します。
3. 手順 2 で得られた値の上位 7 または 8 ビットを取得します。

レジスタの対応するビット値が 1 の場合、パケットは受け入れられます。そうでない場合は、拒否されます。ETH_MACFFR で PM ビットがセットされた場合、マルチキャストハッシュ値に関係なく、すべてのマルチキャストパケットが受け入れられます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT31T0 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT31T0 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **HT31T0[31:0]** : MAC ハッシュテーブルの最初の 32 ビット

このフィールドは、ハッシュテーブルの最初の 32 ビット [31:0] を含みます。

ハッシュテーブル 1 レジスタ (ETH_MACHT1R)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

ハッシュテーブル 1 レジスタは、ハッシュテーブル (64 ビット) の上位 32 ビットを含みます。

ハッシュテーブルは、グループアドレスフィルタリングに使用されます。

ハッシュフィルタの場合、着信パケットのデスティネーションアドレスの内容は、CRC ロジックを通して渡され、CRC レジスタの上位 6 ビットがハッシュテーブルの内容のインデックスに使用されます。最上位ビットは、使用されるレジスタ (ハッシュテーブルレジスタ 0 または 1) を決め、下位 5 ビットはレジスタ内のビットを決めます。たとえば、6'b100000 のハッシュ値は、ハッシュテーブルレジスタ 1 のビット 0 を選択します。

デスティネーションアドレスのハッシュ値は、次の方法で計算されます。

1. DA の 32 ビット CRC を計算します (CRC32 の計算手順については IEEE 802.3、セクション 3.2.8 を参照)。
2. 手順 1 で得られた値のビットの反転を実行します。
3. 手順 2 で得られた値の上位 7 または 8 ビットを取得します。

レジスタの対応するビット値が 1 の場合、パケットは受け入れられます。そうでない場合は、拒否されます。ETH_MACFFR で PM ビットがセットされた場合、マルチキャストハッシュ値に関係なく、すべてのマルチキャストパケットが受け入れられます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT63T32 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT63T32 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **HT63T32[31:0]** : MAC ハッシュテーブルの 2 番目の 32 ビット

このフィールドは、ハッシュテーブルの 2 番目の 32 ビット [63:32] を含みます。

VLAN タグレジスタ (ETH_MACVTR)

アドレスオフセット : 0x0050

リセット値 : 0x0000 0000

VLAN タグレジスタは、IEEE 802.1Q VLAN タイプの packets を特定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIVLRXS	Res.	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EIVLRXS	Res.	EIVLS[1:0]		DOVLT	ERSVLM	ESVL	VTHM	ETV
rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VL[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **EIVLRXS** : Rx ステータスでの内部 VLAN タグの有効化

このビットがセットされた場合、MAC は Rx ステータスで内部 VLAN タグを提供します。このビットがリセットされた場合、MAC は Rx ステータスで内部 VLAN タグを提供しません。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **EIVLS[1:0]** : 受信時の内部 VLAN タグ削除の有効化

このフィールドは、受信したパケットの内部 VLAN タグでの削除操作を示します。

00 : 削除しない

01 : VLAN フィルタを通過した場合に削除

10 : VLAN フィルタに失敗した場合に削除

11 : 常に削除

ビット 27 **ERIVLT** : 内部 VLAN タグの有効化

このビットおよび EDVLP フィールドがセットされた場合、MAC レシーバは内部 VLAN タグ（ある場合）の操作を有効にします。このビットリセットされた場合、MAC レシーバは外部 VLAN タグ（ある場合）の操作を有効にします。ERSVLM ビットは、フィルタや一致に対して有効にする VLAN タイプを決定します。ERSVLM ビットと DOVLT ビットは、フィルタに対して有効にする VLAN タイプを決定します。

ビット 26 **EDVLP** : ダブル VLAN 処理の有効化

このビットがセットされた場合、MAC は Tx および Rx で最大 2 つの VLAN タグ（ある場合）の処理を有効にします。このビットがリセットされた場合、MAC は Tx および Rx で最大 1 つの VLAN タグ（ある場合）の処理を有効にします。

ビット 25 **VTHM** : VLAN タグハッシュテーブル一致の有効化

このビットがセットされた場合、VLAN タグの CRC の最上位 4 ビットを使用して、ETH_MACVLANHTR レジスタの内容をインデックスします。インデックスに対応する VLAN ハッシュテーブルレジスタでの値 1 は、パケットが VLAN ハッシュテーブルに一致したことを示します。

ETV ビットがセットされた場合、12 ビット VLAN 識別子 (VID) の CRC が比較に使用されます。ETV ビットがリセットされた場合、16 ビット VLAN タグの CRC が比較に使用されます。

このビットがリセットされた場合、VLAN ハッシュ一致操作は実行されません。

ビット 24 **EIVLRXS** : Rx ステータスでの VLAN タグの有効化

このビットがセットされた場合、MAC は Rx ステータスで外部 VLAN タグを提供します。このビットがリセットされた場合、MAC は Rx ステータスで外部 VLAN タグを提供しません。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **EVLS[1:0]** : 受信時の VLAN タグ削除の有効化

このフィールドは、受信したパケットの外部 VLAN タグでの削除操作を示します。

00 : 削除しない

01 : VLAN フィルタを通過した場合に削除

10 : VLAN フィルタに失敗した場合に削除

11 : 常に削除

ビット 20 **DOVLTC** : VLAN タイプチェックの無効化

このビットがセットされた場合、MAC は ERIVLT ビットによって指定された VLAN タグが S-VLAN または C-VLAN のどのタイプであるかをチェックしません。

このビットがリセットされた場合、VLAN タグタイプが ERSVLM ビットで指定されたものと同様であるときのみ、MAC は ERIVLT ビットで指定された VLAN タグのフィルタまたは一致を実行します。

ビット 19 **ERSVLM** : 受信 S-VLAN 一致の有効化

このビットがセットされた場合、MAC レシーバは S-VLAN (タイプ = 0x88A8) のフィルタまたは一致を有効にします。このビットがリセットされた場合、MAC レシーバは C-VLAN (タイプ = 0x8100) のフィルタまたは一致を有効にします。

ERIVLT ビットは、フィルタや一致に対して考慮される VLAN タグの位置を決定します。

ビット 18 **ESVL** : S-VLAN の有効化

このビットがセットされた場合、MAC トランスミッタおよびレシーバは S-VLAN パケット (タイプ = 0x88A8) を、有効な VLAN タグ付きパケットとみなします。

ビット 17 **VTIM** : VLAN タグ反転一致の有効化

このビットがセットされた場合、VLAN タグの反転一致が有効になります。一致する VLAN タグがないパケットが、一致とマークされます。リセットされた場合、このビットは VLAN タグの完全一致が有効になります。一致する VLAN タグがあるパケットが、一致とマークされます。

ビット 16 **ETV** : 12 ビット VLAN タグ比較の有効化

このビットがセットされた場合、完全な 16 ビット VLAN タグの代わりに 12 ビット VLAN 識別子が比較およびフィルタに使用されます。VLAN タグのビット [11:0] が、受信した VLAN タグ付きパケット内の対応するフィールドと比較されます。同様に、有効化された場合、受信パケットの VLAN タグの 12 ビットだけがハッシュベース VLAN フィルタに使用されます。

このビットがリセットされた場合、受信した VLAN パケットの 15 番目と 16 番目のバイトの全 16 ビットが比較および VLAN ハッシュフィルタに使用されます。

ビット 15:0 **VL[15:0]** : 受信パケット用 VLAN タグ識別子

このフィールドは、VLAN パケットを識別する 802.1Q VLAN タグを含んでいます。この VLAN タグ識別子は、VLAN パケットのために受信されるパケットの 15 番目および 16 番目のバイトと比較されます。次のリストでは、このフィールドのビットを示しています。

ビット [15:13] : ユーザ優先度

ビット 12 : 正規フォーマットインジケータ (CFI) またはドロップ適格インジケータ (DEI)

ビット [11:0] : VLAN タグの VLAN 識別子 (VID) フィールド

ETV ビットがセットされると、VID のみが比較に使用されます。

このフィールド (ETV がセットされている場合、[11:0]) がすべて 0 の場合、MAC は、VLAN タグの比較のために 15 番目および 16 番目のバイトをチェックせず、タイプフィールドの値が 0x8100 または 0x88a8 のすべてのパケットを VLAN パケットとして宣言します。

VLAN ハッシュテーブルレジスタ (ETH_MACVHTR)

アドレスオフセット : 0x0058

リセット値 : 0x0000 0000

VLAN タグレジスタ (ETH_MACVTR) レジスタの VTHM ビットがセットされた場合、16 ビット VLAN ハッシュテーブルレジスタが VLAN タグに基づいたグループアドレスフィルタに使用されます。ハッシュフィルタでは、着信パケットの 16 ビット VLAN タグまたは 12 ビット VLAN ID の内容 (**VLAN タグレジスタ (ETH_MACVTR)** レジスタの ETV ビットに基づく) は、CRC ロジックを通じて渡されます。計算された CRC の上位 4 ビットを使用して、VLAN ハッシュテーブルの内容をインデックスします。たとえば、1000 のハッシュ値は、VLAN ハッシュテーブルのビット 8 を選択します。

デスティネーションアドレスのハッシュ値は、次の方法で計算されます。

1. VLAN タグまたは ID の 32 ビット CRC を計算します (CRC32 を計算する手順については、IEEE 802.3 のセクション 3.2.8 を参照してください)。
2. 手順 1 で得られた値のビットの反転を実行します。
3. 手順 2 で得られた値の上位 4 ビットを取得します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLHT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **VLHT[15:0]** : VLAN ハッシュテーブル

このフィールドは、16 ビット VLAN ハッシュテーブルを含みます。

VLAN 内包レジスタ (ETH_MACVIR)

アドレスオフセット : 0x0060

リセット値 : 0x0000 0000

VLAN タグ内包または書き換えレジスタには、送信パケットで挿入または書き換えを行うための VLAN タグが含まれます。これには、VLAN タグ挿入制御も含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLT	CSVL	VLP	VLC[1:0]	
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **VLT** : VLAN タグ入力

このビットがセットされた場合、Tx パケットで挿入または書き換える VLAN タグを、Tx ディスクリプタから取得する必要があることを示します。

ビット 19 **CSVL** : C-VLAN または S-VLAN

このビットがセットされた場合、S-VLAN タイプ (0x88A8) が送信パケットの 13 番目および 14 番目のバイトで挿入または書き換えられます。このビットがリセットされた場合、C-VLAN タイプ (0x8100) が送信パケットの 13 番目および 14 番目のバイトで挿入または書き換えられます。

0 : C-LAN

1 : S-LAN

ビット 18 **VLP** : VLAN 優先順位の制御

このビットがセットされた場合、制御ビット [17:16] が VLAN の削除、挿入、書き換えに使用されます。このビットがリセットされた場合、ビット [17:16] は無視されます。

ビット 17:16 **VLC[1:0]** : 送信パケットの VLAN タグ制御

00 : VLAN タグの削除、挿入、書き換えはありません。

01 : VLAN タグを削除します。MAC が VLAN タグ付きで送信されたパケットすべての VLAN タイプ (バイト 13 および 14) と VLAN タグ (バイト 15 および 16) を削除します。

10 : VLAN タグを挿入します。MAC がタイプ値 (0x8100 または 0x88a8) をバイト 13 および 14 に挿入した後に、VLT をパケットのバイト 15 および 16 に挿入します。この操作は、VLAN タグがすでにあるかどうかに関係なく、すべての送信パケットで実行されます。

11 : VLAN タグを書き換えます。MAC がすべての VLAN タイプ送信パケットのバイト 15 および 16 の VLT を書き換えます (バイト 13 および 14 は 0x8100 または 0x88a8 です)。

注 : このフィールドへの変更は、パケットの開始時にのみ有効になります。パケットが送信されているときに、このレジスタフィールドに書き込む場合、後続のパケットのみが更新された値を使用でき、現在のパケットは更新された値を使用しません。

ビット 15:0 **VLT[15:0]** : 送信パケット用 VLAN タグ

このフィールドは、挿入または書き換える VLAN タグの値を含んでいます。この値は、送信ラインが非アクティブである場合や初期化フェーズ中のみ変更する必要があります。

次のリストでは、このフィールドのビットを示しています。

ビット [15:13] : ユーザ優先度

ビット 12 : 正規フォーマットインジケータ (CFI) またはドロップ適格インジケータ (DEI)

ビット [11:0] : VLAN タグの VLAN 識別子 (VID) フィールド

内部 VLAN 内包レジスタ (ETH_MACIVIR)

アドレスオフセット : 0x0064

リセット値 : 0x0000 0000

内部 VLAN タグ内包または書き換えレジスタには、送信パケットで挿入または書き換えを行うための内部 VLAN タグが含まれます。これには、内部 VLAN タグ挿入制御も含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **VLTI** : VLAN タグ入力

このビットがセットされた場合、Tx パケットで挿入または書き換える VLAN タグを、Tx ディスクリプタから取得する必要があることを示します。

ビット 19 **CSVL** : C-VLAN または S-VLAN

このビットがセットされた場合、S-VLAN タイプ (0x88A8) が送信パケットの 13 番目および 14 番目のバイトで挿入または書き換えられます。このビットがリセットされた場合、C-VLAN タイプ (0x8100) が送信パケットの 13 番目および 14 番目のバイトで挿入または書き換えられます。

0 : C-LAN

1 : S-LAN

ビット 18 **VLP** : VLAN 優先順位の制御

このビットがセットされた場合、VLC フィールドが VLAN の削除、挿入、書き換えに使用されます。このビットがリセットされた場合、VLC フィールドは無視されます。

ビット 17:16 **VLC[1:0]** : 送信パケットの VLAN タグ制御

00 : VLAN タグの削除、挿入、書き換えはありません。

01 : VLAN タグを削除します。

MAC が VLAN タグ付きで送信されたパケットすべての VLAN タイプ (バイト 17 および 18) と VLAN タグ (バイト 19 および 20) を削除します。

10 : VLAN タグを挿入します。

MAC がタイプ値 (0x8100 または 0x88a8) をバイト 17 および 18 に挿入した後に、VLT をパケットのバイト 19 および 20 に挿入します。この操作は、VLAN タグがすでにあるかどうかに関係なく、すべての送信パケットで実行されます。

11 : VLAN タグを書き換えます。

MAC がすべての VLAN タイプ送信パケットのバイト 19 および 20 の VLT を書き換えます (バイト 17 および 18 は 0x8100 または 0x88a8 です)。

注 : このフィールドへの変更は、パケットの開始時にのみ有効になります。パケットが送信されているときに、このレジスタフィールドに書き込む場合、後続のパケットのみが更新された値を使用でき、現在のパケットは更新された値を使用しません。

ビット 15:0 **VLT[15:0]** : 送信パケット用 VLAN タグ

このフィールドは、挿入または書き換える VLAN タグの値を含んでいます。この値は、送信ラインが非アクティブである場合や初期化フェーズ中のみ変更する必要があります。

次のリストでは、このフィールドのビットを示しています。

ビット [15:13] : ユーザ優先度

ビット 12 : 正規フォーマットインジケータ (CFI) またはドロップ適格インジケータ (DEI)

ビット [11:0] : VLAN タグの VLAN 識別子 (VID) フィールド

Tx キューフロー制御レジスタ (ETH_MACQTXFCR)

アドレスオフセット : 0x0070

リセット値 : 0x0000 0000

フロー制御レジスタは、MAC のフロー制御モジュールによる制御パケット（ポーズコマンド）の生成と受信を制御します。ビジービットが 1 にセットされているレジスタに書き込むと、フロー制御ブロックがトリガされ、ポーズパケットを生成します。制御パケットのフィールドは、802.3x 仕様での規定に従って選択され、このレジスタのポーズ時間の値が制御パケットのポーズ時間フィールドで使用されます。ビジービットは、制御パケットがケーブルに転送されるまでセット状態を維持します。アプリケーションは、レジスタに書き込む前に、ビジービットがクリアされていることを確認する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]				Res.	Res.	TFE
								rw	rw	rw	rw			rw	rw

ビット 31:16 PT[15:0] : ポーズ時間

このフィールドは、Tx 制御パケットのポーズ時間フィールドで使用される値を保持します。I

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 DZPQ : ゼロクオンタムポーズの無効化

このビットがセットされた場合、ゼロクオンタムポーズパケットの自動生成が無効になります。
このビットがリセットされた場合、ゼロクオンタムポーズパケットの自動生成を伴う通常動作が有効になります。

ビット 6:4 PLT[2:0] : ポーズロー閾値

このフィールドは、入力フローがポーズパケットの自動再送信についてチェックされるポーズタイマの閾値を設定します。

この閾値は、常にビット [31:16] で設定されたポーズ時間未満でなければなりません。たとえば、PT = 100H (256 スロット時間) であり、PLT = 001 の場合、最初のポーズパケットの送信後、228 (256-28) スロット時間の時点で 2 番目のポーズパケットが自動的に送信されます。

次のリストでは異なる値での閾値を示しています。

000 : ポーズ時間 - 4 スロット時間 (PT - 4 スロット時間)

001 : ポーズ時間 - 28 スロット時間 (PT - 28 スロット時間)

010 : ポーズ時間 - 36 スロット時間 (PT - 36 スロット時間)

011 : ポーズ時間 - 144 スロット時間 (PT - 144 スロット時間)

100 : ポーズ時間 - 256 スロット時間 (PT - 256 スロット時間)

101 : ポーズ時間 - 512 スロット時間 (PT - 512 スロット時間)

110 から 111 : 予約済み。使用できません。

スロット時間は、MII インタフェース上で 512 ビット (64 バイト) を送信するための所要時間として定義されます。

この (概算の) 計算は、パケットサイズ (64、1518、2000、9018、16384、または 32768) + 2 ポーズパケットサイズ + スロット時間の IPG です。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TFE** : 送信フロー制御の有効化

全二重モード : このビットがセットされた場合、MAC は、Tx ポーズパケットに対するフロー制御動作を有効にします。このビットがリセットされた場合、MAC のフロー制御動作は無効になり、MAC はポーズパケットを送信しません。

半二重モード : このビットがセットされた場合、MAC はバックプレッシャ機能を有効にします。このビットがリセットされた場合、バックプレッシャ機能は無効になります。

ビット 0 **FCB_BPA** : フロー制御ビジーまたはバックプレッシャ有効

このビットは、全二重モードで、ポーズパケットを開始し、半二重モードでは、TFE ビットがセットされている場合、バックプレッシャ機能を有効にします。

全二重モード : このレジスタに書き込む前に、このビットが 0 として読み出されなければなりません。ポーズパケットを開始するには、アプリケーションはこのビットを 1 にセットする必要があります。制御パケットの転送時、このビットは、パケット送信が進行中であることを示すために、セットされたままです。ポーズパケットの送信が完了すると、MAC はこのビットを 0 にリセットします。このビットがクリアされるまで、このレジスタへの書き込みが禁止されます。

半二重モード : このビットが半二重モードでセットされた場合（および TFE ビットがセットされた場合）、MAC はバックプレッシャをアサートします。バックプレッシャ時、MAC が新しいパケットを受信すると、トランスミッタは、JAM パターンの送出を開始し、結果として衝突が発生します。MAC が全二重モードに設定されると、BPA は自動的に無効になります。

Rx フロー制御レジスタ (ETH_MACRXFCR)

アドレスオフセット : 0x0090

リセット値 : 0x0000 0000

受信フロー制御レジスタは、受信したポーズパケットに基づいて MAC 送信のポーズを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UP	RFE
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **UP** : ユニキャストポーズフレーム検出

ポーズパケットは、IEEE 802.3 で指定されている一意なマルチキャストアドレスを持つ場合に処理されます。このビットがセットされた場合、MAC はステーションのユニキャストアドレスを持つポーズパケットも検出できます。このユニキャストアドレスは **MACアドレス 0 上位レジスタ (ETH_MACA0HR)** と MAC アドレス 0 下位レジスタ **MACアドレス x 下位レジスタ (ETH_MACAxLR)** での指定に従う必要があります。

このビットがリセットされた場合、MAC は一意なマルチキャストアドレスを持つポーズフレームのみを検出します。

注 : マルチキャストアドレスが一意なマルチキャストアドレスではない場合、MAC はポーズパケットを処理しません。これは、優先フロー制御 (PFC) が有効になっている場合、受信した PFC パケットにも適用できます。一意なマルチキャストアドレス (0x01_80_C2_00_00_01) は、IEEE 802.1 Qbb-2011 での指定に従います。

ビット 0 **RFE** : 受信フロー制御有効化

このビットがセットされ、MAC が全二重モードで動作している場合、受信したポーズパケットをデコードして、指定された時間 (ポーズ時間) の間、トランスミッタを無効にします。このビットがリセットされた場合や MAC が半二重モードで動作している場合、ポーズパケットのデコード機能は無効です。PFC が有効になっている場合、フロー制御は PFC パケットに対して有効です。MAC は、受信した PFC パケットをデコードして、受信したポーズ時間の間、一致優先度で送信キューを無効にします。

割込みステータスレジスタ (ETH_MACISR)

アドレスオフセット : 0x00B0

リセット値 : 0x0000 0000

割込みステータスレジスタには、割込みのステータスが含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RXSTIS	TXSTIS	TSIS	Res.	MMCTXIS	MMCRXIS	MMCIS	Res.	Res.	LPIS	PMTIS	PHYIS	Res.	Res.	Res.
	rc_r	rc_r	rc_r		r	r	r			r	r	r			

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 RXSTIS : 受信ステータス割込み

このビットは、受信パケットのステータスを示します。このビットは、[Rx Tx ステータスレジスタ \(ETH_MACRXTXSR\)](#) で RWT ビットがセットされた場合にセットされます。このビットは、対応する割込みソースビットが ETH_MACISR レジスタで読み出される（または [CSR ソフトウェア制御レジスタ \(ETH_MACCSRWC\)](#) の RCWE ビットがセットされた状態で、対応する割込みソースビットが 1 に書き込まれる）ときにクリアされます。

ビット 13 TXSTIS : 送信ステータス割込み

このビットは、送信パケットのステータスを示します。このビットは、[Rx Tx ステータスレジスタ \(ETH_MACRXTXSR\)](#) で次のいずれかのビットがセットされた場合にセットされます。

- 過剰衝突 (EXCOL)
- 遅れ衝突 (LCOL)
- 過剰遅延 (EXDEF)
- キャリア喪失 (LCARR)
- キャリアなし (NCARR)
- ジャバertimeアウト (TJT)

このビットは、対応する割込みソースビットが ETH_MACISR レジスタで読み出される（または [CSR ソフトウェア制御レジスタ \(ETH_MACCSRWC\)](#) の RCWE ビットがセットされた状態で、対応する割込みソースビットが 1 に書き込まれる）ときにクリアされます。

ビット 12 **TSIS** : タイムスタンプ割込みステータス

タイムスタンプ機能が有効になっている場合、以下の条件のいずれかが当てはまる場合に、このビットがセットされます。

- システム時間の値が、目標時間上位および下位レジスタで指定された値以上になっています。
- 秒レジスタにオーバーフローがあります。
- 目標時間エラーが発生しており、プログラムされた目標時間をすでに過ぎています。

補助スナップショット機能が有効になっている場合、補助スナップショットトリガがアサートされた場合に、このビットがセットされます。

ドロップ送信ステータスが MTL で有効になっている場合、キャプチャされた送信タイムスタンプが **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** および **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** レジスタで更新されると、このビットがセットされます。

PTP オフロード機能が有効になっている場合、キャプチャされた送信タイムスタンプが **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** および **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** レジスタで更新されたとき、PTO 生成遅延リクエストおよび Pdelay リクエストパケットに対して、このビットがセットされます。

このビットは、対応する割込みソースビットが **タイムスタンプステータスレジスタ (ETH_MACTSSR)** で読み出される（または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** の RCWE ビットがセットされた状態で、対応する割込みソースビットが 1 に書き込まれる）ときにクリアされます。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **MMCTXIS** : MMC 送信割込みステータス

このビットは、**MMC Tx 割込みレジスタ (ETH_MMC_TX_INTERRUPT)** で割込みが生成されると、ハイにセットされます。このビットは、この割込みレジスタのすべてのビットがクリアされたときにクリアされます。

ビット 9 **MMCRXIS** : MMC 受信割込みステータス

このビットは、**MMC Rx 割込みレジスタ (ETH_MMC_RX_INTERRUPT)** で割込みが生成されると、ハイにセットされます。このビットは、この割込みレジスタのすべてのビットがクリアされたときにクリアされます。

ビット 8 **MMCIS** : MMC 割込みステータス

このビットは、MMCTXIS または MMCRXIS がハイにセットされたときに、ハイにセットされます。このビットは、それらのビットすべてがローのときのみクリアされます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **LPIIS** : LPI 割込みステータス

このビットは、MAC トランスミッタまたはレシーバでの LPI 状態への移行または終了に対してセットされます。**LPI 制御およびステータスレジスタ (ETH_MACLCSR)** の TLPIEN ビットが読み出されると、このビットはクリアされます。

ビット 4 **PMTIS** : PMT 割込みステータス

このビットは、パワーダウンモードで、マジックパケットまたは Wake-on-LAN パケットが受信されたときに、セットされます (**ETH_MACPCSR** レジスタの RWKPRCVD および MGKPRCVD ビット)。このビットは、ビット [6:5] が **PMT 制御ステータスレジスタ (ETH_MACPCSR)** の読出し動作のためにクリアされたときにクリアされます。

ビット 3 **PHYIS** : PHY 割込み

このビットは、立上りエッジが **ETH_PHY_INTN** 入力で検出されたときにセットされます。このレジスタが読み出されると、このビットはクリアされます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

割込み有効レジスタ (ETH_MACIER)

アドレスオフセット : 0x00B4

リセット値 : 0x0000 0000

割込み有効レジスタには、割込みを生成するためのマスクが含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RXSTSIE	TXSTSIE	TSIE	Res.	Res.	Res.	Res.	Res.	Res.	LPIIE	PMTIE	PHYIE	Res.	Res.	Res.
	rw	rw	rw							rw	rw	rw			

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **RXSTSIE** : 受信ステータス割込み有効化このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の RXSTSIS ビットの設定のため、割込み信号のアサートが有効になります。ビット 13 **TXSTSIE** : 送信ステータス割込み有効化このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の TXSTSIS ビットの設定のため、割込み信号のアサートが有効になります。ビット 12 **TSIE** : タイムスタンプ割込み有効化このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の TSIS ビットの設定のため、割込み信号のアサートが有効になります。

ビット 11:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **LPIIE** : LPI 割込み有効このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の LPIIS ビットの設定のため、割込み信号のアサートが有効になります。ビット 4 **PMTIE** : PMT 割込み有効このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の PMTIS ビットの設定のため、割込み信号のアサートが有効になります。ビット 3 **PHYIE** : PHY 割込み有効このビットがセットされた場合、[割込みステータスレジスタ \(ETH_MACISR\)](#) の PHYIS ビットの設定のため、割込み信号のアサートが有効になります。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

Rx Tx ステータスレジスタ (ETH_MACRXTXSR)

アドレスオフセット : 0x00B8

リセット値 : 0x0000 0000

送受信ステータスレジスタには、送受信のエラーステータスが含まれます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
							rc_r			rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **RWT** : 受信ウォッチドッグタイムアウト

このビットは、2,048 バイト (ジャンボパケットモードが有効の場合は 10,240 バイト) より長いパケットが受信され、**動作モード設定レジスタ (ETH_MACCCR)** で WD ビットがリセットされたときにセットされます。このビットは、16,383 バイトより長いパケットが受信され、**動作モード設定レジスタ (ETH_MACCCR)** で WD ビットがセットされたときにセットされます。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **EXCOL** : 過剰衝突

動作モードレジスタ (ETH_MTL0MR) で DTXSTS ビットがセットされた場合、このビットは、現在のパケットの送信を試みている間に 16 の連続した衝突が発生した後、送信が中止されたことを示します。**動作モード設定レジスタ (ETH_MACCCR)** で DR ビットがセットされている場合、このビットが最初の衝突後にセットされて、パケットの送信が中止されます。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 4 **LCOL** : 遅れ衝突

動作モードレジスタ (ETH_MTL0MR) で DTXSTS ビットがセットされた場合、このビットは、MII モードで衝突ウィンドウ (プリアンプルを含めて 64 バイト) の後で衝突が発生したためにパケット送信が中止されたことを示します。

このビットは、アンダーフローエラーが発生した場合は無効です。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 3 **EXDEF** : 過剰遅延

動作モードレジスタ (ETH_MTL0MR) で DTXSTS ビットがセットされ、**動作モード設定レジスタ (ETH_MACCCR)** で DC ビットがセットされた場合、このビットは、24,288 ビット時間 (ジャンボパケットが有効になっている場合は 155,680) を超える過剰遅延があったために送信が終了したことを示します。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 2 **LCARR** : キャリア喪失

動作モードレジスタ (ETH_MTLOMR) で DTXSTS ビットがセットされた場合、このビットは、パケット送信中にキャリアの喪失が発生した、すなわち、パケット送信中に 1 つ以上の送信クロック周期の間、ETH_CRS 信号が無効であったことを示します。このビットは、衝突なしで送信されたパケットにのみ有効です。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 1 **NCARR** : キャリアなし

動作モードレジスタ (ETH_MTLOMR) で DTXSTS ビットがセットされた場合、このビットは、PHY からのキャリア信号がプリアンプル送信の最後に存在しないことを示します。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

ビット 0 **TJT** : 送信ジャバertimeアウト

このビットは、パケットサイズが 2,048 バイト (ジャンボパケットモードが有効の場合は 10,240 バイト) を超えるときに発生する送信ジャバertimeが終了して、**動作モード設定レジスタ (ETH_MACCR)** で JD ビットがリセットされたことを示します。このビットは、パケットサイズが 16,383 バイトを超え、**動作モード設定レジスタ (ETH_MACCR)** で JD ビットがセットされたときにセットされます。

読出し (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされた場合は 1 の書き込み) 時にクリアされます。

PMT 制御ステータスレジスタ (ETH_MACPCSR)

アドレスオフセット : 0x00C0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWKFILTRST	Res.	Res.	RWKPTR[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw			r	r	r	r	r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RWKPF	GLBLUCAST	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN
					rw	rw			r	rc_r			rw	rw	rw

ビット 31 **RWKFILTRST** : リモートウェイクアップパケットフィルタレジスタのポインタのリセット

このビットがセットされた場合、リモートウェイクアップパケットフィルタレジスタのポインタを 0 にリセットします。1 クロックサイクル後に自動的にクリアされます。

ビット 30:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **RWKPTR[4:0]** : リモートウェイクアップ FIFO ポインタ

このフィールドは、リモートウェイクアップパケットフィルタレジスタのポインタの現在の値 (0 ~ 7) を提供します。このポインタが 7 である場合、リモートウェイクアップパケットフィルタレジスタへの書き込みが発生すると、その内容が eth_mii_rx_clk ドメインへ転送されます。

ビット 23:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 RWKPFPE : リモートウェイクアップパケット転送の有効化

このビットが RWKPKTEN とともにセットされた場合、MAC レシーバは期待されるウェイクアップフレームを受信するまで、すべての受信フレームをドロップします。受信したウェイクアップフレームを含むそのイベント以降のすべてのフレームはアプリケーションに転送されます。このビットは、ウェイクアップパケットの受信時にセルフクリアされます。

アプリケーションで、期待されるウェイクアップフレームが受信される前でも、このビットをクリアできます。このような場合、MAC は受信パケットがアプリケーションに転送されるデフォルトの動作に戻ります。このビットは、RWKPKTEN がハイに、PWRDWN がローにセットされているときのみ、セットされなければなりません。このビットの設定は、PWRDWN がハイにセットされているときには効果がありません。

注 : マジックパケットの有効化およびウェイクアップフレームの有効化が、このビットの設定に合わせてセットされ、マジックパケットがウェイクアップフレームより先に受信される場合、このビットはマジックパケットの受信時にセルフクリアされ、受信したマジックパケットはドロップされ、受信したマジックパケット以降のすべてのフレームは、アプリケーションに転送されます。

ビット 9 GLBLUCAST : グローバルユニキャスト

このビットがセットされた場合、MAC (DAF) アドレス認識によってフィルタされたすべてのユニキャストパケットは、リモートウェイクアップパケットとして検出されます。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 RWKPRCVD : リモートウェイクアップパケット受信

このビットがセットされた場合、リモートウェイクアップパケットを受信したため、電源管理イベントが生成されることを示します。このレジスタが読み出されると、このビットはクリアされます。

ビット 5 MGKPRCVD : マジックパケットの受信

このビットがセットされた場合、マジックパケットを受信したため、電源管理イベントが生成されることを示します。このビットは、このレジスタが読み出される（または [CSR ソフトウェア制御レジスタ \(ETH_MACCSRSWCR\)](#) で RCWE ビットがセットされた状態で、1 に書き込まれる）ときにクリアされます。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 RWKPKTEN : リモートウェイクアップパケットの有効化

このビットがセットされた場合、MAC がリモートウェイクアップパケットを受信したときに、電源管理イベントが生成されます。

ビット 1 MGKPKTEN : マジックパケットの有効化

このビットがセットされた場合、MAC がマジックパケットを受信したときに、電源管理イベントが生成されます。

ビット 0 PWRDWN : パワーダウン

このビットがセットされた場合、MAC レシーバは期待されるマジックパケットまたはリモートウェイクアップパケットを受信するまで、すべての受信パケットをドロップします。このビットはセルフクリアされ、パワーダウンモードは無効になります。ソフトウェアで、期待されるマジックパケットまたはリモートウェイクアップパケットが受信される前に、このビットをクリアできます。このビットがクリアされた後に MAC によって受信されたパケットは、アプリケーションに転送されます。このビットは、マジックパケット有効、グローバルユニキャスト、またはリモートウェイクアップパケット有効ビットがハイにセットされているときのみ、セットされなければなりません。

注 : パワーダウンモード中に CSR クロックをゲートオフできます。ただし、CSR クロックがゲートオフされると、このレジスタで読み出しまたは書き込み操作を実行できません。そのため、ソフトウェアはこのビットをクリアできません。

リモートウェイクアップパケットフィルタレジスタ (ETH_MACRWKPFRR)

アドレスオフセット : 0x00C4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACRWKPFRR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACRWKPFRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **MACRWKPFRR[31:0]** : リモートウェイクアップパケットフィルタ

レジスタの内容とプログラミングシーケンスについての詳細は、表 649、表 650、および表 651 を参照してください。

アドレス 0x00C4 の ETH_MACRWKPFRR レジスタは、ウェイクアップパケットフィルタレジスタをロードします。

ウェイクアップパケットフィルタレジスタで値をロードするには、レジスタ全体 (ETH_MACRWKPFRR) が書き込まれる必要があります。ETH_MACRWKPFRR 値 (0~7) それぞれに対して、アドレス (0x00C4) で 8、16、または 32 のレジスタ値を連続でロードすることによって、ETH_MACRWKPFRR レジスタをロードします。ETH_MACRWKPFRR レジスタは、類似した方法で読み出されます。Ethernet ペリフェラルは、ETH_MACPCSR レジスタのビット [26:24] で ETH_MACRWKPFRR レジスタの現在のポインタ値を更新します。

LPI 制御およびステータスレジスタ (ETH_MACLCSR)

アドレスオフセット : 0x00D0

リセット値 : 0x0000 0000

LPI 制御およびステータスレジスタは、LPI 機能を制御し、LPI 割込みステータスを提供します。このレジスタが読み出されると、このステータスビットはクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPITE	LPITXA	Res.	PLS	LPIEN
										rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	RLPIEX	RLPIEN	TLPIEX	TLPIEN
						r	r					r	r	r	r

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **LPITCSE** : LPI Tx クロック停止の有効化

このビットがセットされた場合、MAC は Tx LPI モードに移行した後で sbd_tx_clk_gating_ctrl_o 信号のハイをアサートして、MAC への Tx クロックを停止できることを示します。このビットがリセットされた場合、MAC は Tx LPI モードに移行した後で sbd_tx_clk_gating_ctrl_o 信号のハイをアサートしません。RGMII インタフェースが選択されている場合、LPI パターンを送信するには Tx クロックが必要となります。Tx クロックはゲートできないため、LPITCSE ビットをプログラムすることはできません。

ビット 20 LPITE : LPI タイマ有効化

このビットは、MAC トランスミッタの LPI 状態への自動移行および自動終了を制御します。LPITE、LPITXA および LPIEN ビットがセットされた場合、完全な MAC TX データパスが ETH_MACLETR レジスタで示された期間、アイドル状態であるときのみ、MAC トランスミッタが LPI 状態に移行します。LPI 状態に移行後、データパスが（送信する新しいパケットを受け入れたために）アイドル状態ではなくなった場合、トランスミッタは LPI 状態を終了しますが、LPIEN ビットをクリアしません。これにより、再びアイドル状態になったときに LPI 状態に再移行できます。LPITE が 0 の場合、LPI 自動タイマは無効になり、MAC トランスミッタが LPITXA および LPIEN ビットの説明にある設定に基づいて LPI 状態に移行します。

ビット 19 LPITXA : LPI Tx 自動化

このビットは、送信側で LPI モードの移行または終了を行う場合に MAC の動作を制御します。LPITXA および LPIEN ビットが 1 にセットされた場合、MAC は（コア内の）すべての未処理なパケットと（アプリケーションインタフェースでの）ペンディングパケットが送信された後、LPI モードに移行します。アプリケーションが送信用のパケットを送信する場合や Tx FIFO 一掃コマンドを発行する場合に、MAC は LPI モードを終了します。さらに、MAC は LPI 状態を終了するとき、LPIEN ビットを自動的にクリアします。Tx FIFO 一掃が ETH_MTLTxQOMR の FTQ ビットでセットされた場合、MAC が LPI モードであるときは LPI モードを終了します。このビットが 0 である場合、LPIEN ビットが LPI モードの移行時または終了時の MAC の動作を直接制御します。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 PLS : PHY リンクステータス

このビットは、PHY のリンクステータスを示します。リンクステータスが少なくとも LPI LS TIMER で示された時間、アップ (OKAY) であるときのみ、MAC トランスミッタは LPI パターンをアサートします。このビットがセットされた場合、リンクは問題なし (UP) とみなされ、このビットがリセットされた場合、リンクがダウンしているとみなされます。

ビット 16 LPIEN : LPI 有効化

このビットがセットされた場合、MAC トランスミッタに LPI 状態に移行するよう命令します。このビットがリセットされた場合、MAC に LPI 状態を終了して通常の送信を再開するよう命令します。LPITXA ビットがセットされ、MAC が送信用の新しいパケットを受け取ったために LPI 状態を終了するときに、このビットはクリアされます。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 RLPIST : 受信 LPI の状態

このビットがセットされた場合、MAC が MII インタフェースで LPI パターンを受信していることを示します。

ビット 8 TLPIST : 送信 LPI の状態

このビットがセットされた場合、MAC が MII インタフェースで LPI パターンを送信していることを示します。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 RLPIEX : 受信 LPI の終了

このビットがセットされた場合、MAC レシーバが MII インタフェースで LPI パターンの受信をやめ、LPI 状態を終了し、通常の受信を再開したことを示します。このビットは、このレジスタの読出しによって（または、CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR) で RCWE ビットがセットされている場合は 1 を書き込むことによって）クリアされます。

注 : MAC が、CSR クロックの 3 クロックサイクル未満といったような非常に短い時間、LPI パターンの受信をやめる場合、このビットはセットできません。

ビット 2 **RLPIEN** : 受信 LPI の移行

このビットがセットされた場合、MAC レシーバが LPI パターンを受信して LPI 状態に移行したことを示します。このビットは、このレジスタの読出しによって（または、[CSR ソフトウェア制御レジスタ \(ETH_MACCSRSWCR\)](#) で RCWE ビットがセットされている場合は 1 を書き込むことによって）クリアされます。

注： MAC が、CSR クロックの 3 クロックサイクル未満といったような非常に短い時間、LPI パターンの受信をやめる場合、このビットはセットできません。

ビット 1 **TLPIEX** : 送信 LPI の終了

このビットがセットされた場合、アプリケーションが LPIEN ビットをクリアして LPI TW タイマが終了した後に、MAC トランスミッタが LPI 状態を終了したことを示します。このビットは、このレジスタの読出しによって（または、[CSR ソフトウェア制御レジスタ \(ETH_MACCSRSWCR\)](#) で RCWE ビットがセットされている場合は 1 を書き込むことによって）クリアされます。

ビット 0 **TLPIEN** : 送信 LPI の移行

このビットがセットされた場合、LPIEN ビットの設定のため、MAC トランスミッタが LPI 状態に移行したことを示します。このビットは、このレジスタの読出しによって（または、[CSR ソフトウェア制御レジスタ \(ETH_MACCSRSWCR\)](#) で RCWE ビットがセットされている場合は 1 を書き込むことによって）クリアされます。

LPI タイマ制御レジスタ (ETH_MACLTCR)

アドレスオフセット : 0x00D4

リセット値 : 0x03E8 0000

LPI タイマ制御レジスタは、LPI 状態でのタイムアウト値を制御します。MAC が LPI パターンを送信する時間と、通常の送信を再開するまで MAC が待機する時間も指定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:16 **LST[9:0]** : LPI LS タイマ

このフィールドは、LPI パターンを PHY に送信できるようになる前に、PHY からのリンクステータスをアップ (OKAY) にする必要がある最低時間（ミリ秒単位）を指定します。LPI LS タイマがプログラムされたターミナルカウントに達していない限り、MAC は LPIEN ビットがセットされている場合でも LPI パターンを送信しません。LPI LS タイマのデフォルト値は、IEEE 標準で定義された 1000（1 秒）です。

ビット 15:0 **TWT[15:0]** : LPI TW タイマ

このフィールドは、MAC が LPI パターンを PHY に送信した後から、通常の送信を再開するまで MAC が待機する最低時間（マイクロ秒単位）を指定します。TLPIEX ステータスビットは、このタイマが終了した後にセットされます。

LPI 移行タイマレジスタ (ETH_MACLETR)

アドレスオフセット : 0x00D8

リセット値 : 0x0000 0000

このレジスタは、Tx LPI 移行タイマを制御します。このカウンタは、[LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) レジスタの LPITE ビットが 1 にセットされているときにのみ有効です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPIET[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:0 **LPIET[19:0]** : LPI 移行タイマ

このフィールドは、MAC がすべてのフレームを送信した後から、LPI モードに移行するまで MAC が待機する時間をマイクロ秒単位で指定します。このフィールドは、LPITE および LPITXA が 1 にセットされたときにのみ有効です。

ビット [2:0] は、このタイマのステップが 8 マイクロ秒単位となる読み出し専用です。

1 マイクロ秒ティックカウンタレジスタ (ETH_MAC1USTCR)

アドレスオフセット : 0x00DC

リセット値 : 0x0000 0000

このレジスタは、すべての LPI タイマの基準時間（1 マイクロ秒ティック）の生成を制御します。このタイマは最初にソフトウェアでプログラムする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TIC_1US_CNTR [11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **TIC_1US_CNTR[11:0]** : 1 μ s ティックカウンタ

アプリケーションは、CSR クロックのクロックサイクル数が 1 μ s となるように、このカウンタをプログラムする必要があります（プログラムする前に値から 1 を引きます）。

たとえば、CSR クロックが 100 MHz の場合、このフィールドを 100 - 1 = 99（つまり、0x63）にプログラムする必要があります。

これは、一部の EEE 関連カウンタを更新するために使用される 1 μ s イベントを生成するために必要です。

バージョンレジスタ (ETH_MACVR)

アドレスオフセット : 0x0110

リセット値 : 0x0000 3242

バージョンレジスタは Ethernet ペリフェラルのバージョンを特定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERVER[7:0]								SNPSVER[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **USERVER[7:0]** : ST 定義のバージョンビット 7:0 **SNPSVER[7:0]** : IP バージョン**デバッグレジスタ (ETH_MACDR)**

アドレスオフセット : 0x0114

リセット値 : 0x0000 0000

デバッグレジスタは、さまざまな MAC ブロックのデバッグステータスを提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]		TPESTS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													r	r	r

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]		RPESTS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													r	r	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:17 **TFCSTS[1:0]** : MAC 送信パケットコントローラステータス

このフィールドは、MAC 送信パケットコントローラモジュールの状態を示します。

00 : アイドル状態

01 : 次のいずれかを待ちます。

— 前のパケットのステータス

— IPG またはバックオフ時間の経過

10 : ポーズ制御パケット (全二重モード) の生成と送信

11 : 送信用に入力パケットを転送

ビット 16 **TPESTS** : MAC MII 送信プロトコルエンジンステータス

このビットがセットされた場合、MAC MII 送信プロトコルエンジンがアクティブにデータを送信しているため、アイドル状態にないことを示します。

ビット 15:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **RFCFCSTS[1:0]** : MAC 受信パケットコントローラ FIFO ステータス

このビットがセットされた場合、このフィールドは、MAC 受信パケットコントローラモジュールのモジュール FIFO 読み出しおよび書き込みコントローラのアクティブ状態を示します。

ビット 0 **RPESTS** : MAC MII 受信プロトコルエンジンステータス

このビットがセットされた場合、MAC MII 受信プロトコルエンジンがアクティブにデータを受信しているため、アイドル状態にないことを示します。

HW 機能 0 レジスタ (ETH_MACHWF0R)

アドレスオフセット : 0x011C

リセット値 : 0x0A0D 73F7

このレジスタは、Ethernet ペリフェラルのオプションの特徴や機能の最初のセットの存在を示します。ソフトウェアドライバは、このレジスタを使用して、オプションのブロックに関連するプログラムの有効化や無効化を動的に行えます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	ACTPHYSEL[2:0]			SAVLANINS	TSSTSSEL[1:0]		MACADR64SEL	MACADR32SEL	ADDMACADRSEL[4:0]				Res.	RXCOESEL	
	r	r	r	r	r	r	r	r	r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXCOESEL	EEESEL	TSSEL	Res.	Res.	ARPOFFSEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	PCSSEL	HDSEL	GMIISEL	MIISEL
	r	r	r			r	r	r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **ACTPHYSEL[2:0]** : アクティブな PHY が選択済み

複数の PHY インタフェースが設定されている場合、このフィールドはリセットネグート中の phy_intf_sel_i のサンプル値を示します。

000 : GMII または MII

001 : RGMII

010 : SGMII

011 : TBI

100 : RMII

101 : RTBI

110 : SMII

その他 : 予約済み。使用できません。

ビット 27 **SAVLANINS** : ソースアドレスまたは VLAN 挿入の有効化

Tx での SA および VLAN 挿入オプションが選択されている場合、このビットは 1 にセットされます。

- ビット 26:25 **TSSTSEL[1:0]** : タイムスタンプシステムの時間ソース
このビットは、タイムスタンプシステム時間のソースを示します。
01 : 内部
10 : 外部
11 : 両方
00 : 予約済み。使用できません。
このビットは、IEEE 1588 タイムスタンプサポートの有効化オプションが選択されたときに 1 にセットされます。
- ビット 24 **MACADR64SEL** : MAC アドレス 64 ~ 127 が選択済み
追加の 64 MAC アドレスレジスタ (64 ~ 127) の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 23 **MACADR32SEL** : MAC アドレス 32 ~ 63 が選択済み
追加の 32 MAC アドレスレジスタ (32 ~ 63) の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 22:18 **ADDMACADRSEL[4:0]** : MAC アドレス 1 ~ 31 が選択済み
追加の 1 ~ 31 MAC アドレスレジスタの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 17 予約済みであり、リセット値に保持する必要があります。
- ビット 16 **RXCOESEL** : 受信チェックサムオフロードが有効
受信 TCP/IP チェックサムチェックの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **TXCOESEL** : 送信チェックサムオフロードが有効
送信 TCP/IP チェックサム挿入の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 13 **EEESEL** : 省電力型 Ethernet が有効
省電力型 Ethernet (EEE) の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 12 **TSSEL** : IEEE 1588-2008 タイムスタンプが有効
このビットは、IEEE 1588 タイムスタンプサポートの有効化オプションが選択されたときに 1 にセットされます。
- ビット 11:10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **ARPOFFSEL** : ARP オフロードが有効
IPv4 ARP オフロードの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 8 **MMCSEL** : RMON モジュールの有効化
MAC 管理カウンタ (MMC) が選択されている場合、このビットは 1 にセットされます。
- ビット 7 **MGKSEL** : PMT マジックパケットの有効化
マジックパケット検出の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 6 **RWKSEL** : PMT リモートウェイクアップパケットの有効化
リモートウェイクアップパケット検出の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 5 **SMASEL** : SMA (MDIO) インタフェース
ステーション管理 (MDIO インタフェース) の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 4 **VLHASH** : VLAN Hash フィルタが選択済み
VLAN Hash テーブルベースのフィルタリングの有効化オプションが選択されている場合、このビットは 1 にセットされます。

ビット 3 **PCSSEL** : PCS レジスタ (TBI、SGMII、または RTBI PHY インタフェース)

TBI、SGMII、または RTBI PHY インタフェースオプションが選択されている場合、このビットは 1 にセットされます。

ビット 2 **HDSEL** : 半二重のサポート

半二重モードが選択されている場合、このビットは 1 にセットされます。

ビット 1 **GMISEL** : 1000 Mbps のサポート

動作モードとして 1000 Mbps が選択されている場合、このビットは 1 にセットされます。

ビット 0 **MIISEL** : 10 または 100 Mbps のサポート

動作モードとして 10/100 Mbps が選択されている場合、このビットは 1 にセットされます。

HW 機能 1 レジスタ (ETH_MACHWF1R)

アドレスオフセット : 0x0120

リセット値 : 0x1104 1904

このレジスタは、Ethernet ペリフェラルのオプションの特徴や機能の 2 番目のセットの存在を示します。ソフトウェアドライバは、このレジスタを使用して、オプションのブロックに関連するプログラムの有効化や無効化を動的に行えます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	L3L4FNUM [3:0]				Res.	HASHTBLSZ[1:0]		POUOST	Res.	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN
	r	r	r	r		r	r	r		r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR64 [1:0]		ADVTHWORD		PTOEN	OSTEN	TXFIFOSIZE[4:0]				Res.	RXFIFOSIZE[4:0]				
r	r	r	r	r	r	r	r	r	r		r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:27 **L3L4FNUM[3:0]** : L3 または L4 フィルタの総数

このフィールドは、L3 または L4 フィルタの総数を示します。

0000 : L3 または L4 フィルタなし

0001 : L3 または L4 フィルタ 1 個

0010 : L3 または L4 フィルタ 2 個

..

1000 : L3 または L4 フィルタ 8 個

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **HASHTBLSZ[1:0]** : ハッシュテーブルサイズ

このフィールドは、ハッシュテーブルのサイズを示します。

00 : ハッシュテーブルなし

01 : 64

10 : 128

11 : 256

- ビット 23 **POUOST** : UDP/IP 経由 PTP 用のワンステップ機能の有効化
UDP/IP 経由 PTP のワンステップタイムスタンプ機能の有効化機能が選択されている場合、このビットは 1 にセットされます。
- ビット 22 予約済みであり、リセット値に保持する必要があります。
- ビット 21 **RAVSEL** : Rx 側のみ AV 機能の有効化
Rx 側でのみオーディオビデオブリッジングの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 20 **AVSEL** : AV 機能有効化
オーディオビデオブリッジングの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 19 **DBGMEMA** : DMA デバッグレジスタ有効化
デバッグモードの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 18 **TSOEN** : TCP セグメンテーションオフロード有効化
TCP/IP パケットの TCP セグメンテーションオフロード有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 17 **SPHEN** : スプリットヘッダ機能有効化
スプリットヘッダ構造の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 16 **DCBEN** : DCB 機能有効化
データセンターブリッジングの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 15:14 **ADDR64[1:0]** : アドレスの幅
このフィールドは、設定されたアドレスの幅を示します。
00 : 32 ビット
その他 : 予約済み。使用できません。
- ビット 13 **ADVTHWORD** : IEEE 1588 上位ワードレジスタ有効化
IEEE 1588 上位ワードレジスタの有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 12 **PTOEN** : PTP オフロード有効化
PTP タイムスタンプオフロード機能の有効化オプションが選択されている場合、このビットは 1 にセットされます。
- ビット 11 **OSTEN** : ワンステップタイムスタンプ有効化
ワンステップタイムスタンプ機能の有効化オプションが選択されている場合、このビットは 1 にセットされます。

ビット 10:6 **TXFIFOSIZE[4:0]** : MTL 送信 FIFO サイズ

このフィールドには、底を 2 とする対数から 7 を引いてバイト単位で表される MTL Tx FIFO の設定された値、つまり $\text{Log}_2(\text{TXFIFO_SIZE}) - 7$ が含まれます。

00000 : 128 バイト
00001 : 256 バイト
00010 : 512 バイト
00011 : 1,024 バイト
00100 : 2,048 バイト
00101 : 4,096 バイト
00110 : 8,192 バイト
00111 : 16,384 バイト
01000 : 32 KB
01001 : 64 KB
01010 : 128 KB
01011 から 11111 : 予約済み。使用できません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **RXFIFOSIZE[4:0]** : MTL 受信 FIFO サイズ

このフィールドには、底を 2 とする対数から 7 を引いてバイト単位で表される MTL Rx FIFO の設定された値、つまり $\text{Log}_2(\text{RXFIFO_SIZE}) - 7$ が含まれます。

00000 : 128 バイト
00001 : 256 バイト
00010 : 512 バイト
00011 : 1,024 バイト
00100 : 2,048 バイト
00101 : 4,096 バイト
00110 : 8,192 バイト
00111 : 16,384 バイト
01000 : 32 KB
01001 : 64 KB
01010 : 128 KB
01011 : 256KB
01100 から 11111 : 予約済み。使用できません。

HW 機能 2 レジスタ (ETH_MACHWF2R)

アドレスオフセット : 0x0124

リセット値 : 0x4100 0000

このレジスタは、Ethernet ペリフェラルのオプションの特徴や機能の 3 番目のセットの存在を示します。ソフトウェアドライバは、このレジスタを使用して、オプションのブロックに関連するプログラムの有効化や無効化を動的に行えます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			TDCSZ[1:0]		TXCHCNT[3:0]				RDCSZ[1:0]	
	r	r	r		r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCHCNT[3:0]				Res.	Res.	TXQCNT[3:0]				Res.	Res.	RXQCNT[3:0]			
r	r	r	r			r	r	r	r			r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **AUXSNAPNUM[2:0]** : 補助スナップショット入力数

このフィールドは、補助スナップショット入力数を示します。

000 : 補助入力なし

001 : 補助入力 1 個

010 : 補助入力 2 個

011 : 補助入力 3 個

100 : 補助入力 4 個

101 から 111 : 予約済み。使用できません。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:24 **PPSOUTNUM[2:0]** : PPS 出力数

このフィールドは、PPS 出力数を示します。

000 : PPS 出力なし

001 : PPS 出力 1 個

010 : PPS 出力 2 個

011 : PPS 出力 3 個

100 : PPS 出力 4 個

101 から 111 : 予約済み。使用できません。

ビット 23:22 **TDCSZ[1:0]** : 16 バイトディスクリプタに関する Tx DMA ディスクリプタキャッシュサイズ

00 : キャッシュ未設定

01 : 4 つの 16 バイトディスクリプタ

10 : 8 つの 16 バイトディスクリプタ

11 : 16 個の 16 バイトディスクリプタ

ビット 21:18 **TXCHCNT[3:0]** : DMA 送信チャネル数

このフィールドは、DMA 送信チャネル数を示します。

0000 : DMA Tx チャネル 1 個

0001 : DMA Tx チャネル 2 個

...

0111 : DMA Tx チャネル 8 個

ビット 17:16 **RDCSZ[1:0]** : 16 バイトディスクリプタに関する Rx DMA ディスクリプタキャッシュサイズ

00 : キャッシュ未設定

01 : 4 つの 16 バイトディスクリプタ

10 : 8 つの 16 バイトディスクリプタ

11 : 16 個の 16 バイトディスクリプタ

ビット 15:12 **RXCHCNT[3:0]** : DMA 受信チャネル数

このフィールドは、DMA 受信チャネル数を示します。

0000 : DMA Rx チャネル 1 個

0001 : DMA Rx チャネル 2 個

..

0111 : DMA Rx チャネル 8 個

ビット 11:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:6 **TXQCNT[3:0]** : MTL 送信キュー数

このフィールドは、MTL 送信キュー数を示します。

0000 : MTL Tx キュー 1 個

0001 : MTL Tx キュー 2 個

..

0111 : MTL Tx キュー 8 個

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **RXQCNT[3:0]** : MTL 受信キュー数

このフィールドは、MTL 受信キュー数を示します。

0000 : MTL Rx キュー 1 個

0001 : MTL Rx キュー 2 個

..

0111 : MTL Rx キュー 8 個

HW 機能 3 レジスタ (ETH_MACHWF3R)

アドレスオフセット : 0x0128

リセット値 : 0x0000 0020

このレジスタは、Ethernet ペリフェラルのオプションの特徴や機能の 4 番目のセットの存在を示します。ソフトウェアドライバは、このレジスタを使用して、オプションのブロックに関連するプログラムの有効化や無効化を動的に行えます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVLAN	CBTISEL	Res.	NRVF[2:0]		
										r	r		r	r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **DVLAN** : ダブル VLAN 処理の有効化

ダブル VLAN 処理が有効化されている場合、このビットは 1 にセットされます。

ビット 4 **CBTISEL** : Tx でのキュー / チャネルベースの VLAN タグ挿入の有効化

Tx での キュー / チャネルベースの VLAN タグ挿入の有効化機能が選択されている場合、このビットは 1 にセットされます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **NRVF[2:0]** : 拡張 VLAN タグフィルタの数が有効

このフィールドは、拡張 VLAN タグフィルタの数が選択されていることを示します。

000 : 拡張 Rx VLAN フィルタなし

001 : 4 つの拡張 Rx VLAN フィルタ

010 : 8 つの拡張 Rx VLAN フィルタ

011 : 16 個の拡張 Rx VLAN フィルタ

100 : 24 個の拡張 Rx VLAN フィルタ

101 : 32 個の拡張 Rx VLAN フィルタ

110 から 111 : 予約済み。使用できません。

MDIO アドレスレジスタ (ETH_MACMDIOAR)

アドレスオフセット : 0x0200

リセット値 : 0x0000 0000

MDIO アドレスレジスタは、管理インタフェースを通じて外部 PHY の管理サイクルを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]					RDA[4:0]				
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NTC[2:0]			CR[3:0]				Res.	Res.	Res.	SKAP	GOC[1:0]		C45E	MB
	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **PSE** : プリアンブル抑制有効化

このビットがセットされた場合、SMA は 32 ビットプリアンブルおよび 1 プリアンブルビットだけを持つ送信 MDIO フレームを抑制します。

このビットが 0 である場合、MDIO フレームには必ず IEEE 標準で定義された 32 ビットのプリアンブルがあります。

ビット 26 **BTB** : 連続トランザクション

このビットがセットされ、NTC に 0 を超える値がある場合、MAC はフレーム転送の最後（立下りクロックが送信される前）に読出しまたは書き込みコマンドの完了を通知します。このように、ソフトウェアは、前のフレームに生成された立下りクロックの数に関係なく、ただちに実行される次のコマンドを開始できます。

このビットがリセットされた場合、立下りクロック後でのみ読出し／書き込みコマンド完了（MII ビジーがクリア）が生成されます。このモードでは、各フレーム後に NTC が必ず生成されることを確認します。NTC=0 の場合、このビットはセットしないでください。

ビット 25:21 **PA[4:0]** : 物理層アドレス

このフィールドは、MAC がどの Clause 22 PHY デバイス（32 デバイスのうち）にアクセスしているかを示します。このフィールドは、MAC がどの Clause 45 対応 PHY（32 個の PHY のうち）にアクセスしているかを示します。

ビット 20:16 **RDA[4:0]** : レジスタ／デバイスアドレス

これらのビットは、選択された Clause 22 PHY デバイスで PHY レジスタを選択します。これらのビットは、選択された Clause 45 対応 PHY でデバイス（MMD）を選択します。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **NTC[2:0]** : 立下りクロックの数

このフィールドは、MDIO フレームの送信終了後に ETH_MDC で生成される立下りクロックサイクル数を制御します。有効な値は 0~7 です。値を 011 にプログラムすることで、MDIO フレームの転送終了後に MDC ラインでさらに 3 つのクロックサイクルがあることを示します。

ビット 11:8 **CR[3:0]** : CSR クロック範囲

CSR クロック範囲選択は、設計に使用される CSR クロック周波数に従って MDC クロックの周波数を決めます。

0000 : CSR クロック = 60~100 MHz、MDC クロック = CSR クロック/42

0001 : CSR クロック = 100~150 MHz、MDC クロック = CSR クロック/62

0010 : CSR クロック = 20~35 MHz、MDC クロック = CSR クロック/16

0011 : CSR クロック = 35~60 MHz、MDC クロック = CSR クロック/26

0100 : CSR クロック = 150~250 MHz、MDC クロック = CSR クロック/102

0101 : CSR クロック = 250~300 MHz、MDC クロック = CSR クロック/124

0110 から 0111 : 予約済み。使用できません。

各値に適用可能な CSR クロック周波数の推奨範囲 (ビット 11 = 0 のとき) は、MDC クロックが約 1.0 ~ 2.5 MHz の周波数の範囲になることを保証します。

ビット 11 がセットされた場合、2.5 MHz の周波数制限 (IEEE 802.3 で指定) を上回る MDC クロックの周波数を実現し、より低い値のクロック分周回路をプログラムできます。たとえば、CSR クロックの周波数が 100 MHz で、これらのビットを 1010 でプログラムする場合、その結果の MDC クロックは IEEE 802.3 で指定された範囲を超える 12.5 MHz となります。接続しているチップがより高速な MDC クロックをサポートしている場合のみ、次の値をプログラムします。

1000 : CSR クロック/4

1001 : CSR クロック/6

1010 : CSR クロック/8

1011 : CSR クロック/10

1100 : CSR クロック/12

1101 : CSR クロック/14

1110 : CSR クロック/16

1111 : CSR クロック/18

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **SKAP** : スキップアドレスパケット

このビットがセットされると、SMA は読み出し、書き込み、または事後読み出しインクリメントアドレスパケットの前にアドレスパケットを送信しません。このビットは、C45E がセットされている場合のみ有効です。

ビット 3:2 **GOC[1:0]** : MII 動作コマンド

このビットは、PHY の動作コマンドを示します。

00 : 予約済み。使用できません。

01 : 書き込み

10 : Clause 45 PHY に対する事後読み出しインクリメントアドレス

11 : 読出し

Clause 22 PHY が有効である場合、書き込みおよび読出しコマンドは有効です。

ビット 1 **C45E** : Clause 45 PHY 有効化

このビットがセットされた場合、Clause 45 対応 PHY が MDIO に接続されます。このビットがリセットされた場合、Clause 22 対応 PHY が MDIO に接続されます。

ビット 0 **MB** : MII ビジー

アプリケーションはこのビットをセットして SMA に MDIO の読出しまたは書き込み

アクセスを開始するよう命令します。MAC は、MDIO フレーム転送が完了した後に

このビットをクリアします。そのため、ソフトウェアは、このビットがセットされている限り、

MDIO アドレスレジスタ (ETH_MACMDIOAR) および **MDIO データレジスタ (ETH_MACMDIODR)** のフィールドへの書き込みや変更を行ってはなりません。

書き込み転送の場合、アプリケーションはまず **MDIO データレジスタ (ETH_MACMDIODR)** レジスタの MD フィールド (C45E がセットされている場合は RA フィールドも含む) に 16 ビットデータを書き込んでからこのビットをセットする必要があります。C45E がセットされている場合、**MDIO データレジスタ (ETH_MACMDIODR)** レジスタの RA フィールドにも書き込んでから読出し転送を開始する必要があります。読出し転送が完了すると (MII ビジー = 0)、PHY レジスタから読み出したデータが **MDIO データレジスタ (ETH_MACMDIODR)** の MD フィールドで有効になります。

注 : アドレス指定された PHY が存在しない場合でも、このビットの機能に変化はありません。

MDIO データレジスタ (ETH_MACMDIODR)

アドレスオフセット : 0x0204

リセット値 : 0x0000 0000

MDIO データレジスタは、**MDIO アドレスレジスタ (ETH_MACMDIOAR)** で指定されたアドレスにある PHY レジスタに書き込むデータを格納します。このレジスタは、MDIO アドレスレジスタによって指定されたアドレスにある PHY レジスタから読み出されたデータも格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 **RA[15:0]** : レジスタアドレス

このフィールドは、C45E がセットされている場合のみ有効です。これは、MDIO フレームが対象としている PHY のレジスタアドレスを含みます。

ビット 15:0 **MD[15:0]** : MII データ

このフィールドは、管理読出し動作の後で PHY から読み出された 16 ビットのデータ、または管理書き込み動作の前に PHY に書き込まれる 16 ビットデータを含みます。

ARP アドレスレジスタ (ETH_MACARPAR)

アドレスオフセット : 0x0210

リセット値 : 0x0000 0000

ARP アドレスレジスタは、MAC の IPv4 デスティネーションアドレスを含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPPA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARPPA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **ARPPA[31:0]** : ARP プロトコルアドレス

このフィールドは、MAC の IPv4 デスティネーションアドレスを含みます。このアドレスは、受信した ARP パケットにあるターゲットフィールドのプロトコルアドレスとの完全一致に使用されます。

CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)

アドレスオフセット : 0x0230

リセット値 : 0x0000 0000

このレジスタは、CSR アクセス応答の変更とステータスビットのクリアのためのソフトウェアプログラム可能な制御を格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCWE
							rw								rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SEEN** : スレーブエラー応答の有効化

このビットがセットされている場合、CSR スペースでの予約済みレジスタへのアクセスに対して、MAC はスレーブエラーを返します。

このビットがリセットされている場合、CSR スペースからアクセスされたレジスタに対して、MAC は OK 応答を返します。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **RCWE** : 1 の書込み時にレジスタクリアの有効化

このビットがセットされている場合、一部のレジスタフィールドへのアクセスモードが rc_w1 (書込み時にクリア) に変わります。つまり、クリアするにはアプリケーションで各ビットを 1 にセットする必要がありますということです。

このビットがリセットされている場合、これらのレジスタフィールドへのアクセスモードは rc_r (読出し時にクリア) のままです。

MACアドレス 0 上位レジスタ (ETH_MACA0HR)

アドレスオフセット : 0x0300

リセット値 : 0x8000 FFFF

MAC アドレス 0 上位レジスタは、ステーションの最初の MAC アドレス 6 バイトのうち上位 16 ビットを保持します。MII インタフェース上で受信された最初の DA バイトは、MAC アドレス下位レジスタの LS バイト (ビット [7:0]) に対応します。たとえば、MII 上でデスティネーションアドレスとして 0x112233445566 が受信された場合 (最初の列のレーン 0 で 0x11)、MacAddress0 レジスタ [47:0] は 0x665544332211 と比較されます。

MAC アドレスレジスタが、MII クロックドメインと二重同期するよう設定された場合、MAC Address0 下位レジスタのビット [31:24] (リトルエンディアンモード) またはビット [7:0] (ビッグエンディアンモード) が書き込まれたときのみ、同期がトリガされます。適切に同期更新を行うために、このアドレス下位レジスタへの連続書き込みは、デスティネーションクロックドメインの少なくとも 4 クロックサイクル後に実行されなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **AE** : アドレス有効化

このビットは常に 1 にセットされます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ADDRHI[15:0]** : MAC Address0[47:32]

このフィールドは、最初の MAC アドレス 6 バイトのうち上位 16 ビット [47:32] を含みます。MAC は、受信パケットのフィルタのためと、送信フロー制御 (ポーズ) パケットに MAC アドレスを挿入するために、このフィールドを使用します。

MACアドレス x 下位レジスタ (ETH_MACAxLR)

アドレスオフセット : 0x0304 + 0x8 * x (x = 0 から 3)

リセット値 : 0xFFFF FFFF

MAC アドレス x 下位レジスタは、ステーションの最初の MAC アドレス 6 バイトのうち下位 32 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **ADDRLO[31:0]** : MAC アドレス x [31:0] (x = 0~3)

このフィールドは、最初の MAC アドレス 6 バイトのうち下位 32 ビットを含みます。MAC は、受信パケットのフィルタのためと、送信フロー制御 (ポーズ) パケットに MAC アドレスを挿入するために、このフィールドを使用します。

MACアドレス x 上位レジスタ (ETH_MACAxHR)

アドレスオフセット : $0x0308 + 0x8 * (x-1)$ 、(x = 1 ~ 3)

リセット値 : 0x0000 FFFF

MAC アドレス x 上位レジスタは、ステーションの 2 番目の MAC アドレス 6 バイトのうち上位 16 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **AE** : アドレス有効化

このビットがセットされると、アドレスフィルタモジュールは、完全フィルタに 2 番目の MAC アドレスを使用します。このビットがリセットされると、アドレスフィルタモジュールはフィルタについてアドレスを無視します。

ビット 30 **SA** : ソースアドレス

このビットがセットされると、受信パケットの SA フィールドとの比較に MAC Addressx[47:0] が使用されます。このビットがリセットされると、受信パケットの DA フィールドとの比較に MAC Addressx[47:0] が使用されます。

0 : DA

1 : SA

ビット 29:24 **MBC[5:0]** : マスクバイト制御

これらのビットは、MAC アドレスの各バイトの比較のためのマスク制御ビットです。これらのビットがハイにセットされると、MAC は、受信した DA または SA の対応するバイトと MAC Address1 レジスタの内容とを比較しません。各ビットは、以下のようにバイトのマスキングを制御します。

ビット 29 : ETH_MACAxHR[15:8]

ビット 28 : ETH_MACAxHR[7:0]

ビット 27 : ETH_MACAxLR[31:24]

ビット 26 : ETH_MACAxLR[23:16]

ビット 25 : ETH_MACAxLR[15:8]

ビット 24 : ETH_MACAxLR[7:0]

アドレスの 1 つ以上のバイトをマスクすることで、アドレスのグループをフィルタできます (グループアドレスフィルタとして知られる)。

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ADDRHI[15:0]** : MAC Address1 [47:32]

このフィールドは、2 番目の MAC アドレス 6 バイトのうち上位 16 ビット [47:32] を含みます。

MMC 制御レジスタ (ETH_MMC_CONTROL)

アドレスオフセット : 0x0700

リセット値 : 0x0000 0000

このレジスタは、MMC 動作モードを設定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC	Res.	Res.	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
							rw			rw	rw	rw	rw	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 UCDBC : ドロップされたブロードバンドパケットの MMC カウンタの更新

CNTRST ビットは、CNTPRST ビットより優先されます。そのため、ソフトウェアが同じ書き込みサイクルで両方のビットをセットしようとする、すべてのカウンタがクリアされ、CNTPRST ビットはセットされません。

セットされた場合、MAC は [パケットフィルタ制御レジスタ \(ETH_MACPFR\)](#) の DBF ビットの設定によってドロップされたブロードバンドパケットの関連する MMC カウンタすべてを更新します。

リセットされた場合、MMC カウンタはドロップされたブロードバンドパケットについて更新されません。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 CNTPRSTLVL : フルハーフプリセット

このビットがローで CNTPRST ビットがセットされた場合、すべての MMC カウンタがほぼハーフ値にプリセットされます。また、すべてのオクテットカウンタが 0x7FFF_F800 (ハーフ 2 KB) にプリセットされ、すべてパケットカウンタが 0x7FFF_FFF0 (ハーフ 16) にプリセットされます。

このビットがハイで CNTPRST ビットがセットされた場合、すべての MMC カウンタがほぼフル値にプリセットされます。また、すべてのオクテットカウンタが 0xFFFF_F800 (フル 2 KB) にプリセットされ、すべてパケットカウンタが 0xFFFF_FFF0 (フル 16) にプリセットされます。

16 ビットカウンタの場合、ほぼハーフのプリセット値は、オクテットカウンタおよびパケットカウンタそれぞれに対して 0x7800 および 0x7FF0 です。同様に、16 ビットカウンタのほぼフルのプリセット値は、0xF800 および 0xFFFF0 です。

ビット 4 CNTPRST : カウンタプリセット

このビットがセットされた場合、すべてのカウンタが初期化されるか、CNTPRSTLVL ビットに従ってほぼフルまたはほぼハーフにプリセットされます。このビットは、1 クロックサイクル後に自動的にクリアされます。

CNTPRSTLVL ビットとともに、このビットは MMC カウンタがハーフフルまたはフルになっているために発生する割込みのアサーションのデバッグやテストに有用です。

ビット 3 CNTFREEZ : MMC カウンタ固定

このビットがセットされた場合、すべての MMC カウンタを現在の値で固定します。

このビットが 0 にリセットされた場合、MMC カウンタはパケットの送受信のために更新されます。MMC カウンタのいずれかが読み出し時リセットビットがセットされた状態で読み出された場合、そのカウンタはこのモードでクリアされます。

ビット 2 RSTONRD : 読み出し時リセット

このビットがセットされた場合、MMC カウンタは、読み出し後に 0 にリセットされます (リセット後、自動クリア)。カウンタは、最下位バイトレーン (ビット [7:0]) が読み出されると、クリアされます。

ビット 1 **CNTSTOPRO** : カウンタストップロールオーバー

このビットがセットされた場合、カウンタは、最大値に達した後も 0 にロールオーバーしません。

ビット 0 **CNTRST** : カウンタリセット

このビットがセットされた場合、すべてのカウンタがリセットされます。このビットは、1 クロックサイクル後に自動的にクリアされます。

MMC Rx 割込みレジスタ (ETH_MMC_RX_INTERRUPT)

アドレスオフセット : 0x0704

リセット値 : 0x0000 0000

このレジスタは、すべての受信統計カウンタから生成された割込みを保持します。

MMC 受信割込みレジスタは、次のことが発生した際に生成される割込みを保持します。

- 受信統計カウンタが、最大値の半分に達しています (32 ビットカウンタでは 0x8000_0000、16 ビットカウンタでは 0x8000)。
- 受信統計カウンタが、最大値を超えています (32 ビットカウンタでは 0xFFFF_FFFF、16 ビットカウンタでは 0xFFFF)。

MMC 制御レジスタ (ETH_MMC_CONTROL) で **CNTSTOPRO** がセットされた場合、割込みがセットされますが、カウンタはすべて 1 のままとなります。MMC 受信割込みレジスタは 32 ビットレジスタです。割込みビットは、割込みの原因となったそれぞれの MMC カウンタが読み出されたときにクリアされます。割込みビットをクリアするには、それぞれのカウンタの最下位バイトレーン (ビット [7:0]) が読み出されなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIS	RXLPIUSCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIS	Res.
				rc_r	rc_r									rc_r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNRPIS	RXCRCRPIS	Res.	Res.	Res.	Res.	Res.
									rc_r	rc_r					

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **RXLPITRCIS** : MMC 受信 LPI 遷移カウンタ割込みステータス

このビットは、**Rx LPI 遷移カウンタレジスタ (ETH_RX_LPI_TRAN_CNTR)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 26 **RXLPIUSCIS** : MMC 受信 LPI マイクロ秒カウンタ割込みステータス

このビットは、**Rx LPI マイクロ秒カウンタレジスタ (ETH_RX_LPI_USEC_CNTR)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 25:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RXUCGPIS** : MMC 受信ユニキャスト良好パケットカウンタ割込みステータス

このビットは、**Rx ユニキャストパケット良好レジスタ (ETH_RX_UNICAST_PACKETS_GOOD)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 16:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **RXALGNERPIS** : MMC 受信整列エラーパケットカウンタ割込みステータス

このビットは、**Rx 整列エラーパケットレジスタ (ETH_RX_ALIGNMENT_ERROR_PACKETS)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 5 **RXCRCERPIS** : MMC 受信 CRC エラーパケットカウンタ割込みステータス

このビットは、**Rx CRC エラーパケットレジスタ (ETH_RX_CRC_ERROR_PACKETS)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

MMC Tx 割込みレジスタ (ETH_MMC_TX_INTERRUPT)

アドレスオフセット : 0x0708

リセット値 : 0x0000 0000

このレジスタは、すべての送信統計カウンタから生成された割込みを保持します。

MMC 送信割込みレジスタは、送信統計カウンタが最大値の半分 (32 ビットカウンタでは 0x8000_0000、16 ビットカウンタでは 0x8000) に達した場合、および最大値 (32 ビットカウンタでは 0xFFFF_FFFF、16 ビットカウンタでは 0xFFFF) を超えた場合に生成される割込みを保持します。

MMC 制御レジスタ (ETH_MMC_CONTROL) で CNTSTOPRO がセットされた場合、割込みがセットされますが、カウンタはすべて 1 のままとなります。

MMC 送信割込みレジスタは 32 ビットレジスタです。割込みビットは、割込みの原因となったそれぞれの MMC カウンタが読み出されたときにクリアされます。

割込みビットをクリアするには、それぞれのカウンタの最下位バイトレーン (ビット [7:0]) が読み出されなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCS	TXLPIUSCIS	Res.	Res.	Res.	Res.	TXGPKTIS	Res.	Res.	Res.	Res.	Res.
				rc_r	rc_r					rc_r					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOGPIS	TXSCOLGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rc_r	rc_r														

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **TXLPITRCS** : MMC 送信 LPI 遷移カウンタ割込みステータス

このビットは、**Tx LPI 遷移カウンタレジスタ (ETH_TX_LPI_TRAN_CNTR)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 26 **TXLPIUSCIS** : MMC 送信 LPI マイクロ秒カウンタ割込みステータス

このビットは、**Tx LPI マイクロ秒タイマレジスタ (ETH_TX_LPI_USEC_CNTR)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TXGPKTIS** : MMC 送信良好パケットカウンタ割込みステータス

このビットは、**Tx パケットカウンタ良好レジスタ (ETH_TX_PACKET_COUNT_GOOD)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 20:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TXMCOLGPIS** : MMC 送信複数衝突良好パケットカウンタ割込みステータス

このビットは、**Tx 複数衝突良好パケットレジスタ (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 14 **TXSCOLGPIS** : MMC 送信単一衝突良好パケットカウンタ割込みステータス

このビットは、**Tx 単一衝突良好パケットレジスタ (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)** カウンタが最大値の半分または最大値に達したときにセットされます。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

MMC Rx 割込みマスクレジスタ (ETH_MMC_RX_INTERRUPT_MASK)

アドレスオフセット : 0x070C

リセット値 : 0x0000 0000

MMC 受信割込みマスクレジスタは、受信統計カウンタが最大値の半分または最大値に達したときに生成される割込みのマスクを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.
				rw	rw									rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNRPIM	RXCRCPIM	Res.	Res.	Res.	Res.	Res.
									rw	rw					

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **RXLPITRCIM** : MMC 受信 LPI 遷移カウンタ割込みマスク

このビットがセットされると、**Rx LPI 遷移カウンタレジスタ (ETH_RX_LPI_TRAN_CNTR)** カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 26 **RXLPIUSCIM** : MMC 受信 LPI マイクロ秒カウンタ割込みマスク

このビットがセットされると、**Rx LPI マイクロ秒カウンタレジスタ (ETH_RX_LPI_USEC_CNTR)** カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 25:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RXUCGPIM** : MMC 受信ユニキャスト良好パケットカウンタ割込みマスク

このビットがセットされると、**Rx ユニキャストパケット良好レジスタ (ETH_RX_UNICAST_PACKETS_GOOD)** カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 16:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **RXALGNERPIM** : MMC 受信整列エラーパケットカウンタ割込みマスク

このビットがセットされると、[Rx 整列エラーパケットレジスタ \(ETH_RX_ALIGNMENT_ERROR_PACKETS\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 5 **RXCRCERPIM** : MMC 受信 CRC エラーパケットカウンタ割込みマスク

このビットがセットされると、[Rx CRC エラーパケットレジスタ \(ETH_RX_CRC_ERROR_PACKETS\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

MMC Tx 割込みマスクレジスタ (ETH_MMC_TX_INTERRUPT_MASK)

アドレスオフセット : 0x0710

リセット値 : 0x0000 0000

MMC 送信割込みマスクレジスタは、送信統計カウンタが最大値の半分または最大値に達したときに生成される割込みのマスクを保持します。このレジスタは 32 ビット幅です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.
				rw	rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **TXLPITRCIM** : MMC 送信 LPI 遷移カウンタ割込みマスク

このビットがセットされると、[Tx LPI 遷移カウンタレジスタ \(ETH_TX_LPI_TRAN_CNTR\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 26 **TXLPIUSCIM** : MMC 送信 LPI マイクロ秒カウンタ割込みマスク

このビットがセットされると、[Tx LPI マイクロ秒タイマレジスタ \(ETH_TX_LPI_USEC_CNTR\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TXGPKTIM** : MMC 送信良好パケットカウンタ割込みマスク

このビットがセットされると、[Tx パケットカウンタ良好レジスタ \(ETH_TX_PACKET_COUNT_GOOD\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 20:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TXMCOLGPIM** : MMC 送信複数衝突良好パケットカウンタ割込みマスク

このビットがセットされると、[Tx 複数衝突良好パケットレジスタ \(ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS\)](#) カウンタが最大値の半分または最大値に達したときに割込みをマスクします。

ビット 14 **TXSCOLGPIM** : MMC 送信単一衝突良好パケットカウンタ割込みマスク
 このビットがセットされると、Tx 単一衝突良好パケットレジスタ
 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS) カウンタが最大値の半分または最大値に達した
 ときに割込みをマスクします。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

**Tx 単一衝突良好パケットレジスタ
 (ETH_TX_SINGLE_COLLISION_GOOD_PACKETS)**

アドレスオフセット : 0x074C

リセット値 : 0x0000 0000

このレジスタは、半二重モードで 1 回の衝突後に Ethernet ペリフェラルによって正常に送信された
 パケットの数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSNGLCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSNGLCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXSNGLCOLG[31:0]** : Tx 単一衝突良好パケット
 このフィールドは、半二重モードで 1 回の衝突後に正常に送信されたパケットの数を示します。

**Tx 複数衝突良好パケットレジスタ
 (ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS)**

アドレスオフセット : 0x0750

リセット値 : 0x0000 0000

このレジスタは、半二重モードで複数回の衝突後に Ethernet ペリフェラルによって正常に送信された
 パケットの数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXMULTCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMULTCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXMULTCOLG[31:0]** : Tx 複数衝突良好パケット
 このフィールドは、半二重モードで複数回の衝突後に正常に送信されたパケットの数を示します。

Tx パケットカウント良好レジスタ (ETH_TX_PACKET_COUNT_GOOD)

アドレスオフセット : 0x0768

リセット値 : 0x0000 0000

このレジスタは、Ethernet ペリフェラルによって送信された良好フレームの数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXPKTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXPKTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXPKTG[31:0]** : Tx パケットカウント良好

このフィールドは、送信された良好パケットの数を示します。

Rx CRC エラーパケットレジスタ (ETH_RX_CRC_ERROR_PACKETS)

アドレスオフセット : 0x0794

リセット値 : 0x0000 0000

このレジスタは、CRC エラーがある Ethernet ペリフェラルによって受信されたパケットの数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRCERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRCERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXCRCERR[31:0]** : Rx CRC エラーパケット

このフィールドは、CRC エラーがある受信パケットの数を示します。

Rx 整列エラーパケットレジスタ (ETH_RX_ALIGNMENT_ERROR_PACKETS)

アドレスオフセット : 0x0798

リセット値 : 0x0000 0000

このレジスタは、整列（ドリブル）エラーがある Ethernet ペリフェラルによって受信されたパケットの数を提供します。このレジスタは 10/100 モードでのみ有効です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXALGNERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXALGNERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXALGNERR[31:0]** : Rx 整列エラーパケット

このフィールドは、整列（ドリブル）エラーがある受信パケットの数を示します。このレジスタは 10/100 モードでのみ有効です。

Rx ユニキャストパケット良好レジスタ (ETH_RX_UNICAST_PACKETS_GOOD)

アドレスオフセット : 0x07C4

リセット値 : 0x0000 0000

このレジスタは、Ethernet ペリフェラルによって受信された良好ユニキャストパケットの数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXUCASTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXUCASTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXUCASTG[31:0]** : Rx ユニキャストパケット良好

このフィールドは、受信された良好ユニキャストパケットの数を示します。

Tx LPI マイクロ秒タイマレジスタ (ETH_TX_LPI_USEC_CNTR)

アドレスオフセット : 0x07EC

リセット値 : 0x0000 0000

このレジスタは、Tx LPI が Ethernet ペリフェラルによってアサートされたマイクロ秒数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXLPIUSC[31:0]** : Tx LPI マイクロ秒カウンタ

このフィールドは、Tx LPI がアサートされたマイクロ秒数を示します。それぞれの Tx LPI の移行および終了に対して、タイマ値に +/- 1 マイクロ秒の誤差を設けることができます。

Tx LPI 遷移カウンタレジスタ (ETH_TX_LPI_TRAN_CNTR)

アドレスオフセット : 0x07F0

リセット値 : 0x0000 0000

このレジスタは、Ethernet ペリフェラルが Tx LPI に移行した回数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXLPITRC[31:0]** : Tx LPI 遷移カウンタこのフィールドは、Tx LPI 移行が発生した回数を示します。Tx LPI 移行が ([LPI 制御およびステータスレジスタ \(ETH_MACLCSR\)](#) でセットされた LPITXA ビットによって) 自動化モードで発生しても、カウンタはインクリメントします。

Rx LPI マイクロ秒カウンタレジスタ (ETH_RX_LPI_USEC_CNTR)

アドレスオフセット : 0x07F4

リセット値 : 0x0000 0000

このレジスタは、Rx LPI が Ethernet ペリフェラルによってサンプリングされたマイクロ秒数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPUIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPUIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXLPUIUSC[31:0]** : Rx LPI マイクロ秒カウンタ

このフィールドは、Rx LPI がアサートされたマイクロ秒数を示します。それぞれの Rx LPI の移行および終了に対して、タイマ値に +/- 1 マイクロ秒の誤差を設けることができます。

Rx LPI 遷移カウンタレジスタ (ETH_RX_LPI_TRAN_CNTR)

アドレスオフセット : 0x07F8

リセット値 : 0x0000 0000

このレジスタは、Ethernet ペリフェラルが Rx LPI に移行した回数を提供します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RXLPITRC[31:0]** : Rx LPI 遷移カウンタ

このフィールドは、Rx LPI 移行が発生した回数を示します。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R)

アドレスオフセット : 0x0900

リセット値 : 0x0000 0000

レイヤ 3 およびレイヤ 4 制御レジスタは、レイヤ 3 およびレイヤ 4 のフィルタ 0 の動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Res.	L4PEN0
										rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM0 [4:0]					L3HSBM0 [4:0]					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Res.	L3PEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **L4DPIM0** : レイヤ 4 デスティネーションポート反転一致の有効化

このビットがセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの完全一致が有効になります。

このビットは、L4DPM0 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 20 **L4DPM0** : レイヤ 4 デスティネーションポート一致の有効化

このビットがセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 4 デスティネーションポート番号フィールドの一致を無視します。

ビット 19 **L4SPIM0** : レイヤ 4 ソースポート反転一致の有効化

このビットがセットされた場合、レイヤ 4 ソースポート番号フィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 4 ソースポート番号フィールドの完全一致が有効になります。このビットは、L4SPM0 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 18 **L4SPM0** : レイヤ 4 ソースポート一致の有効化

このビットがセットされた場合、レイヤ 4 ソースポート番号フィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 4 ソースポート番号フィールドの一致を無視します。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **L4PEN0** : レイヤ 4 プロトコルの有効化

このビットがセットされた場合、UDP パケットのソースおよびデスティネーションポート番号が一致に使用されます。このビットがリセットされた場合、TCP パケットのソースおよびデスティネーションポート番号が一致に使用されます。

レイヤ 4 一致は、L4SPM0 または L4DPM0 ビットがセットされたときにのみ実行されます。

ビット 15:11 **L3HDBM0[4:0]** : レイヤ 3 IP DA 上位ビット一致

状態 : IPv4 パケット

このフィールドは、IPv4 パケットでマスクする IP デスティネーションアドレスの上位ビットの数を含みます。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

31 : MSb を除くすべてのビットがマスクされます。

状態 : IPv6 パケット

このフィールドのビット [12:11] は、IPv6 パケットでマスクされる IP ソースまたはデスティネーションアドレスの下位ビットの数を示す L3HSBM0 のビット [6:5] に対応します。マスクされるビット数は、L3HDBM0[1:0] ビットと L3HSBM0 ビットの結合値により与えられます。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

31 : MSb を除くすべてのビットがマスクされます。

このフィールドは、L3DAM0 または L3SAM0 ビットがセットされたときにのみ有効であり、適用されます。

ビット 10:6 **L3HSBM0[4:0]** : レイヤ 3 IP SA 上位ビット一致

状態 : IPv4 パケット

このフィールドは、IPv4 パケットでの一致に対してマスクされる IP ソースアドレスの下位ビットの数を含みます。次のリストでは、このフィールドの値を示しています。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

31 : MSb を除くすべてのビットがマスクされます。

状態 : IPv6 パケット :

このフィールドは、L3HSBM0 のビット [4:0] を含みます。これらのビットは、IPv6 パケットで一致する IP ソースまたはデスティネーションアドレスの上位ビットの数を示します。このフィールドは、L3DAM0 または L3SAM0 ビットがハイにセットされたときにのみ有効であり、適用されます。

ビット 5 **L3DAIM0** : レイヤ 3 IP DA 反転一致の有効化

このビットがセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの完全一致が有効になります。

このビットは、L3DAM0 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 4 **L3DAM0** : レイヤ 3 IP DA 一致の有効化

このビットがセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 3 IP デスティネーションアドレスフィールドの一致を無視します。

注 : **L3PEN0** ビットがセットされた場合、IPv6 DA または SA がフィルタでチェックできるため、このビットが **L3SAM0** ビットをセットする必要があります。

ビット 3 **L3SAIM0** : レイヤ 3 IP SA 反転一致の有効化

このビットがセットされた場合、レイヤ 3 IP ソースアドレスフィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 3 IP ソースアドレスフィールドの完全一致が有効になります。このビットは、L3SAM0 ビットがセットされた場合にのみ有効であり、適用されます。

ビット 2 **L3SAM0** : レイヤ 3 IP SA 一致の有効化

このビットがセットされた場合、レイヤ 3 IP ソースアドレスフィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 3 IP ソースアドレスフィールドの一致を無視します。

注 : **L3PEN0** ビットがセットされた場合、IPv6 SA または DA がフィルタでチェックできるため、このビットか **L3DAM0** ビットをセットする必要があります。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **L3PEN0** : レイヤ 3 プロトコルの有効化

このビットがセットされた場合、レイヤ 3 IP ソースまたはデスティネーションアドレスの一致が IPv6 パケットに対して有効になります。このビットがリセットされた場合、レイヤ 3 IP ソースまたはデスティネーションアドレスの一致が IPv4 パケットに対して有効になります。

レイヤ 3 一致は、**L3SAM0** または **L3DAM0** ビットがセットされたときにのみ実行されます。

レイヤ 4 アドレスフィルタ 0 レジスタ (ETH_MACL4A0R)

アドレスオフセット : 0x0904

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP0 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP0 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **L4DP0[15:0]** : レイヤ 4 デスティネーションポート番号フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で **L4PEN0** ビットがリセットされ、**L4DPM0** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの TCP デスティネーションポート番号フィールドと一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で **L4PEN0** および **L4DPM0** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの UDP デスティネーションポート番号フィールドと一致する値を含みます。

ビット 15:0 **L4SP0[15:0]** : レイヤ 4 ソースポート番号フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で **L4PEN0** ビットがリセットされ、**L4DPM0** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの TCP ソースポート番号フィールドと一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で **L4PEN0** および **L4DPM0** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの UDP ソースポート番号フィールドと一致する値を含みます。

レイヤ 3 アドレス 0 フィルタ 0 レジスタ (ETH_MACL3A00R)

アドレスオフセット : 0x0910

リセット値 : 0x0000 0000

IPv4 パケットでは、レイヤ 3 アドレス 0 フィルタ 0 レジスタは、32 ビットの IP ソースアドレスフィールドを含みます。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [31:0] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A00 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A00 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **L3A00[31:0]** : レイヤ 3 アドレス 0 フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3SAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [31:0] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3DAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [31:0] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 ビットがリセットされ、L3SAM0 ビットがセットされた場合、このフィールドは IPv4 パケットの IP ソースアドレスフィールドと一致する値を含みます。

レイヤ 3 アドレス 1 フィルタ 0 レジスタ (ETH_MACL3A10R)

アドレスオフセット : 0x0914

リセット値 : 0x0000 0000

IPv4 パケットでは、レイヤ 3 アドレス 1 フィルタ 0 レジスタは、32 ビットの IP デスティネーションアドレスフィールドを含みます。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [63:32] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A10 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A10 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **L3A10[31:0]** : レイヤ 3 アドレス 1 フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3SAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [63:32] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3DAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [63:32] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 ビットがリセットされ、L3SAM0 ビットがセットされた場合、このフィールドは IPv4 パケットの IP デスティネーションアドレスフィールドと一致する値を含みます。

レイヤ 3 アドレス 2 フィルタ 0 レジスタ (ETH_MACL3A20R)

アドレスオフセット : 0x0918

リセット値 : 0x0000 0000

レイヤ 3 アドレス 2 フィルタ 0 レジスタは、IPv4 パケットでは予約済みとなります。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [95:64] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A20 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A20 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **L3A20[31:0]** : レイヤ 3 アドレス 2 フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3SAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [95:64] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3DAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [95:64] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 ビットがリセットされた場合、このフィールドは使用されません。

レイヤ 3 アドレス 3 フィルタ 0 レジスタ (ETH_MACL3A30R)

アドレスオフセット : 0x091C

リセット値 : 0x0000 0000

レイヤ 3 アドレス 3 フィルタ 0 レジスタは、IPv4 パケットでは予約済みとなります。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [127:96] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A30 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A30 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **L3A30[31:0]** : レイヤ 3 アドレス 3 フィールド

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3SAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [127:96] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 および L3DAM0 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [127:96] と一致する値を含みます。

L3 および L4 制御 0 レジスタ (ETH_MACL3L4C0R) で L3PEN0 ビットがリセットされた場合、このフィールドは使用されません。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R)

アドレスオフセット : 0x0930

リセット値 : 0x0000 0000

レイヤ 3 およびレイヤ 4 制御レジスタは、レイヤ 3 およびレイヤ 4 のフィルタ 1 の動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1
										rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM1 [4:0]					L3HSBM1 [4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **L4DPIM1** : レイヤ 4 デスティネーションポート反転一致の有効化

このビットがセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの完全一致が有効になります。

このビットは、L4DPM1 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 20 **L4DPM1** : レイヤ 4 デスティネーションポート一致の有効化

このビットがセットされた場合、レイヤ 4 デスティネーションポート番号フィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 4 デスティネーションポート番号フィールドの一致を無視します。

ビット 19 **L4SPIM1** : レイヤ 4 ソースポート反転一致の有効化

このビットがセットされた場合、レイヤ 4 ソースポート番号フィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 4 ソースポート番号フィールドの完全一致が有効になります。このビットは、L4SPM1 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 18 **L4SPM1** : レイヤ 4 ソースポート一致の有効化

このビットがセットされた場合、レイヤ 4 ソースポート番号フィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 4 ソースポート番号フィールドの一致を無視します。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **L4PEN1** : レイヤ 4 プロトコルの有効化

このビットがセットされた場合、UDP パケットのソースおよびデスティネーションポート番号が一致に使用されます。このビットがリセットされた場合、TCP パケットのソースおよびデスティネーションポート番号が一致に使用されます。

レイヤ 4 一致は、L4SPM1 または L4DPM1 ビットがセットされたときにのみ実行されます。

ビット 15:11 **L3HDBM1[4:0]** : レイヤ 3 IP DA 上位ビット一致

状態 : IPv4 パケット

このフィールドは、IPv4 パケットでの一致に対してマスクされる IP デスティネーションアドレスの下位ビットの数を含みます。次のリストでは、このフィールドの値を示しています。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

31 : MSb を除くすべてのビットがマスクされます。

状態 : IPv6 パケット

このフィールドのビット [12:11] は、IPv6 パケットでマスクされる IP ソースまたはデスティネーションアドレスの下位ビットの数を示す L3HSBM1 のビット [6:5] に対応します。次のリストでは、L3HDBM1[1:0] および L3HSBM1 ビットの連結された値を示しています。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

127 : MSb を除くすべてのビットがマスクされます。

このフィールドは、L3DAM1 または L3SAM1 ビットがセットされたときにのみ有効であり、適用されます。

ビット 10:6 **L3HSBM1[4:0]** : レイヤ 3 IP SA 上位ビット一致

状態 : IPv4 パケット

このフィールドは、IPv4 パケットでの一致に対してマスクされる IP ソースアドレスの下位ビットの数を含みます。次のリストでは、このフィールドの値を示しています。

0 : ビットはマスクされません。

1 : LSb[0] がマスクされます。

2 : 2 つの LSb [1:0] がマスクされます。

..

31 : MSb を除くすべてのビットがマスクされます。

状態 : IPv6 パケット

このフィールドは、L3HSBM1 のビット [4:0] を含みます。これらのビットは、IPv6 パケットで一致する IP ソースまたはデスティネーションアドレスの上位ビットの数を示します。このフィールドは、L3DAM1 または L3SAM1 ビットがハイにセットされたときにのみ有効であり、適用されます。

ビット 5 **L3DAIM1** : レイヤ 3 IP DA 反転一致の有効化

このビットがセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの完全一致が有効になります。

このビットは、L3DAM1 ビットがハイにセットされた場合にのみ有効であり、適用されます。

ビット 4 **L3DAM1** : レイヤ 3 IP DA 一致の有効化

このビットがセットされた場合、レイヤ 3 IP デスティネーションアドレスフィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 3 IP デスティネーションアドレスフィールドの一致を無視します。

注 : L3PEN1 ビットがセットされた場合、IPv6 DA または SA がフィルタでチェックできるため、このビットか L3SAM1 ビットをセットする必要があります。

ビット 3 **L3SAIM1** : レイヤ 3 IP SA 反転一致の有効化

このビットがセットされた場合、レイヤ 3 IP ソースアドレスフィールドの反転一致が有効になります。このビットがリセットされた場合、レイヤ 3 IP ソースアドレスフィールドの完全一致が有効になります。このビットは、L3SAM1 ビットがセットされた場合にのみ有効であり、適用されます。

ビット 2 **L3SAM1** : レイヤ 3 IP SA 一致の有効化

このビットがセットされた場合、レイヤ 3 IP ソースアドレスフィールドの一致が有効になります。このビットがリセットされた場合、MAC はレイヤ 3 IP ソースアドレスフィールドの一致を無視します。

注 : **L3PEN01** ビットがセットされた場合、IPv6 SA または DA がフィルタでチェックできるため、このビットか **L3DAM1** ビットをセットする必要があります。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **L3PEN1** : レイヤ 3 プロトコルの有効化

このビットがセットされた場合、レイヤ 3 IP ソースまたはデスティネーションアドレスの一致が IPv6 パケットに対して有効になります。このビットがリセットされた場合、レイヤ 3 IP ソースまたはデスティネーションアドレスの一致が IPv4 パケットに対して有効になります。

レイヤ 3 一致は、**L3SAM1** または **L3DAM1** ビットがセットされたときにのみ実行されます。

レイヤ 4 アドレスフィルタ 1 レジスタ (ETH_MACL4A1R)

アドレスオフセット : 0x0934

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP1 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP1 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **L4DP1[15:0]** : レイヤ 4 デスティネーションポート番号フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で **L4PEN1** ビットがリセットされ、**L4DPM1** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの TCP デスティネーションポート番号フィールドと一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で **L4PEN1** および **L4DPM1** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの UDP デスティネーションポート番号フィールドと一致する値を含みます。

ビット 15:0 **L4SP1[15:0]** : レイヤ 4 ソースポート番号フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で **L4PEN1** ビットがリセットされ、**L4DPM1** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの TCP ソースポート番号フィールドと一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で **L4PEN1** および **L4DPM1** ビットがセットされた場合、このフィールドは IPv4 または IPv6 パケットの UDP ソースポート番号フィールドと一致する値を含みます。

レイヤ 3 アドレス 0 フィルタ 1 レジスタ (ETH_MACL3A01R)

アドレスオフセット : 0x0940

リセット値 : 0x0000 0000

IPv4 パケットでは、レイヤ 3 アドレス 0 フィルタ 1 レジスタは、32 ビットの IP ソースアドレスフィールドを含みます。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [31:0] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A01 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A01 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **L3A01[31:0]** : レイヤ 3 アドレス 0 フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3SAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [31:0] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3DAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [31:0] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 ビットがリセットされ、L3SAM1 ビットがセットされた場合、このフィールドは IPv4 パケットの IP ソースアドレスフィールドと一致する値を含みます。

レイヤ 3 アドレス 1 フィルタ 1 レジスタ (ETH_MACL3A11R)

アドレスオフセット : 0x0944

リセット値 : 0x0000 0000

IPv4 パケットでは、レイヤ 3 アドレス 1 フィルタ 1 レジスタは、32 ビットの IP デスティネーションアドレスフィールドを含みます。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [63:32] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A11 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A11 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **L3A11[31:0]** : レイヤ 3 アドレス 1 フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3SAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [63:32] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3DAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [63:32] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 ビットがリセットされ、L3SAM1 ビットがセットされた場合、このフィールドは IPv4 パケットの IP デスティネーションアドレスフィールドと一致する値を含みます。

レイヤ 3 アドレス 2 フィルタ 1 レジスタ (ETH_MACL3A21R)

アドレスオフセット : 0x0948

リセット値 : 0x0000 0000

レイヤ 3 アドレス 2 フィルタ 1 レジスタは、IPv4 パケットでは予約済みとなります。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [95:64] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A21 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A21 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **L3A21[31:0]** : レイヤ 3 アドレス 2 フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3SAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [95:64] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3DAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [95:64] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 ビットがリセットされた場合、このフィールドは使用されません。

レイヤ 3 アドレス 3 フィルタ 1 レジスタ (ETH_MACL3A31R)

アドレスオフセット : 0x94C

リセット値 : 0x0000 0000

レイヤ 3 アドレス 3 フィルタ 1 レジスタは、IPv4 パケットでは予約済みとなります。IPv6 パケットでは、128 ビットの IP ソースアドレスまたはデスティネーションアドレスフィールドのビット [127:96] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A31 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A31 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **L3A31[31:0]** : レイヤ 3 アドレス 3 フィールド

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3SAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP ソースアドレスフィールドのビット [127:96] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 および L3DAM1 ビットがセットされた場合、このフィールドは IPv6 パケットの IP デスティネーションアドレスフィールドのビット [127:96] と一致する値を含みます。

L3 および L4 制御 1 レジスタ (ETH_MACL3L4C1R) で L3PEN1 ビットがリセットされた場合、このフィールドは使用されません。

タイムスタンプ制御レジスタ (ETH_MACTSCR)

アドレスオフセット : 0x0B00

リセット値 : 0x0000 2000

このレジスタは、システム時間生成回路の動作およびレシーバのタイムスタンプの PTP パケットの処理を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	AV8021ASMEN	Res.	Res.	Res.	TXTSSTSM	Res.	Res.	Res.	Res.	Res.	TSENMACADDR	SNAPTYPSEL[1:0]	
			rw				rw						rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSMSTRENA	TSEVENTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDRREG	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **AV8021ASMEN** : AV 802.1AS モードの有効化

このビットがセットされた場合、MAC は PTP ステータスの提供とタイムスタンプスナップショットのキャプチャ、つまり IEEE 802.1AS 動作モードのために、タグなしの Ethernet 経由 PTP パケットだけを処理します。

PTP オフロードの目的で PTP オフロード機能が有効である場合、PTP ヘッダの転送固有フィールドが生成され、このビットの値に基づいてチェックされます。

ビット 27:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TXTSSTSM** : 送信タイムスタンプステータスモード

このビットがセットされた場合、MAC はソフトウェアによって読み出されない場合でも、以前の送信タイムスタンプステータスを上書きします。MAC は、[Tx タイムスタンプステータスナノ秒レジスタ \(ETH_MACTXTSSNR\)](#) レジスタの TXTSSMIS ビットをセットして示します。

このビットがリセットされると、MAC はソフトウェアによって前のパケットのタイムスタンプステータスが読み出されない場合、現在のタイムスタンプステータスを無視します。MAC は、[Tx タイムスタンプステータスナノ秒レジスタ \(ETH_MACTXTSSNR\)](#) の TXTSSMIS ビットをセットして示します。

ビット 23:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TSENMACADDR** : PTP パケットフィルタ用の MAC アドレスの有効化

このビットがセットされているとき、PTP が Ethernet 経由で直接送信される場合、(MAC アドレスレジスタと一致する) DA MAC アドレスは PTP パケットをフィルタするために使用されます。

このビットがセットされているとき、PTP が Ethernet 経由で直接送信される場合、MAC アドレスレジスタでプログラムされたものと一致する特別なマルチキャストまたはユニキャストアドレスを含む DA を持って受信される PTP パケットは、以下に示す処理を考慮します。

通常のタイムスタンプ動作の場合、MAC アドレスレジスタ 0 ~ 31 はユニキャストデスティネーションアドレスの一致で考慮されます。

PTP オフロードの場合、MAC アドレスレジスタ 0 だけがユニキャストデスティネーションアドレスの一致で考慮されます。

ビット 17:16 **SNAPTYPSEL[1:0]** : スナップショット取得のための PTP パケットの選択

これらのビットは、ビット 15 および 14 とともに、スナップショットを取得する必要がある PTP パケットタイプのセットを定義します。エンコードは表 635 : ETH_MACTSCR ビットへのタイムスタンプスナップショットの依存に示します。

ビット 15 **TSMSTRENA** : マスタに関係するメッセージのスナップショットの有効化

このビットがセットされている場合、マスタノードに関係するメッセージのみスナップショットが取られます。セットされていない場合は、スレーブノードに関係するメッセージのみスナップショットが取られます。

ビット 14 **TSEVNTENA** : イベントメッセージのタイムスタンプスナップショットの有効化

このビットがセットされた場合、イベントメッセージ (SYNC、Delay_Req、Pdelay_Req または Pdelay_Resp) に対してのみタイムスタンプスナップショットが取られます。このビットがリセットされた場合は、アナウンス、管理、シグナリング以外の全メッセージのスナップショットが取られます。タイムスタンプのスナップショットについては、表 635 : ETH_MACTSCR ビットへのタイムスタンプスナップショットの依存を参照してください。

ビット 13 **TSIPV4ENA** : IPv4-UDP 経由で送信された PTP パケットの処理の有効化

このビットがセットされた場合、MAC レシーバは IPv4-UDP パケットにカプセル化された PTP パケットを処理します。このビットがリセットされた場合、MAC は IPv4-UDP 経由で転送された PTP パケットを無視します。このビットはデフォルトでセットされます。

ビット 12 **TSIPV6ENA** : IPv6-UDP 経由で送信された PTP パケットの処理の有効化

このビットがセットされた場合、MAC レシーバは IPv6-UDP パケットにカプセル化された PTP パケットを処理します。このビットがクリアされた場合、MAC は IPv6-UDP 経由で転送された PTP パケットを無視します。

ビット 11 **TSIPENA** : Ethernet 経由 PTP パケットの処理の有効化

このビットがセットされた場合、MAC レシーバは Ethernet パケットに直接カプセル化された PTP パケットを処理します。このビットがリセットされた場合、MAC は Ethernet 経由 PTP パケットを無視します。

ビット 10 **TSVER2ENA** : バージョン 2 フォーマット用 PTP パケット処理の有効化

このビットがセットされた場合、IEEE 1588 バージョン 2 フォーマットを使用して PTP パケットを処理します。このビットがリセットされた場合、IEEE 1588 バージョン 1 フォーマットを使用して PTP パケットを処理します。IEEE 1588 フォーマットは、「PTP の処理および制御」で説明しています。

ビット 9 **TSCTRLSSR** : タイムスタンプデジタルまたはバイナリロールオーバー制御

このビットがセットされた場合、0x3B9A_C9FF の値の後 (1 ナノ秒精度) にタイムスタンプ下位レジスタがロールオーバーし、タイムスタンプ (ハイ) セカンドをインクリメントさせます。このビットがリセットされた場合、サブセカンドレジスタのロールオーバー値は 0x7FFF_FFFF です。サブセカンドのインクリメントは、PTP のリファレンスクロック周波数とこのビットの値に応じて正しくプログラムされる必要があります。

ビット 8 **TSENALL** : すべてのパケットに対するタイムスタンプの有効化

このビットがセットされた場合、MAC が受信する全パケットに対してタイムスタンプスナップショットが有効化されます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TSADDREG** : 加数レジスタの更新

このビットがセットされた場合、タイムスタンプ加数レジスタの内容が、微調整のために PTP ブロックで更新されます。このビットは、更新が完了するとクリアされます。このビットはセット前に 0 にする必要があります。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 TSUPDT : タイムスタンプの更新

このビットがセットされた場合、システム時間は、[システム時間秒更新レジスタ \(ETH_MACSTSUR\)](#) および [システム時間ナノ秒更新レジスタ \(ETH_MACSTNUR\)](#) レジスタで指定された値で更新されます (加算または減算されます)。

このビットは更新前に 0 にする必要があります。このビットは、ハードウェアで更新が完了するとリセットされます。

ビット 2 TSINIT : タイムスタンプの初期化

このビットがセットされた場合、システム時間は、[システム時間秒更新レジスタ \(ETH_MACSTSUR\)](#) および [システム時間ナノ秒更新レジスタ \(ETH_MACSTNUR\)](#) レジスタで指定された値で初期されます (上書きされます)。

このビットは更新前に 0 にする必要があります。このビットは、初期化が完了するとリセットされます。

ビット 1 TSCFUPDT : 粗密タイムスタンプ更新

このビットがセットされた場合、密更新法を使用してシステムタイムスタンプが更新されます。このビットがリセットされた場合、粗更新法を使用してシステムタイムスタンプが更新されます。

ビット 0 TSENA : タイムスタンプの有効化

このビットがセットされると、送信および受信パケットにタイムスタンプが追加されます。無効化された場合、送信および受信パケットにタイムスタンプは追加されず、タイムスタンプ生成回路もサスペンドされます。このモードを有効にした後、タイムスタンプ (システム時間) を初期化する必要があります。受信側では、このビットがセットされた場合のみ、MAC は 1588 パケットを処理します。

サブセカンドインクリメントレジスタ (ETH_MACSSIR)

アドレスオフセット : 0x0B04

リセット値 : 0x0000 0000

粗更新モードでは ([タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) の TSCFUPDT ビット)、clk_ptp_ref_i クロックサイクルごとに、このレジスタの値がシステム時間に加算されます。密更新モードでは、加算器がオーバーフローするたびに、このレジスタの値がシステム時間に加算されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 SSINC[7:0] : サブセカンドインクリメント値

このフィールドにプログラムされる値は、サブセカンドレジスタの内容で (clk_ptp_i の) 各クロックサイクルごとに累積されます。たとえば、PTP クロックが 50 MHz (周期は 20 ns) であるとき、システム時間ナノ秒レジスタに 1 ns の精度がある場合 ([タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) で TSCTRLSSR ビットがセットされている場合)、20 (0x14) をプログラムする必要があります。TSCTRLSSR がクリアされた場合、ナノ秒レジスタの分解能は ~0.465 ns です。この場合、20 ns/0.465 を基にした 43 (0x2B) の値をプログラムする必要があります。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

システム時間秒レジスタ (ETH_MACSTSR)

アドレスオフセット : 0x0B08

リセット値 : 0x0000 0000

システム時間秒レジスタは、システム時間ナノ秒レジスタとともに、MAC によって維持されているシステム時間の現在値を示します。これは連続的に更新されますが、クロックドメイン転送レイテンシ (clk_ptp_ref_i から CSR へ) によって実際の時間から少し遅延します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 TSS[31:0] : タイムスタンプの秒

このフィールドの値は、MAC によって維持されているシステム時間の現在値を秒で示します。

システム時間ナノ秒レジスタ (ETH_MACSTNR)

アドレスオフセット : 0x0B0C

リセット値 : 0x0000 0000

システム時間ナノ秒レジスタは、システム時間秒レジスタとともに、MAC によって維持されているシステム時間の現在値を示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TSSS[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:0 TSSS[30:0] : タイムスタンプサブセカンド

このフィールドの値は、0.46 ns 精度のサブセカンド表示です。[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) で TCTRLSSR がセットされた場合、各ビットは 1 ns を表します。最大値は 0x3B9A_C9FF で、これ以降は 0 にロールオーバーします。

システム時間秒更新レジスタ (ETH_MACSTSUR)

アドレスオフセット : 0x0B10

リセット値 : 0x0000 0000

システム時間秒更新レジスタは、システム時間ナノ秒更新レジスタとともに、MAC によって維持されているシステム時間を初期化または更新します。[タイムスタンプ制御レジスタ \(ETH_MACTSCR\)](#) で TSINIT または TSUPDT ビットをセットする前に、両方のレジスタを書き込む必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 TSS[31:0] : タイムスタンプの秒

このフィールドの値は、更新の秒部分です。

ADDSUB がセットされている場合は、このフィールドは更新値の秒部分によりプログラムする必要があります。

ADDSUB がセットされた場合、このフィールドには更新値の秒部分の補数をプログラムする必要があります。

たとえば、システム時間から 2.000000001 秒を引くには、ETH_MACSTSUR レジスタの TSS フィールドを 0xFFFF_FFFE (つまり、 $2^{32}-2$) にする必要があります。

注意 : **ADDSUB ビットがセットされている場合、システム時間ナノ秒更新レジスタ (ETH_MACSTNUR) で TSS[30:0] フィールドを 0 にセットすることはできません。0 ns を引く場合でも、TSS ビットフィールドには 0x7FFF_FFFF (結果は -0.46 ns) をプログラムする必要があります。**
 たとえば、システム時間から 2.000000000 秒を引くには、**システム時間秒更新レジスタ (ETH_MACSTSUR) の TSS フィールドを 0xFFFF_FFFE (つまり、 $2^{32}-1$) とし、システム時間ナノ秒更新レジスタ (ETH_MACSTNUR) を 0xFFFF_FFFF (ADDSUB = 1 で TSS[30:0] フィールド = 0x7FFF_FFFF) とする必要があります。**

システム時間ナノ秒更新レジスタ (ETH_MACSTNUR)

アドレスオフセット : 0x0B14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD SUB	TSS[30:16]														
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 ADDSUB : 時間の加算または減算

このビットがセットされた場合、時間値は更新レジスタの内容で減算されます。このビットがリセットされた場合、時間値は更新レジスタの内容で加算されます。

ビット 30:0 **TSSS[30:0]** : タイムスタンプサブセカンド

このフィールドの値は、更新のサブセカンド部分です。

- ADDSUB が 1 : このフィールドにはここに示す更新値のサブセカンド部分の補数をプログラムする必要があります。
- ADDSUB が 0 : このフィールドには**タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCTRLSSR ビットに基づいた精度で、更新値のサブセカンド部分をプログラムする必要があります。
- **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCTRLSSR フィールドが 1 :
 - プログラムされた値は、 $10^9 - \text{<サブセカンド値>}$ でなければなりません。
 - 各ビットは、1 ns を示し、プログラムされた値は 0x3B9A_C9FF を超えてはなりません。
- **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCTRLSSR フィールドが 0 :
 - プログラムされた値は、 $2^{31} - \text{<サブセカンド値>}$ でなければなりません。
 - 各ビットは 0.46 ns の精度を表します。

たとえば、システム時間から 2.000000001 秒を引くには、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCTRLSSR ビットがリセットされている場合は ETH_MACSTNUR レジスタの TSSS フィールドが 0x7FFF_FFFF (つまり、 $2^{31} - 1$) でなければならず、**タイムスタンプ制御レジスタ (ETH_MACTSCR)** で TSCTRLSSR ビットがセットされている場合は 0x3B9A_C9FF (つまり、 $10^9 - 1$) でなければなりません。

注意 : ADDSUB ビットがセットされている場合、TSSS[30:0] フィールドを 0 にセットすることはできません。0 ns を引く場合でも、TSSS ビットフィールドには 0x7FFF_FFFF (結果は -0.46 ns) をプログラムする必要があります。
 たとえば、システム時間から 2.000000000 秒を引くには、**システム時間ナノ秒更新レジスタ (ETH_MACSTNUR)** は 0xFFFF_FFFF (ADDSUB = 1 で TSSS[30:0] = 0) でなければならず、**システム時間秒更新レジスタ (ETH_MACSTSUR)** の TSS フィールドは 0xFFFF_FFFE (つまり、 $2^{32} - 1$) でなければなりません。

タイムスタンプ加数レジスタ (ETH_MACTSAR)

アドレスオフセット : 0x0B18

リセット値 : 0x0000 0000

このレジスタ値は、システム時間が密更新モード (**タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TSCFUPDT ビット) で設定されているときにのみ使用されます。このレジスタの内容は、clk_ptp_ref_i のクロックサイクルごとに 32 ビットの加算器に加算され、加算器がオーバーフローするたびにシステム時間が更新されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **TSAR[31:0]** : タイムスタンプ加数レジスタ

このフィールドは、時間同期を達成するために、加算器レジスタに加算される 32 ビットの時間値を示します。

タイムスタンプステータスレジスタ (ETH_MACTSSR)

アドレスオフセット : 0x0B20

リセット値 : 0x0000 0000

アプリケーションがこのレジスタを読み出す場合、ビット [27:25] を除くすべてのビットがクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ATSNS[4:0]					ATSS TM	Res.	Res.	Res.	Res.	ATSSTN[3:0]			
		r	r	r	r	r	rc_r					rc_r	rc_r	rc_r	rc_r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXT SSIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRG TERR0	AUXTS TRIG	TSTAR GT0	TSSOV F
rc_r												rc_r	rc_r	rc_r	rc_r

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:25 **ATSNS[4:0]** : 補助タイムスタンプスナップショットの数

このフィールドは、FIFO で使用できるスナップショットの数を示します。FIFO の深さに等しい値 (4) は、補助スナップショット FIFO がフルであることを示します。これらのビットは、補助スナップショット FIFO クリアビットがセットされたときに (00000 に) クリアされます。

ビット 24 **ATSSTM** : 補助タイムスタンプスナップショットトリガの見逃し

補助タイムスタンプスナップショット FIFO がフルで、外部トリガがセットされたときに、このビットがセットされます。これは、最新のスナップショットが FIFO に保存されないことを示します。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **ATSSTN[3:0]** : 補助タイムスタンプスナップショットトリガの識別子

これらのビットは、補助スナップショットレジスタで使用するタイムスタンプが適用できる補助トリガ入力を識別します。複数のビットが同時にセットされた場合、対応する補助トリガが同じクロックでサンプリングされたことを示します。これらのビットは、補助スナップショットの数が複数である場合のみ適用できます。1 ビットは次のリストに示すとおり各トリガに割り当てられます。

ビット 16 : 補助トリガ 0

ビット 17 : 補助トリガ 1

ビット 18 : 補助トリガ 2

ビット 19 : 補助トリガ 3

ソフトウェアはこのレジスタを読み出して、タイムスタンプが取得されたときにセットされるトリガを検索できます。

ビット 15 **TXTSSIS** : Tx タイムスタンプステータス割込みステータス

ドロップ送信ステータスが MTL で有効になっている場合、キャプチャされた送信タイムスタンプが **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** および **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** で更新されると、このビットがセットされます。

PTP オフロード機能が有効になっている場合、キャプチャされた送信タイムスタンプが **Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)** および **Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** で更新されたとき、PTO 生成遅延リクエストおよび Pdelay リクエストパケットに対して、このビットがセットされます。

このビットは、**Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)** が読み出される (または **CSR ソフトウェア制御レジスタ (ETH_MACCSRWCRCR)** で RCWE ビットがセットされた状態で、1 に書き込まれる) ときにクリアされます。

ビット 14:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TSTRGTERR0** : タイムスタンプ目標時間エラー

このビットは、ETH_MACPPSTTSR および ETH_MACPPSTTSNR でプログラムした最新の目標時間が経過したときにセットされます (**PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)** を参照)。このビットは、このアプリケーションでこのビットを読み出すとき (または、**CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされている場合は 1 を書き込むとき) クリアされます。

ビット 2 **AUXSTRIG** : 補助タイムスタンプトリガスナップショット

このビットは、補助スナップショットが FIFO に書き込まれたときにハイにセットされます。
このビットは、このアプリケーションでこのビットを読み出すとき (または、**CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされている場合は 1 を書き込むとき) クリアされます。

ビット 1 **TSTARGET0** : タイムスタンプ目標時間到達

セットされている場合、このビットは、システム時間の値が ETH_MACPPSTTSR および ETH_MACPPSTTSNR レジスタで指定される値以上であることを示します (**PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)** および **PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)** を参照)。
このビットは、このアプリケーションでこのビットを読み出すとき (または、**CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされている場合は 1 を書き込むとき) クリアされます。

ビット 0 **TSSOVF** : タイムスタンプの秒オーバーフロー

このビットがセットされた場合、タイムスタンプの秒の値 (バージョン 2 フォーマットをサポートしている場合) が 32'hFFFF_FFFF を超えてオーバーフローしたことを示します。
このビットは、このアプリケーションでこのビットを読み出すとき (または、**CSR ソフトウェア制御レジスタ (ETH_MACCSRSWCR)** で RCWE ビットがセットされている場合は 1 を書き込むとき) クリアされます。

Tx タイムスタンプステータスナノ秒レジスタ (ETH_MACTXTSSNR)

アドレスオフセット : 0x0B30

リセット値 : 0x0000 0000

このレジスタは、Tx ステータスが無効化されたときに送信パケットにキャプチャされたタイムスタンプのナノ秒部分を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSS MIS	TXTSSLO[30:16]														
	r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	TXTSSLO[15:0]														
	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

ビット 31 **TXSSMIS** : 送信タイムスタンプステータスの見逃し

このビットがセットされた場合、以下のいずれかを示します。

- **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TXSSTSM ビットがリセットされた場合、現在のパケットのタイムスタンプは無視されます。
- **タイムスタンプ制御レジスタ (ETH_MACTSCR)** の TXSSTSM がセットされた場合、前のパケットのタイムスタンプが現在のパケットのタイムスタンプで上書きされます。

ビット 30:0 **TXSSLO[30:0]** : 送信タイムスタンプステータスロー

このフィールドは、送信パケットのキャプチャされたタイムスタンプのナノ秒フィールドの 31 ビットを含みます。

Tx タイムスタンプステータス秒レジスタ (ETH_MACTXTSSSR)

アドレスオフセット : 0x0B34

リセット値 : 0x0000 0000

このレジスタは PTP パケットが送信されるときにキャプチャされるタイムスタンプ (秒単位) の上位 32 ビットを含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **TXTSSHI[31:0]** : 送信タイムスタンプステータスハイ

このフィールドは、送信パケットのキャプチャされたタイムスタンプの秒フィールドの下位 32 ビットを含みます。

補助制御レジスタ (ETH_MACACR)

アドレスオフセット : 0x0B40

リセット値 : 0x0000 0000

補助タイムスタンプ制御レジスタは、補助タイムスタンプスナップショットを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEN3	ATSEN2	ATSEN1	ATSEN0	Res.	Res.	Res.	ATSFC
								rw	rw	rw	rw				rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **ATSEN3** : 補助スナップショット 3 有効化

- このビットは、補助スナップショットトリガ 3 のキャプチャを制御します。このビットがセットされた場合、eth_ptp_trg3 入力 入力でのイベントの補助スナップショットが有効化されます。このビットがリセットされた場合、この入力でのイベントは無視されます。

ビット 6 **ATSEN2** : 補助スナップショット 2 有効化

- このビットは、補助スナップショットトリガ 2 のキャプチャを制御します。このビットがセットされた場合、eth_ptp_trg2 入力 入力でのイベントの補助スナップショットが有効化されます。このビットがリセットされた場合、この入力でのイベントは無視されます。

ビット 5 **ATSEN1** : 補助スナップショット 1 有効化

- このビットは、補助スナップショットトリガ 1 のキャプチャを制御します。このビットがセットされた場合、eth_ptp_trg1 入力 入力でのイベントの補助スナップショットが有効化されます。このビットがリセットされた場合、この入力でのイベントは無視されます。

ビット 4 **ATSEN0** : 補助スナップショット 0 有効化

このビットは、補助スナップショットトリガ 0 のキャプチャを制御します。このビットがセットされた場合、eth_ptp_trg0 入力 入力でのイベントの補助スナップショットが有効化されます。このビットがリセットされた場合、この入力でのイベントは無視されます。

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **ATSFC** : 補助スナップショット FIFO クリア

セットされると、このビットは補助スナップショット FIFO のポインタをリセットします。このビットは、ポインタがリセットされ FIFO がエンプティになったときにクリアされます。このビットがハイになった場合、補助スナップショットが FIFO に保存されます。

補助タイムスタンプナノ秒レジスタ (ETH_MACATSNR)

アドレスオフセット : 0x0B48

リセット値 : 0x0000 0000

補助タイムスタンプナノ秒レジスタ (ETH_MACATSNR) は、補助タイムスタンプ秒レジスタ (ETH_MACATSSR) とともに補助スナップショットとして保存される 64 ビットタイムスタンプを提供します。これら 2 つのレジスタは、4 ワードの深さを持つ 64 ビット幅 FIFO の読出しポートを形成します。

この FIFO には複数のスナップショットを保存できます。タイムスタンプステータスレジスタ (ETH_MACTSSR) のビット [29:25] は、FIFO のフィルレベルを示します。FIFO の最上部は、補助タイムスタンプ秒レジスタ (ETH_MACATSSR) が読み出されるときのみ、削除されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	AUXTSLO[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSLO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:0 **AUXTSLO[30:0]** : 補助タイムスタンプ

補助タイムスタンプの下位 31 ビット (ナノ秒フィールド) を含みます。

補助タイムスタンプ秒レジスタ (ETH_MACATSSR)

アドレスオフセット : 0x0B4C

リセット値 : 0x0000 0000

補助タイムスタンプ秒レジスタは、補助タイムスタンプレジスタの秒フィールドの下位 32 ビットを含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUXTSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **AUXTSHI[31:0]** : 補助タイムスタンプ

補助タイムスタンプの秒フィールドの下位 32 ビットを含みます。

タイムスタンプ入力非対称補正レジスタ (ETH_MACTSIACR)

アドレスオフセット : 0x0B50

リセット値 : 0x0000 0000

MAC タイムスタンプ入力非対称補正レジスタは、PDelay_Resp PTP メッセージの補正フィールドを更新する際に使用する入力非対称補正值を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTIAC[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTIAC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **OSTIAC[31:0]** : ワンステップタイムスタンプ入力非対称補正

このフィールドは、PDelay_Resp PTP パケットの correctionField に追加する入力パス非対称値を含みます。プログラムされる値はナノ秒単位とし、 2^{16} で掛ける必要があります。たとえば、2.5 ns は 0x00028000 で表されます。

この値は負にもでき、符号ビットを表すビット 31 で 2 の補数形式で表わすことができます。

タイムスタンプ出力非対称補正レジスタ (ETH_MACTSEACR)

アドレスオフセット : 0x0B54

リセット値 : 0x0000 0000

MAC タイムスタンプ出力非対称補正レジスタは、PDelay_Req PTP メッセージの補正フィールドを更新する際に使用する出力非対称補正值を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTEAC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTEAC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **OSTEAC[31:0]** : ワンステップタイムスタンプ出力非対称補正

このフィールドは、Pdelay_Resp PTP パケットの correctionField から差し引かれる出力パス非対称値を含みます。プログラムされる値は 2^{16} で掛けられたナノ秒単位のネゲートされた値にする必要があります。

たとえば、必要な補正值が +2.5 ns の場合、プログラムされる値は 0xFFFFD_8000 でなければならず、0x0002_8000 ($2.5 * 2^{16}$) という 2 の補数となります。同様に、必要な補正值が -3.3 ns の場合、プログラムされる値は 0x0003_4CCC ($3.3 * 2^{16}$) となります。

タイムスタンプ入力補正ナノ秒レジスタ (ETH_MACTSICNR)

アドレスオフセット : 0x0B58

リセット値 : 0x0000 0000

このレジスタは、入力パスでキャプチャされたタイムスタンプ値とともに使用されるナノ秒単位の補正值を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSIC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **TSIC[31:0]** : タイムスタンプ入力補正

このフィールドは、入力補正の式によって定義される入力パス補正值を含みます。

タイムスタンプ出力補正ナノ秒レジスタ (ETH_MACTSECNR)

アドレスオフセット : 0x0B5C

リセット値 : 0x0000 0000

このレジスタは、出力パスでキャプチャされたタイムスタンプ値とともに使用されるナノ秒単位の補正値を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSEC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSEC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 TSEC[31:0] : タイムスタンプ出力補正

このフィールドは、出力補正の式によって定義される出力パス補正値のナノ秒部分を含みます。

PPS 制御レジスタ (ETH_MACPPSCR)

アドレスオフセット : 0x0B70

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0 [1:0]		PPSEN 0	PPSCTRL[3:0]			
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:5 TRGTMODSEL0[1:0] : PPS 出力の目標時間レジスタモード

このフィールドは、PPS 出力信号の目標時間レジスタ (PPS 目標時間秒レジスタ (ETH_MACPPSTTSR) および PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTTNR)) モードを示します。

00 : 目標時間レジスタは、割込みイベントを生成するためだけにプログラムされます。

01 : 予約済み。使用できません。

10 : 目標時間レジスタは、割込みイベントを生成して、PPS 出力信号生成を開始または停止するためにプログラムされます。

11 : 目標時間レジスタは、PPS 出力信号生成を開始または停止するためだけにプログラムされます。割込みはアサートされません。

ビット 4 PPSSEN0 : フレキシブル PPS 出力モード有効化

このビットがセットされた場合、PPSCTRL[3:0] が PPSCMD[3:0] として機能します。このビットがリセットされた場合、PPSCTRL[3:0] が PPSCTRL (固定 PPS モード) として機能します。

ビット 3:0 **PPSCTRL[3:0]** : PPS 出力周波数制御

このレジスタは、PPS 出力 (eth_ptp_pps_out) 信号の周波数を制御します。PPSCTRL のデフォルト値は 0000 で、PPS 出力は毎秒 1 パルス (width clk_ptp_i) です。PPSCTRL のその他の値では、PPS 出力は次の周波数のクロック生成となります。

0001 : バイナリロールオーバーは 2 Hz で、デジタルロールオーバーは 1 Hz です。

0010 : バイナリロールオーバーは 4 Hz で、デジタルロールオーバーは 2 Hz です。

0011 : バイナリロールオーバーは 8 Hz で、デジタルロールオーバーは 4 Hz です。

0100 : バイナリロールオーバーは 16 Hz で、デジタルロールオーバーは 8 Hz です。

..

1111 : バイナリロールオーバーは 32.768 KHz で、デジタルロールオーバーは 16.384 KHz です。

注 : バイナリロールオーバーモードでは、PPS 出力 (eth_ptp_pps_out) は、これらの周波数で 50% のデューティサイクルとなります。バイナリロールオーバーモードでは、PPS 出力周波数は平均数です。実際のクロックは、毎秒同期される異なった周波数となります。例 :

- PPSCTRL = 0001 の場合、PPS (1 Hz) には、537 ms のロー周期と、463 ms のハイ周期があります。
- PPSCTRL = 0010 の場合、PPS (2 Hz) は、50% のデューティサイクルの一連の 1 クロックと、537 ms の周期となります。463 ms 周期の 2 番目のクロック (268 ms ローおよび 195 ms ハイ)
- PPSCTRL = 0011 の場合、PPS (4 Hz) は、50% のデューティサイクルの一連の 3 クロックと、268 ms の周期となります。195 ms 周期の 4 番目のクロック (134 ms ローおよび 61 ms ハイ)

この動作は、ETH_MACSTNR レジスタのデジタルロールオーバーモードでのビットのノンリニア切り替えによるためです。

PPS 制御レジスタ[オルタネート] (ETH_MACPPSCR)

アドレスオフセット : 0x0B70

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSELO [1:0]		PPSEN 0	PPSCMD[3:0]			
									r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:5 **TRGTMODSELO[1:0]** : PPS 出力の目標時間レジスタモード

このフィールドは、PPS 出力信号の目標時間レジスタ (PPS 目標時間秒レジスタ (ETH_MACPPSTTSR) および PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTNTR)) モードを示します。

00 : 目標時間レジスタは、割込みイベントを生成するためだけにプログラムされます。

01 : 予約済み。使用できません。

10 : 目標時間レジスタは、割込みイベントを生成して、PPS 出力信号生成を開始または停止するためにプログラムされます。

11 : 目標時間レジスタは、PPS 出力信号生成を開始または停止するためだけにプログラムされます。割込みはアサートされません。

ビット 4 **PPSEN0** : フレキシブル PPS 出力モードの有効化

このビットがセットされた場合、ビット [3:0] が PPSCMD[3:0] として機能します。このビットがリセットされた場合、ビット [3:0] が PPSCTRL (固定 PPS モード) として機能します。

ビット 3:0 **PPSCMD[3:0]** : フレキシブル PPS 出力 (eth_ptp_pps_out) 制御

これらのビットを 0 以外の値にプログラムして、MAC にイベントを開始するよう命令します。コマンドが PTP クロックドメインに転送または同期されるときに、これらのビットは自動的にクリアされます。ソフトウェアは、これらのビットが「すべて 0」である場合のみプログラムされることを確認する必要があります。次のリストでは、PPSCMD0 の値を示しています。

0000 : コマンドなし

0001 : シングルパルス開始。

このコマンドは、目標時間レジスタ (レジスタ 455 および 456) で定義された開始位置および PPS 幅レジスタで定義された時間の開始位置で立ち上がるシングルパルスを生成します。

0010 : パルス列開始。

このコマンドは、目標時間レジスタで定義された開始位置および PPS 幅レジスタで定義された時間の開始位置で立ち上がり、PPS 間隔レジスタで定義された間隔で繰り返されるパルスの列を生成します。デフォルトでは、PPS パルス列は「パルス列の同時停止」または「パルス列の即時停止」コマンドによって停止されない限り、フリーランニングです。

0011 : 開始のキャンセル。

このコマンドは、システム時間がプログラムされた開始時間を超えなかった場合に、シングルパルス開始およびパルス列開始コマンドをキャンセルします。

0100 : パルス列の同時停止。

このコマンドは、目標時間レジスタでプログラムされた時間が経過した後にパルス列開始コマンド (PPSCMD[3:0] = 0010) で開始されたパルス列を停止します。

0101 : パルス列の即時停止。

このコマンドは、パルス列開始コマンド (PPSCMD[3:0] = 0010) で開始されたパルス列をただちに停止します。

0110 : パルス列停止のキャンセル。

このコマンドは、プログラムされた停止時間が経過していない場合に、パルス列の同時停止のコマンドをキャンセルします。PPS パルス列は、このコマンドが正常に実行されるとフリーランニングになります。

0111 から 1111 : 予約済み。使用できません。

PPS 目標時間秒レジスタ (ETH_MACPPSTTSR)

アドレスオフセット : 0x0B80

リセット値 : 0x0000 0000

PPS 出力目標時間秒レジスタは、[PPS 目標時間ナノ秒レジスタ \(ETH_MACPPSTTNR\)](#) とともに、システム時間がこれらのレジスタでプログラムされた値を超えたときに割り込みイベント [[タイムスタンプステータスレジスタ \(ETH_MACTSSR\)](#)] のビット TSSOVF をスケジュールするために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSTRH0 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTRH0 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **TSTRH0[31:0]** : PPS 目標時間秒レジスタ

このフィールドは、時間を秒単位で格納します。タイムスタンプ値が両方の目標タイムスタンプレジスタと一致するか、それを超えた場合に、MAC は PPS 信号出力を開始または停止して、[PPS 制御レジスタ \(ETH_MACPPSCR\)](#) の対応する PPS 出力に選択された目標時間モードに基づいて割り込み (有効である場合) を生成します。

PPS 目標時間ナノ秒レジスタ (ETH_MACPPSTNR)

アドレスオフセット : 0x0B84

リセット値 : 0x0000 0000

PPS 目標時間ナノ秒レジスタは、複数のフレキシブル PPS 出力が選択されている場合のみ存在します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRGTBUSY0	TTSL0 [30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSL0 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **TRGTBUSY0** : PPS 目標時間レジスタビジー

MAC は、**PPS 制御レジスタ (ETH_MACPPSCR)** の PPSCMD0 フィールドが 010 か 011 にプログラムされている場合に、このビットをセットします。PPSCMD0 フィールドを 010 か 011 にプログラムして、MAC に目標時間レジスタを PTP クロックドメインと同期するよう命令します。

目標時間レジスタを PTP クロックドメインと同期した後に、MAC はこのビットをクリアします。このビットが 1 で読み出される場合、アプリケーションは目標時間レジスタを更新してはなりません。そうしないと、以前にプログラムされた時間の同期が破損します。

ビット 30:0 **TTSL0[30:0]** : PPS 出力の目標時間ロー

このレジスタは、時間値を (符号付き) ナノ秒単位で格納します。タイムスタンプの値が両方の目標タイムスタンプレジスタの値と一致する場合に、MAC は PPS 信号出力を開始または停止して、**PPS 制御レジスタ (ETH_MACPPSCR)** の TRGTMODSEL0 フィールド (ビット [6:5]) に基づいて割込み (有効である場合) を生成します。

TSCTRLSSR ビットが**タイムスタンプ制御レジスタ (ETH_MACTSCR)** でセットされている場合、この値は 0x3B9A_C9FF を超えてはなりません。PPS 信号出力の実際の開始または停止時間に、サブセカンドインクリメント値の 1 単位までのエラーマージンが発生する可能性があります。

PPS 間隔レジスタ (ETH_MACPPSIR)

アドレスオフセット : 0x0B88

リセット値 : 0x0000 0000

PPS 間隔レジスタは、PPS 出力の立上りエッジ間のサブセカンドインクリメント値の単位数を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSINT0 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSINT0 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **PPSINT0[31:0]** : PPS 出力信号間隔

これらのビットは、PPS 信号出力の立上りエッジ間の間隔を格納します。この間隔は、サブセカンドインクリメント値の単位数で格納されます。

必要な間隔未満の 1 つの値をプログラムする必要があります。たとえば、PTP リファレンスクロックが 50 MHz (20 ns の周期) であり、PPS 信号出力の立上りエッジの期待する間隔が 100 ns (サブセカンドインクリメント値の 5 単位) である場合、このレジスタで値 4 (5-1) をプログラムする必要があります。

PPS 幅レジスタ (ETH_MACPPSWR)

アドレスオフセット : 0x0B8C

リセット値 : 0x0000 0000

PPS 幅レジスタは、PPS 出力の立上りエッジと、対応する立下りエッジの間のサブセカンドインクリメント値の単位数を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSWIDTH0 [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSWIDTH0 [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **PPSWIDTH0[31:0]** : PPS 出力信号幅

これらのビットは、PPS 信号出力の立上りエッジと、対応する立下りエッジの間の幅を格納します。この幅は、セカンドインクリメント値の単位数で格納されます。

必要な間隔未満の 1 つの値をプログラムする必要があります。たとえば、PTP リファレンスクロックが 50 MHz (20 ns の周期) であり、PPS 信号出力の立上りエッジと対応する立下りエッジの間の幅が 80 ns (サブセカンドインクリメント値の 4 単位) である場合、このレジスタで値 3 (4-1) をプログラムする必要があります。

注 : このレジスタにプログラムされる値は、**PPS 間隔レジスタ (ETH_MACPPSIR)** にプログラムされる値未満にする必要があります。

PTP オフロード制御レジスタ (ETH_MACPOCR)

アドレスオフセット : 0x0BC0

リセット値 : 0x0000 0000

このレジスタは、PTP オフロードエンジン動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN[7:0]								Res.	DRRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNQEN	PTOEN
rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW		rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DN[7:0]** : ドメイン数

このフィールドは、PTP ノードが動作しているドメイン数を示します。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DRRDIS** : PTO 遅延リクエスト／レスポンス生成の無効化

このビットがセットされた場合、プログラムされたモードの必要に応じて、受信 SYNC および遅延リクエストパケットそれぞれに対して遅延リクエストおよび遅延レスポンスが生成されません。

- ビット 5 **APDREQTRIG** : 自動 PTP Pdelay_Req メッセージトリガ
このビットがセットされた場合、1 つの PTP Pdelay_Req メッセージが送信されます。このビットは PTP Pdelay_Req メッセージが送信された後に自動的にクリアされます。アプリケーションは、この動作に APDREQEN ビットをセットする必要があります。
- ビット 4 **ASYNCTRIG** : 自動 PTP SYNC メッセージトリガ
このビットがセットされた場合、1 つの PTP SYNC メッセージが送信されます。このビットは PTP SYNC メッセージが送信された後に自動的にクリアされます。アプリケーションは、この動作に ASYNCEN ビットをセットする必要があります。
- ビット 3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **APDREQEN** : 自動 PTP Pdelay_Req メッセージ有効化
このビットがセットされた場合、MAC がピアツーピアトランスペアレントモードになるようプログラムされているときに、プログラムされた間隔またはアプリケーションからのトリガに基づいて、PTP Pdelay_Req メッセージが周期的に生成されます。
- ビット 1 **ASYNCEN** : 自動 PTP SYNC メッセージ有効化
このビットがセットされた場合、MAC がクロックマスタモードになるようプログラムされているときに、プログラムされた間隔またはアプリケーションからのトリガに基づいて、PTP SYNC メッセージが周期的に生成されます。
- ビット 0 **PTOEN** : PTP オフロード有効化
このビットがセットされた場合、PTP オフロード機能が有効になります。

PTP ソースポートアイデンティティ 0 レジスタ (ETH_MACSPI0R)

アドレスオフセット : 0x0BC4

リセット値 : 0x0000 0000

このレジスタは、PTP ノードの 80 ビットソースポートアイデンティティのビット [31:0] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI0 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:0 **SPI0[31:0]** : ソースポートアイデンティティ 0
このフィールドは、PTP ノードの sourcePortIdentity のビット [31:0] を示します。



PTP ソースポートアイデンティティ 1 レジスタ (ETH_MACSPI1R)

アドレスオフセット : 0x0BC8

リセット値 : 0x0000 0000

このレジスタは、PTP ノードの 80 ビットソースポートアイデンティティのビット [63:32] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI1 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **SPI1[31:0]** : ソースポートアイデンティティ 1

このフィールドは、PTP ノードの sourcePortIdentity のビット [63:32] を示します。

PTP ソースポートアイデンティティ 2 レジスタ (ETH_MACSPI2R)

アドレスオフセット : 0x0BCC

リセット値 : 0x0000 0000

このレジスタは、PTP ノードの 80 ビットソースポートアイデンティティのビット [79:64] を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **SPI2[15:0]** : ソースポートアイデンティティ 2

このフィールドは、PTP ノードの sourcePortIdentity のビット [79:64] を示します。

ログメッセージ間隔レジスタ (ETH_MACLMIR)

アドレスオフセット : 0x0BD0

リセット値 : 0x0000 0000

このレジスタは自動 PTP パケット生成の周期的間隔を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMPDRI[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]			LSI[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **LMPDRI[7:0]** : ログ最小 Pdelay_Req 間隔

このフィールドは、PTP ノードの logMinPdelayReqInterval を示します。このビットは、周期的 Pdelay リクエストパケット送信のスケジュールに使用されます。許可される値は -15~15 です。負の値は 2 の補数形式で表す必要があります。たとえば、必要な値が -1 の場合、プログラムされる値は、0xFF でなければなりません。

ビット 23:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **DRSYNCR[2:0]** :

SYNC に対する Delay_Req の比率
スレーブモードでは、送信される Delay_Req メッセージの周波数制御に使用されます。
0 : DelayReq が受信 SYNC ごとに生成されます。
1 : DelayReq が SYNC のオルタネート受信ごとに生成されます。
2 : 4 個の SYNC メッセージごとに生成されます。
3 : 8 個の SYNC メッセージごとに生成されます。
4 : 16 個の SYNC メッセージごとに生成されます。
5 : 32 個の SYNC メッセージごとに生成されます。
その他 : 予約済み。使用できません。

マスタは、DelayResp PTP メッセージのこの情報 (logMinDelayReqInterval) をスレーブに送信します。受信時に、受信した DelayResp メッセージのこの値を処理し、それに応じてこのフィールドを更新します。スレーブモードでは、ホストは受信した値を上書きする必要がない限り、このレジスタの書き込み／更新を行ってはなりません。マスタモードでは、このフィールドと logSyncInterval (LSI) フィールドの合計が、生成されたマルチキャスト Delay_Resp PTP メッセージの logMinDelayReqInterval フィールドで提供されます。

ビット 7:0 **LSI[7:0]** :

ログ同期間隔
このフィールドは、PTP ノードがマスタであるときに自動生成された SYNC メッセージの周期性を示します。許可される値は -15~15 です。負の値は 2 の補数形式で表す必要があります。たとえば、必要な値が -1 の場合、プログラムされる値は、0xFF でなければなりません。

Ethernet MAC レジスタマップとリセット値

表 682. Ethernet MAC レジスタマップとリセット値

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0000	ETH_MACCR	ARPEN :	SARC[2:0]		IPC		IPG[2:0]		GPSLCE		S2KP	CST	ACS	WD	Res	JD	JE	Res	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res	BL[1:0]		DC		PRELEN[1:0]		TE	RE
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0004	ETH_MACECR	Res	Res	EIPG[4:0]				EIPGEN		Res	Res	Res	Res	Res	USP	SPEN	DCRCC	Res	Res	GPSL[13:0]														
	リセット値			0	0	0	0	0	0						0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	ETH_MACPFR	RA	Res	Res	Res	Res	Res	Res	Res	Res	Res	DNTU	IPFE	Res	Res	Res	VTFE	Res	Res	Res	Res	Res	Res	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
	リセット値	0										0	0				0						0	0	0	0	0	0	0	0	0	0	0	0
0x000C	ETH_MACWTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PWE	Res	Res	Res	Res	WTO[3:0]				
	リセット値																								0					0	0	0	0	0
0x0010	ETH_MACHT0R	HT31T0 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	ETH_MACHT1R	HT63T32 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018 - 0x004C	予約済みです。																																	
0x0050	ETH_MACVTR	EIVLRXS	Res	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	Res	EVL[S[1:0]		DOVLT	ERSVLM	ESVL	VTIM	ETV	VL[15:0]																
	リセット値	0		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	予約済みです。																																	
0x0058	ETH_MACVHTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	VLHT[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	予約済みです。																																	
0x0060	ETH_MACVIR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	VLT	CSV	VLP	VLC[1:0]	VLT[15:0]																	
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x0064	ETH_MACiVIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTi	CSVL	VLP	VLC[1:0]	VLT[15:0]																					
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0068 - 0x006C	予約済みです。																																					
0x0070	ETH_ MACQTXFCR	PT[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]		Res.	Res.	TFE	RPA								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0			0	0					
0x0074 - 0x008C	予約済みです。																																					
0x0090	ETH_MACRXFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UP	RFE						
	リセット値																														0	0						
0x0094- 0x00AC	予約済みです。																																					
0x00B0	ETH_MACISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIS	TXSTSIS	TSIS	MMCTXIS												LPIIS	PMTIS	PHYSIS	Res.	Res.	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x00B4	ETH_MACIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIE	TXSTSIE	TSIE	MMCTXIS												LPIE	PMTIE	PHYIE	Res.	Res.	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x00B8	ETH_ MACRXTXSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT						
	リセット値																							0			0	0	0	0	0	0	0	0	0			
0x00BC	予約済みです。																																					
0x00C0	ETH_MACPCSR	RWKFLTRST	Res.	Res.	RWKPTR[4:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWKPF	GLBLUCAS	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN					
	リセット値	0			0	0	0	0	0														0	0			0	0			0	0	0	0	0			
0x00C4	ETH_ MACRWKPFR	MACRWKPFR[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x00C8 - 0x00CC	予約済みです。																																					

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00D0	ETH_MACLCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPITE	LPITXA	Res.	PLS	LPIEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値											0	0	0		0	0								0	0					0	0	0	0
0x00D4	ETH_MACLTCR	Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]									TWT[15:0]																	
	リセット値							1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D8	ETH_MACLETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[19:0]																					
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00DC	ETH_ MAC1USTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIC_1US_CNTR [11:0]												
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0	
0x00E0 - 0x010C	予約済みです。																																	
0x0110	ETH_MACVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USERVER[7:0]							SNPSVER[7:0]										
	リセット値																0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	0	
0x0114	ETH_MACDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]	TPESTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]	RPESTS		
	リセット値														0	0	0													0	0	0		
0x0118	予約済みです。																																	
0x011C	ETH_MACHWF0R	Res.	ACTPHYSEL[2:0]			SAVLANINS		TSSTSSEL[1:0]		MACADR64SEL		MACADR32SEL		ADDMACADRSEL[4:0]				Res.	RXCOSSEL	Res.	TXCOSSEL	EESEL	TSSEL	Res.	Res.	ARPOFFSEL	MMCSSEL	MGKSEL	RWKSEL	SMASEL	VLHASH	PCSSEL	HDSEL	GMISEL
	リセット値		0	0	0	1	0	1	0	0	0	0	0	0	1	1		1		1	1	1			1	1	1	1	1	1	1	0	1	1
0x0120	ETH_MACHWF1R	Res.	L3L4FNUM [3:0]			Res.	HASHTBLSZ[1:0]		POUOST		Res.	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN	ADDR64 [1:0]		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE[4:0]				Res.	RXFIFOSIZE[4:0]						
	リセット値		0	0	1	0		0	1	0		0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0		0	0	1	0

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x0124	ETH_MACHWF2R	Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			TDCSZ[1:0]		TXCHCNT[3:0]			RDCSZ[1:0]		RXCHCNT[3:0]			Res.	Res.	TXQCNT[3:0]			Res.	Res.	RXQCNT[3:0]			0	0	0	0		
	リセット値		1	0	0			0	0	1	0	0	0	0	0	0	0	0	0			0	0	0			0	0	0					0	0
0x0128	ETH_MACHWF3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVLAN	CBTSEL	Res.	NRVFI[2:0]		0	0	0
	リセット値																										1	0	0		0	0			
0x012C - 0x01FC	予約済みです。																																		
0x0200	ETH_MACMDIOAR	Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]				RDA[4:0]				Res.	NTC[2:0]			CR[3:0]			Res.	Res.	Res.	SKAP	GOC[1:0]	C45E	MR	0	0	0	0		
	リセット値					0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0												0	0
0x0204	ETH_MACMDIODR	RA[15:0]															MD[15:0]																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0208 - 0x020C	予約済みです。																																		
0x0210	ETH_MACARPAR	ARPPA[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0214 - 0x022C	予約済みです。																																		
0x0230	ETH_MACCSRSW CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCWE		
	リセット値																							0								0			
0x0234 - 0x02FC	予約済みです。																																		
0x0300	ETH_MACA0HR	AE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
	リセット値	1																																	
0x0304	ETH_MACA0LR	ADDRLO[31:0]																																	
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0308	ETH_MACA1HR	AE	SA	MBC[5:0]					Res.					Res.					Res.					ADDRHI[15:0]									
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x030C	ETH_MACA1LR	ADDRLO[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0310	ETH_MACA2HR	AE	SA	MBC[5:0]					Res.					Res.					Res.					ADDRHI[15:0]									
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0314	ETH_MACA2LR	ADDRLO[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0318	ETH_MACA3HR	AE	SA	MBC[5:0]					Res.					Res.					Res.					ADDRHI[15:0]									
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x031C	ETH_MACA3LR	ADDRLO[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0320 - 0x06FC	予約済みです。																																
0x0700	ETH_MMC_ CONTROL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC										Res.	Res.	Res.	Res.	Res.	
	リセット値																	0															
0x0704	ETH_MMC_RX_ INTERRUPT	Res.	Res.	Res.	Res.	RXLPI	RXCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXLGN	RXCIS	Res.	Res.	Res.	Res.
	リセット値					0	0									0											0	0					
0x0708	ETH_MMC_TX_ INTERRUPT	Res.	Res.	Res.	Res.	TXLPIT	RXCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXMCO	TXSCOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値					0	0					0						0	0														

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x070C	ETH_MMCRX_INTERRUPT_MASK	Res.	Res.	Res.	Res.	RXLPIITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIM	RXCRCERPIM	Res.	Res.	Res.	Res.	Res.
	リセット値					0	0									0											0	0					
0x0710	ETH_MMCTX_INTERRUPT_MASK	Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.	TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値					0	0					0						0	0														
0x0714 - 0x0748	予約済みです。																																
0x074C	ETH_TX_SINGLE_COLLISION_GOOD_PACKETS	TXSNGLCOLG[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0750	ETH_TX_MULTIPLE_COLLISION_GOOD_PACKETS	TXMULTCOLG[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0754 - 0x0764	予約済みです。																																
0x0768	ETH_TX_PACKET_COUNT_GOOD	TXPKTG[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x076C - 0x0790	予約済みです。																																
0x0794	ETH_RX_CRC_ERROR_PACKETS	RXCRCERR[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0798	ETH_RX_ALIGNMENT_ERROR_PACKETS	RXALGNERR[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x079C - 0x07C0	予約済みです。																																

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x07C4	ETH_RX_UNICAST_PACKETS_GOOD	RXUCASTG[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C8 - 0x07E8	予約済みです。																																
0x07EC	ETH_TX_LPI_USEC_CNTR	TXLPIUSC[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F0	ETH_TX_LPI_TRAN_CNTR	TXLPITRC[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F4	ETH_RX_LPI_USEC_CNTR	RXLPIUSC[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F8	ETH_RX_LPI_TRAN_CNTR	RXLPITRC[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07FC - 0x08FC	予約済みです。																																
0x0900	ETH_MACL3L4C0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPM0	L4DPM0	L4SPM0	L4SPM0	Res.	L4PEN0	L3HDBM0 [4:0]				L3HSBM0 [4:0]				L3DAM0	L3DAM0	L3SAM0	L3SAM0	Res.	L3PEN0		
	リセット値											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0904	ETH_MACL4A0R	L4DP0 [15:0]															L4SP0 [15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0908 - 0x090C	予約済みです。																																
0x0910	ETH_MACL3A00R	L3A00 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0914	ETH_MACL3A10R	L3A10 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0918	ETH_MACL3A20R	L3A20 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x091C	ETH_MACL3A30R	L3A30 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0920 - 0x092C	予約済みです。																																			
0x0930	ETH_ MACL3L4C1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1			L3HDBM1 [4:0]					L3HSBM1 [4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1	
	リセット値											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0934	ETH_MACL4A1R	L4DP1 [15:0]															L4SP1 [15:0]																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0938 - 0x093C	予約済みです。																																			
0x0940	ETH_MACL3A01R	L3A01 [31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0944	ETH_MACL3A11R	L3A11 [31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0948	ETH_MACL3A21R	L3A21 [31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x094C	ETH_MACL3A31R	L3A31 [31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0950 - 0x0AFC	予約済みです。																																			
0x0B00	ETH_MACTSCR	Res.	Res.	Res.	AV8021ASMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSENMACADDR	SNAPTYPSEL[1:0]	TSMSTRENA	TSEVNTENA	TSPV4ENA	TSPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDRESS	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA			
	リセット値				0							0					0	0	0	1	0	0	0	0	0			0		0	0	0	0			
0x0B04	ETH_MACSSIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値									0	0	0	0	0	0	0	0																			
0x0B08	ETH_MACSTSR	TSS[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0B0C	ETH_MACSTNR	Res.	TSSS[30:0]																																	
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0B10	ETH_MACSTSUR	TSS[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x0B14	ETH_MACSTNUR	ADDSUB	TSSS[30:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B18	ETH_MACTSAR	TSAR[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B1C	予約済みです。																																					
0x0B20	ETH_MACTSSR	Res.	Res.	ATSNS[4:0]				ATSSTM		Res.	Res.	Res.	Res.	ATSSTN[3:0]			TXTSSIS		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRGTERR0	AUTSTRIG	TSTARGT0	TSSOVF					
	リセット値			0	0	0	0	0	0					0	0	0	0	0											0	0	0	0						
0x0B24 - 0x0B2C	予約済みです。																																					
0x0B30	ETH_MACTXTSSNR	TXTSSMIS	TXTSSLO[30:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B34	ETH_MACTXTSSSR	TXTSSHI[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B38 - 0x0B3C	予約済みです。																																					
0x0B40	ETH_MACACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEAC					
	リセット値																																0					
0x0B44	予約済みです。																																					
0x0B48	ETH_MACATSNR	Res.	AUXTSLO[30:0]																																			
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B4C	ETH_MACATSSR	AUXTSHI[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B50	ETH_MACTSIACR	OSTIAC[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0B54	ETH_MACTSEAC R	OSTEAC[31:0]																																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0B58	ETH_ MACTSICNR	TSIC[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B5C	ETH_ MACTSECNR	TSEC[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B60 - 0x0B6C	予約済みです。																																	
0x0B70	ETH_MACPPSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																	
0x0B70	ETH_MACPPSCR (オルタネート)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																	
0x0B74 - 0x0B7C	予約済みです。																																	
0x0B80	ETH_ MACPPSTTSR	TSTRH0 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B84	ETH_ MACPPSTTNR	TRGTBUSY0	TTSL0 [30:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B88	ETH_MACPPSIR	PPSINT0 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B8C	ETH_MACPPSWR	PPSWIDTH0 [31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0B90 - 0x0BBC	予約済みです。																																	

表 682. Ethernet MAC レジスタマップとリセット値 (続き)

オフ セット	レジスタ名と リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0BC0	ETH_MACPOCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DN[7:0]							Res.	Res.	DRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNCEEN	PTOEN
	リセット値																	0	0	0	0	0	0	0	0		0	0	0		0	0	0
0x0BC4	ETH_MACSPI0R	SPI0 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BC8	ETH_MACSPI1R	SPI1 [31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BCC	ETH_MACSPI2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI2 [15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BD0	ETH_MACLMIR	LMPDR[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]				LSI[7:0]						
	リセット値	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[110ページのセクション 2.3](#) を参照してください。

58 デバッグサポート (DBG)

58.1 概要

以下に示すデバッグの機能を統合させたもので、ソフトウェア開発およびシステムインテグレーションをサポートするために提供されます。

- CPU コアのブレークポイントのデバッグ
- コード実行トレース
- ソフトウェア計測
- クロストリガ

デバッグ機能は JTAG／シリアルワイヤデバッグ用のアクセスポートを介し、業界標準のデバッグツールを使用して制御することができます。トレース用のポートを使用すると、ログおよび分析のためにデータをキャプチャすることができます。

デバッグ機能は Arm® CoreSight™ コンポーネントに基づいています。

- SWJ-DP : JTAG／シリアルワイヤデバッグポート
- AHB-AP : AHB アクセスポート
- ROM テーブル
- システム制御空間 (SCS)
- ブレークポイントユニット (BPU)
- データウォッチポイントおよびトレースユニット (DWT)
- 計装トレースマクロセル (ITM)
- Embedded Trace Macrocell™ (ETM)
- クロストリガインタフェース (CTI)
- トレースポートインタフェースユニット (TPU)

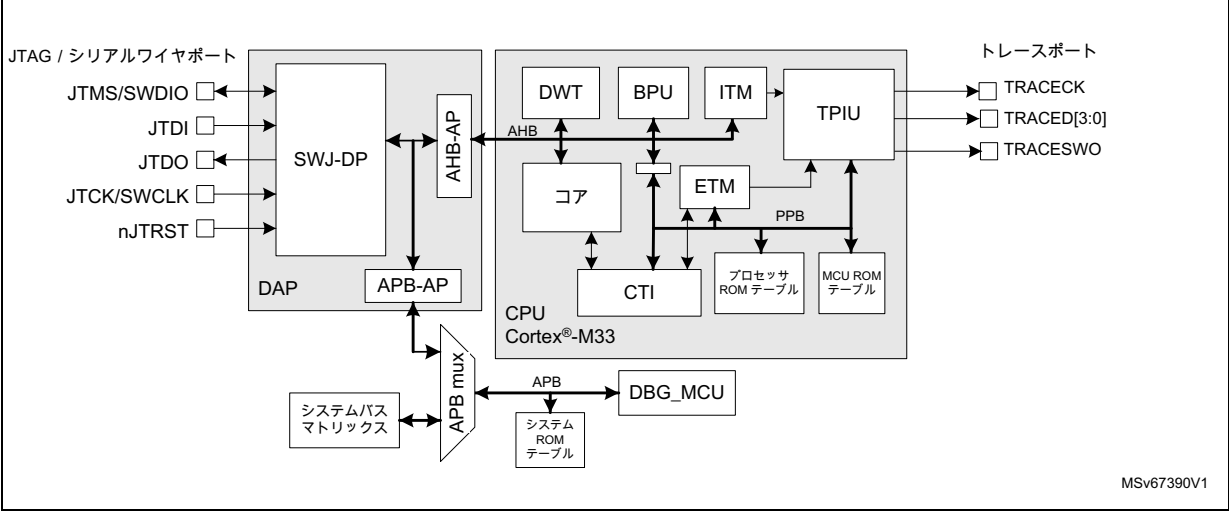
デバッグ機能には、デバッグによって AHB-AP 経由でアクセスできます。

追加情報は[セクション 58.13](#)に掲載されている Arm® の資料をご覧ください。

58.2 DBG の機能説明

58.2.1 DBG ブロック図

図 829. デバッグサポートインフラストラクチャのブロック図



58.2.2 DBG ピンおよび内部信号

表 683. JTAG／シリアルワイヤデバッグポートのピン

ピン名	JTAG デバッグポート		SW デバッグポート		ピン割り当て
	タイプ	説明	タイプ	説明	
JTMS／SWDIO	I	JTAG テストモード選択	IO	シリアルワイヤデータ入出力	PA13
JTCK／SWCLK	I	JTAG テストクロック	I	シリアルワイヤクロック	PA14
JTDI ⁽¹⁾	I	JTAG テストデータ入力	-	-	PA15
JTDO	O	JTAG テストデータ出力	-	-	PB3
nJTRST	I	JTAG テストリセット	-	-	PB4

1. TDI は、同じ IO 上に USBPD-CC ラインとしてホストされています。プルアップ/プルダウンの競合を避けるため、パッドを TDI として使用するか CC として使用するかをユーザオプションで決定できます。

表 684. トレースポートピン

ピン名	タイプ	説明	ピン割り当て
TRACED0	O	トレース同期データ出力 0	データシートを参照してください
TRACED1		トレース同期データ出力 1	
TRACED2		トレース同期データ出力 2	
TRACED3		トレース同期データ出力 3	
TRACECK		トレースクロック	

表 685. シングルワイヤトレースポートのピン

ピン名	タイプ	説明	ピン割り当て
TRACESWO	O	シングルワイヤトレース非同期データ出力	PB3 ⁽¹⁾

1. TRACESWO が JTDO とマルチプレクスされます。つまり、シングルワイヤトレースは JTAG ではなく、シリアルワイヤデバッグ用のインタフェースを使用する場合にしか使用できません。

58.2.3 DBG のリセットおよびクロック

デバッグポート (SWJ-DP) は、STANDBY モードからウェイクアップするときに、パワーオンリセットによってのみリセットされます。

デバッグは、デバッグインタフェースのピン、JTCK/SWCLK を介して、デバッグポート用のクロックを提供します。このクロックを使用して、シリアル-ワイヤおよび JTAG モードでシリアル入力データを登録し、同時にデバッグポートのステートマシンや内部ロジックを運用します。したがって、デバッグポートを確実にアイドルステートに戻すためには、このクロックはアクセス終了後に数回のサイクルの間トグルし続ける必要があります。

SWJ-DP は DCLK ドメインとの非同期インタフェースを内蔵しており、このインタフェースは残りの SWJ-DP およびアクセスポートを範囲に含みます。

DCLK は、システムクロックのゲート付きバージョンです。

DCLK ドメインは、[DP 制御およびステータスレジスタ \(DP_CTRLSTATR\)](#) の CDBGPWRUPREQE ビットを使用して、デバッグによって有効にできます。デバッグがデバイスのデバッグ機能のいずれかにアクセスするには、先にクロックを有効にする必要があります。クロックの有効性は、DP_CTRL/STATR の CDBGPWRUPACK ビットに反映されます。エネルギー節約のため、DCLK はパワーアップ時に無効化され、またデバッグが切断されたときに無効化する必要があります。

プロセッサに内蔵されているデバッグおよびトレースコンポーネントは、プロセッサクロックでクロック供給されます。

58.2.4 DBG 電力ドメイン

デバッグコンポーネントは、コア電力ドメインに配置されています。これは、SHUTDOWN あるいは STANDBY 低電力モードではデバッグ接続を使用できないことを意味しています。デバイスが STANDBY モードになったときに接続が切断されないよう、[DBGMCU 設定レジスタ \(DBGMCU_CR\)](#) にビットをセットしてコアへの電源供給を維持することができます。これにより、プロセッサクロックをアクティブな状態で維持しながらリセットを保留して、デバッグセッションを維持することができます。

58.2.5 デバッグおよび低電力モード

デバイスには省電力機能が内蔵されており、必要ないときにはコア電力ドメインを電源オフまたは停止することができます。電源オフ、またはコアがクロック供給されていない場合、デバッグコンポーネントはすべてデバッグにアクセスできません。このような事態を防ぐために、省電力モードのエミュレーションが実装されています。あるドメインでエミュレーションが有効な場合、そのドメインは省電力モードに移行しますが、そのクロックと電力は維持されます。言い換えれば、ドメインは省電力モードと同様に動作しながら、デバッグとの接続は維持します。

エミュレーションモードは、マイクロコントローラデバッグ (DBGMCU) ユニットでプログラムされます。詳細については、[セクション 58.12 : マイクロコントローラデバッグユニット \(DBGMCU\)](#) を参照してください。

58.2.6 セキュリティ

トレースおよびデバッグのコンポーネントにより、製品開発中のプロセッサやシステムへの高度なアクセスが可能です。ユーザコードを保護するため、またデバッグ機能を使用して完成した製品の通常動作が変更されたり危険にさらされないことがないよう、これらの機能は無効にするか、範囲を制限することができます。たとえば、セキュアソフトウェアのデバッグおよびトレースの無効化は、非セキュアコードのデバッグを妨げることなく実現できます。

デバッグのセキュアメモリへのアクセスは（許可されている場合）、デバッグ AHB のセキュアなトランザクションを使用して（つまり **AP1 制御/ステータスワードレジスタ (AP1_CSWR)** で PROT[6] ビットをセットして）実行する必要があります。

プロセッサがシステム Flash メモリから起動されている場合、デバッグアクセスは無効になります。

システムは次の認証信号を使用して、有効化/無効化するデバッグ機能を判断します。

- **dbgen** : すべてのデバッグ機能のグローバル有効
 - 0 : すべてのデバッグ機能が無効になります。
 - 1 : 非セキュアステートのデバッグ機能が有効になります。セキュアステートのデバッグ機能は **spiden** 信号の状態に依存します。
- **spiden** : **dbgen** = 1 の場合にセキュアステートでのデバッグを有効にします。
 - 0 : デバッグ機能がセキュアステートで無効になります。
 - 1 : デバッグ機能がセキュアステートで有効になります。
- **niden** : トレースと性能の監視を有効にします（非侵入型デバッグ）。
 - 0 : トレース生成が無効になります。
 - 1 : 非セキュアステートのトレース生成が有効になります。セキュアステートのトレース生成は **spniden** 信号の状態に依存します。
- **spniden** : **niden** = 1 の場合にセキュアステートでのトレースと性能の監視を有効にします。
 - 0 : トレース生成がセキュアステートで無効になります。
 - 1 : トレース生成がセキュアステートで有効になります。

認証信号の状態に応じた各コンポーネントの動作の詳細については、関連する章または関連する Arm® のテクニカルドキュメントを参照してください。

信号の状態は、TrustZone が有効になっている（TZEN = 0xB4）場合は表 686 に示すようなデバッグの状態に従ってセットされ、TrustZone が無効になっている（TZEN = 0xC3）場合は表 687 に従ってセットされます。

表 686. TrustZone が有効な状態（TZEN = 0xB4）での認証信号の状態

デバッグ状態	認証信号の状態	説明
OPEN	dbgen = 1 spiden = 1 niden = 1 spniden = 1	デバッグおよびトレースは、プロセッサの状態にかかわらず有効になります。セキュアメモリへのデバッグアクセスが許可されます。
CLOSED SECURE	dbgen = 1 spiden = 0 niden = 1 spniden = 0	デバッグおよびトレースは、プロセッサが非セキュアステートである場合に有効になります。非セキュアメモリへのデバッグアクセスが許可されます。セキュアメモリへのデバッグアクセスが無効になります。

表 686. TrustZone が有効な状態 (TZEN = 0xB4) での認証信号の状態 (続き)

デバッグ状態	認証信号の状態	説明
CLOSED	dbgen = 0 spiden = 0 niden = 0 spniden = 0	デバッグおよびトレースは無効になります。

表 687. TrustZone が無効な状態 (TZEN = 0xC3) での認証信号の状態

デバッグ状態	認証信号の状態	説明
OPEN	dbgen = 1 spiden = 1 niden = 1 spniden = 1	デバッグおよびトレースは、プロセッサの状態にかかわらず有効になります。メモリとリソースはすべて非セキュアと見なされ、デバッグにアクセス可能です。
CLOSED	dbgen = 0 spiden = 0 niden = 0 spniden = 0	デバッグおよびトレースは無効になります。

認証信号の状態は、Cortex[®]-M33 のシステム制御空間 (SCS) の DAUTHSTATUS レジスタから読み出すことができます。

デバッグ状態は、製品のライフサイクル状態 (「製品のライフサイクル」の章を参照) とデバッグの認証状態によって異なります ([セクション 58.2.7 : デバッグ認証](#)を参照)。

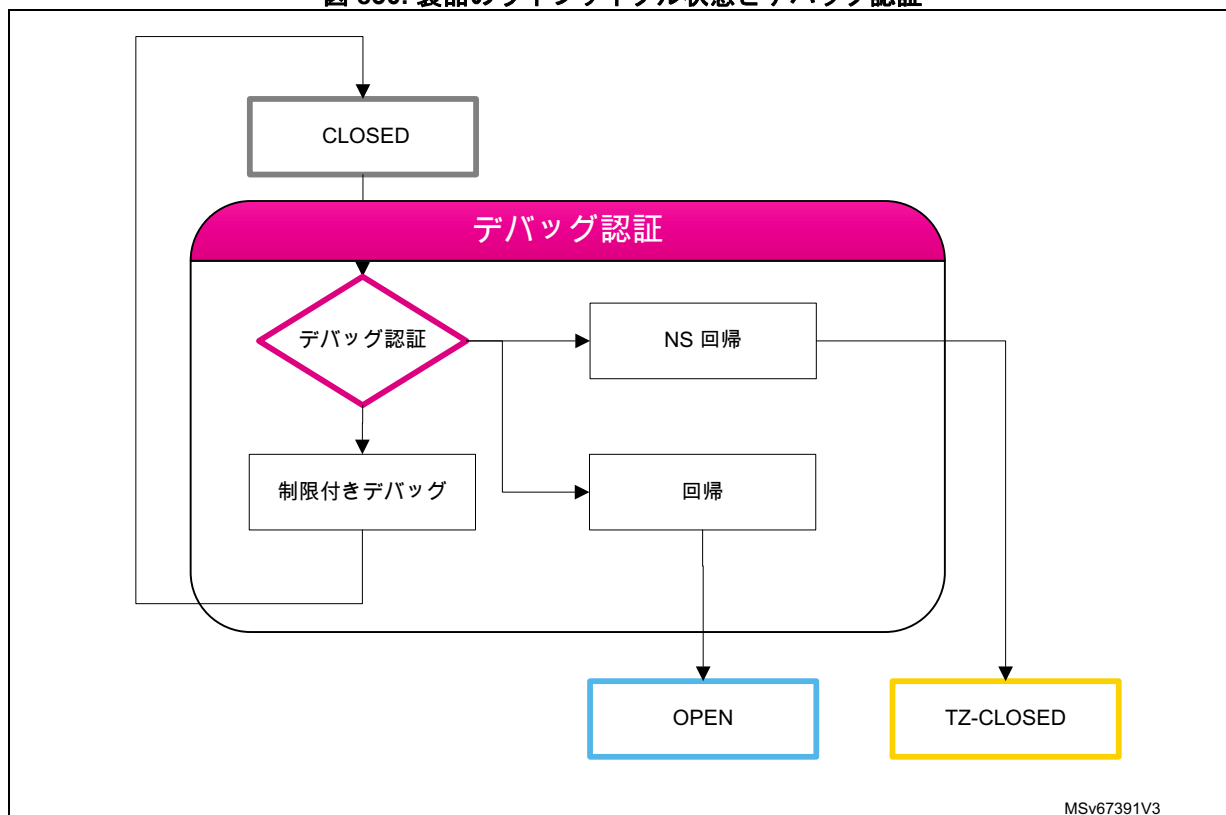
表 688. ライフサイクル状態とデバッグ状態

製品のライフサイクル状態	デバッグ状態
OPEN	OPEN
TZ_CLOSED	CLOSED-SECURE
CLOSED (デバッグが認証されない)	CLOSED
CLOSED (制限付きのデバッグ)	OPEN/CLOSED-SECURE ⁽¹⁾
LOCKED	CLOSED

1. 認証レベルによって異なります。

58.2.7 デバッグ認証

図 830. 製品のライフサイクル状態とデバッグ認証



MSv67391V3

図 830 に、製品のライフサイクルとデバッグ認証の状態を示します。

デバイスが CLOSED（製品のライフサイクル）状態の場合、デバッグ状態も CLOSED になります。デバッグ認証手順を使用すると、信頼の起点（ROT）と呼ばれる機密情報を損傷することなく、信頼できるデバッグでアクセスを再開することができます。

デバッグを再開できるのは、機密アセットのセキュリティが保証されている（そして、TrasZone が有効（TZEN =0xB4）になっている）場合のみです。これは制限付きデバッグと呼ばれます。制限により、ROT 情報のセキュリティが保証されるためです。

あるいは、機密情報のセキュリティが保証できない場合は、部分または完全回帰メカニズムを使用できます。これが回帰と呼ばれるのは、回帰により、デバッグの再開前に機密情報を削除できるためです。

- 部分的回帰では、非セキュアコードおよびアセットの解放が行われます。部分的な回帰管理が行える中間状態は、NS 回帰と呼ばれます。
- 完全回帰では、すべてのコードおよびアセットの解放が行われます。完全な回帰管理が行える中間状態は、回帰と呼ばれます。

デバッグ認証制御原理

デバッグによりユーザがシステムの大部分にアクセスできることを考えると、デバッグ検証はシステムの最も重要なセキュリティ機能の 1 つと言えます。

デバッグの再開を制御するため、デバイスはデバッグ認証プロトコルを適用します。

このプロトコルは、ホスト認証のため、非対称暗号化に基づくチャレンジレスポンスメカニズムを実装します。これは、デバイスに格納された公開鍵とホストライブラリの秘密鍵のペアを使用して、デバイスによって生成されるランダム値（チャレンジ）に署名します。

このプロトコルは、DBGMCU に配置されたメールボックスインタフェースを通じて、ホストとデバイス間の双方向通信を実装します。

ホストは、JTAG/SWD インタフェースを通じてメールボックスへの書込みができます。同じメールボックスを通じて、デバイスから応答とメッセージが返されることを想定します。

デバッグ認証プロトコルは、デバイスのパワーオンリセット時、ホストによって「オープンリクエスト」メッセージがポストされるときに、起動されます。

プロトコルは以下に基づいています。

- 初期メッセージ：ホストによってポストされ、リセットと合わせて、デバイスのデバッグ認証プロセスを起動します。
- チャレンジメッセージ：デバイスでランダム値が生成され、応答の返送時にホストによって署名されます。
- 応答：ホストによって、認証のためのメッセージがデバイスに送信されます。これは、トークンの生成ツールを使用して行われます。

実装は、システム Flash に内蔵されたコードで行われます。ホストによってメールボックスに初期メッセージがポストされている場合、リセット後に自動的にこのコードが呼び出されます。

プロトコルのバージョン、デバイスのタイプ、類似のパラメータを調整するための相互認証の最初のシーケンスの後、デバイスではランダム値を生成します。これに対して、ホストでは、応答の作成時に秘密鍵を使用して署名します。

ST マイクロエレクトロニクスの実装は、デバッグ認証のための Arm[®] PSA-ADAC ソリューションに基づいて行われます。

デバッグ認証は、独自のプロトコルまたはオープンソースプロトコルを使用して実装できます。

この機能はセキュリティ上重要であるため、STM32H5 デバイスではデバッグ認証がシステム Flash でプロビジョニングされます。ST マイクロエレクトロニクスでは、デバッグ環境に統合されたホストツールを用意しています (IDE)。

デバッグ認証のプロビジョニング

デバッグ認証は、STM32H5 プラットフォームでネイティブにサポートされています。つまり、ST デバッグ認証 (ST-DA) で使用されるデータはセキュアキーストレージ領域内の定義された場所でプロビジョニングする必要があるということです (Flash メモリで定義された OBKey)。

デバッグ認証の設定が行えるのは、PRODUCT_STATE が「Provisioning」のときのみです。PRODUCT_STATE が「Open」のときは行えません。

以下のデータをプロビジョニングする必要があります (セクション 7.7.2 : RSS ユーザ機能を参照)。

- データのプロビジョニングは、OBK_HDPL1 マッピングの最初の方で行う必要があります (0x0FFD0100) (セクション 7.5.2 : HDPL レベルごとの OBK アクセスを参照)
- STM32H57x デバイスに対して RSS_Lib 暗号化オプションをセットする必要があります
- STM32H56x デバイスに対して RSS_Lib 暗号化オプションをリセットする必要があります

プロビジョニングするデータを表 689 で定義します。これは、TZEN オプションバイト設定 (有効または無効) によって変わります。

表 689. プロビジョニングするデータの定義

データ サイズ	TZEN が有効 (認証)	TZEN が無効 (パスワード : 完全回帰のみが使用可能)
32 バイト	BLOB 全体の SHA256 (つまり、2 つの次のフィールド)	
	ROT 認証公開鍵の SHA256	パスワードの SHA256
16 バイト	128 ビットの許可マスク (表 690を参照)	128 ビットが予約済みで、 0 にセットされます

注意 : Open または Provisioning 以外の状態に変更する前に、上記に従ってデバッグ認証を設定してください。

表 690. 許可マスク (エンディアン形式 : リトルエンディアン)

ビット	説明
ビット 0	セットした場合、HDPL1 - NS からの浸入型デバッグが可能になります
ビット 1	セットした場合、HDPL2 - NS からの浸入型デバッグが可能になります
ビット 2	セットした場合、HDPL3 - NS からの浸入型デバッグが可能になります
ビット 3	予約済みです。
ビット 4	セットした場合、HDPL1 - S からの浸入型デバッグが可能になります
ビット 5	セットした場合、HDPL2 - S からの浸入型デバッグが可能になります
ビット 6	セットした場合、HDPL3 - S からの浸入型デバッグが可能になります
ビット 07:11	予約済みです。
ビット 12	TZ-Closed への回帰。1 の場合、TZ-Closed への回帰が可能です
ビット 13	予約済みです。
ビット 14	回帰 (完全回帰)。1 の場合、完全回帰が可能です
ビット 15:127	予約済みです。

58.3 シリアルワイヤ/JTAG デバッグポート (SWJ-DP)

SWJ-DP は CoreSight™ コンポーネントの 1 つで、デバッグ装置との接続用の外部アクセスポートを実装しています。

2 種類のインタフェースを設定することができます。

- 5 ピンの標準 JTAG インタフェース (JTAG-DP)
- 2 ピン (クロック + データ) のシリアルワイヤデバッグポート (SW-DP)

これらの 2 つのモードは同じ IO ピンを共有しているため、相互に排他的です。

デフォルトでは、JTAG-DP はシステムリセットまたはパワーオンリセット後に選択されます。5 つの IO ピンはデバッグ代替機能モードでハードウェアによって設定されます。SWJ-DP は、JTDI、JTMS/SWDIO、および nJTRST にプルアップレジスタ、JTCK/SWCLK にプルダウンレジスタを搭載しています。

デバッグは JTMS/SWDIO に次のようなシリアルデータシーケンスを送信することによって、SW-DP を選択することができます。

..... (1 が 50 個以上) ..., 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, ... (1 が 50 個以上) ...

JTCK/SWCLK はデータビットごとに循環させる必要があります。

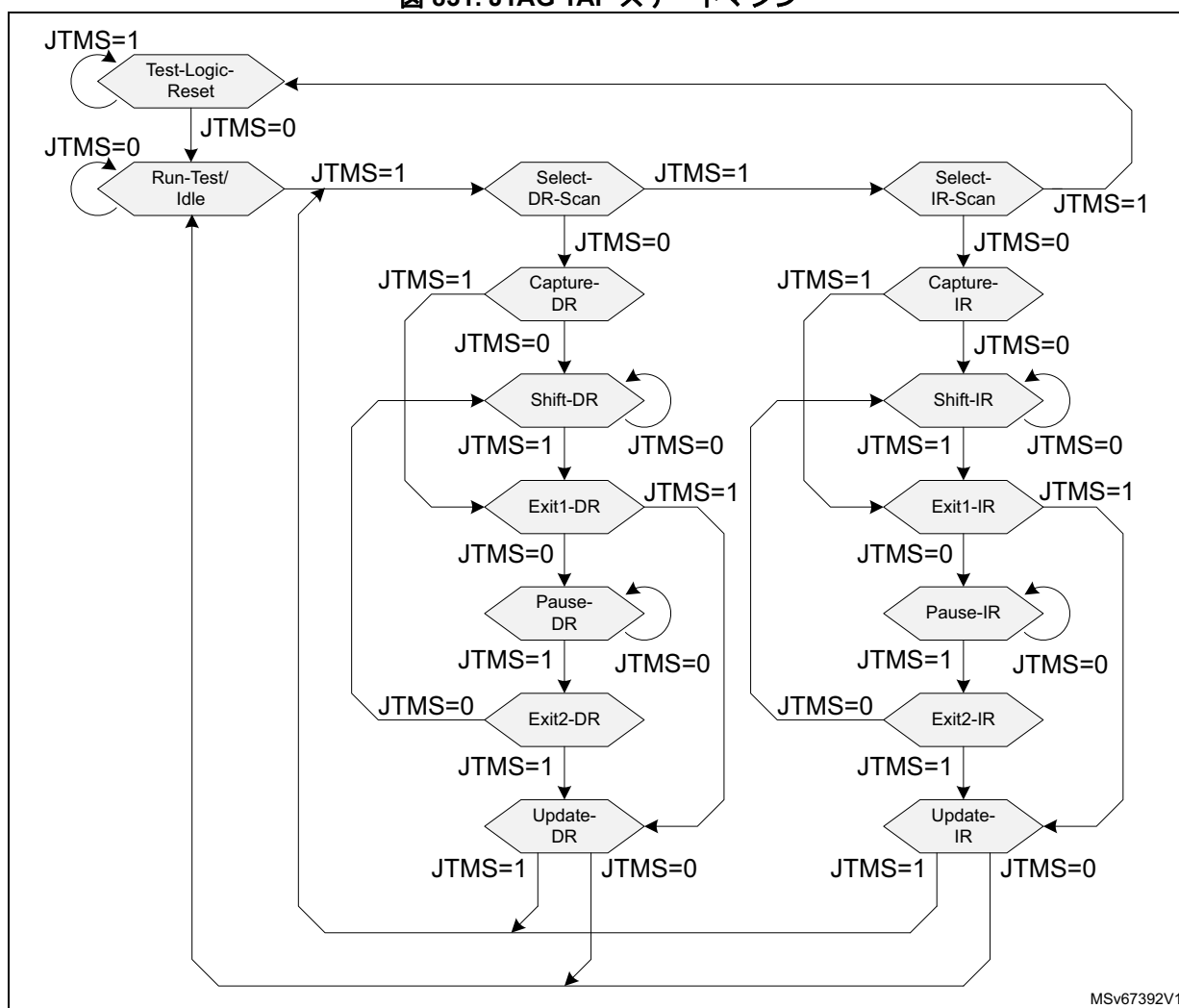
SW-DP モードでは、使用していない JTAG ピン、JTDI、JTDO、および nJTRST を他の機能に使用することができます。

注： SWJ ポートのすべての IO はソフトウェアで他の機能に合わせて再設定できますが、デバッグはできなくなります。

58.3.1 JTAG デバッグポート

JTAG-DP は IEEE 1149.1-1990 に基づく、下図に示す TAP ステートマシン (TAPSM) を実装しています。ステートマシンは 2 つのスキャンチェーンを制御します。一方は 1 つの命令レジスタ (IR) に、もう一方は複数のデータレジスタ (DR) に関連付けられています。

図 831. JTAG TAP ステートマシン



MSv67392V1

JTAG-DP の動作は次のとおりです。

1. TAPSM が Capture-IR ステートに入ると、0b0001 が命令レジスタ (IR) のスキャンチェーンに転送されます。IR スキャンチェーンは JTDI と JTDO の間に接続されます。
2. TAPSM が Shift-IR ステートにあるとき、IR スキャンチェーンは JTCK の各立上りエッジで 1 ビットだけシフトします。つまり、最初の単位時間で以下のアクションが起こります。
 - IR スキャンチェーンの LSB が JTDO に出力されます。
 - IR スキャンチェーンのビット [n] がビット [n-1] に転送されます。
 - JTDI の値が IR スキャンチェーンの MSB に転送されます。
3. TAPSM が Update-IR ステートに入ると、IR スキャンチェーンにスキャンされた値が命令レジスタに転送されます。
4. TAPSM が Capture-DR ステートに入ると、ある値がデータレジスタの 1 つから、JTDI と JTDO の間に接続された DR スキャンチェーンの 1 つに転送されます。
5. 命令レジスタに保持された値によって、どのデータレジスタ、および該当する DR スキャンチェーンを選択するかが決まります。
6. 次に、このデータは、TAPSM が Shift-DR ステートにある間に、Shift-IR ステートで IR がシフトしたのと同じようにシフトされます。
7. TAPSM が Update-DR ステートに入ると、DR スキャンチェーンにスキャンされた値が選択されたデータレジスタに転送されます。
8. TAPSM が Run-Test/Idle ステートにあるときは、特別なアクションは起こりません。IDCODE 命令は IR にロードされます。

アクティブな場合、nJTRST 信号は Test-Logic-Reset ステートとは非同期にステートマシンをリセットします。

以下の表に、4 ビットの IR 命令に対応するデータレジスタを記載します。

表 691. JTAG-DP データレジスタ

IR 命令	DR レジスタ	スキャン チェーン長	説明
0000~0111	(BYPASS)	1	実装されません。BYPASS が選択されます。
1000	ABORT	35	アボートレジスタ – ビット 34:3 = APABORT[31:0] : 実行中のアクセスポートトランザクションをアボートするには、0x0000 0001 を書き込みます – ビット 2:1 = A[3:2] = 00 – ビット 0 = RnW = 0 : 書き込み専用
1001	(BYPASS)	1	予約済み。BYPASS が選択されます。

表 691. JTAG-DP データレジスタ (続き)

IR 命令	DR レジスタ	スキャン チェーン長	説明
1010	DPACC	35	<p>デバッグポートアクセスレジスタ</p> <p>デバッグポートを初期化し、デバッグポートレジスタへのアクセスを可能にします。</p> <p>– 入力データ転送時 :</p> <p>ビット 34:3 = DATA[31:0] = 書き込みリクエスト用に転送する 32 ビットデータ</p> <p>ビット 2:1 = A[3:2] = デバッグポートレジスタの 2 ビットアドレス</p> <p>ビット 0 = RnW = 読出しリクエスト (1) または書き込みリクエスト (0)</p> <p>– 出力データ転送時 :</p> <p>ビット 34:3 = DATA[31:0] = 読出しリクエストに続いて読み出される 32 ビットデータ</p> <p>ビット 2:0 = ACK[2:0] = 3 ビット確認応答 :</p> <ul style="list-style-type: none"> – 010 = OK/FAULT – 001 = WAIT – その他 : 予約済み
1011	APACC	35	<p>アクセスポートアクセスレジスタ</p> <p>アクセスポートを初期化し、アクセスポートレジスタへのアクセスを可能にします。</p> <p>– 入力データ転送時 :</p> <p>ビット 34:3 = DATA[31:0] = 書き込みリクエスト用にシフトインする 32 ビットデータ</p> <p>ビット 2:1 = A[3:2] = アクセスポートレジスタの 2 ビットサブアドレス</p> <p>ビット 0 = RnW = 読出しリクエスト (1) または書き込みリクエスト (0)</p> <p>– 出力データ転送時 :</p> <p>ビット 34:3 = DATA[31:0] = 読出しリクエストに続いて読み出される 32 ビットデータ</p> <p>ビット 2:0 = ACK[2:0] = 3 ビット確認応答 :</p> <ul style="list-style-type: none"> – 010 = OK/FAULT – 001 = WAIT – その他 : 予約済み
1100	(BYPASS)	1	予約済み。BYPASS が選択されます。
1101	(BYPASS)	1	予約済み。BYPASS が選択されます。
1110	IDCODE	32	<p>識別コード</p> <p>0x6BA0 0477 : Cortex[®]-M33 JTAG デバッグポートの ID コード</p>
1111	BYPASS	1	<p>バイパス</p> <p>1 JTCK サイクル遅延が JTDI と JTDO の間に挿入されます。</p>

DR レジスタの詳細は、Arm[®] Debug Interface Architecture Specification を参照してください ([セクション 58.13](#) を参照)。

58.3.2 シリアルワイヤデバッグポート

シリアルワイヤデバッグプロトコルでは、次のピンを使用します。

- SWCLK : ホストからターゲットへのクロック
- SWDIO : 双方向シリアルデータ (100kΩプルアップが必要)

シリアルデータはクロックと同期して LSB が先に転送されます。

転送は以下の 3 フェーズで構成されます。

1. ホストによって送信されるパケットリクエスト (8 ビット) (表 692 を参照)
2. ターゲットによって送信される確認応答 (3 ビット) (表 693 を参照)
3. ホスト (書込みの場合) またはターゲット (読出しの場合) によって送信されるデータ転送 (33 ビット) (表 694 を参照)

確認応答が OK である場合にのみ、データ転送が実行されます。

データ方向が逆転した場合、各フェーズの間に 1 クロックサイクルのターンアラウンド時間が挿入されます。

表 692. パケットリクエスト

ビットフィールド	名前	説明
0	スタート	1 である必要があります。
1	APnDP	– 0 : DP レジスタアクセス - セクション 58.3.3 : デバッグポートレジスタを参照のこと。 – 1 : AP レジスタアクセス - セクション 58.4 : アクセスポートを参照のこと。
2	RnW	– 0 : 書込みリクエスト – 1 : 読出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 696 または 表 698 を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。ターゲットによって 1 と読み出される必要があります。

表 693. ACK 応答

ビットフィールド	名前	説明
2:0	ACK	– 000 : FAULT – 010 : ウェイト – 100 : OK

表 694. データ転送

ビットフィールド	名前	説明
31:0	WDATA または RDATA	書込み/読出しデータ
32	Parity	32 データビットのうちの 1 ビットパリティ

ターゲットからフォールトまたはウェイトの ACK 応答があった場合、オーバーラン検出が有効でなければ、データ転送フェーズは取り消されます。この場合、データはターゲットから無視されるか (書込みの場合)、駆動されません (読出しの場合)。

初回接続時またはプロトコルエラー発生時には、ホストはラインリセットを生成する必要があります。ラインリセットは、SWDIO がハイの状態の 50 回以上の SWCLK サイクルと、これに続く SWDIO がローの状態の 2 回の SWCLK サイクルで構成されています。

シリアルワイヤデバッグプロトコルの詳細については、Arm® Debug Interface Architecture Specification[1] を参照してください。

注： SWJ-DP は SWD プロトコルのバージョン 2 を実装しています。

58.3.3 デバッグポートレジスタ

SW-DP と JTAG-DP は両方とも表 696 に示されたデバッグポート (DP) レジスタにアクセスします。

デバッグは次のように DP レジスタへアクセスすることができます。

1. JTAG を使用する場合、バンク内にあるレジスタアドレスを DPACC レジスタの A(3:2) フィールドに書き込みます。読出しか書き込みかを選択するために、RnW ビットをプログラムします。書き込みの場合は、データフィールドに書き込みデータをプログラムします。SWD を使用する場合、A(3:2) および RnW フィールドは、APnDP ビットをリセットした状態で SW-DP に送信されるパケットリクエストワードの一部となります (表 692 を参照)。書き込みデータはデータフェーズで送信されます。
2. アドレス 0x4 の DP バンクレジスタの 1 つにアクセスするには、まずアドレス 0x8 の DP_SELECTR レジスタにレジスタ番号を書き込む必要があります。アドレス 0x4 への後続の読出しまたは書き込みはすべて、DP_SELECTR レジスタの内容に対応するレジスタにアクセスします。

表 695. デバッグポートレジスタ

アドレス	A[3:2] の値	読出し／書き込み	説明
0x0	00	R	DP デバッグポートの識別レジスタ (DP_DPIDR) は、デバッグポート用の IDCODE を格納します。
		W	DP アバートレジスタ (DP_ABORTR) ⁽¹⁾ 現在の AP トランザクションをアバートします。また、このレジスタを使用して、DP_CTRLSTATR レジスタのエラーフラグをクリアします。
0x4	01	読出し／書き込み	DP_SELECTR.DPBANKSEL[3:0] = 0x0 の場合、DP 制御およびステータスレジスタ (DP_CTRLSTATR) で DP を制御し、ステータス情報を提供します。
			DP_SELECTR.DPBANKSEL[3:0] = 0x1 の場合、DP のデータリンク制御レジスタ (DP_DLCR) ⁽²⁾ で SWD データリンクの動作モードを制御します。
			DP_SELECTR.DPBANKSEL[3:0] = 0x2 の場合、DP ターゲット識別レジスタ (DP_TARGETIDR) でターゲット識別情報を提供します。
			DP_SELECTR.DPBANKSEL[3:0] = 0x3 の場合、DP データリンクプロトコル識別レジスタ (DP_DLPIDR) ⁽²⁾ で SWD プロトコルバージョンを提供します。
0x8	10	R	DP イベントステータスレジスタ (DP_RESENDER) ⁽²⁾ は、最後の AP 読出しまたは DP_RDBUFF 読出しによって返された値を返します。読み出し転送の破損時に使用されます。
		W	DP アクセスポート選択レジスタ (DP_SELECTR) はアクセスポート、アクセスポートのレジスタバンク、および DP レジスタのアドレス 0x4 を選択します。
0xC	11	R	DP 読出しバッファレジスタ (DP_RDBUFFR) – JTAG-DP 経由でこのレジスタを使用して、デバッグが一連の動作のあとで最終結果を得られるようにします (新しい JTAG-DP 動作のリクエストなし)。 – SW-DP 経由でこのレジスタを使用する場合は、直前の AP 読出しアクセスの結果を格納し、新たな AP アクセスを防ぎます。

1. JTAG-DP から AP ABORT レジスタへアクセスするには ABORT 命令を使用します。
2. SW-DP 経由でのみアクセス可能。レジスタは JTAG-DP 経由で「予約済み」です。

DP デバッグポートの識別レジスタ (DP_DPIDR)

アドレスオフセット : 0x0

リセット値 : 0x6BA0 2477

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				PARTNO[7:0]								Res.	Res.	Res.	MIN
r	r	r	r	r	r	r	r	r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION[3:0]				DESIGNER[10:0]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

ビット 31:28 **REVISION[3:0]** : リビジョンコード

0x6 : リビジョン 6

ビット 27:20 **PARTNO[7:0]** : デバッグポートの部品番号

0xBA

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **MIN** : 最小デバッグポート (MINDP) の実装

0x0 : MINDP は実装されません

ビット 15:12 **VERSION[3:0]** : デバッグポートアーキテクチャバージョン

0x2 : DPv2

ビット 11:1 **DESIGNER[10:0]** : JEDEC 設計者 ID コード

0x23B : Arm® JEDEC コード

ビット 0 予約済みであり、リセット値に保持する必要があります。

DP アボートレジスタ (DP_ABORTR)

アドレスオフセット : 0x0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORUNERRCLR	WDERRCLR	STKERRCLR	Res.	DAPABORT
											w	w	w		w

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **ORUNERRCLR** : 受信オーバーランエラークリア

0 : 影響なし。

1 : DP_CTRL/STATR レジスタの STICKYORUN ビットがクリアされます。

ビット 3 **WDERRCLR** : 書き込みデータエラークリア

0 : 影響なし。

1 : DP_CTRL/STATR レジスタの WDATAERR ビットがクリアされます。

ビット 2 **STKERRCLR** : スティックエラークリア

0 : 影響なし。

1 : DP_CTRL/STATR レジスタの STICKYERR ビットがクリアされます。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DAPABORT** : 過剰な数のウェイト応答が返された場合の現在の AP トランザクションのアボート

このビットはトランザクションがストールしていることを示します。

0 : 影響なし。

1 : トランザクションがアボートされます。

DP 制御およびステータスレジスタ (DP_CTRLSTATR)

アドレスオフセット : 0x4

リセット値 : 0x0000 0000

このレジスタは DP_SELECTR.DPBANKSEL[3:0] = 0x0 の場合にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	CDBGPWRUPACK	CDBGPWRUPREQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		r	r												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATAERR	READOK	STICKYERR	Res.	Res.	Res.	STICKYORUN	ORUNDETECT
								r	r	r				r	r

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **CDBGPWRUPACK** : デバッグパワーアップの確認応答[セクション 58.2.5 : デバッグおよび低電力モード](#)の説明を参照してください。

0 : DCLK がゲートされます。

1 : DCLK は有効です。

ビット 28 **CDBGPWRUPREQ** : デバッグパワーアップリクエスト

このビットは DCLK イネーブルリクエスト信号を制御します。

0 : DCLK のゲート操作をリクエストします。

1 : DCLK の有効化をリクエストします。

ビット 27:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDATAERR** : SW-DP の書込みデータエラー (読出し専用)

このビットは、書込みのデータフェーズでパリティエラーまたはフレーミングエラーが発生しているか、DP によって受け付けられた書込みが、その後 AP に提供されることなく破棄されたことを示します。

このビットは、ABORT.WDERRCLR ビットに 1 を書き込むことによってリセットされます。

0 : エラーはありません。

1 : エラーが発生しました。

注 : このビットは JTAG-DP に予約済みです。ビット 6 **READOK** : SW-DP の AP 読出し応答 (読出し専用)

このビットは、最後の AP 読出しアクセスへの応答を示します。

0 : 読出しは NG です。

1 : 読出しは OK です。

注 : このビットは JTAG-DP に予約済みです。ビット 5 **STICKYERR** : トランザクションエラー (SW-DP では読出し専用、JTAG-DP では読み書き)

このビットは、AP トランザクション中にエラーが発生したことを示します。このビットは、DP_ABORTR.STKERRCLR ビット (SW-DP および JTAG-DP) に 1 を書き込むことによってリセットされます。

0 : エラーはありません。

1 : エラーが発生しました。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **STICKYORUN** : オーバーラン (SW-DP では読出し専用、JTAG-DP では読み書き)

このビットは、オーバーラン (前のトランザクションの完了前に新たなトランザクションを受信) が発生したことを示します。このビットは、ORUNDETECT ビットがセットされている場合のみセットされます。DP_ABORTR.ORUNERRCLR ビット (SW-DP および JTAG-DP) に 1 を書き込むことによってリセットされます。

0 : オーバーランはありません。

1 : オーバーランが発生しました。

ビット 0 **ORUNDETECT** : オーバーラン検出モード有効

0 : 無効

1 : 有効オーバーランが発生した場合、STICKYORUN ビットがセットされ、このビットがクリアされるまで次のトランザクションはブロックされます。

DP のデータリンク制御レジスタ (DP_DLCR)

アドレスオフセット : 0x4

リセット値 : 0x0000 0000

このレジスタは DP_SELECTR.DPBANKSEL[3:0] = 0x1 の場合にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TURNROUND[1:0]		WIREMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.
						r	r	r	r						

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **TURNROUND[1:0]** : SWDIO のトライステート周期

0x0 : 1 データビット周期

ビット 7:6 **WIREMODE[1:0]** : SW-DP モード

0x0 : 同期モード

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

DP ターゲット識別レジスタ (DP_TARGETIDR)

アドレスオフセット : 0x4

リセット値 : 0xFFFF 0041

このレジスタは DP_SELECTR.DPBANKSEL[3:0] = 0x2 の場合にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TREVISION[3:0]				TPARTNO[15:4]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPARTNO[3:0]				TDESIGNER[10:0]										Res.	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

ビット 31:28 **TREVISION[3:0]** : ターゲットのリビジョンビット 27:12 **TPARTNO[15:0]** : ターゲットの部品番号

0x4840 : STM32H5

ビット 11:1 **TDESIGNER[10:0]** : ターゲットの設計者 JEDEC コード

0x020 : STMicroelectronics

ビット 0 予約済みであり、リセット値に保持する必要があります。

DP データリンクプロトコル識別レジスタ (DP_DLPIDR)

アドレスオフセット : 0x4

リセット値 : 0x0000 0001

このレジスタは DP_SELECTR.DPBANKSEL[3:0] = 0x3 の場合にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TINSTANCE[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROTSVN[3:0]			
												r	r	r	r

ビット 31:28 **TINSTANCE[3:0]** : ターゲットのインスタンス番号

このフィールドは、マルチドロップシステムにおけるデバイスのインスタンス番号を定義します。

0x0 : インスタンス番号 0

ビット 27:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PROTSVN[3:0]** : シリアルワイヤデバッグプロトコルのバージョン

0x1 : バージョン 2

DP イベントステータスレジスタ (DP_RESENDER)

アドレスオフセット : 0x8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESEND[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESEND[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RESEND[31:0]** : 最後の AP 読出しまたは DP_RDBUFF 読出しによって返された値
このレジスタは、読出し転送の破損時に使用されます。

DP アクセスポート選択レジスタ (DP_SELECTR)

アドレスオフセット : 0x8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSEL[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w	w	w	w	w	w	w	w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APBANKSEL[3:0]				DPBANKSEL[3:0]			
								w	w	w	w	w	w	w	w

ビット 31:24 **APSEL[7:0]** : アクセスポート選択

このフィールドによって次のトランザクションのアクセスポートを選択します。

0x00 : AP0 - システムデバッグアクセスポート (APB-AP)

0x01 : AP1 - Cortex[®]-M33 デバッグアクセスポート (AHB-AP)

その他 : 予約済み

ビット 23:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **APBANKSEL[3:0]** : AP レジスタバンク選択

このフィールドによって、次のトランザクションのアクティブな AP で 4 ワードレジスタバンクを選択します。

ビット 3:0 **DPBANKSEL[3:0]** : DP レジスタバンク選択

このフィールドによって、デバッグポートのアドレス 0x4 のレジスタを選択します。

0x0 : DP_CTRLSTAT レジスタ

0x1 : DP_DLCR レジスタ

0x2 : DP_TARGETID レジスタ

0x3 : DP_DLPIDR レジスタ

その他 : 予約済み

DP 読出しバッファレジスタ (DP_RDBUFFR)

アドレスオフセット : 0xC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDBUFF[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDBUFF[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **RDBUFF[31:0]** : 最後の AP 読出しアクセスによって返された値

AP 読出しアクセスによって返された値は、同じアドレスに対する 2 回目の読出しアクセスで取得でき、これによって対応するバス上で新しいトランザクションを開始することができます。値はこのレジスタから読み出すこともできますが、その場合、新しい AP トランザクションは発生しません。

58.3.4 デバッグポートのレジスタマップとリセット値

これらのレジスタは CPU メモリバス上にはありません。SW-DP および JTAG-DP のデバッグインタフェースからのみアクセスできます。

デバッグポートアドレスオフセットは 4 ビット幅で、上位 2 ビットは JTAG-DP レジスタの DPACC または SW-DP パケットリクエストの A[3:2] フィールドで定義されています。下位 2 ビットは 00 です。

表 696. デバッグポートのレジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	DP_DPIDR	REVISION [3:0]				PARTNO[7:0]								Res	Res	Res	MIN	VERSION [3:0]			DESIGNER[10:0]										Res		
	リセット値	0	1	1	0	1	0	1	1	1	0	1	0		Res	Res	Res	0	0	0	1	0	0	1	0	0	0	1	1	1	0	1	1
0x0	DP_ABORTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ORUNERRCLR	WDERRCLR	STKERRCLR	Res	DAPABORT
	リセット値																												0	0	0		0
0x4 ⁽¹⁾	DP_CTRLSTATR	Res	Res	CDBGPWRUPACK	CDBGPWRUPREQ	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDATAERR	READOK	STICKYERR	Res	Res	STICKYORUN	ORUNDETECT	
	リセット値			0	0																					0	0	0				0	0
0x4 ⁽²⁾	DP_DLCCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TURNROUND[1:0]			WIREMODE[1:0]			Res	Res	Res	Res
	リセット値																							0	0	0	0						
0x4 ⁽³⁾	DP_TARGETIDR	TREVISION [3:0]				TPARTNO[15:0]										TDESIGNER[10:0]										Res							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

表 696. デバッグポートのレジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4 ⁽⁴⁾	DP_DLPIDR	TINSTANCE[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROTSELVN[3:0]			
	リセット値	0	0	0	0																									0	0	0	1
0x8	DP_RESENDER	RESENDER[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	DP_SELECTOR	APSEL[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APBANKSEL[3:0]				DPBANKSEL[3:0]				
	リセット値	x	x	x	x	x	x	x	x																	x	x	x	x	x	x	x	x
0xC	DP_RDBUFFR	RDBUFF[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. DP_SELECTOR.DPBANKSEL[3:0] = 0x0
2. DP_SELECTOR.DPBANKSEL[3:0] = 0x1
3. DP_SELECTOR.DPBANKSEL[3:0] = 0x2
4. DP_SELECTOR.DPBANKSEL[3:0] = 0x3

58.4 アクセスポート

DP に接続される 2 つのアクセスポート (AP) があります。

- システムデバッグアクセスポート (AP0) : APB バス経由での DBGMCU と システムの ROM テーブルへのアクセスを有効にします。
- Cortex[®]-M33 デバッグアクセスポート (AP1) : 内部 AHB バスを介して、Cortex[®]-M33 プロセッサコアに組み込まれているデバッグおよびトレース機能へのアクセスを可能にします。

58.4.1 アクセスポートレジスタ

アクセスポートはすべて MEM-AP タイプです。つまり、デバッグおよびトレースコンポーネントのレジスタは AHB のアドレス空間にマップされます。AP は、デバッグより、それぞれ 4 つのレジスタから成るバンクとして構成された一連の 32 ビットレジスタとみなされます。これらのレジスタの一部を使用して AP 自体を設定または監視し、その他のレジスタはバスで転送を行うために使用します。AP レジスタを [表 698](#)に記載します。

AP レジスタのアドレスは、以下のフィールドによって構成されます。

- ビット [7:4] : [DP アクセスポート選択レジスタ \(DP_SELECTOR\)](#) の APBANKSEL[3:0] フィールドの内容
- ビット [3:2] : 使用するデバッグインタフェースに応じて、JTAG-DP の APACC データレジスタの A(3:2) フィールドの内容 ([表 691](#) を参照)、または SW-DP パケットリクエストの内容 ([表 692](#) を参照) のいずれか。
- ビット [1:0] = 常に 0 にセットされます。

DP_SELECTOR.APSEL[3:0] フィールドの内容は、アクセスされる MEM-AP を定義します。

表 697. MEM-AP レジスタ

アドレス	APBANKSEL	A[3:2]	名前	説明
0x00	0x0	0	CSWR	制御／ステータスワードレジスタ
0x04	0x0	1	TAR	転送アドレスレジスタ バストランザクションのターゲットアドレス
0x08	-	-	-	予約済みです。
0x0C	0x0	3	DRWR	データ読出し／書込みレジスタ このレジスタへアクセスすると、TAR[31:0] に格納されているアドレスに対して、デバッグバス上で該当するトランザクションが開始されます。
0x10	0x1	0	BD0R	バンクデータ 0 レジスタ このレジスタにアクセスすると、TAR[31:4] + 0x0 に格納されたアドレスに対して、デバッグバス上で該当するトランザクションが開始されます。
0x14	0x1	1	BD1R	バンクデータ 1 レジスタ このレジスタにアクセスすると、TAR[31:4] + 0x4 に格納されたアドレスに対して、デバッグバス上で該当するトランザクションが開始されます。
0x18	0x1	2	BD2R	バンクデータ 2 レジスタ このレジスタにアクセスすると、TAR[31:4] + 0x8 に格納されたアドレスに対して、デバッグバス上で該当するトランザクションが開始されます。
0x1C	0x1	3	BD3R	バンクデータ 3 レジスタ このレジスタにアクセスすると、TAR[31:4] + 0xC に格納されたアドレスに対して、デバッグバス上で該当するトランザクションが開始されます。
0x20	-	-	-	予約済みです。
0x24 から 0xEC	-	-	-	予約済みです。
0xF0	-	-	-	予約済みです。
0xF4	0xF	1	CFGR	設定レジスタ（読出し専用）
0xF8	0xF	2	BASER	デバッグ用ベースアドレスレジスタ（読出し専用） ROM テーブルのベースアドレス
0xFC	0xF	3	IDR	識別レジスタ（読出し専用）

デバッグは次のように AP レジスタへアクセスすることができます。

1. AP を選択するために [DP アクセスポート選択レジスタ \(DP_SELECTR\)](#) の APSEL[3:0] フィールドをプログラムし、アクセスするレジスタバンクを選択するために DP_SELECTR の APBANKSEL[3:0] フィールドをプログラムします。
2. JTAG を使用する場合は、バンク内にあるレジスタアドレスを APACC データレジスタの A(3:2) フィールドに書き込みます。読出しか書込みかを選択するために、RnW ビットをプログラムします。書込みの場合は、DATA フィールドに書込みデータをプログラムします。SWD を使用する場合は、A(3:2) および RnW フィールドは、APnDP ビットをセットした状態で SW-DP に送信されるパケットリクエストワードの一部となります（表 692 を参照）。書込みデータはデータフェーズで送信されます。

デバッグは、AP レジスタを使用して、以下の手順でメモリマップドデバッグコンポーネントレジスタにアクセス可能です（上記の AP レジスタアクセス手順を使用）。

1. **APx 転送アドレスレジスタ (APx_TAR)** ($x = 0, 1$) のトランザクションターゲットアドレスをプログラムします。
2. 必要であれば、転送パラメータ (AddrInc など) を **AP1 制御/ステータスワードレジスタ (AP1_CSWR)** に書き込みます。
3. AP_TAR に保持されているアドレスのバストランザクションを開始するには、**APx データ読み/書き込みレジスタ (APx_DRWR)** ($x = 0, 1$) に対する書き込みまたは読み出しを行います。あるいは、**APx バンクデータ n レジスタ (APx_BDnR)** ($x = 0, 1$) に対する読み出し/書き込みを行うことで、アドレス TAR[31:4] + n へのアクセスを開始します (AP_TAR レジスタのアドレスを変更せずに、最大 4 つの連続したアドレスへのアクセスを可能にします)。

MEM-AP の詳細については、Arm® Debug Interface Architecture Specification を参照してください [1]。

AP0 制御/ステータスワードレジスタ (AP0_CSWR)

アドレスオフセット : 0x0

リセット値 : 0x8000 0042

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBGSWEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r/w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MODE[3:0]				TRINPROG	DEVICEEN	ADDRINC[1:0]		Res.	SIZE[2:0]		
				r/w	r/w	r/w	r/w	r	r	r/w	r/w		r	r	r

ビット 31 **DBGSWEN** : ソフトウェアアクセスの有効化

APB バスへのソフトウェアアドレスを有効化または無効化します

0 : ソフトウェアアクセスを無効化します

1 : ソフトウェアアクセスを有効化します

ビット 30:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **MODE[3:0]** : 動作モード

0b0000 : 通常のダウンロードまたはアップロード

その他 : 予約済み

ビット 7 **TRINPROG** : 転送の実行中 (読み出し専用)

このフィールドは、APB マスタポートで現在転送が行われているかどうかを示します。

0 : APB の転送が実行中ではありません

1 : APB の転送が実行中です

ビット 6 **DEVICEEN** : デバイスのイネーブルステータス (読み出し専用)

1 : APB 転送は常に有効です

ビット 5:4 **ADDRINC[1:0]** : 自動インクリメントモード

TAR アドレスがトランザクション後に自動的にインクリメントされるかどうかを定義します。

0x0 : 自動インクリメントされません。

0x1 : アドレスがトランザクションのバイトサイズ (SIZE フィールド) でインクリメントされます。

その他 : 予約済み

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SIZE[2:0]** : 次のメモリアクセストランザクションのサイズ
0x2 : ワード (32 ビット)

AP1 制御/ステータスワードレジスタ (AP1_CSWR)

アドレスオフセット : 0x0

リセット値 : 0x43X0 : 00X2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SPROT	Res.	Res.	PROT[3:0]				SPISTATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw			rw	rw	rw	rw	r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRINPROG	DBGSTATUS	ADDRINC[1:0]		Res.	SIZE[2:0]		
								r	r	rw	rw		rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SPROT** : セキュア転送リクエスト

このフィールドにはバス転送の保護属性 HPROT[6] をセットします。

0 : セキュア転送、HPROT[6] = low

1 : 非セキュア転送、HPROT[6] = high

SPIDEN = 0 で SPROT = 0 の場合、バスの転送は行われません

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **PROT[3:0]** : バス転送保護

このフィールドにはバス転送の保護属性 HPROT[3:0] をセットします。

0bXXX0 : 命令アクセス

0bXXX1 : データアクセス

0bXX0X : ユーザモード

0bXX1X : 特権モード

0bX0XX : バッファ格納不可

0bX1XX : バッファ格納可

0b0XXX : 共有不可、ルックアップ不可、変更不可

0b1XXX : 共有可、ルックアップ可、変更可

ビット 23 **SPISTATUS** : セキュアなデバッグ認証ステータス

このフィールドは、SPIDEN信号の状態を示します

0 : セキュア AHB 転送は許可されません

1 : セキュア AHB 転送が許可されます

ビット 22:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TRINPROG** : 転送の実行中 (読出し専用)

このフィールドは、APB マスタポートで現在転送が行われているかどうかを示します。

0 : AHB の転送が実行中ではありません

1 : AHB の転送が実行中です

ビット 6 **DBGSTATUS** : デバッグイネーブル (DBGEN) ステータス

0 : AHB 転送がブロックされます。

1 : AHB 転送が有効になります。

ビット 5:4 **ADDRINC[1:0]** : 自動インクリメントモード

TAR アドレスがトランザクション後に自動的にインクリメントされるかどうかを定義します。

0x0 : 自動インクリメントされません。

0x1 : アドレスがバイト単位のトランザクションサイズ (SIZE フィールド) でインクリメントされます。シングル転送

0x2 : アドレスがバイト単位のトランザクション (SIZE フィールド) でインクリメントされます。4 つの 8 ビット転送または 2 つの 16 ビット転送を 1 つの 32 ビット DAP 転送にパックします。複数のトランザクションが AHB インタフェースで実行されます。

その他 : 予約済み

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SIZE[2:0]** : 次のメモリアクセストランザクションのサイズ

0x0 : バイト (8 ビット)

0x1 : ハーフワード (16 ビット)

0x2 : ワード (32 ビット)

その他 : 予約済み

APx 転送アドレスレジスタ (APx_TAR) (x = 0, 1)

アドレスオフセット : 0x04

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **TA[31:0]** : 現在の転送のアドレス。AP0 では、TA[1:0] は 0 に固定されています。

APx データ読出し/書込みレジスタ (APx_DRWR) (x = 0, 1)

アドレスオフセット : 0x0C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TD[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TD[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **TD[31:0]** : 現在の転送のデータ

APx バンクデータ n レジスタ (APx_BDnR) (x = 0, 1)アドレスオフセット : $0x10 + 4 * n$, ($n = 0 \sim 3$)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **DATA[31:0]** : アドレス TA [31:4] + $4 * n$ への現在の転送のバンクデータ。

自動アドレスのインクリメントは APx_BD0-3R では行われません。バンク転送は、ワード転送の場合にのみサポートされます。

APx ベースアドレスレジスタ (APx_BASER) (x = 0, 1)

アドレスオフセット : 0xF8

リセット値 : AP 0 : 0xE00E 0003

リセット値 : AP 1 : 0xE00F : E003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:12]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FORMAT	ENTRYPRESENT
r	r	r	r											r	r

ビット 31:12 **BASEADDR[31:12]** : 最初の ROM テーブルのベースアドレス (31 から 12)

ROM テーブルは 4 KB の境界上に整列させる必要があるため、12 LSB はゼロです。

0xE00E0 : AP0

0xE00FE : AP1

ビット 11:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **FORMAT** : ベースアドレスレジスタフォーマット

1 : Arm® Debug Interface V5

ビット 0 **ENTRYPRESENT** : デバッグコンポーネントの有無

アクセスポートバス上のデバッグコンポーネントの有無を示します。

1 : デバッグコンポーネントあり

APx 識別レジスタ (APx_IDR) (x = 0, 1)

アドレスオフセット : 0xFC

リセット値 : AP 0 : 0x5477 0002

リセット値 : AP 1 : 0x8477 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				JEDEC BANK[3:0]				JEDEC CODE[6:0]						CLASS[3]	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLASS[2:0]			Res.	Res.	Res.	Res.	Res.	IDENTITY[7:0]							
r	r	r						r	r	r	r	r	r	r	r

ビット 31:28 **REVISION[3:0]** : リビジョン番号

0x5 : r1p0

0x8 : r0p9

ビット 27:24 **JEDEC BANK[3:0]** : JEDEC バンク

0x4 : Arm®

ビット 23:17 **JEDEC CODE[6:0]** : JEDEC コード

0x3B : Arm®

ビット 16:13 **CLASS[3:0]** :

0x1 : MEM-AP

ビット 12:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **IDENTITY[7:0]** :

0x1 : AHB-AP

0x2 : APB-AP

58.4.2 アクセスポートレジスタマップ

これらのレジスタは CPU メモリバス上にはありません。SW-DP および JTAG-DP のデバッグインタフェースからのみアクセスできます。

アクセスポートアドレスは 8 ビット幅で、DP_SELECTR.APBANKSEL[3:0] フィールドならびに JTAG-DP レジスタの DPACC または SW-DP パケットリクエストの A[3:2] フィールドで定義されます。

表 698. アクセスポートのレジスタマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
0x00	AP0_CSWR	DBGSWEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[3:0]				TRINPROG	DEVICEN	ADDRINC[1:0]		Res.	SIZE[2:0]																
	リセット値	1																				0	0	0	0	0	0	1	0	0	0	1	0														
0x00	AP1_CSWR	Res.	SPROT	Res.	Res.	PROT[3:0]				SPISTATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[3:0]				TRINPROG	DBGSTATUS	ADDRINC[1:0]		Res.	SIZE[2:0]																
	リセット値		1			0	0	1	1	X												0	0	0	0	0	0	X	0	0	0	0	1	0													
0x04	APx_TAR	TA[31:0]																																													
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X														
0x08	予約済みです。	予約済みです。																																													
0x0C	APx_DRWR	TD[31:0]																																													
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X														
0x10	APx_BD0R	DATA[31:0]																																													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0x14	APx_BD1R	DATA[31:0]																																													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0x18	APx_BD2R	DATA[31:0]																																													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0x1C	APx_BD3R	DATA[31:0]																																													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0x20 から 0xF4	予約済みです。	予約済みです。																																													
0xF8	AP0_BASER	BASEADDR[31:12]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0											1	1													
0xF8	AP1_BASER	BASEADDR[31:12]																				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0											1	1												
0xFC	AP0_IDR	REVISION[3:0]			JEDEC BANK[3:0]			JEDEC CODE[6:0]						CLASS[3:0]			Res.	Res.	Res.	Res.	Res.	IDENTITY[7:0]																									
	リセット値	0	1	0	1	0	1	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0					0	0	0	0	0	0	0	1	0												

表 698. アクセスポートのレジスタマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFC	AP0_IDR	REVISION[3:0]				JEDEC BANK[3:0]				JEDEC CODE[6:0]				CLASS[3:0]				Res.	Res.	Res.	Res.	Res.	IDENTITY[7:0]										
	リセット値	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0						0	0	0	0	0	0	0	1

58.5 ROM テーブル

ROM テーブルは CoreSight™ コンポーネントの一つで、設置先のアクセスポートを介してアクセス可能な CoreSight™ デバッグコンポーネントのベースアドレスを格納します。このようなテーブルを使用することにより、デバッグは CoreSight™ システムのトポロジを自動検出することができます。

各アクセスポート (APn) の背後には、上位レベルの 1 つの ROM テーブルがあります。この ROM テーブルのベースアドレスは、アクセスポートの APn_BASER レジスタを読み出すことで取得できます。上位レベルの ROM テーブルは、順に次の ROM テーブルをポイントすることができます。

システム ROM テーブルは、AP0 ベースレジスタ AP0_BASER によりポイントされます。ここには、DBGMCU のベースアドレスポインタが格納されます。

システム ROM テーブルは、デバッグからアクセスしたときに 4 KB、すなわちアドレス空間の 32 ビット幅の領域 (0xE00E 0000 ~ 0xE00E 0FFC) を占有します。0x4402 0000 から 0x4402 0FFC までのアドレス範囲で CPU によってアクセスできます。

表 699. システム ROM テーブル

ROM テーブルの アドレスオフセット	コンポーネント名	コンポーネントベース アドレス	コンポーネント アドレスオフセット	サイズ (KB)	エントリ
0x000	DBGMCU	0xE00E 4000 (デバッグ) 0x4402 4000 (CPU)	0x0000 4000	4	0x0000 4003
0x004	テーブルの先頭	-	-	-	0x0000 0000
0x008 から 0xFC8	予約済みです。	-	-	-	0x0000 0000
0xFCC から 0xFFC	ROM テーブルレ ジスタ	-	-	-	表 702 を参照

CPU サブシステムには 2 つの ROM テーブルがあります。MCU ROM テーブルは、AP1 ベースレジスタ AP1_BASER によりポイントされます。これには、プロセッサ ROM テーブルおよび TPIU レジスタのベースアドレスポインタが格納されます。

MCU ROM テーブル (以下の表を参照) は 4 KB、すなわちアドレス空間の 32 ビット幅の領域 (0xE00F E000 ~ 0xE00F EFFC) を占有します。

表 700. MCU ROM テーブル

ROM テーブルの アドレスオフセット	コンポーネント名	コンポーネント ベースアドレス	コンポーネント アドレスオフセット	サイズ (KB)	エントリ
0x000	プロセッサ ROM テーブル	0xE00F F000	0x0000 1000	4	0x0000 1003
0x004	TPIU	0xE004 0000	0xFFFF 2000	4	0xFFFF 2003

表 700. MCU ROM テーブル (続き)

ROM テーブルの アドレスオフセット	コンポーネント名	コンポーネント ベースアドレス	コンポーネント アドレスオフセット	サイズ (KB)	エントリ
0x008	予約済みです。	-	-	-	0x1FF0 2002
0x00C	予約済みです。	-	-	-	0x1FF0 2002
0x010	テーブルの先頭	-	-	-	0x0000 0000
0x014 から 0xFC8	予約済みです。	-	-	-	0x0000 0000
0xFCC から 0xFFC	ROM テーブルレジスタ	-	-	-	表 703 を参照

プロセッサ ROM テーブルには、ベースアドレスポインタが、システム制御空間 (SCS) レジスタ用に加え、BPU、DWT、ITM、ETM、および CTI 用に用意されています。これにより、デバッグは CPU コアを識別可能になります。

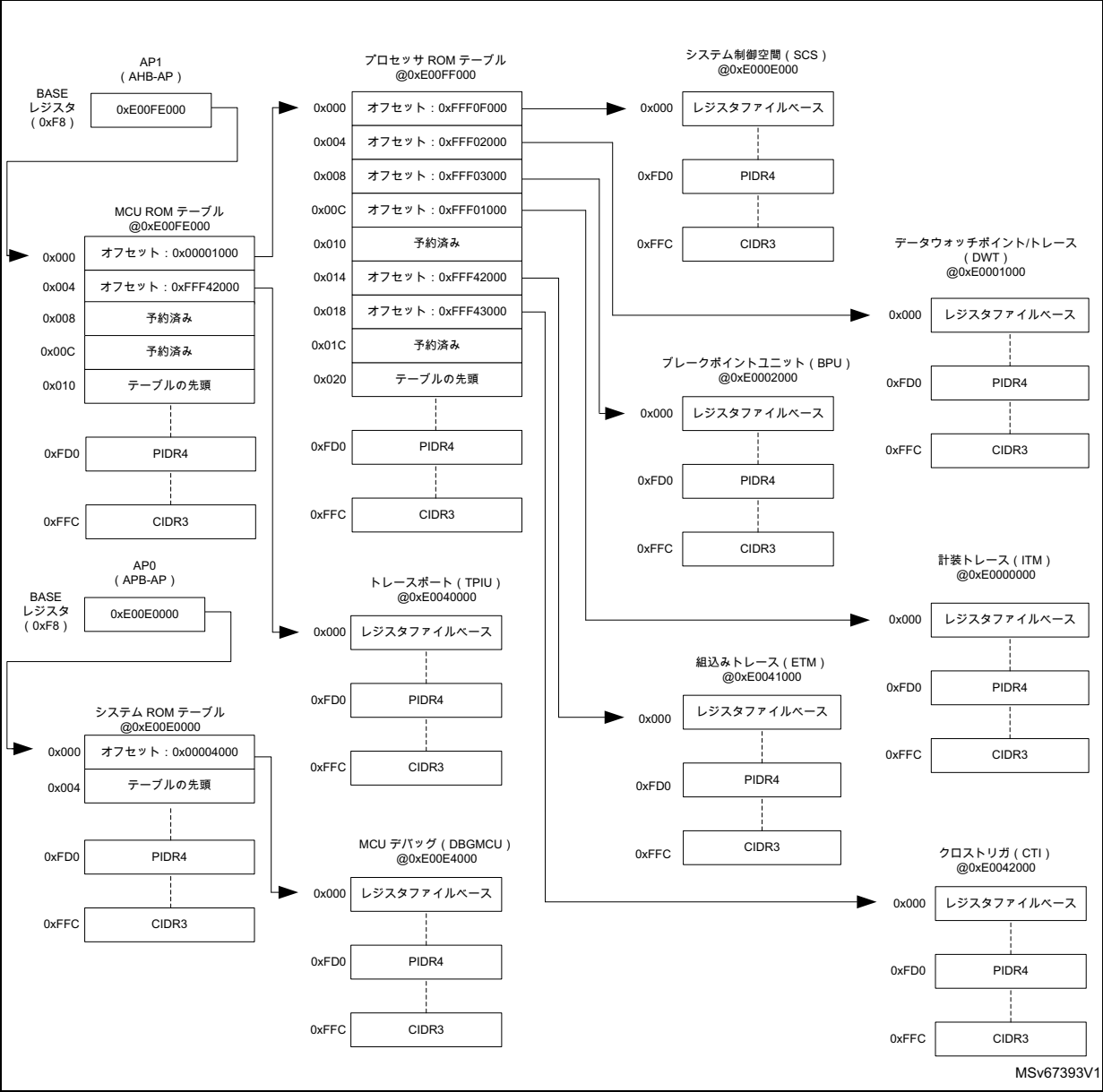
プロセッサ ROM テーブル (以下の表を参照) は 4 KB、すなわちアドレス空間の 32 ビット幅の領域 (0xE00F F000 ~ 0xE00F FFFC) を占有します。

表 701. プロセッサ ROM テーブル

ROM テーブルの アドレス	コンポーネント名	コンポーネント ベースアドレス	コンポーネント アドレスオフセット	サイズ (KB)	エントリ
0xE00F F000	SCS	0xE000 E000	0xFFFF0 F000	4	0xFFFF0 F003
0xE00F F004	DWT	0xE000 1000	0xFFFF0 2000	4	0xFFFF0 2003
0xE00F F008	BPU	0xE000 2000	0xFFFF0 3000	4	0xFFFF0 3003
0xE00F F00C	ITM	0xE000 0000	0xFFFF0 1000	4	0xFFFF0 1003
0xE00F F010	予約済みです。	-	-	-	0xFFFF4 1002
0xE00F F014	ETM	0xE004 1000	0xFFFF4 2000	4	0xFFFF4 2003
0xE00F F018	CTI	0xE004 2000	0xFFFF4 3000	4	0xFFFF4 3003
0xE00F F01C	予約済みです。	-	-	-	0xFFFF4 4002
0xE00F F020	テーブルの先頭	-	-	-	0x0000 0000
0xE00F F024 ~ 0xE00F FFC8	予約済みです。	-	-	-	0x0000 0000
0xE00F FFCC ~ 0xE00F FFFC	ROM テーブルレジスタ	-	-	-	表 704 を参照

CoreSight™ コンポーネントのトポロジを下図に示します。

図 832. CoreSight トポロジ



MSv67393V1

58.5.1 システム ROM テーブルレジスタ

システム ROM メモリタイプレジスタ (SYSROM_MEMTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSTEM
															r

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **SYSTEM** : システムメモリ

0x1 : このバス上にシステムメモリがあります。

システム ROM CoreSight ペリフェラル ID レジスタ 4 (SYSROM_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタ ファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x0 : STMicroelectronics JEDEC 継続コード

システム ROM CoreSight ペリフェラル ID レジスタ 0 (SYSROM_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x84 : STM32H563/H573および STM32H562

システム ROM CoreSight ペリフェラル ID レジスタ 1 (SYSROM_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0x0 : STMicroelectronics JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0x4 : STM32H563/H573および STM32H562

システム ROM CoreSight ペリフェラル ID レジスタ 2 (SYSROM_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : リビジョン r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x2 : STMicroelectronics JEDEC コード

システム ROM CoreSight ペリフェラル ID レジスタ 3 (SYSROM_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

システム ROM CoreSight コンポーネント ID レジスタ 0 (SYSROM_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

システム ROM CoreSight ペリフェラル ID レジスタ 1 (SYSROM_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント識別ビット [15:12] - コンポーネントクラス

0x1 : ROM テーブルコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

システム ROM CoreSight コンポーネント ID レジスタ 2 (SYSROM_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

システム ROM CoreSight コンポーネント ID レジスタ 3 (SYSROM_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.5.2 システム ROM テーブルレジスタマップ

表 702. システム ROM テーブルレジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0xFCC	SYSROM_MEMTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	1		
	リセット値																																	1		
0xFD0	SYSROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]			JEP106CON [3:0]							
	リセット値																									0	0	0	0	0	0	0	0			
0xFD4 ~ FDC	予約済みです。	予約済みです。																																		
0xFE0	SYSROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]										
	リセット値																									X	X	X	X	X	X	X	X			
0xFE4	SYSROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]			PARTNUM [11:8]							
	リセット値																									0	0	0	0	X	X	X	X			
0xFE8	SYSROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]	JEDEC			JEP106ID [6:4]						
	リセット値																							0	0								0	0	1	0
0xFEC	SYSROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]			CMOD[3:0]							
	リセット値																									0	0	0	0	0	0	0	0			
0xFF0	SYSROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]										
	リセット値																									0	0	0	0	1	1	0	1			
0xFF4	SYSROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]			PREAMBLE [11:8]							
	リセット値																									0	0	0	1	0	0	0	0			
0xFF8	SYSROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]										
	リセット値																									0	0	0	0	0	1	0	1			
0xFFC	SYSROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]										
	リセット値																									1	0	1	1	0	0	0	1			

レジスタ境界アドレスについては表 699 : システム ROM テーブルを参照してください。

58.5.3 MCU ROM テーブルレジスタ

MCU ROM メモリタイプレジスタ (MCUROM_MEMTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSTEM
															r

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **SYSTEM** : システムメモリ

0x1 : このバス上にシステムメモリがあります。

MCU ROM CoreSight ペリフェラル ID レジスタ 4 (MCUROM_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタ ファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x0 : STMicroelectronics JEDEC 継続コード

MCU ROM CoreSight ペリフェラル ID レジスタ 0 (MCUROM_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x84 : STM32H563/H573および STM32H562

MCU ROM CoreSight ペリフェラル ID レジスタ 1 (MCUROM_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0x0 : STMicroelectronics JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0x4 : STM32H563/H573および STM32H562

MCU ROM CoreSight ペリフェラル ID レジスタ 2 (MCUROM_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : リビジョン r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x2 : STMicroelectronics JEDEC コード

MCU ROM CoreSight ペリフェラル ID レジスタ 3 (MCUROM_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

MCU ROM CoreSight コンポーネント ID レジスタ 0 (MCUROM_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

MCU ROM CoreSight ペリフェラル ID レジスタ 1 (MCUROM_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント識別ビット [15:12] - コンポーネントクラス

0x1 : ROM テーブルコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

MCU ROM CoreSight コンポーネント ID レジスタ 2 (MCUROM_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

MCU ROM CoreSight コンポーネント ID レジスタ 3 (MCUROM_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.5.4 MCU ROM テーブルレジスタマップ

表 703. MCU ROM テーブルレジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0xFCC	MCUROM_ MEMTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																	1		
0xFD0	MCUROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]				JEP106CON [3:0]						
	リセット値																									0	0	0	0	0	0	0	0			
0xFD4 ~ FDC	予約済みです。	予約済みです。																																		
0xFE0	MCUROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]										
	リセット値																									X	X	X	X	X	X	X	X			
0xFE4	MCUROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]						
	リセット値																									0	0	0	0	X	X	X	X			
0xFE8	MCUROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]	JEDEC		JEP106ID [6:4]							
	リセット値																							0	0				0	0	1	0	1	0		
0xFEC	MCUROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]						
	リセット値																									0	0	0	0	0	0	0	0			
0xFF0	MCUROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]										
	リセット値																									0	0	0	0	0	1	1	0	1		
0xFF4	MCUROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]						
	リセット値																									0	0	0	1	0	0	0	0			
0xFF8	MCUROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]										
	リセット値																									0	0	0	0	0	1	0	1			
0xFFC	MCUROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]										
	リセット値																									1	0	1	1	0	0	0	1			

レジスタ境界アドレスについては表 700 : MCU ROM テーブルを参照してください。

58.5.5 プロセッサ ROM テーブルレジスタ

CPU ROM メモリタイプレジスタ (CPUROM_MEMTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSTEM
															r

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **SYSTEM** : システムメモリ

1 : このバス上にシステムメモリがあります。

CPU ROM CoreSight ペリフェラル ID レジスタ 4 (CPUROM_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタ ファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : ARM® JEDEC 継続コード

CPU ROM CoreSight ペリフェラル ID レジスタ 0 (CPUROM_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 00C9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0xC9 : Cortex®-M33

CPU ROM CoreSight ペリフェラル ID レジスタ 1 (CPUROM_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : ARM® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0x4 : Cortex®-M33

CPU ROM CoreSight ペリフェラル ID レジスタ 2 (CPUROM_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : リビジョン r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

1 : JEDEC によって指定される設計者 ID

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

CPU ROM CoreSight ペリフェラル ID レジスタ 3 (CPUROM_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

CPU ROM CoreSight コンポーネント ID レジスタ 0 (CPUROM_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

CPU ROM CoreSight ペリフェラル ID レジスタ 1 (CPUROM_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント識別ビット [15:12] - コンポーネントクラス

0x1 : ROM テーブルコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

CPU ROM CoreSight コンポーネント ID レジスタ 2 (CPUROM_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

CPU ROM CoreSight コンポーネント ID レジスタ 3 (CPUROM_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.5.6 プロセッサ ROM テーブルレジスタマップ

表 704. CPU ROM テーブルレジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0xFCC	CPUROM_MEMENTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSTEM
	リセット値																																	1
0xFD4 ~ FDC	予約済みです。	予約済みです。																																
0xFD0	CPUROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]			JEP106CON [3:0]					
	リセット値																									0	0	0	0	0	1	0	0	
0xFE0	CPUROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]								
	リセット値																									1	1	0	0	1	0	0	1	
0xFE4	CPUROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]			PARTNUM [11:8]					
	リセット値																									1	0	1	1	0	1	0	0	
0xFE8	CPUROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]			JEDEC	JEP106ID [6:4]				
	リセット値																									0	0	0	0	1	0	1	1	
0xFEC	CPUROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]			CMOD[3:0]					
	リセット値																									0	0	0	0	0	0	0	0	
0xFF0	CPUROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]								
	リセット値																									0	0	0	0	1	1	0	1	
0xFF4	CPUROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]			PREAMBLE [11:8]					
	リセット値																									0	0	0	1	0	0	0	0	
0xFF8	CPUROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]								
	リセット値																									0	0	0	0	0	1	0	1	
0xFFC	CPUROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]								
	リセット値																									1	0	1	1	0	0	0	1	

レジスタ境界アドレスについては表 701：プロセッサ ROM テーブルを参照してください。

58.6 データウォッチポイントおよびトレースユニット (DWT)

DWT には 4 つのコンパレータがあり、それぞれ以下の役割に使用できます。

- ウォッチポイント
- ETM トリガ
- PC サンプリングトリガ
- データアドレスサンプリングトリガ
- データコンパレータ (COMP 1 のみ)
- クロックサイクルカウンタコンパレータ (COMP 0 のみ)

DWT は以下をカウントするためのカウンタも内蔵しています。

- クロックサイクル数
- フォールドされた命令数
- ロードストアユニット (LSU) の動作数
- スリープサイクル数
- 命令当たりのサイクル数
- 割込みオーバーヘッドの回数

DWT コンパレータは、[DWT コンパレータ x レジスタ \(DWT_COMPxR\)](#) に保持されている値と次のいずれかを比較します。

- データアドレス
- 命令アドレス
- データ値
- サイクルカウント値 (COMP 0 のみ)

アドレス照合の場合、コンパレータはマスクを使用することができるため、あるレンジのアドレスと一致します。

照合が成立すると、コンパレータは以下のうちの 1 つを生成します。

- 1つ以上の DWT データトレースパケットで、以下を 1 つ以上含むもの。
 - データアクセスを伴う命令のアドレス
 - アドレスオフセット (データアクセスアドレスのビット [15:0])
 - 一致したデータ値
- PC 値またはアクセスしたデータアドレスのいずれかで発生するウォッチポイントデバッグイベント
- DWT ユニット外での一致を信号で伝える CMPMATCH[N] イベント

ウォッチポイントデバッグイベントが発生すると、DebugMonitor 例外が生成されるか、またはプロセッサが停止してデバッグステートに入ります。

DWT の使用方法の詳細については、Arm[®] v8-M Architecture Reference Manual を参照してください [\[4\]](#)。

58.6.1 DWT レジスタ

DWT レジスタはアドレスレンジ 0xE000 1000 ~ 0xE000 1FFC に配置されています。

DWT 制御レジスタ (DWT_CTRLR)

アドレスオフセット : 0x000

リセット値 : 0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCOMP[3:0]				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	CYCDISS	CYCEVTENA	FOLDEVTEANA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA
r	r	r	r	r	r	r	r	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSTINIT[3:0]				POSTRESET[3:0]				CYCCNTENA
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 **NUMCOMP[3:0]** : 実装されるコンパレータの数 (読出し専用)

0x4 : 4 つのコンパレータ

ビット 27 **NOTRCPKT** : トレースサンプリングおよび例外トレースのサポート (読出し専用)

0 : サポートされます。

ビット 26 **NOEXTTRIG** : 外部一致信号、CMPMATCH のサポート (読出し専用)

0 : サポートされます。

ビット 25 **NOCYCCNT** : サイクルカウンタのサポート (読出し専用)

0 : サポートされます。

ビット 24 **NOPRFCNT** : プロファイリングカウンタのサポート (読出し専用)

0 : サポートされます。

ビット 23 **CYCDISS** : セキュアなサイクルカウンタの無効化

サイクルカウンタをセキュアモードで無効化するかどうかを制御します。

0 : 影響なし。

1 : プロセッサがセキュアステートである場合にサイクルカウンタのインクリメントを無効にします。

ビット 22 **CYCEVTENA** : POSTCNT アンダーフローイベントカウンタのパケット生成の有効化

0 : 無効

1 : 有効

ビット 21 **FOLDEVTEANA** : フォールドされた命令数カウンタのオーバーフローイベント生成の有効化

0 : 無効

1 : 有効

ビット 20 **LSUEVTENA** : LSU カウンタのオーバーフローイベント生成の有効化

0 : 無効

1 : 有効

ビット 19 **SLEEPEVTENA** : スリープカウンタのオーバーフローイベント生成の有効化

0 : 無効

1 : 有効

ビット 18 **EXCEVTENA** : 例外オーバーヘッドカウンタのオーバーフローイベント生成の有効化

0 : 無効

1 : 有効

ビット 17 **CPIEVTENA** : CPI カウンタのオーバーフローイベント生成の有効化

0 : 無効

1 : 有効

ビット 16 **EXCTRCENA** : 例外トレース生成の有効化

0 : 無効

1 : 有効

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **PCSAMPLENA** : 周期的 PC サンプルパケット生成用のタイマとして使用するための POSTCNT カウンタの有効化

0 : 無効

1 : 有効

ビット 11:10 **SYNCTAP[1:0]** : CYCCNT カウンタ上の同期パケットカウンタのタップ位置

ここでのフィールドが、同期パケットの速度を決定します。

00 : 無効、同期パケットはありません。

01 : CYCCNT[24] でタップします。

10 : CYCCNT[26] でタップします。

11 : CYCCNT[28] でタップします。

ビット 9 **CYCTAP** : **CYCCNT カウンタ上の POSTCNT のタップ位置の選択**

0 : CYCCNT[6] でタップします。

1 : CYCCNT[10] でタップします。

ビット 8:5 **POSTINIT[3:0]** : POSTCNT カウンタの初期値

このフィールドへの書込みは POSTCNT カウンタが有効であれば、無視されます。POSTINIT を書き込む前に CYCEVTENA または PCSAMPLENA をリセットする必要があります。

ビット 4:1 **POSTRESET[3:0]** : POSTCNT カウンタの値の再ロード

ビット 0 **CYCCNTENA** : CYCCNT カウンタ有効化

0 : 無効

1 : 有効

DWT サイクルカウントレジスタ (DWT_CYCCNTR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CYCCNT[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCCNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **CYCCNT[31:0]** : プロセッサのクロックサイクルカウンタ

DWT CPI カウントレジスタ (DWT_CPICNTR)

アドレスオフセット : 0x008

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **CPICNT[7:0]** : CPI カウンタ

複数サイクルの命令を実行するために必要な追加サイクル数 (DWT_LSUCNTR によって記録されるものを除く) をカウントし、命令フェッチストールがあればカウントします。

DWT 例外カウントレジスタ (DWT_EXCCNTR)

アドレスオフセット : 0x00C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **EXCCNT[7:0]** : 例外オーバーヘッドサイクルカウンタ

例外処理に費やされるサイクル数をカウントします。

DWT スリープカウントレジスタ (DWT_SLPCNTR)

アドレスオフセット : 0x010

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEP CNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **SLEEP CNT[7:0]** : スリープサイクルカウンタ

SLEEPモードに費やされるサイクル数をカウントします (WFI、WFE、Sleep-on-exit)。

DWT LSU カウントレジスタ (DWT_LSUCNTR)

アドレスオフセット : 0x014

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **LSUCNT[7:0]** : ロードストアカウンタ

ロードおよびストア命令を実行するために必要な追加サイクル数をカウントします。

DWT フォールドカウントレジスタ (DWT_FOLDCNTR)

アドレスオフセット : 0x018

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDCNT[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **FOLDCNT[7:0]** : フォールドされた命令数のカウンタ
 サイクル数 0 の命令のたびにインクリメントします。

DWT プログラムカウンタサンプルレジスタ (DWT_PCSR)

アドレスオフセット : 0x01C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIASAMPLE[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIASAMPLE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **EIASAMPLE[31:0]** : 実行された命令アドレスのサンプル値
 プログラムカウンタの現在値をサンプリングします。

DWT コンパレータ x レジスタ (DWT_COMPxR)

アドレスオフセット : 0x020 + 0x010 * x、(x = 0 から 3)

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **COMP[31:0]** : 比較のための基準値

DWT 機能レジスタ 0 (DWT_FUNCTR0)

アドレスオフセット : 0x028

リセット値 : 0x5800 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[4:0]					Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r			r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DATAVSIZE[1:0]		Res.	Res.	Res.	Res.	ACTION[1:0]		MATCH[3:0]			
				rw	rw					rw	rw	rw	rw	rw	rw

ビット 31:27 **ID[4:0]** : 機能識別

コンパレータ 0 に適した機能を識別します。

0b01011 : サイクルカウンタ、命令アドレス、データアドレス、および値が入力されたデータアドレス

ビット 26:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **MATCHED** : コンパレータ一致

レジスタが最後に読み出されたあと、コンパレータの一致が発生したかどうかを示します。

0 : 一致していません。

1 : 一致が発生しました。

ビット 23:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **DATAVSIZE[1:0]** : データ値のサイズ

データ値およびデータアドレスのコンパレータによって監視されているオブジェクトのサイズを定義します。

0x0 : 1 バイト

0x1 : 2 バイト

0x2 : 4 バイト

0x3 : 予約済み

ビット 9:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ACTION[1:0]** : 一致時のアクション

0x0 : トリガのみ。

0x1 : デバッグイベントを生成します。

0x2 : サイクルカウンタ、命令アドレス、データアドレス、データ値、またはリンクされたデータ値のコンパレータ用に、データトレース一致パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレースデータ値パケットを生成します。

0x3 : データアドレス制限コンパレータ用に、データトレース一致パケットを生成します。サイクルカウンタ、命令アドレス制限、またはデータアドレスのコンパレータ用に、データトレース PC 値パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレース PC 値パケットとデータトレースデータ値パケットの両方を生成します。

ビット 3:0 **MATCH[3:0]** : 一致タイプ

コンパレータ 0 によって生成された一致タイプを制御します。

このフィールドに設定可能な値については、[\[4\]](#) を参照してください。

DWT 機能レジスタ 1 (DWT_FUNCTR1)

アドレスオフセット : 0x038

リセット値 : 0xD000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[4:0]					Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r			r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DATAVSIZE[1:0]		Res.	Res.	Res.	Res.	ACTION[1:0]		MATCH[3:0]			
				rw	rw					rw	rw	rw	rw	rw	rw

ビット 31:27 ID[4:0] : 機能識別

コンパレータ 1 に適した機能を識別します。

0b11010 : 命令アドレス、命令アドレス制限、データアドレス、データアドレス制限、および値が入力されたデータアドレス

ビット 26:25 予約済みであり、リセット値に保持する必要があります。**ビット 24 MATCHED** : コンパレータの一致

レジスタが最後に読み出されたあと、コンパレータの一致が発生したかどうかを示します。

0 : 一致していません。

1 : 一致が発生しました。

ビット 23:12 予約済みであり、リセット値に保持する必要があります。**ビット 11:10 DATAVSIZE[1:0]** : データ値のサイズ

データ値およびデータアドレスのコンパレータによって監視されているオブジェクトのサイズを定義します。

0x0 : 1 バイト

0x1 : 2 バイト

0x2 : 4 バイト

0x3 : 予約済み

ビット 9:6 予約済みであり、リセット値に保持する必要があります。**ビット 5:4 ACTION[1:0]** : 一致時のアクション

0x0 : トリガのみ。

0x1 : デバッグイベントを生成します。

0x2 : サイクルカウンタ、命令アドレス、データアドレス、データ値、またはリンクされたデータ値のコンパレータ用に、データトレース一致パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレースデータ値パケットを生成します。

0x3 : データアドレス制限コンパレータ用に、データトレース一致パケットを生成します。サイクルカウンタ、命令アドレス制限、またはデータアドレスのコンパレータ用に、データトレース PC 値パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレース PC 値パケットとデータトレースデータ値パケットの両方を生成します。

ビット 3:0 MATCH[3:0] : 一致タイプ

コンパレータ 1 によって生成された一致タイプを制御します。

このフィールドに設定可能な値については、[\[4\]](#) を参照してください。

DWT 機能レジスタ 2 (DWT_FUNCTR2)

アドレスオフセット : 0x048

リセット値 : 0x5000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[4:0]					Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r			r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DATAVSIZE[1:0]		Res.	Res.	Res.	Res.	ACTION[1:0]		MATCH[3:0]			
				rw	rw					rw	rw	rw	rw	rw	rw

ビット 31:27 **ID[4:0]** : 機能識別

コンパレータ 2 に適した機能を識別します。

0b01010 : 命令アドレス、データアドレス、および値が入力されたデータアドレス

ビット 26:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **MATCHED** : コンパレータ一致

レジスタが最後に読み出されたあと、コンパレータの一致が発生したかどうかを示します。

0 : 一致していません。

1 : 一致が発生しました。

ビット 23:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **DATAVSIZE[1:0]** : データ値サイズ :

データ値およびデータアドレスのコンパレータによって監視されているオブジェクトのサイズを定義します。

0x0 : 1 バイト

0x1 : 2 バイト

0x2 : 4 バイト

0x3 : 予約済み

ビット 9:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ACTION[1:0]** : 一致時のアクション

0x0 : トリガのみ。

0x1 : デバッグイベントを生成します。

0x2 : サイクルカウンタ、命令アドレス、データアドレス、データ値、またはリンクされたデータ値のコンパレータ用に、データトレース一致パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレースデータ値パケットを生成します。

0x3 : データアドレス制限コンパレータ用に、データトレース一致パケットを生成します。サイクルカウンタ、命令アドレス制限、またはデータアドレスのコンパレータ用に、データトレース PC 値パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレース PC 値パケットとデータトレースデータ値パケットの両方を生成します。

ビット 3:0 **MATCH[3:0]** : 一致タイプ

コンパレータ 2 によって生成された一致タイプを制御します。

このフィールドに設定可能な値については、[\[4\]](#) を参照してください。

DWT 機能レジスタ 3 (DWT_FUNCTR3)

アドレスオフセット : 0x058

リセット値 : 0xF000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID[4:0]					Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r			r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DATAVSIZE[1:0]		Res.	Res.	Res.	Res.	ACTION[1:0]		MATCH[3:0]			
				rw	rw					rw	rw	rw	rw	rw	rw

ビット 31:27 **ID[4:0]** : 機能識別

コンパレータ 2 に適した機能を識別します。

0b11110 : 命令アドレス、命令アドレス制限、データアドレス、データアドレス制限、データ値、リンクされたデータ値、および値が入力されたデータアドレス

ビット 26:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **MATCHED** : コンパレータ一致

レジスタが最後に読み出されたあと、コンパレータの一致が発生したかどうかを示します。

0 : 一致していません。

1 : 一致が発生しました。

ビット 23:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **DATAVSIZE[1:0]** : データ値のサイズ

データ値およびデータアドレスのコンパレータによって監視されているオブジェクトのサイズを定義します。

0x0 : 1 バイト

0x1 : 2 バイト

0x2 : 4 バイト

0x3 : 予約済み

ビット 9:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **ACTION[1:0]** : 一致時のアクション

0x0 : トリガのみ。

0x1 : デバッグイベントを生成します。

0x2 : サイクルカウンタ、命令アドレス、データアドレス、データ値、またはリンクされたデータ値のコンパレータ用に、データトレース一致パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレースデータ値パケットを生成します。

0x3 : データアドレス制限コンパレータ用に、データトレース一致パケットを生成します。サイクルカウンタ、命令アドレス制限、またはデータアドレスのコンパレータ用に、データトレース PC 値パケットを生成します。値が入力されたデータアドレスのコンパレータ用に、データトレース PC 値パケットとデータトレースデータ値パケットの両方を生成します。

ビット 3:0 **MATCH[3:0]** : 一致タイプ

コンパレータ 2 によって生成された一致タイプを制御します。

このフィールドに設定可能な値については、[\[4\]](#) を参照してください。

DWT デバイスタイプアーキテクチャレジスタ (DWT_DEVARCHR)

アドレスオフセット : 0xFC8

リセット値 : 0x4770 : 1A02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARCHITECT[10:0]											PRESENT	REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARCHVER[3:0]					ARCHPART[11:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:21 **ARCHITECT[10:0]** : アーキテクチャ JEP106 コード

0x23B : JEP106 継続コード 0x4、JEP106 ID コード 0x3BArm® 限定

ビット 20 **PRESENT** : DWT_DEVARCH レジスタ有無

0x1 : あり

ビット 19:16 **REVISION[3:0]** : アーキテクチャのリビジョン

0x0 : DWT アーキテクチャ v2.0

ビット 15:12 **ARCHVER[3:0]** : アーキテクチャバージョン

0x1 : DWT アーキテクチャ v2.0

ビット 11:0 **ARCHPART[11:0]** : アーキテクチャパート

0xA02 : DWT アーキテクチャ

DWT デバイスタイプレジスタ (DWT_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUB[3:0]** : サブタイプ

0x0 : その他

ビット 3:0 **MAJOR[3:0]** : メジャータイプ

0x0 : その他

DWT CoreSight ペリフェラル ID レジスタ 4 (DWT_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : Arm® JEDEC コード

DWT CoreSight ペリフェラル ID レジスタ 0 (DWT_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : Cortex®-M33 DWT 部品番号

DWT CoreSight ペリフェラル ID レジスタ 1 (DWT_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : Cortex®-M33 DWT 部品番号

DWT CoreSight ペリフェラル ID レジスタ 2 (DWT_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのレビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

DWT CoreSight ペリフェラル ID レジスタ 3 (DWT_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

DWT CoreSight コンポーネント ID レジスタ 0 (DWT_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

DWT CoreSight ペリフェラル ID レジスタ 1 (DWT_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0x9 : デバッグコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

DWT CoreSight コンポーネント ID レジスタ 2 (DWT_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

DWT CoreSight コンポーネント ID レジスタ 3 (DWT_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値



58.6.2 DWT レジスタマップ

DWT レジスタはアドレスレンジ 0xE000 1000 ~ 0xE000 1FFC に配置されています。

表 705. DWT レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DWT_CTRLR	NUMCOMP[3:0]				NOTRCPKT	NOEXTTRIG	NOCYCNT	NOPRFCNT	CYCDISS	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXTRCENA	Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]	CYCTAP	POSTINIT[3:0]				POSTPRESET[3:0]				CYCCNTENA	
	リセット値	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	DWT_CYCCNTR	CYCCNT[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	DWT_CPICNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]								
	リセット値																									X	X	X	X	X	X	X	X
0x00C	DWT_EXCCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]								
	リセット値																									X	X	X	X	X	X	X	X
0x010	DWT_SLPCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEPcnt[7:0]								
	リセット値																									X	X	X	X	X	X	X	X
0x014	DWT_LSUCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]								
	リセット値																									X	X	X	X	X	X	X	X
0x018	DWT_FOLDcnTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDcnt[7:0]								
	リセット値																									X	X	X	X	X	X	X	X
0x01C	DWT_PCSR	EIASAMPLE[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x020	DWT_COMP0R	COMP[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x024	予約済みです。	予約済みです。																															
0x028	DWT_FUNCTR0	ID[4:0]				Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA/size[1:0]	Res.	Res.	Res.	Res.	ACTION[1:0]	MATCH[3:0]					
	リセット値	0	1	0	1	1		0														0	0				0	0	0	0	0	0	0
0x02C	予約済みです。	予約済みです。																															
0x030	DWT_COMP1R	COMP[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x034	予約済みです。	予約済みです。																															
0x038	DWT_FUNCTR1	ID[4:0]				Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA/size[1:0]	Res.	Res.	Res.	Res.	ACTION[1:0]	MATCH[3:0]					
	リセット値	1	1	0	1	0		0														0	0				0	0	0	0	0	0	0
0x03C	予約済みです。	予約済みです。																															
0x040	DWT_COMP2R	COMP[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x044	予約済みです。	予約済みです。																															

表 705. DWT レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x048	DWT_FUNCTR2	ID[4:0]								MATCHED													DATA/SIZE[1:0]						ACTION[1:0]	MATCH[3:0]									
	リセット値	0	1	0	1	0			0													0	0					0	0	0	0	0	0						
0x04C	予約済みです。	予約済みです。																																					
0x050	DWT_COMP3R	COMP[31:0]																																					
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X						
0x054	予約済みです。	予約済みです。																																					
0x058	DWT_FUNCTR3	ID[4:0]								MATCHED													DATA/SIZE[1:0]						ACTION[1:0]	MATCH[3:0]									
	リセット値	1	1	1	1	0			0													0	0					0	0	0	0	0	0						
0x05C から 0xFC4	予約済みです。	予約済みです。																																					
0xFC8	DWT_DEVARCHR	ARCHITECT[10:0]										PRESENT		REVISION[3:0]				ARCHVER[3:0]			ARCHPART[11:0]																		
	リセット値	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0						
0xFCC	DWT_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]										
	リセット値																									0	0	0	0	0	0	0	0						
0xFD0	DWT_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]								
	リセット値																										0	0	0	0	0	1	0	0					
0xFD4 から 0xFDC	予約済みです。	予約済みです。																																					
0xFE0	DWT_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]													
	リセット値																										0	0	1	0	0	0	0	1					
0xFE4	DWT_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]									
	リセット値																										1	0	1	1	1	1	0	1					
0xFE8	DWT_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC		JEP106ID [6:4]							
	リセット値																										0	0	0	0	1	0	1	1					
0xFEC	DWT_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]									
	リセット値																										0	0	0	0	0	0	0	0					
0xFF0	DWT_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]														
	リセット値																										0	0	0	0	1	1	0	1					

表 705. DWT レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF4	DWT_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]				
	リセット値																									1	0	0	1	0	0	0	0
0xFF8	DWT_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]								
	リセット値																									0	0	0	0	0	1	0	1
0xFFC	DWT_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]								
	リセット値																									1	0	1	1	0	0	0	1

レジスタ境界アドレスについては表 701 : プロセッサ ROM テーブルを参照してください。

58.7 計装トレースマクロセル (ITM)

ITM はトレース情報をパケットとして生成します。3 つのソースでパケットを生成できます。パケットが複数のソースから同時に生成される場合、ITM はパケットの出力順番についてアービトレーションを行います。3 つのソースを優先度の高いものから順に挙げます。

- ソフトウェアトレース

ソフトウェアは、32 x 32 ビットの ITM スティムラスレジスタのいずれかに直接書き込みを行って、パケットを生成します。各ポートの許可レベルはプログラムで設定できます。ソフトウェアが有効なスティムラポートに書き込むと、ITM は FIFO に書き込むパケットの中に、ポートの ID、書き込みアクセスのサイズ、および書き込まれたデータを統合します。ITM は FIFO からトレースバスにパケットを出力します。スティムラポートレジスタを読み出すと、ビット 0 のスティムラレジスタのステータス（エンプティまたはペンディング）を返します。

- ハードウェアトレース

DWT はデータトレースイベント、すなわち PC サンプルやパフォーマンスプロファイリングカウンタの折り返しに応じて、トレースパケットを生成します。ITM がこれらのパケットをトレースバスに出力します。

- ローカルタイムスタンプ

ITM は（分周済みの）プロセッサクロックが供給される 21 ビットカウンタを内蔵しています。カウンタの値はトレースバスのタイムスタンプパケットに出力されます。タイムスタンプパケットが生成されるたびに、カウンタはゼロにリセットされます。よって、タイムスタンプは前のタイムスタンプパケットからの経過時間となります。

ITM とその使用方法の詳細については、[4] を参照してください。

58.7.1 ITM レジスタ

ITM レジスタはアドレスレンジ 0xE000 0000 ~ 0xE000 0FFC に配置されています。

ITM スティムラスレジスタ x (ITM_STIMRx)

アドレスオフセット : $0x000 + 0x004 * x$ 、(x = 0 から 31)

リセット値 : 0XXXXX XXXX

条件 : 書込み時

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STIMULUS[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIMULUS[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	rw	rw

ビット 31:0 **STIMULUS[31:0]** : トレース出力データ

書込みデータは、ソフトウェアイベントパケットとしてトレースバスに出力されます。

ITM スティムラスレジスタ x [オルタネート] (ITM_STIMRx)

アドレスオフセット : $0x000 + 0x004 * x$ 、(x = 0 から 31)

リセット値 : 0XXXXX XXXX

条件 : 読出し時

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DISABLE	FIFO_READY
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DISABLE** : 無効化フラグ

0 : スティムラスポートおよび ITM は有効です。

1 : スティムラスポートおよび ITM は無効です。

ビット 0 **FIFO_READY** : FIFO レディインジケータ

0 : スティムラスポートのバッファはフルです (またはポートが無効)。

1 : スティムラスポートは新しい書込みデータを受け付けることができます。

ITM トレース有効レジスタ (ITM_TER)

アドレスオフセット : 0xE00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STIMENA[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIMENA[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **STIMENA[31:0]** : スティムラポートの有効化

各ビット x (0 ~ 31) は、ITM_STIMRx レジスタと関連のあるスティムラポートを有効にします。

0 : ポートを無効化

1 : ポートを有効化

ITM トレース特権レジスタ (ITM_TPR)

アドレスオフセット : 0xE40

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIVMASK[3:0]			
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PRIVMASK[3:0]** : ITM スティムラポートへの非特権のアクセスを無効にします。

各ビットは 8 つのスティムラポートを制御します。

XXX0 : ポート 0~7 では非特権のアクセスが許可されます。

XXX1 : ポート 0~7 では特権アクセスのみが許可されます。

XX0X : ポート 8~15 では非特権のアクセスが許可されます。

XX1X : ポート 8~15 では特権アクセスのみが許可されます。

X0XX : ポート 16~23 では非特権のアクセスが許可されます。

X1XX : ポート 16~23 では特権アクセスのみが許可されます。

0XXX : ポート 24~31 では非特権のアクセスが許可されます。

1XXX : ポート 24~31 では特権アクセスのみが許可されます。

ITM トレース制御レジスタ (ITM_TCR)

アドレスオフセット : 0xE80

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]						
								r	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSPRESCALE[1:0]		Res.	Res.	STALLENA	SWOENA	TXENA	SYNCENA	TSENA	ITMENA
						rw	rw			rw	r	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **BUSY** : ITM が現在イベント処理中かどうかを示します。

0 : 非ビジー

1 : ビジー

ビット 22:16 **TRACEBUSID[6:0]** : マルチソーストレースストリームのフォーマット識別子

マルチソーストレースを使用している場合、デバッガはこのフィールドにゼロ以外の値を書き込まなければなりません。

注 : システム内のトレースソースにはそれぞれ異なる ID を使用してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **TSPRESCALE[1:0]** : トレースパケットのリファレンスクロックとともに使用するローカルタイムスタンププリスケラ

0x0 : プリスケールなし

0x1 : 4 分周

0x2 : 16 分周

0x3 : 64 分周

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **STALLENA** : ストールの有効化

0 : ITM 出力がストールしている場合に、ハードウェアソースパケットをドロップし、オーバーフローを生成します。

1 : プロセッサをストールし、データトレースパケットの供給を保証します。

ビット 4 **SWOENA** : SWO 有効化

タイムスタンプカウンタの非同期クロックを有効にします (読出し専用)。

0 : タイムスタンプカウンタはプロセッサクロックを使用します。

ビット 3 **TXENA** : 送信の有効化

DWT ユニットからトレースポートへのハードウェアイベントパケットの転送を有効にします。

0 : 無効

1 : 有効

ビット 2 **SYNCENA** : 同期パケット送信の有効化

このビットをセットするデバッガは、正しい同期速度にするために、DWT_CTRLR.SYNCTAP フィールドも設定する必要があります。

0 : 無効

1 : 有効

ビット 1 **TSENA** : ローカルタイムスタンプ生成の有効化
 0 : 無効
 1 : 有効

ビット 0 **ITMENA** : ITM 有効化
 0 : 無効
 1 : 有効

ITM デバイスタイプアーキテクチャレジスタ (ITM_DEVARCHR)

アドレスオフセット : 0xFBC
 リセット値 : 0x4770 1A01

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARCHITECT[10:0]											PRESENT	REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARCHVER[3:0]					ARCHPART[11:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:21 **ARCHITECT[10:0]** : アーキテクチャ JEP106 コード
 0x23B : JEP106 継続コード 0x4、JEP106 ID コード 0x3BArm® 限定

ビット 20 **PRESENT** : DEVARCH レジスタの有無
 0x1 : あり

ビット 19:16 **REVISION[3:0]** : アーキテクチャのリビジョン
 0x0 : ITM アーキテクチャ v2.0

ビット 15:12 **ARCHVER[3:0]** : アーキテクチャバージョン
 0x1 : ITM アーキテクチャ v2.0

ビット 11:0 **ARCHPART[11:0]** : アーキテクチャパート
 0xA01 : ITM アーキテクチャ

ITM デバイスタイプレジスタ (ITM_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0043

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUB[3:0]** : サブタイプ

0x4 : バスアクティビティから生成されたスティムラスであるバスに関連します。

ビット 3:0 **MAJOR[3:0]** : メジャータイプ

0x3 : トレースソース

ITM CoreSight ペリフェラル ID レジスタ 4 (ITM_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : Arm® JEDEC コード

ITM CoreSight ペリフェラル ID レジスタ 0 (ITM_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : ITM 部品番号

ITM CoreSight ペリフェラル ID レジスタ 1 (ITM_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : ITM 部品番号

ITM CoreSight ペリフェラル ID レジスタ 2 (ITM_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

ITM CoreSight ペリフェラル ID レジスタ 3 (ITM_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

ITM CoreSight コンポーネント ID レジスタ 0 (ITM_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

ITM CoreSight ペリフェラル ID レジスタ 1 (ITM_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 00E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント識別ビット [15:12] - コンポーネントクラス

0xE : トレースジェネレータのコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

ITM CoreSight コンポーネント ID レジスタ 2 (ITM_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

ITM CoreSight コンポーネント ID レジスタ 3 (ITM_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.7.2 ITM レジスタマップ

ITM レジスタはアドレスレンジ 0xE000 0000 ~ 0xE000 0FFC に配置されています。

表 706. ITM レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000 から 0x07C	ITM_STIMR0 から ITM_STIMR31	STIMULUS[31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x07C から 0xD7C	予約済みです。	予約済みです。																																
0xE00	ITM_TER	STIMENA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0xE04 から 0xE3C	予約済みです。	予約済みです。																																
0xE40	ITM_TPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIVMASK [3:0]				
	リセット値																													0	0	0	0	
0xE44 から 0xE7C	予約済みです。	予約済みです。																																
0xE80	ITM_TCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSPRESCALE [1:0]	Res.	Res.	Res.	Res.	STALLENA	SWOENA	TXENA	SYNCENA	TSENA	ITMENA
	リセット値									0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0	
0xE84 から 0xFB8	予約済みです。	予約済みです。																																
0xFBC	ITM_DEVARCHR	ARCHITECT[10:0]										PRESENT	REVISION [3:0]			ARCHVER [3:0]			ARCHPART[3:0]															
	リセット値	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1
0xFC0 から 0xFC8	予約済みです。	予約済みです。																																
0xFCC	ITM_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]			MAJOR[3:0]						
	リセット値																								0	1	0	0	0	0	0	1	1	
0xFD0	ITM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]			JEP106CON [3:0]					
	リセット値																								0	0	0	0	0	1	0	0		
0xFD4 から 0xFDC	予約済みです。	予約済みです。																																
0xFE0	ITM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]								
	リセット値																									0	0	1	0	0	0	0	1	
0xFE4	ITM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]			PARTNUM [11:8]					
	リセット値																									1	0	1	1	1	1	0	1	
0xFE8	ITM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]			JEDEC	JEP106ID [6:4]				
	リセット値																									0	0	0	0	1	0	1	1	
0xFEC	ITM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]			CMOD[3:0]					
	リセット値																									0	0	0	0	0	0	0	0	
0xFF0	ITM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]								
	リセット値																									0	0	0	0	1	1	0	1	

表 706. ITM レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0xFF4	ITM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]							
	リセット値																									1	1	1	0	0	0	0	0				
0xFF8	ITM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]											
	リセット値																									0	0	0	0	0	1	0	1				
0xFFC	ITM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]											
	リセット値																									1	0	1	1	0	0	0	1				

レジスタ境界アドレスについては表 701 : プロセッサ ROM テーブルを参照してください。

58.8 ブレークポイントユニット (BPU)

BPU によって、ハードウェアブレークポイントをセットできます。命令フェッチアドレスを監視する 8 つのコンパレータが含まれます。一致の発生時に、命令コンパレータはブレークポイント命令を生成するよう設定できます。

ブレークポイントユニットとその使用方法の詳細については、[4] を参照してください。

58.8.1 BPU レジスタ

BPU レジスタはアドレスレンジ 0xE0002000~0xE0002FFC に配置されています。

BPU 制御レジスタ (BPU_CTRLR)

アドレスオフセット : 0x000

リセット値 : 0x1000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NUM_CODE[6:4]			Res.	Res.	Res.	Res.	NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	r	r	r					r	r	r	r			rw	rw

ビット 31:28 **REV[3:0]** : リビジョン番号

0x1 : BPU バージョン 2

ビット 27:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12、7:4 **NUM_CODE[6:0]** : サポートされる命令アドレスコンパレータの数

0x08 : 8つの命令コンパレータがサポートされます。

ビット 11:8、3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **KEY** : 書き込み保護キー

FPB_CTRLR レジスタへの書き込みは、このビットが「1」にセットされていない場合は無視されます。

ビット 0 **ENABLE** : FPB 有効化

0 : 無効

1 : 有効

BPU コンパレータ x レジスタ (BPU_COMPxR)アドレスオフセット : $0x008 + 0x004 * x$ 、(x = 0 から 7)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BPADDR[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BPADDR[15:1]															BE
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:1 **BPADDR[31:1]** : ブレークポイントアドレスビット 0 **BE** : ブレークポイントの有効化

0 : 無効

1 : 有効

BPU デバイスタイプアーキテクチャレジスタ (BPU_DEVARCHR)

アドレスオフセット : 0xFBC

リセット値 : 0x4770 1A03

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARCHITECT[10:0]											PRESENT	REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARCHVER[3:0]				ARCHPART[11:0]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:21 **ARCHITECT[10:0]** : アーキテクチャ JEP106 コード

0x23B : JEP106 継続コード 0x4、JEP106 ID コード 0x3BArm® 限定

ビット 20 **PRESENT** : DEVARCH レジスタの有無

0x1 : あり

ビット 19:16 **REVISION[3:0]** : アーキテクチャのリビジョン

0x0 : BPU アーキテクチャ v2.0

ビット 15:12 **ARCHVER[3:0]** : アーキテクチャバージョン

0x1 : BPU アーキテクチャ v2.0

ビット 11:0 **ARCHPART[11:0]** : アーキテクチャパート

0xA03 : BPU アーキテクチャ

BPU デバイスタイプレジスタ (BPU_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUB[3:0]** : サブタイプ

0x0 : その他

ビット 3:0 **MAJOR[3:0]** : メジャータイプ

0x0 : その他

BPU CoreSight ペリフェラル ID レジスタ 4 (BPU_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : Arm® JEDEC コード

BPU CoreSight ペリフェラル ID レジスタ 0 (BPU_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : BPU 部品番号

BPU CoreSight ペリフェラル ID レジスタ 1 (BPU_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : BPU 部品番号

BPU CoreSight ペリフェラル ID レジスタ 2 (BPU_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]0x3 : Arm[®] JEDEC コード**BPU CoreSight ペリフェラル ID レジスタ 3 (BPU_PIDR3)**

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

BPU CoreSight コンポーネント ID レジスタ 0 (BPU_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

BPU CoreSight ペリフェラル ID レジスタ 1 (BPU_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0x9 : デバッグコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

BPU CoreSight コンポーネント ID レジスタ 2 (BPU_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

BPU CoreSight コンポーネント ID レジスタ 3 (BPU_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.8.2 BPU レジスタマップ

BPU レジスタはアドレスレンジ 0xE000 2000 ~ 0xE000 2FFC に配置されています。

表 707. BPU レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x000	BPU_CTRLR	REV[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_CODE [6:4]	0	0	0	Res.	Res.	Res.	Res.	NUM_CODE [3:0]	1	0	0	0	Res.	Res.	KEY	ENABLE		
	リセット値	0	0	0	1																																
0x004	予約済みです。	予約済みです。																																			
0x008 から 0x024	BPU_COMP0R から BPU_COMP7R	BPADDR[31:1]																															BE				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x028 から 0xFB8	予約済みです。	予約済みです。																																			
0xFBC	BPU_DEVARCHR	ARCHITECT[10:0]										PRESENT	REVISION [3:0]			ARCHVER [3:0]			ARCHPART[11:0]																		
	リセット値	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	1			
0xFC0 から 0xFC8	予約済みです。	予約済みです。																																			
0xFCC	BPU_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]			MAJOR[3:0]								
	リセット値																									0	0	0	0	0	0	0	0				
0xFD0	BPU_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]			JEP106CON [3:0]								
	リセット値																									0	0	0	0	0	1	0	0				
0xFD4 から 0xFDC	予約済みです。	予約済みです。																																			
0xFE0	BPU_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]											
	リセット値																									0	0	1	0	0	0	0	1				
0xFE4	BPU_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]			PARTNUM [11:8]								
	リセット値																									1	0	1	1	1	0	1					
0xFE8	BPU_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]			JEDEC	JEP106ID [6:4]							
	リセット値																									0	0	0	0	1	0	1	1				
0xFEC	BPU_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]			CMOD[3:0]								
	リセット値																									0	0	0	0	0	0	0	0				
0xFF0	BPU_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]											
	リセット値																									0	0	0	0	1	1	0	1				
0xFF4	BPU_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]			PREAMBLE [11:8]								
	リセット値																									1	0	0	1	0	0	0	0				
0xFF8	BPU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]											
	リセット値																									0	0	0	0	0	1	0	1				
0xFFC	BPU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]											
	リセット値																									1	0	1	1	0	0	0	1				

レジスタ境界アドレスについては表 701 : プロセッサ ROM テーブルを参照してください。

58.9 組み込みトレースマクロセル (ETM)

ETM は CPU と密接に連結された CoreSight™ コンポーネントです。ETM は Cortex®-M33 コアの実行をトレースすることができるトレースパケットを生成します。STM32H563/H573 および STM32H562 では、ETM は命令トレース専用を設定されています。トレース情報にデータアクセスは含まれません。

ETM はプロセッサのトレースインタフェース経由で CPU から以下の情報を受信します。

- 同一サイクルで実行される命令の数
- プログラムフローの変更
- プロセッサの現在の命令ステート
- ロードおよびストア命令によってアクセスされるメモリ位置のアドレス
- 転送のタイプ、方向、およびサイズ
- 条件コードに関する情報
- 例外情報
- 割込み待ちステートに関する情報

詳細については、Arm® CoreSight™ ETM-M33 Technical Reference Manual を参照してください [6]。

58.9.1 ETM レジスタ

ETM レジスタはアドレスレンジ 0xE004 1000 ~ 0xE004 1FFC に配置されています。

ETM プログラミング制御レジスタ (ETM_PRGCTLR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EN** : トレースユニットの有効化

0 : 無効

1 : 有効

ETM ステータスレジスタ (ETM_STATR)

アドレスオフセット : 0x00C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PMSTABLE	IDLE
														r	r

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **PMSTABLE** : 安定性ステータス

ETM-M33 レジスタが安定しており、読出し可能であることを示します。

0 : 安定していません。

1 : 安定しています。

ビット 0 **IDLE** : トレースユニットステータス

トレースユニットが非アクティブであることを示します。

0 : アイドル状態ではありません。

1 : アイドル状態です。

ETM 設定レジスタ (ETM_CONFIGR)

アドレスオフセット : 0x010

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RS	Res.	COND[5:0]						CCI	BB	Res.	Res.	Res.
			rw		rw	rw	rw	rw	rw	rw	rw	rw			

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **RS** : リターンスタックの有効化

0 : 無効

1 : 有効

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:5 **COND[5:0]** : 条件付き命令のトレース

0x0 : 条件付き命令のトレースは無効です。

0x1 : 条件付きロード命令をトレースします。

0x2 : 条件付きストア命令をトレースします。

0x3 : 条件付きロードおよびストア命令をトレースします。

0x7 : すべての条件付き命令をトレースします。

ビット 4 **CCI** : 命令トレース時のサイクルカウント

0 : 無効

1 : 有効

ビット 3 **BB** : 分岐ブロードキャストモード

0 : 無効

1 : 有効

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

ETM イベント制御 0 レジスタ (ETM_EVENTCTL0R)

アドレスオフセット : 0x020

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE1	Res.	Res.	Res.	SEL1 [3:0]				TYPE0	Res.	Res.	Res.	SEL0 [3:0]			
rw				rw	rw	rw	rw	rw				rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TYPE1** : event1 のリソースタイプ

0 : 単一選択リソース

1 : ブール組み合わせリソースペア

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **SEL1[3:0]** : TYPE1 に基づくリソース番号

TYPE1 の値に基づいて、リソース番号を選択します。

TYPE1 = 0 のときは、SEL1[3:0] で定義された 0 ~ 15 の中から単一リソースが選択されます。

TYPE1 = 1 のときは、SEL1[2:0] で定義されたブール組み合わせリソースペアが選択されます。

ビット 7 **TYPE0** : event0 のリソースタイプ

0 : 単一選択リソース

1 : ブール組み合わせリソースペア

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **SEL0[3:0]** : TYPE0 に基づくリソース番号

TYPE0 の値に基づいて、リソース番号を選択します。

TYPE0 = 0 のときは、SEL0[3:0] で定義された 0 ~ 15 の中から単一リソースが選択されます。

TYPE0 = 1 のときは、SEL0[2:0] で定義されたブール組み合わせリソースペアが選択されます。

ETM イベント制御 1 レジスタ (ETM_EVENTCTL1R)

アドレスオフセット : 0x024

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	LPOVERRIDE	ATB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSTEN[1:0]	
			rw	rw										rw	rw

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **LPOVERRIDE** : 低電力ステートでの動作のオーバーライド

0 : 通常の低電力ステートの動作を行います。

1 : リソースやイベントトレースの生成は、低電力ステートへの移行による影響を受けません。

ビット 11 **ATB** : ATB トリガ有効

0 : 無効

1 : 有効

ビット 10:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **INSTEN[1:0]** : 命令イベント生成

命令ストリームでのイベント要素の生成を有効にします。

0bX0 : Event0 はイベント要素の原因とはなりません。

0bX1 : Event0 は、その発生時にイベント要素の原因となります。

0b0X : Event1 はイベント要素の原因とはなりません。

0b1X : Event1 は、その発生時にイベント要素の原因となります。

ETM ストール制御レジスタ (ETM_STALLCTLR)

アドレスオフセット : 0x02C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	INSTPRIORITY	Res.	ISTALL	Res.	Res.	Res.	Res.	LEVEL[3:0]			
					rw		rw					rw	rw	rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **INSTPRIORITY** : 命令トレースの優先順位

命令トレースのバッファスペースが LEVEL[3:0] を下回る場合に、命令トレースを優先します。

0 : ETM は命令トレースを優先してはなりません。

1 : ETM は命令トレースを優先することができます。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ISTALL** : プロセッサのストール

命令トレースのバッファスペースに基づいてプロセッサをストールします。

0 : ETM はプロセッサをストールしてはなりません。

1 : ETM はプロセッサをストールすることができます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **LEVEL[3:0]** : ストールがアクティブになる閾値

このフィールドには 4 つのレベルがあります。このレベルは、ストールによって生じた侵入のレベルを最適化して FIFO オーバーフローのリスクとのバランスを取るために変更できます。

0x0 : 侵入はありませんが、FIFO オーバーフローのリスクが大きくなります。

.....

0xF : 侵入レベルは最大ですが、FIFO オーバーフローのリスクは小さくなります。

ETM 同期周期レジスタ (ETM_SYNCPR)

アドレスオフセット : 0x034

リセット値 : 0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD[4:0]				
											r	r	r	r	r

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **PERIOD[4:0]** : 同期周期

トレース同期リクエスト間のトレースのバイト数を、命令ストリームによって生成されたバイト数の合計として定義します。

0xA : 1024 バイト

ETM サイクルカウント制御レジスタ (ETM_CCCTL)

アドレスオフセット : 0x038

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	THRESHOLD[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **THRESHOLD[11:0]** : 命令トレースサイクルカウントの閾値

命令トレースサイクルカウントの閾値をセットします。この閾値は、サイクルカウントトレースパケットの最小間隔を表します。

ETM トレース識別レジスタ (ETM_TRACEIDR)

アドレスオフセット : 0x040

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[6:0]						
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **TRACEID[6:0]** : トレースバスに出力するトレース識別

このフィールドは、一意の値でプログラムすることで、システム内の他のトレースソースと区別する必要があります。

値 0x00 および 0x70 ~ 0x7F は予約済みです。

ETM ViewInst メイン制御レジスタ (ETM_VICTLR)

アドレスオフセット : 0x080

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXLEVEL_S[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRCERR	TRCRESET	SSSTATUS	Res.	EVENT[7:0]							
				rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **EXLEVEL_S[3:0]** : セキュアステートの例外レベル

セキュアステートで、対応する例外レベルについて命令トレースを有効にするかどうかを制御します。

0bXXX0 : 例外レベル 0 のセキュアステートにおいて、命令トレースは生成されません。

0bXXX1 : 例外レベル 0 のセキュアステートにおいて、命令トレースが生成されます。

0b0XXX : 例外レベル 3 のセキュアステートにおいて、命令トレースは生成されません。

0b1XXX : 例外レベル 3 のセキュアステートにおいて、命令トレースが生成されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **TRCERR** : システムエラー例外のトレース

0 : システムエラー例外の直前の命令または例外がトレースされる場合のみ、システムエラー例外はトレースされます。

1 : システムエラー例外は常にトレースされます。

ビット 10 **TRCRESET** : リセット例外のトレース

0 : リセット例外の直前の命令または例外がトレースされる場合のみ、リセット例外はトレースされます。

1 : リセット例外は常にトレースされます。

ビット 9 **SSSTATUS** : 開始／停止ロジックのステータス

0 : 停止されます。

1 : 開始されます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **EVENT[7:0]** : イベントセレクタ

ETM カウンタ再ロード値レジスタ 0 (ETM_CNTRLDVR0)

アドレスオフセット : 0x140

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **VALUE[15:0]** : カウンタ再ロード値

この値は、再ロードイベント発生の際にカウンタにロードされます。

ETM 識別レジスタ 8 (ETM_IDR8)

アドレスオフセット : 0x180

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAXSPEC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAXSPEC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **MAXSPEC[31:0]** : 最大スペキュレーション深さ

命令トレースストリームの最大スペキュレーション深さを示します。これは、トレースストリームで同時にコミットされることがない P0 要素の最大数です。

0x0 : トレーススペキュレーションの最大深さは 0 です。

ETM 識別レジスタ 9 (ETM_IDR9)

アドレスオフセット : 0x184

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP0KEY [31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP0KEY [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **NUMP0KEY[31:0]** : 使用する P0 右側キーの数

0x0 : 命令トレースで使用する P0 右側キーはありません。

ETM 識別レジスタ 10 (ETM_IDR10)

アドレスオフセット : 0x188

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP1KEY [31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP1KEY [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **NUMP1KEY[31:0]** : 使用する P1 右側キーの総数 (通常キーと特殊キーを含む)

0x0 : 命令トレースで使用する P1 右側キーはありません。

ETM 識別レジスタ 11 (ETM_IDR11)

アドレスオフセット : 0x18C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP1SPC [31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP1SPC [15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **NUMP1SPC[31:0]** : 使用する特殊な P1 右側キーの数

0x0 : 設定で使用されている特殊な P1 右側キーはありません。

ETM 識別レジスタ 12 (ETM_IDR12)

アドレスオフセット : 0x190

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCONDKEY[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMCONDKEY[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **NUMCONDKEY[31:0]** : 使用する条件付き命令右側キーの数 (通常キーと特殊キーを含む)

0x1 : 条件付き命令右側キーが 1 つ実装されます。

ETM 識別レジスタ 13 (ETM_IDR13)

アドレスオフセット : 0x194

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCONDSPC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMCONDSPC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **NUMCONDSPC[31:0]** : 使用する特殊な条件付き命令右側キーの数

0x0 : 特殊な条件付き命令右側キーは実装されません。

ETM 実装固有レジスタ 0 (ETM_IMSPECR0)

アドレスオフセット : 0x1C0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUPPORT[3:0]			
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **SUPPORT[3:0]** : 実装固有の拡張機能のサポート

0x0 : 実装固有の拡張機能はサポートされません。

ETM 識別レジスタ 0 (ETM_IDR0)

アドレスオフセット : 0x1E0

リセット値 : 0x2800 06E1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	COMMOPT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCEXDATA	QSUPP[1]
		r												r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QSUPP[0]	Res.	CONDTYPE[1:0]		NUMEVENT[1:0]		RETSTACK	Res.	TRCCCI	TRCCOND	TRCBB	TRCDATA[1:0]		INSTP0 [1:0]		Res.
r		r	r	r	r	r		r	r	r	r	r	r	r	

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **COMMOPT** : コミットフィールドの意味

一部のバケットにおけるコミットフィールドの意味を示します。

1 : コミットモード 1

ビット 28:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **TRCEXDATA** : 例外のデータ転送のトレース

例外および返された例外のデータ転送のトレースに対するサポートを示します。

0 : 実装されません。

ビット 16:15 **QSUPP[1:0]** : Q 要素のサポート

0 : サポートされません。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:12 **CONDTYPE[1:0]** : 条件付き結果のトレース

条件付き結果のトレース方法を示します。

0 : トレースユニットは、条件付き命令が条件コードチェックを通過または失敗したときにのみ示されます。

ビット 11:10 **NUMEVENT[1:0]** : サポートされるイベント数

0x1 : 2 つのイベント

ビット 9 **RETSTACK** : リターンスタックのサポート

1 : 2 つのエントリリターンスタック

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TRCCCI** : サイクルカウンタのサポート

1 : サイクルカウンタが実装されます。

ビット 6 **TRCCOND** : 条件付き命令のサポート

1 : 条件付き命令のトレースが実装されます。

ビット 5 **TRCBB** : 分岐ブロードキャストのサポート

1 : 分岐ブロードキャストのトレースが実装されます。

ビット 4:3 **TRCDATA[1:0]** : データトレースのサポート

0x0 : データトレースはサポートされません。

ビット 2:1 **INSTP0[1:0]** : P0 要素としてのロードおよびストア命令のトレースのサポート

0x0 : サポートされません。

ビット 0 予約済みであり、リセット値に保持する必要があります。

ETM 識別レジスタ 1 (ETM_IDR1)

アドレスオフセット : 0x1E4

リセット値 : 0x4100 F421

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DESIGNER[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRCARCHMAJ[3:0]				TRCARCHMIN[3:0]				REVISION[3:0]			
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DESIGNER[7:0]** : トレースユニット設計者

0x41 : Arm®

ビット 23:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **TRCARCHMAJ[3:0]** : メジャーなトレースユニットアーキテクチャのバージョン番号

0x4 : ETMv4

ビット 7:4 **TRCARCHMIN[3:0]** : マイナーなトレースユニットアーキテクチャのバージョン番号

0x2 : マイナーリビジョン 2

ビット 3:0 **REVISION[3:0]** : 実装のリビジョン番号

0x1 : 実装のリビジョン 1

ETM 識別レジスタ 2 (ETM_IDR2)

アドレスオフセット : 0x1E8

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CCSIZE[3:0]				DVSIZE[4:0]				DASIZE[4:1]				
			r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DASIZE[0]	VMDSIZE[4:0]					CIDSIZE[4:0]					IASIZE[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:25 **CCSIZE[3:0]** : サイクルカウンタのサイズ

0x0 : 12 ビット

ビット 24:20 **DVSIZE[4:0]** : データ値のサイズ

0x0 : データ値のサイズはサポートされません。

ビット 19:15 **DASIZE[4:0]** : データアドレスのサイズ
0x0 : データアドレスのサイズはサポートされません。

ビット 14:10 **VMIDSIZE[4:0]** : 仮想マシン ID サイズ
0x0 : 仮想マシン ID のトレースは実装されません。

ビット 9:5 **CIDSIZE[4:0]** : コンテキスト ID サイズ
0x0 : コンテキスト ID のトレースは実装されません。

ビット 4:0 **IASIZE[4:0]** : 命令アドレスのサイズ
0x4 : 最大 32 ビットのアドレスサイズ

ETM 識別レジスタ 3 (ETM_IDR3)

アドレスオフセット : 0x1EC

リセット値 : 0x0F09 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOOVERFLOW	NUMPROC[2:0]				SYSSTALL	STALLCTL	SYNCPR	TRCERR	Res.	Res.	Res.	Res.	EXLEVEL_S[3:0]		
r	r	r	r	r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CCITMIN[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 **NOOVERFLOW** : ETM_STALLCTLR.NOOVERFLOW の実装
0 : 実装されません。

ビット 30:28 **NUMPROC[2:0]** : トレースに利用できるプロセッサの数
0x0 : 1 つのプロセッサ

ビット 27 **SYSSTALL** : プロセッサのストール制御のシステムサポート
1 : システムでストール制御をサポートします。

ビット 26 **STALLCTL** : ストール制御のサポート
1 : ETM_STALLCTLR が実装されます。

ビット 25 **SYNCPR** : トレース同期周期のサポート
1 : 命令トレースのみの設定では ETM_SYNCPR は読み出し専用です。トレース同期周期は固定です。

ビット 24 **TRCERR** : ETM_VICTLR.TRCERR の実装
0x1 : 実装されます。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **EXLEVEL_S[3:0]** : 特権レベルの実装
0x9 : 特権レベル、スレッドとハンドラーが実装されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **CCITMIN[11:0]** : TRCCCCTLR.THRESHOLD にプログラム可能な最小値
最小サイクルカウンターの閾値を定義します。
0x4 : 最小値は 4 命令トレースサイクルです。

ETM 識別レジスタ 4 (ETM_IDR4)

アドレスオフセット : 0x1F0

リセット値 : 0x0011 4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMVMIDC[3:0]				NUMCIDC[3:0]				NUMSSCC[3:0]				NUMRSPAIR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMPC[3:0]				Res.	Res.	Res.	SUPPDAC	NUMDVC[3:0]				NUMACPAIRS[3:0]			
r	r	r	r				r	r	r	r	r	r	r	r	r

ビット 31:28 **NUMVMIDC[3:0]** : 仮想マシン ID (VMID) コンパレータの数

0x0 : VMID コンパレータは実装されません。

ビット 27:24 **NUMCIDC[3:0]** : コンテキスト ID コンパレータの数

0x0 : コンテキスト ID コンパレータはサポートされません。

ビット 23:20 **NUMSSCC[3:0]** : シングルショットコンパレータ制御の数

0x1 : シングルショットコンパレータ制御が 1 つ実装されます。

ビット 19:16 **NUMRSPAIR[3:0]** : リソース選択ペアの数

0x1 : リソース選択ペアが 2 つ実装されます。

ビット 15:12 **NUMPC[3:0]** : DWT 用のプロセッサコンパレータ入力の数

0x4 : プロセッサコンパレータ入力が 4 つ実装されます。

ビット 11:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SUPPDAC** : データアドレス比較

0 : データアドレスの比較はサポートされません。

ビット 7:4 **NUMDVC[3:0]** : データ値コンパレータの数

0x0 : データ値コンパレータは実装されません。

ビット 3:0 **NUMACPAIRS[3:0]** : アドレスコンパレータペアの数

0x0 : アドレスコンパレータペアは実装されません。

ETM 識別レジスタ 5 (ETM_IDR5)

アドレスオフセット : 0x1F4

リセット値 : 0x90C7 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REDFUNCNTR	NUMCNTR[2:0]			NUMSEQSTATE[2:0]			Res.	LPOVERRIDE	ATBTRIG	TRACEIDSIZE[5:0]					
r	r	r	r	r	r	r		r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NUMEXTINSEL[2:0]			NUMEXTIN[8:0]								
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31 **REDFUNCNTR** : 低減機能カウンタ

1 : 低減機能カウンタとしてカウンタ 0 が実装されます。

ビット 30:28 **NUMCNTR[2:0]** : カウンタの数

0x1 : カウンタが 1 つ実装されます。

ビット 27:25 **NUMSEQSTATE[2:0]** : シーケンサのステート数

0x0 : シーケンサのステートは実装されません。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **LPOVERRIDE** : 低電力ステートのオーバーライドのサポート

1 : 低電力ステートのオーバーライドのサポートが実装されます。

ビット 22 **ATBTRIG** : ATB トリガのサポート

1 : ATB トリガのサポートが実装されます。

ビット 21:16 **TRACEIDSIZE[5:0]** : トレース識別のビット数

0x7 : 7 ビットトレース識別が実装されます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:9 **NUMEXTINSEL[2:0]** : 外部入力セレクトの数

0x0 : 外部入力セレクトは実装されません。

ビット 8:0 **NUMEXTIN[8:0]** : 外部入力の数

0x004 : 4 つの外部入力の実装されます。

ETM リソースレジスタ 2 (ETM_RSCTLR2)

アドレスオフセット : 0x208

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAIRINV	INV	Res.	GROUP[2:0]		
										rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **PAIRINV** : 組み合わせリソースペアの結果の反転

0 : 反転されません。

1 : 反転されます。

ビット 20 **INV** : 選択されたリソースの反転

0 : 反転されません。

1 : 反転されます。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **GROUP[2:0]** : リソース選択のグループ

0x0 : 外部入力セレクタ (0 ~ 3 を選択)

0x1 : プロセッサ DWT コンパレータ要素からの入力 (0 ~ 3 を選択)

0x2 : ゼロのカウンタ (0 を選択)

0x3 : シングルショットコンパレータ (0 を選択)

その他 : 予約済み

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **SELECT[7:0]** : さらなるリソース選択

GROUP[2:0] で選択したグループから 1 つ以上のリソースを選択します。

ETM リソースレジスタ 3 (ETM_RSCTLR3)

アドレスオフセット : 0x20C

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INV	Res.	GROUP[2:0]		
											rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **INV** : 選択されたリソースの反転

0 : 反転されません。

1 : 反転されます。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **GROUP[2:0]** : リソース選択のグループ

0x0 : 外部入力セレクタ (0 ~ 3 を選択)

0x1 : プロセッサ DWT コンパレータ要素からの入力 (0 ~ 3 を選択)

0x2 : ゼロのカウンタ (0 を選択)

0x3 : シングルショットコンパレータ (0 を選択)

その他 : 予約済み

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **SELECT[7:0]** : さらなるリソース選択

GROUP[2:0] で選択したグループから 1 つ以上のリソースを選択します。

ETM シングルショットコンパレータ制御レジスタ 0 (ETM_SSCCR0)

アドレスオフセット : 0x280

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **RST** : シングルショットコンパレータのリセット

シングルショットコンパレータリソースの発生時にそのリセットを有効にし、別のコンパレータの一致を検出できるようにします。

1 : リセットを有効化

ビット 23:0 予約済みであり、リセット値に保持する必要があります。

ETM シングルショットコンパレータのステータスレジスタ 0 (ETM_SSCSR0)

アドレスオフセット : 0x2A0

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC	DV	DA	INST
												r	r	r	r

ビット 31 **STATUS** : シングルショットコンパレータのステータス

選択されたコンパレータのいずれかで一致が発生したかどうかを示します。

0 : 一致は発生しませんでした。

1 : 最低でも 1 回の一致が発生しました。

ビット 30:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **PC** : プロセッサコンパレータ入力の感度

1 : プロセッサコンパレータ入力に反応するシングルショットコンパレータ

ビット 2 **DV** : データ値コンパレータのサポート

0 : シングルショットデータ値の比較はサポートされません。

ビット 1 **DA** : データアドレスコンパレータのサポート

0 : シングルショットデータアドレスの比較はサポートされません。

ビット 0 **INST** : 命令アドレスコンパレータのサポート

0 : シングルショット命令アドレスの比較はサポートされません。

ETM シングルショットプロセッサコンパレータの入力制御レジスタ 0 (ETM_SSPCICR0)

アドレスオフセット : 0x2C0

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PC[3:0]** : シングルショット制御のプロセッサコンパレータ入力の選択

0xxx0 : プロセッサコンパレータ入力 0 は選択されません。

0xxx1 : プロセッサコンパレータ入力 0 が選択されます。

0xx0x : プロセッサコンパレータ入力 1 は選択されません。

0xx1x : プロセッサコンパレータ入力 1 が選択されます。

0x0xx : プロセッサコンパレータ入力 2 は選択されません。

0x1xx : プロセッサコンパレータ入力 2 が選択されます。

00xxx : プロセッサコンパレータ入力 3 は選択されません。

01xxx : プロセッサコンパレータ入力 3 が選択されます。

ETM パワーダウン制御レジスタ (ETM_PDCR)

アドレスオフセット : 0x310

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU	Res.	Res.	Res.
												rw			

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **PU** : パワーアップリクエスト

0 : パワーアップはリクエストされません。

1 : パワーアップがリクエストされます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

ETM パワーダウンステータスレジスタ (ETM_PDSR)

アドレスオフセット : 0x314

リセット値 : 0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STICKYPD	POWER
														r	r

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **STICKYPD** : スティックキーなパワーダウンステート

0 : トレースレジスタの電源は、ETM_PDSR が最後に読み出されたあとで削除されていません。

1 : トレースレジスタの電源は、ETM_PDSR が最後に読み出されたあとで削除されます。

ビット 0 **POWER** : ETM パワーアップステータス

1 : ETM のパワーアップ

ETM 要求タグ設定レジスタ (ETM_CLAIMSETR)

アドレスオフセット : 0xFA0

リセット値 : 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CLAIMSET[3:0]** : 要求タグビットのセット

書込み :

0000 : 影響なし。

xxx1 : ビット 0 をセットします。

xx1x : ビット 1 をセットします。

x1xx : ビット 2 をセットします。

1xxx : ビット 3 をセットします。

読出し :

0xF : 要求タグには 4 つのビットがあることを示します。

リセット値: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw	rw	rw	rw

読出し：要求タグの現在値を返します。

リセット値：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r	r	r	r	r	r	r	r

0x0 : 実装されません。

ETM デバイスタイプアーキテクチャレジスタ (ETM_DEVARCHR)

アドレスオフセット : 0xFBC

リセット値 : 0x4772 4A13

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARCHITECT[10:0]											PRESENT	REVISION[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARCHVER[3:0]				ARCHPART[11:0]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:21 **ARCHITECT[10:0]** : アーキテクチャ JEP106 コード

0x23B : JEP106 継続コード 0x4、JEP106 ID コード 0x3BArm® 限定

ビット 20 **PRESENT** : DEVARCH レジスタの有無

0x1 : あり

ビット 19:16 **REVISION[3:0]** : アーキテクチャのリビジョン

0x2 : ETM アーキテクチャ v4.2

ビット 15:12 **ARCHVER[3:0]** : アーキテクチャバージョン

0x4 : ETM アーキテクチャ v4.2

ビット 11:0 **ARCHPART[11:0]** : アーキテクチャパート

0xA13 : ETM アーキテクチャ

ETM CoreSight デバイスタイプレジスタ (ETM_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0013

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUBTYPE[3:0]** : デバイスのサブタイプ識別子

0x1 : プロセッサのトレース

ビット 3:0 **MAJORTYPE[3:0]** : デバイスのメインタイプ識別子

0x3 : トレースソース

ETM CoreSight ペリフェラル ID レジスタ 4 (ETM_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード0x4 : Arm[®] JEDEC コード**ETM CoreSight ペリフェラル ID レジスタ 0 (ETM_PIDR0)**

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : ETM 部品番号

ETM CoreSight ペリフェラル ID レジスタ 1 (ETM_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : ETM 部品番号

ETM CoreSight ペリフェラル ID レジスタ 2 (ETM_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x1 : r0p1

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

ETM CoreSight ペリフェラル ID レジスタ 3 (ETM_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

ETM CoreSight コンポーネント ID レジスタ 0 (ETM_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

ETM CoreSight ペリフェラル ID レジスタ 1 (ETM_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0x9 : トレースジェネレータのコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

ETM CoreSight コンポーネント ID レジスタ 2 (ETM_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

ETM CoreSight コンポーネント ID レジスタ 3 (ETM_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 PREAMBLE[27:20] : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.9.2 ETM レジスタマップ

ETM レジスタへはデバッガからアクセスでき、アドレスレンジは 0xE0041000 ~ 0xE0041FFC です。

表 708. ETM レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x004	ETM_PRGCTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN	
	リセット値																																0	
0x008	予約済みです。	予約済みです。																																
0x00C	ETM_STATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PMSTABLE	IDLE	
	リセット値																															X	X	
0x010	ETM_CONFIGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RS	COND[5:0]					CCI	BB	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																				X		X	X	X	X	X	X	X	X				
0x014 から 0x01C	予約済みです。	予約済みです。																																
0x020	ETM_EVENTCTL0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TYPE1	Res.	Res.	Res.	SEL1 [3:0]	TYPE0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEL0 [3:0]			
	リセット値																				X	X	X	X	X					X	X	X	X	
0x024	ETM_EVENTCTL1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPOVERRIDE	ATB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSTEN[1:0]		
	リセット値																				X	X										X	X	
0x028	予約済みです。	予約済みです。																																
0x02C	ETM_STALLCTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSTPRIORITY	ISTALL		Res.	Res.	Res.	Res.	LEVEL[3:0]				
	リセット値																					X		X						X	X	X	X	
0x030	予約済みです。	予約済みです。																																

表 708. ETM レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x034	ETM_SYNCPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD[4:0]						
	リセット値																												0	1	0	1	0		
0x038	ETM_CCCTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																					X	X	X	X	X	X	X	X	X	X	X	X		
0x03C	予約済みです。	予約済みです。																																	
0x040	ETM_TRACEIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[6:0]						
	リセット値																											X	X	X	X	X	X		
0x044 から 0x07C	予約済みです。	予約済みです。																																	
0x080	ETM_VICTLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXLEVEL_S [3:0]			Res.	Res.	Res.	Res.	TRCERR	TRCRESET	SSSTATUS	Res.	EVENT[7:0]									
	リセット値													X	X	X	X					X	X	X		X	X	X	X	X	X	X	X		
0x084 から 0x13C	予約済みです。	予約済みです。																																	
0x140	ETM_CNTRLDVR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VALUE[15:0]																	
	リセット値																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
0x144 から 0x17C	予約済みです。	予約済みです。																																	
0x180	ETM_IDR8	MAXSPEC[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x184	ETM_IDR9	NUMP0KEY [31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x188	ETM_IDR10	NUMP1KEY [31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18C	ETM_IDR11	NUMP1SPC [31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x190	ETM_IDR12	NUMCONDKEY[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
0x194	ETM_IDR13	NUMCONDSPC[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x198 から 0x1BC	予約済みです。	予約済みです。																																	
0x1C0	ETM_IMSPECR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUPPORT [3:0]				
	リセット値																													0	0	0	0		
0x1C4 から 0x1DC	予約済みです。	予約済みです。																																	
0x1E0	ETM_IDR0	Res.	Res.	COMMOPT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCADATA	QSUPP[1:0]	Res.	Res.	CONDTYPE [1:0]	Res.	NUMEVENT [1:0]	RETSTACK	Res.	TRCCCI	TRCCOND	TRCBB	TRCADATA[1:0]	INSTP0 [1:0]			Res.				
	リセット値			1											0	0	0		0	0	0	1	1		1	1	1	0	0	0	0	0			
0x1E4	ETM_IDR1	DESIGNER[7:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCARCHMAJ [3:0]	TRCARCHMIN [3:0]				REVISION [3:0]									
	リセット値	0	1	0	0	0	0	0	1												0	1	0	0	0	0	1	0	0	0	0	1			
0x1E8	ETM_IDR2	Res.	Res.	Res.	CCSIZE[3:0]			DVSIZE[4:0]				DASIZE[4:0]				VMIDSIZE[4:0]				CIDSIZE[4:0]				IASIZE[4:0]											
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		

表 708. ETM レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x1EC	ETM_IDR3	NOOVERFLOW	NUMPROC[2:0]		SYSSTALL	STALLCTL	SYNCPR	TRCERR	Res.	Res.	Res.	Res.	EXLEVEL_S [3:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0	0	0		0	1	1	1	1						1	0																		0
0x1F0	ETM_IDR4	NUMVMIDC [3:0]			NUMCIDC [3:0]			NUMSSCC [3:0]			NUMRSPAIR [3:0]			NUMPC [3:0]			Res.	Res.	Res.	SUPPDAC	NUMDVC [3:0]			NUMACPAIRS [3:0]											
	リセット値	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0					0	0	0	0	0	0	0	0	0	0
0x1F4	ETM_IDR5	REDFUNCNTR	NUMCNTR[2:0]		NUMSEQSTATE [2:0]		Res.	LPOVERRIDE	ATBTRIG	TRACEIDSIZE [5:0]					Res.	Res.	Res.	Res.	NUMEXTINSEL [2:0]	NUMEXTIN[8:0]															
	リセット値	1	0	0	1	0	0	0		1	1	0	0	0	1	1	1						0	0	0	0	0	0	0	0	0	0	1	0	0
0x1F8 から 0x204	予約済みです。	予約済みです。																																	
0x208	ETM_RSCTLR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAIRINV	INV	Res.	GROUP [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]									
	リセット値										X	X		X	X	X										X	X	X	X	X	X	X	X	X	
0x20C	ETM_RSCTLR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INV	Res.	GROUP [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]									
	リセット値											X		X	X	X										X	X	X	X	X	X	X	X	X	
0x210 から 0x27C	予約済みです。	予約済みです。																																	
0x280	ETM_SSCCR0	Res.	Res.	Res.	Res.	Res.	Res.	RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値							X																											
0x284 から 0x29C	予約済みです。	予約済みです。																																	
0x2A0	ETM_SSCSR0	STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC	DV	DA	INST		
	リセット値	X																												X	X	X	X		
0x2A4 から 0x2BC	予約済みです。	予約済みです。																																	
0x2C0	ETM_SSPICR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC[3:0]				
	リセット値																													X	X	X	X		
0x2C4 から 0x30C	予約済みです。	予約済みです。																																	
0x310	ETM_PDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU	Res.	Res.		
	リセット値																													0					
0x314	ETM_PDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STICKYPD	POWER		
	リセット値																															1	1		
0x318 から 0xF9C	予約済みです。	予約済みです。																																	
0xFA0	ETM_CLAIMSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]					
	リセット値																													1	1	1	1		

表 708. ETM レジスタのマップとリセット値 (続き)

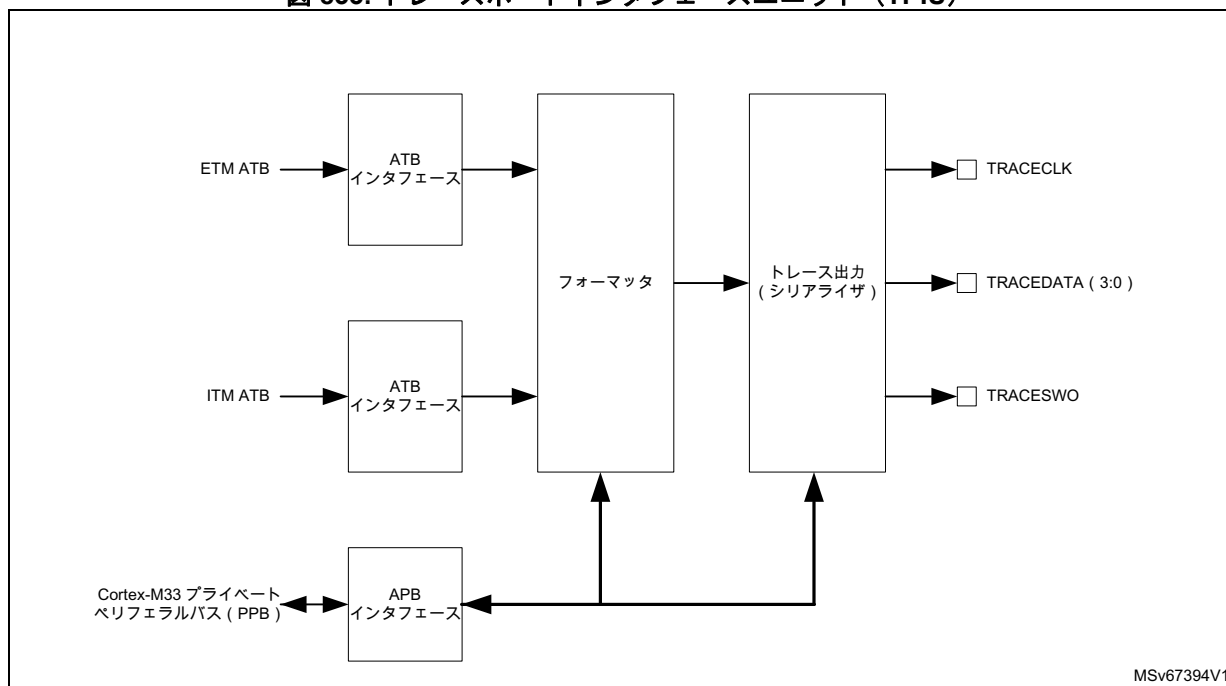
オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0xFA4	ETM_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]				
	リセット値																													0	0	0	0	
0xFA8 から 0xFB4	予約済みです。	予約済みです。																																
0xFB8	ETM_AUTHSTATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFBC	ETM_DEVARCHR	ARCHITECT[10:0]										PRESENT	REVISION [3:0]			ARCHVER [3:0]			ARCHPART[11:0]															
	リセット値	0	1	0	0	0	1	1	1	0	1		1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1
0xFC0to : 0xFC8	予約済みです。	予約済みです。																																
0xFCC	ETM_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																	
0xFD0	ETM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFD4to : 0xFDC	予約済みです。	予約済みです。																																
0xFE0	ETM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFE4	ETM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFE8	ETM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFEC	ETM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFF0	ETM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFF4	ETM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFF8	ETM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0xFFC	ETM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	

レジスタ境界アドレスについては表 701 : プロセッサ ROM テーブル を参照してください。

58.10 トレースポートインタフェースユニット (TPIU)

TPIU はトレースストリームをフォーマットし、それを外部トレースポート信号に出力します。下図に示すように、TPIU には ETM と ITM の受信トレースデータ用に 2 つの ATB スレーブポートが搭載されています。トレースポートは同期パラレルポートで、クロック出力 TRACECLK と 4 つのデータ出力 TRACEDATA(3:0) から成ります。トレースポートの幅はレンジ 1~4 でプログラム可能です。より幅の狭いポート設定を使用すると、必要なテスト点／コネクタピンの数が減り、他の目的のために出力ポートを解放することができます。しかし、それによりトレースポートの帯域幅が制限され、リアルタイムで出力できるトレース情報量が減少します。

図 833. トレースポートインタフェースユニット (TPIU)



トレースデータはシリアルワイヤ出力 TRACESWO に出力することもできます。

Cortex®-M33 のトレースポートインタフェースの詳細については、Arm® Cortex®-M33 Technical Reference Manual を参照してください [\[5\]](#)。

58.10.1 TPIU レジスタ

TPIU でサポートされるポートサイズのレジスタ (TPIU_SSPSR)

アドレスオフセット : 0x000

リセット値 : 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **PORTSIZE[31:0]** : 1 ~ 32 ピンのトレースポートのサイズ

ビット n-1 がセットされている場合、ポートサイズ n がサポートされることを示します。

0x0000 000F : ポートサイズ 1~4 がサポートされます。

TPIU の現在のポートサイズのレジスタ (TPIU_CSPSR)

アドレスオフセット : 0x004

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **PORTSIZE[31:0]** : 現在のトレースポートサイズ

ビット n-1 がセットされている場合、現在のポートサイズが n ピンであることを示します。n の値はサポートされるポートサイズレンジ内（1~4）でなければなりません。1 ビットのみをセットしてください。そうしないと、予期せぬ動作を引き起こす場合があります。

このレジスタの変更は必ずフォーマッタの停止時に行ってください。

TPIU 非同期クロックプリスケアラレジスタ (TPIU_ACPR)

アドレスオフセット : 0x010

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PRESCALER[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:0 **PRESCALER[12:0]** : 非同期出力 TRACESWO のボーレート

ボーレートは (PRESCALER +1) で割った TRACELKIN 周波数で与えられます。

TPIU 選択ピンプロトコルレジスタ (TPIU_SPPR)

アドレスオフセット : 0x0F0

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXMODE[1:0]	
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TXMODE[1:0]** : トレース出力に使用されるプロトコル

0x0 : パラレルトレースポートモード

0x1 : マンチェスタ符号化を使用した非同期 SWO

0x2 : NRZ 符号化を使用した非同期 SWO

0x3 : 予約済み

TPIU フォーマッタおよび一掃のステータスレジスタ (TPIU_FFSR)

アドレスオフセット : 0x300

リセット値 : 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTNONSTOP	TCPRESENT	FTSTOPPED	FLINPROG
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **FTNONSTOP** : フォーマッタ停止

フォーマッタを停止できるかどうかを示します。

1 : フォーマッタは停止できません。

ビット 2 **TCPRESENT** : TRACECTL 出力ピンの使用可能性

オプションの TRACECTL 出力ピンを使用可能かどうかを示します。

0 : TRACECTL ピンはこのデバイスには存在しません。

ビット 1 **FTSTOPPED** : フォーマッタ停止

フォーマッタは停止リクエスト信号を受信しており、すべてのトレースデータおよびポストアンブルが送信されます。ATB インタフェースでの追加のトレースデータは無視されます。

0 : フォーマッタは停止していません。

ビット 0 **FLINPROG** : 一掃の実行中

ATB スレーブポートで一掃を実行中かどうかを示します。このビットは、AFVALIDS 出力の状態を反映します。一掃は TPIU_FFCR レジスタの一掃制御ビットによって開始することができます。

0 : 実行中の一掃はありません。

1 : 一掃の実行中です。

TPIU フォーマッタおよび一掃の制御レジスタ (TPIU_FFCR)

アドレスオフセット : 0x304

リセット値 : 0x0000 0102

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGIN	Res.	FONMAN	Res.	Res.	Res.	Res.	ENFCONT	Res.
							r		rW					rW	

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TRIGIN** : トリガ入力時のトリガ

1 : TRIGIN 入力のアサートされたときにトレースストリームにあるトリガを示します。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FONMAN** : マニュアル一掃

0 : 一掃が完了しました。

1 : 一掃を生成します。

ビット 5:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **ENFCONT** : 連続フォーマットの有効化

SWO モードでこのビットを 0 にセットすることでフォーマッタはバイパスされ、ITM/DWT トレースのみが出力されて、ETM トレースは破棄されます。

0 : 連続フォーマットは無効です。

1 : 連続フォーマットは有効です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

TPIU 周期的同期カウンタのレジスタ (TPIU_PSCR)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PSCOUNT[12:0]												
			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:0 **PSCOUNT[12:0]** : フォーマットフレームカウンタ

キャプチャデバイスの大容量のストレージを無駄にすることなく、サイズの異なる複数の TPA を効果的に使用できるようにします。このカウンタには、最後の 128 ビット同期パケット以降のフォーマットフレームの数が格納されます。12 ビットのカウンタで、最大カウント値は 4096 です。この値は 65536 バイトごとの同期、すなわち 4096 パケット x パケット当たり 16 バイトに等しい数です。デフォルトは 1024 バイトごと、すなわち 64 フォーマットフレームごとの同期パケットに対して設定されます。フォーマッタが連続モード用に設定されている場合は、フルワードとハーフワードの同期フレームが通常動作中に挿入されます。このような状況では、カウント値はフル同期パケット間の完了フレームの最大数となります。

TPIU 要求タグ設定レジスタ (TPIU_CLAIMSETR)

アドレスオフセット : 0xFA0

リセット値 : 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rW	rW	rW	rW

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CLAIMSET[3:0]** : 要求タグビットのセット

書込み :

0000 : 影響なし。

xxx1 : ビット 0 をセットします。

xx1x : ビット 1 をセットします。

x1xx : ビット 2 をセットします。

1xxx : ビット 3 をセットします。

読出し :

0xF : 要求タグには 4 つのビットがあることを示します。

TPIU 要求タグクリアレジスタ (TPIU_CLAIMCLR)

アドレスオフセット : 0xFA4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
													rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CLAIMCLR[3:0]** : 要求タグビットのリセット

書込み :

0000 : 影響なし。

xxx1 : ビット 0 をクリアします。

xx1x : ビット 1 をクリアします。

x1xx : ビット 2 をクリアします。

1xxx : ビット 3 をクリアします。

読出し : 要求タグの現在値を返します。

TPIU デバイス設定レジスタ (TPIU_DEVIDR)

アドレスオフセット : 0xFC8

リセット値 : 0x0000 0CA1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWOUARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MAXNUM[4:0]				
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SWOUARTNRZ** : シリアルワイヤ出力、NRZ のサポート

0x1 : サポートされます。

ビット 10 **SWOMAN** : シリアルワイヤ出力、マンチェスタ符号化フォーマットのサポート

0x1 : サポートされます。

ビット 9 **TCLKDATA** : トレースクロックおよびデータのサポート

0x0 : サポートされます。

ビット 8:6 **FIFOSIZE[2:0]** : FIFO のサイズ (2 の累乗値)

0x2 : FIFO のサイズ = 4 バイト

ビット 5 **CLKRELAT** : ATB クロックと TRACECLKIN の関係 (同期または非同期)

0x1 : 非同期

ビット 4:0 **MAXNUM[4:0]** : ATB 入力ポートマルチプレクスの数/タイプ

0x1 : 2 つの入力ポート

TPIU デバイスタイプ識別子レジスタ (TPIU_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUBTYPE[3:0]** : 下位分類

0x1 : トレースポートコンポーネント

ビット 3:0 **MAJORTYPE[3:0]** : 大分類

0x1 : トレースシンクコンポーネント

TPIU CoreSight ペリフェラル ID レジスタ 4 (TPIU_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : Arm® JEDEC コード

TPIU CoreSight ペリフェラル ID レジスタ 0 (TPIU_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : TPIU 部品番号

TPIU CoreSight ペリフェラル ID レジスタ 1 (TPIU_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : TPIU 部品番号

TPIU CoreSight ペリフェラル ID レジスタ 2 (TPIU_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

TPIU CoreSight ペリフェラル ID レジスタ 3 (TPIU_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

TPIU CoreSight コンポーネント ID レジスタ 0 (TPIU_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

TPIU CoreSight ペリフェラル ID レジスタ 1 (TPIU_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0x9 : CoreSight™ コンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

TPIU CoreSight コンポーネント ID レジスタ 2 (TPIU_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

TPIU CoreSight コンポーネント ID レジスタ 3 (TPIU_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.10.2 TPIU レジスタマップ

表 709. TPIU レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x000	TPIU_SSPSR	PORTSIZE[31:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1				
0x004	TPIU_CSPSR	PORTSIZE[31:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
0x008	予約済みです。	予約済みです。																																			
0x010	TPIU_ACPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[12:0]																	
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0				
0x014 から 0x0EC	予約済みです。	予約済みです。																																			
0x0F0	TPIU_SPPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXMODE [1:0]					
	リセット値																															0	1				
0x0F4 から 0x2FC	予約済みです。	予約済みです。																																			
0x300	TPIU_FFSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTNONSTOP	TCPRESENT	FTSTOPPED	FLINPROG				
	リセット値																													1	0	0	0				
0x304	TPIU_FFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGIN	FONMAN	Res.	Res.	Res.	Res.	ENFCON	Res.				
	リセット値																									1	0					1					
0x308	TPIU_PSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSCOUNT[12:0]																
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0				
030C から 0xF9C	予約済みです。	予約済みです。																																			
0xFA0	TPIU_CLAIMSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]					
	リセット値																														1	1	1	1			
0xFA4	TPIU_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]					
	リセット値																														0	0	0	0			
0FA8 から 0xFC4	予約済みです。	予約済みです。																																			
0xFC8	TPIU_DEVIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWOARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]		CLKRELAT		MaXNUM[4:0]									
	リセット値																				1	1	0	0	1	0	1	0	0	0	0	1					
0xFCC	TPIU_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]			MAJORTYPE [3:0]									
	リセット値																								0	0	0	1	0	0	0	1					
0xFD0	TPIU_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]			JEP106CON [3:0]								
	リセット値																									0	0	0	0	0	1	0	0				
0xFE0	TPIU_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]												
	リセット値																									0	0	1	0	0	0	0	1				

表 709. TPIU レジスタのマップとリセット値 (続き)

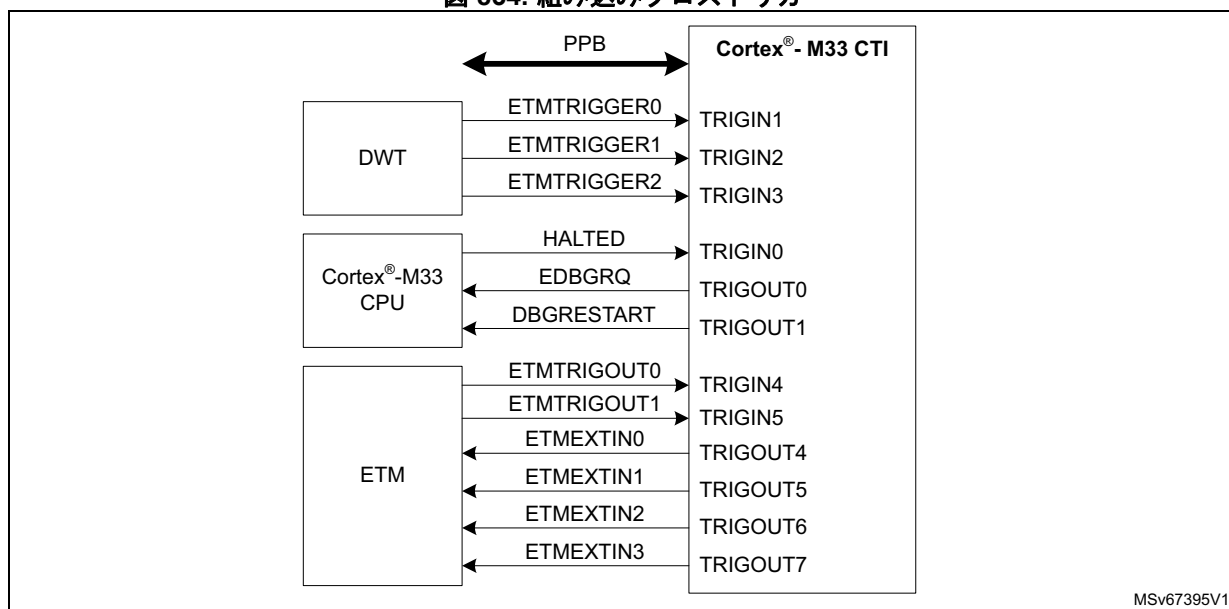
オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0xFE4	TPIU_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]								
	リセット値																									1	0	1	1	1	1	0	1					
0xFE8	TPIU_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]						
	リセット値																									0	0	0	0	1	0	1	1					
0xFEC	TPIU_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]							
	リセット値																									0	0	0	0	0	0	0	0					
0xFF0	TPIU_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]											
	リセット値																										0	0	0	0	1	1	0	1				
0xFF4	TPIU_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]							
	リセット値																									1	0	0	1	0	0	0	0					
0xFF8	TPIU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]											
	リセット値																										0	0	0	0	0	1	0	1				
0xFFC	TPIU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]											
	リセット値																										1	0	1	1	0	0	0	1				

レジスタ境界アドレスについては表 700 : MCU ROM テーブル を参照してください。

58.11 クロストリガインタフェース (CTI)

CTI は、プロセッサと ETM の間のクロストリガを可能にします (下図を参照)。

図 834. 組み込みクロストリガ



CTI は、さまざまな転送元からのイベントによるデバッグ/トレース動作、またはその両方のトリガを有効にします。たとえば、プロセッサで到達したウォッチポイントがコードトレースを開始または停止するか、トレースコンパレータがプロセッサを停止することができます。

CTI のトリガ入力および出力信号は下図に記載されています。

表 710. CTI 入力

番号	転送元信号	転送元コンポーネント	コメント
0	HALTED	CPU	プロセッサ停止 - CPU はデバッグモードです。
1	ETMTRIGGER0	DWT	DWT コンパレータ出力 0
2	ETMTRIGGER1	DWT	DWT コンパレータ出力 1
3	ETMTRIGGER2	DWT	DWT コンパレータ出力 2
4	ETMTRIGOUT0	ETM	ETM イベント出力 0
5	ETMTRIGOUT1	ETM	ETM イベント出力 1
6	-	-	未使用
7	-	-	未使用

表 711. CTI 出力

番号	転送元信号	転送先コンポーネント	コメント
0	EDBGRQ	CPU	CPU 停止リクエスト - CPU をデバッグモードにします。
1	DBGRESTART	CPU	CPU 再開リクエスト - CPU はデバッグモードを終了します。
2	ETMEXTIN0	ETM	ETM イベント入力 0
3	ETMEXTIN1	ETM	ETM イベント入力 1
4	ETMEXTIN2	ETM	ETM イベント入力 2
5	ETMEXTIN3	ETM	ETM イベント入力 3
6	-	-	未使用
7	-	-	未使用

クロストリガインタフェース CoreSight™ コンポーネントの詳細情報は、Arm® CoreSight™ SoC-400 Technical Reference Manual を参照してください [\[2\]](#)。

58.11.1 CTI レジスタ

CTI のレジスタファイルのベースアドレスは 0xE004 2000 です。

CTI 制御レジスタ (CTI_CONTROLR)

アドレスオフセット : 0x000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
															rw

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **GLBEN** : グローバル CTI 有効化

0 : 無効

1 : 有効

CTI トリガ確認応答レジスタ (CTI_INTACKR)

アドレスオフセット : 0x010

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **INTACK[7:0]** : トリガ確認応答

CTITRIGOUT 出力ごとに レジスタの 1 ビットが割り当てられます。このレジスタの 1 ビットに 1 が書き込まれると、対応する CTITRIGOUT 出力が確認応答され、クリアされます。

CTI アプリケーショントリガ設定レジスタ (CTI_APPSETR)

アドレスオフセット : 0x014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **APPSET[3:0]** : チャネルイベント設定

読出し :

XXX0 : チャネル 0 イベントはインアクティブです。

XXX0 : チャネル 0 イベントはアクティブです。

XX0X : チャネル 1 イベントはインアクティブです。

XX1X : チャネル 1 イベントはアクティブです。

X0XX : チャネル 2 イベントはインアクティブです。

X1XX : チャネル 2 イベントはアクティブです。

0XXX : チャネル 3 イベントはインアクティブです。

1XXX : チャネル 3 イベントはアクティブです。

書込み :

XXX0 : 影響はありません。

XXX0 : チャネル 0 にイベントを設定します。

XX0X : 影響はありません。

XX1X : チャネル 1 にイベントを設定します。

X0XX : 影響はありません。

X1XX : チャネル 2 にイベントを設定します。

0XXX : 影響はありません。

1XXX : チャネル 3 にイベントを設定します。

CTI アプリケーショントリガクリアレジスタ (CTI_APPCLEAR)

アドレスオフセット : 0x018

リセット値: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR[3:0]			
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **APPCLEAR[3:0]** : チャネルイベントのクリア

0000：影響なし。

XXX1: チャンネル 0 のイベントをクリアします。

XX1X: チャンネル 1 のイベントをクリアします。

X1XX: チャンネル 2 のイベントをクリアします。

1XXX: チャンネル 3 のイベントをクリアします。

CTI アプリケーションパルスレジスタ (CTI APPPULSER)

アドレスオフセット : 0x01C

リセット値: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE[3:0]			
												w	w	w	w

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **APPPULSE[3:0]** : チャネルイベントのパルス出力

このレジスタ自体がただちにクリアされます。

0000：影響なし。

XXX1: チャンネル 0 のパルスを生成します。

XX1X: チャンネル 1 のパルスを生成します。

X1XX: チャンネル 2 のパルスを生成します。

1XXX: チャンネル 3 のパルスを生成します。

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
												rw	rw	rw	rw

1XXX: トリガ x はチャンネル 3 のイベントを生成します。

リセット値: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
												rw	rw	rw	rw

1XXX: チャンネル 3 イベントはトリガ出力 x のトリガを生成します。

CTI トリガ入カステータスレジスタ (CTI_TRGISTSR)

アドレスオフセット : 0x130

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TRIGINSTATUS[7:0]** : トリガ入カステータス

CTITRIGINx 入力ごとにレジスタの 1 ビットが割り当てられます。このビットが 1 にセットされている場合は、対応するトリガ入力アクティブであることを示します。このビットが 0 に設定されている場合は、対応するトリガ入力はインアクティブです。

CTI トリガ出カステータスレジスタ (CTI_TRGOSTSR)

アドレスオフセット : 0x134

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TRIGOUTSTATUS[7:0]** : トリガ出カステータス

CTITRIGOUT 出力ごとにレジスタの 1 ビットが割り当てられます。このビットが 1 にセットされている場合は、対応するトリガ出力がアクティブであることを示します。このビットが 0 に設定されている場合は、対応するトリガ出力はインアクティブです。

CTI チャンネル入カステータスレジスタ (CTI_CHINSTSR)

アドレスオフセット : 0x138

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHINSTSTATUS[3:0]			
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CHINSTSTATUS[3:0]** : チャンネル入カステータス

チャンネル入力ごとに レジスタの 1 ビットが割り当てられます。このビットが 1 にセットされている場合は、対応するチャンネル入力がアクティブであることを示します。このビットが 0 に設定されている場合は、対応するチャンネル入力はインアクティブです。

CTI チャンネル出カステータスレジスタ (CTI_CHOUTSTSR)

アドレスオフセット : 0x13C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOUTSTATUS[3:0]			
												r	r	r	r

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CHOUTSTATUS[3:0]** : チャンネル出カステータス

チャンネル出力ごとに レジスタの 1 ビットが割り当てられます。このビットが 1 にセットされている場合は、対応するチャンネル出力がアクティブであることを示します。このビットが 0 に設定されている場合は、対応するチャンネル出力はインアクティブです。

リセット値 : 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN[3:0]			
												rw	rw	rw	rw

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]			
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMTRIG[7:0]								Res.	Res.	Res.	EXTMUXNUM[4:0]				
r	r	r	r	r	r	r	r				r	r	r	r	r

CTI デバイスタイプ識別子レジスタ (CTI_DEVTYPER)

アドレスオフセット : 0xFCC

リセット値 : 0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SUBTYPE[3:0]** : 下位分類

0x1 : クロストリガコンポーネント

ビット 3:0 **MAJORTYPE[3:0]** : 大分類

0x4 : このコンポーネントによりデバッガが CoreSight™ SoC-400 システムの他のコンポーネントを制御できることを示します。

CTI CoreSight ペリフェラル ID レジスタ 4 (CTI_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x4 : Arm® JEDEC コード

CTI CoreSight ペリフェラル ID レジスタ 0 (CTI_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x21 : CTI 部品番号

CTI CoreSight ペリフェラル ID レジスタ 1 (CTI_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 00BD

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0xB : Arm® JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0xD : CTI 部品番号

CTI CoreSight ペリフェラル ID レジスタ 2 (CTI_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのレビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x3 : Arm® JEDEC コード

CTI CoreSight ペリフェラル ID レジスタ 3 (CTI_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

CTI CoreSight コンポーネント ID レジスタ 0 (CTI_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

CTI CoreSight ペリフェラル ID レジスタ 1 (CTI_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0x9 : CoreSight™ コンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

CTI CoreSight コンポーネント ID レジスタ 2 (CTI_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

CTI CoreSight コンポーネント ID レジスタ 3 (CTI_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]

0xB1 : 共通の識別値

58.11.2 CTI レジスタマップ

表 712. CTI レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x000	CTI_CONTROLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN			
	リセット値																																0			
0x004 から 0x00C	予約済みです。	予約済みです。																																		
0x010	CTI_INTACKR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]										
	リセット値																									X	X	X	X	X	X	X	X			
0x014	CTI_APPSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]						
	リセット値																													0	0	0	0			
0x018	CTI_APPCLEAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR [3:0]						
	リセット値																													0	0	0	0			
0x01C	CTI_APPPULSER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE [3:0]						
	リセット値																													X	X	X	X			
0x020 から 0x03C	CTI_INEN0R から CTI_INEN7R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]						
	リセット値																													0	0	0	0			
0x040 から 0x09C	予約済みです。	予約済みです。																																		
0x0A0 から 0x0BC	CTI_OUTEN0R から CTI_OUTEN7R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]					
	リセット値																													0	0	0	0			
0x0C0 から 0x12C	予約済みです。	予約済みです。																																		
0x130	CTI_TRGISTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]									
	リセット値																										0	0	0	0	0	0	0			
0x134	CTI_TRGOSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]									
	リセット値																										0	0	0	0	0	0	0			
0x138	CTI_CHINSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHINSTATUS [3:0]						
	リセット値																												0						0	0
0x13C	CTI_CHOUTSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOUTSTATUS[3:0]						
	リセット値																																		0	0
0x140	CTI_GATER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN[3:0]						
	リセット値																													1	1	1	1			
0x144 から 0xFC4	予約済みです。	予約済みです。																																		
0xFC8	CTI_DEVIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]				NUMTRIG[7:0]								Res.	Res.	Res.	EXTMUXNUM [4:0]								
	リセット値												0	1	0	0	0	0	0	0	0	1	0	0	0				0	0	0	0	0			

表 712. CTI レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFCC	CTI_DEVTYPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SUB[3:0]			MAJOR[3:0]				
	リセット値																									0	0	0	1	0	1	0	0
0xFD0	CTI_PIDR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SIZE [3:0]			JEP106CON [3:0]				
	リセット値																									0	0	0	0	0	1	0	0
0xFD4 から 0xFDC	予約済みです。	予約済みです。																															
0xFE0	CTI_PIDR0	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PARTNUM[7:0]							
	リセット値																									0	0	0	1	0	0	0	1
0xFE4	CTI_PIDR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	JEP106ID [3:0]			PARTNUM [11:8]				
	リセット値																									1	0	1	1	1	1	0	1
0xFE8	CTI_PIDR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REVISION [3:0]			JEDEC	JEP106ID [6:4]			
	リセット値																									0	0	0	0	1	0	1	1
0xFEC	CTI_PIDR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REVAND[3:0]			CMOD[3:0]				
	リセット値																									0	0	0	0	0	0	0	0
0xFF0	CTI_CIDR0	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREAMBLE[7:0]							
	リセット値																									0	0	0	0	1	1	0	1
0xFF4	CTI_CIDR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLASS[3:0]			PREAMBLE [11:8]				
	リセット値																									1	0	0	1	0	0	0	0
0xFF8	CTI_CIDR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREAMBLE[19:12]							
	リセット値																									0	0	0	0	0	1	0	1
0xFFC	CTI_CIDR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREAMBLE[27:20]							
	リセット値																									1	0	1	1	0	0	0	1

レジスタ境界アドレスについては表 701 : プロセッサ ROM テーブルを参照してください。

58.12 マイクロコントローラデバッグユニット (DBGMCU)

DBGMCU は、デバッグモードで電源やクロックの動作を制御するレジスタが複数内蔵されているコンポーネントです。デバッグ（またはソフトウェア）の以下の動作を可能にします。

- 低電力モード（SLEEP、STOP、STANDBY）時に、プロセッサコアへのクロックや電源の供給を維持します。
- 低電力モード時に、システムデバッグおよびトレースコンポーネントへのクロックと電源の供給を維持します。
- 一方のプロセッサコアがデバッグモードで停止した場合、特定のペリフェラルへのクロック（SMBUS タイムアウト、ウォッチドッグ、タイマ、RTC）の供給を停止します。

58.12.1 デバイス ID

DBGMCU には ID コードレジスタ DBGMCU_IDCODE が含まれます。このレジスタには、デバイス用の ID コードが格納されます。デバッグツールでは、[セクション 58.5 : ROM テーブル](#)で説明する CoreSight™ 検出手順によりこのレジスタの位置を特定できます。

58.12.2 低電力モードエミュレーション

デバイスが STOP モード（クロックが停止）または STANDBY モード（コア電源がオフ）に移行すると、デバッグではデバッグアクセスポートにアクセスできなくなり、デバイスとの接続が切れます。この状態を回避するため、デバッグ（またはソフトウェア）により [DBGMCU 設定レジスタ \(DBGMCU_CR\)](#) で DBG_STANDBY ビットや DBG_STOP ビットをセットすることができます。これらのビットをセットすると、デバイスを対応する低電力モードにした状態で、プロセッサへのクロックと電力の供給を継続できます。プロセッサは SLEEP モードのまま、正常に低電力モードを終了します。ただし、ペリフェラルデバイスは動作を継続するため、デバイスの動作は実際の低電力モードの動作とは異なることがあります。

58.12.3 ペリフェラルクロックフリーズ

DBGMCU ペリフェラルクロックフリーズレジスタを使用すると、特定のペリフェラルの動作をデバッグモードでサスペンドすることができます。この機能に対応しているペリフェラルユニットを、以下の表に示します。

表 713. ペリフェラルクロックフリーズ制御ビット

バス	制御レジスタ	ペリフェラル	説明
APB1L	DBGMCU_APB1LFZR	I3C1	I3C1 SCL ストールタイムアウトカウンタ
		I2C2	I2C2 SMBUS タイムアウト
		I2C1	I2C1 SMBUS タイムアウト
		IWDG	独立型ウォッチドッグ
		WWDG	ウィンドウ型ウォッチドッグ
		TIM14	汎用タイマ 14
		TIM13	汎用タイマ 13
		TIM12	汎用タイマ 12
		TIM7	汎用タイマ 7
		TIM6	汎用タイマ 6
		TIM5	汎用タイマ 5
		TIM4	汎用タイマ 4
		TIM3	汎用タイマ 3
		TIM2	汎用タイマ 2
APB1H	DBGMCU_APB1HFZR	LPTIM2	低電力タイマ 2
APB2	DBGMCU_APB2FZR	TIM17	汎用タイマ 17
		TIM16	汎用タイマ 16
		TIM15	汎用タイマ 15
		TIM8	汎用タイマ 8
		TIM1	汎用タイマ 1

表 713. ペリフェラルクロックフリーズ制御ビット (続き)

バス	制御レジスタ	ペリフェラル	説明
APB3	DBGMCU_APB3FZR	RTC	リアルタイムクロック
		LPTIM6	低電力タイマ 6
		LPTIM5	低電力タイマ 5
		LPTIM4	低電力タイマ 4
		LPTIM3	低電力タイマ 3
		LPTIM1	低電力タイマ 1
		I2C4	I2C4 SMBUS タイムアウト
		I2C3	I2C3 SMBUS タイムアウト
AHB1	DBGMCU_AHB1FZR	GPDMA2 0 ~ 7	汎用 DMA2 チャンネル 0 ~ 7
		GPDMA1 0 ~ 7	汎用 DMA1 チャンネル 0 ~ 7

各ペリフェラルユニットまたは DMA チャンネルには対応する制御ビット DBG_xxx_STOP があります。ここで、xxx はペリフェラル(または DMA チャンネル)の略称です。制御ビットは DBGMCU_zzzFZR レジスタで構成されます。ここで、zzz はバスの名前に対応します (AHB または APB)。たとえば、DBGMCU_APB1LFZR には APB1L バスのペリフェラルの制御ビットが格納されます。

制御ビットをセットすると、以下の表に従って、デバッグで CPU が停止したときに (HALTED = 1)、対応するペリフェラルの動作をサスペンドできます。

表 714. デバッグモードでのペリフェラルの動作

HALTED	DBG_xxx_STOP	ペリフェラルの動作
0	X	動作が継続します。
1	0	動作が継続します。
1	1	動作がサスペンドします。

デバッグからビット DBG_xxx_STOP へのアクセス可能性は、認証信号 spiden の状態によって変わります。

spiden = 1 (セキュア特権デバッグが有効) の場合、すべてのビットをデバッグとソフトウェアの両方から修正し、読み出すことができます (セキュアおよび非セキュア)。

spiden = 0 (セキュア特権デバッグが無効) の場合、非セキュアなペリフェラル (または DMA チャンネル) に対応するビットのみをデバッグまたはソフトウェアから修正できます。すべてのビットの読出しもできます。

TrustZone 対応ペリフェラルまたは DMA チャンネルのステータス (セキュアまたは非セキュア) は、ペリフェラルにより DBGMCU に通知されます。

58.12.4 DBGMCU レジスタ

DBGMCU レジスタは、システムリセットではリセットされず、パワーオンリセットによってのみリセットされます。それらは AHB アクセスポート経由でデバッガのベースアドレス 0xE004 4000、およびソフトウェアのベースアドレス 0x4402 4000 にアクセス可能です。

DBGMCU ID コードレジスタ (DBGMCU_IDCODE)

アドレスオフセット : 0x00

リセット値 : 0XXXXX 6XXX

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **REV_ID[15:0]** : 版

A : 0x1000

Z : 0x1001

X : 0x1007

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DEV_ID[11:0]** : デバイス識別

0x484 : STM32H563/H573および STM32H562

DBGMCU 設定レジスタ (DBGMCU_CR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッガや CPU からこのレジスタにアクセスできます。認証が行われる前は、デバッガのアクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCRT
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_MODE [1:0]		TRACE_EN	TRACE_IOEN	Res.	DBG_STANDBY	DBG_STOP	Res.
								rw	rw	rw	rw		rw	rw	

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DCRT** : デバッグ認証情報のリセットタイプ

このビットでは、デバッグ認証情報を取り消すために使用するリセットのタイプを選択します。

0 : システムリセット

1 : 電源リセット

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **TRACE_MODE[1:0]** : TRACE ピンの割当て

0x0 : 非同期モードに割り当てられた TRACE ピン (TRACESWO)

0x1 : ポート幅が 1 の同期モードに割り当てられた TRACE ピン (TRACECK、TRACED0)

0x2 : ポート幅が 2 の同期モードに割り当てられた TRACE ピン (TRACECK、TRACED0-1)

0x3 : ポート幅が 4 の同期モードに割り当てられた TRACE ピン (TRACECK、TRACED0-3)

ビット 5 **TRACE_EN** : トレースポートおよびクロックの有効化

このビットは、トレースポートクロック TRACECK を有効にします。

0 : 無効

1 : 有効

ビット 4 **TRACE_IOEN** : TRACE ピンの有効化

0 : 無効。TRACE ピンは割り当てられません。

1 : 有効。TRACE ピンは TRACE_MODE フィールドの値に応じて割り当てられます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG_STANDBY** : STANDBY モードでデバッグ許可

0 : 通常動作

STANDBY モードでは、すべてのクロックが無効化され、コアの電源は自動的に切断されます。

1 : 自動クロック停止/パワーダウンは無効。

すべてのアクティブなクロックおよびオシレータは STANDBY モード中でも動作し続け、コア供給は維持され、すべてのデバッグ機能を許可します。STANDBY モードを終了すると、システムリセットが行われます。

ビット 1 **DBG_STOP** : STOP モードでデバッグ許可

0 : 通常動作

STOP モードでは、すべてのクロックが自動的に無効化されます。

1 : 自動クロック停止は無効。

すべてのアクティブなクロックおよびオシレータは STOP モード中でも動作し続け、すべてのデバッグ機能を許可します。STOP モードから抜けると、クロック設定は STOP モード終了ステートに設定されます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

DBGMCU APB1L ペリフェラルフリーズレジスタ (DBGMCU_APB1LFZR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッグが CPU からこのレジスタにアクセスできます。認証が行われる前は、アクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_I3C1_STOP	DBG_I2C2_STOP	DBG_I2C1_STOP	Res.	Res.	Res.	Res.	Res.
								rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	Res.	Res.	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
			rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **DBG_I3C1_STOP** : デバッグ時の I3C1 SCL ストールカウンタ停止

- 0 : 通常動作。I3C1 SCL ストールタイムアウトカウンタは、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。I3C1 SCL ストールタイムアウトカウンタは、CPU がデバッグモードの間は停止します。

ビット 22 **DBG_I2C2_STOP** : デバッグ時の I2C2 SMBUS タイムアウトの停止

- 0 : 通常動作。I2C2 SMBUS タイムアウトは、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。I2C2 SMBUS タイムアウトは、CPU がデバッグモードの間は停止します。

ビット 21 **DBG_I2C1_STOP** : デバッグ時の I2C1 SMBUS タイムアウトの停止

- 0 : 通常動作。I2C1 SMBUS タイムアウトは、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。I2C1 SMBUS タイムアウトは、CPU がデバッグモードの間は停止します。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DBG_IWDG_STOP** : デバッグ時の IWDG 停止

- 0 : 通常動作。IWDG は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。IWDG は、CPU がデバッグモードの間は停止します。

ビット 11 **DBG_WWDG_STOP** : デバッグ時の WWDG 停止

- 0 : 通常動作。WWDG は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。WWDG は、CPU がデバッグモードの間は停止します。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DBG_TIM14_STOP** : デバッグ時の TIM14 停止

- 0 : 通常動作。TIM14 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM14 は、CPU がデバッグモードの間は停止します。

ビット 7 **DBG_TIM13_STOP** : デバッグ時の TIM13 停止

- 0 : 通常動作。TIM13 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM13 は、CPU がデバッグモードの間は停止します。

ビット 6 **DBG_TIM12_STOP** : デバッグ時の TIM12 停止

- 0 : 通常動作。TIM12 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM12 は、CPU がデバッグモードの間は停止します。

ビット 5 **DBG_TIM7_STOP** : デバッグ時の TIM7 停止

- 0 : 通常動作。TIM7 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM7 は、CPU がデバッグモードの間は停止します。

ビット 4 **DBG_TIM6_STOP** : デバッグ時の TIM6 停止

- 0 : 通常動作。TIM6 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM6 は、CPU がデバッグモードの間は停止します。

ビット 3 **DBG_TIM5_STOP** : デバッグ時の TIM5 停止

- 0 : 通常動作。TIM5 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM5 は、CPU がデバッグモードの間は停止します。

ビット 2 **DBG_TIM4_STOP** : デバッグ時の TIM4 停止

- 0 : 通常動作。TIM4 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM4 は、CPU がデバッグモードの間は停止します。

ビット 1 **DBG_TIM3_STOP** : デバッグ時の TIM3 停止

- 0 : 通常動作。TIM3 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM3 は、CPU がデバッグモードの間は停止します。

ビット 0 **DBG_TIM2_STOP** : デバッグ時の TIM2 停止

- 0 : 通常動作。TIM2 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。TIM2 は、CPU がデバッグモードの間は停止します。

DBGMCU APB1H ペリフェラルフリーズレジスタ (DBGMCU_APB1HFZR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッグや CPU からこのレジスタにアクセスできます。認証が行われる前は、アクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	Res.	Res.
										rw					

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **DBG_LPTIM2_STOP** : デバッグ時の LPTIM3 停止

- 0 : 通常動作。LPTIM2 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM2 は、CPU がデバッグモードの間は停止します。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

DBGMCU APB2 ペリフェラルフリーズレジスタ (DBGMCU_APB2FZR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッガや CPU からこのレジスタにアクセスできます。認証が行われる前は、アクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DBG_TIM8_STOP	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rw		rw											

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **DBG_TIM17_STOP** : デバッグ時の TIM17 停止

0 : 通常動作。TIM17 は、CPU がデバッグモードの間は動作し続けます。

1 : デバッグ時の停止。TIM17 は、CPU がデバッグモードの間は停止します。

ビット 17 **DBG_TIM16_STOP** : デバッグ時の TIM16 停止

0 : 通常動作。TIM16 は、CPU がデバッグモードの間は動作し続けます。

1 : デバッグ時の停止。TIM16 は、CPU がデバッグモードの間は停止します。

ビット 16 **DBG_TIM15_STOP** : デバッグ時の TIM15 停止

0 : 通常動作。TIM15 は、CPU がデバッグモードの間は動作し続けます。

1 : デバッグ時の停止。TIM15 は、CPU がデバッグモードの間は停止します。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **DBG_TIM8_STOP** : デバッグ時の TIM8 停止

0 : 通常動作。TIM8 は、CPU がデバッグモードの間は動作し続けます。

1 : デバッグ時の停止。TIM8 は、CPU がデバッグモードの間は停止します。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **DBG_TIM1_STOP** : デバッグ時の TIM1 停止

0 : 通常動作。TIM1 は、CPU がデバッグモードの間は動作し続けます。

1 : デバッグ時の停止。TIM1 は、CPU がデバッグモードの間は停止します。

ビット 10:0 予約済みであり、リセット値に保持する必要があります。

DBGMCU APB3 ペリフェラルフリーズレジスタ (DBGMCU_APB3FZR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッグが CPU からこのレジスタにアクセスできます。認証が行われる前は、アクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DBG_RTC_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM6_STOP	DBG_LPTIM5_STOP	DBG_LPTIM4_STOP	DBG_LPTIM3_STOP	DBG_LPTIM1_STOP	Res.
	rw									rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DBG_I2C4_STOP	DBG_I2C3_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw										

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **DBG_RTC_STOP** : デバッグ時の RTC 停止

- 0 : 通常動作。RTC は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。RTC は、CPU がデバッグモードの間は停止します。

ビット 29:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **DBG_LPTIM6_STOP** : デバッグ時の LPTIM6 停止

- 0 : 通常動作。LPTIM6 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM6 は、CPU がデバッグモードの間は停止します。

ビット 20 **DBG_LPTIM5_STOP** : デバッグ時の LPTIM5 停止

- 0 : 通常動作。LPTIM5 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM5 は、CPU がデバッグモードの間は停止します。

ビット 19 **DBG_LPTIM4_STOP** : デバッグ時の LPTIM4 停止

- 0 : 通常動作。LPTIM4 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM4 は、CPU がデバッグモードの間は停止します。

ビット 18 **DBG_LPTIM3_STOP** : デバッグ時の LPTIM3 停止

- 0 : 通常動作。LPTIM3 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM3 は、CPU がデバッグモードの間は停止します。

ビット 17 **DBG_LPTIM1_STOP** : デバッグ時の LPTIM1 停止

- 0 : 通常動作。LPTIM1 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。LPTIM1 は、CPU がデバッグモードの間は停止します。

ビット 16:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **DBG_I2C4_STOP** : デバッグ時の I2C4 SMBUS タイムアウトの停止

- 0 : 通常動作。I2C4 SMBUS タイムアウトカウンタは、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。I2C4 SMBUS タイムアウトカウンタは、CPU がデバッグモードの間は停止します。

ビット 10 **DBG_I2C3_STOP** : デバッグ時の I2C3 SMBUS タイムアウトの停止

- 0 : 通常動作。I2C3 SMBUS タイムアウトカウンタは、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。I2C3 SMBUS タイムアウトカウンタは、CPU がデバッグモードの間は停止します。

ビット 9:0 予約済みであり、リセット値に保持する必要があります。

DBGMCU_AHB1 ペリフェラルフリーズレジスタ (DBGMCU_AHB1FZR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

認証が正しく行われれば、デバッグや CPU からこのレジスタにアクセスできます。認証が行われる前は、アクセスは無視されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_GPDMA2_7_STOP	DBG_GPDMA2_6_STOP	DBG_GPDMA2_5_STOP	DBG_GPDMA2_4_STOP	DBG_GPDMA2_3_STOP	DBG_GPDMA2_2_STOP	DBG_GPDMA2_1_STOP	DBG_GPDMA2_0_STOP
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_GPDMA1_7_STOP	DBG_GPDMA1_6_STOP	DBG_GPDMA1_5_STOP	DBG_GPDMA1_4_STOP	DBG_GPDMA1_3_STOP	DBG_GPDMA1_2_STOP	DBG_GPDMA1_1_STOP	DBG_GPDMA1_0_STOP
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **DBG_GPDMA2_7_STOP** : デバッグ時の GPDMA2 チャンネル 7 停止

- 0 : 通常動作。GPDMA2 チャンネル 7 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 7 は、CPU がデバッグモードの間は停止します。

ビット 22 **DBG_GPDMA2_6_STOP** : デバッグ時の GPDMA2 チャンネル 6 停止

- 0 : 通常動作。GPDMA2 チャンネル 6 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 6 は、CPU がデバッグモードの間は停止します。

ビット 21 **DBG_GPDMA2_5_STOP** : デバッグ時の GPDMA2 チャンネル 5 停止

- 0 : 通常動作。GPDMA2 チャンネル 5 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 5 は、CPU がデバッグモードの間は停止します。

ビット 20 **DBG_GPDMA2_4_STOP** : デバッグ時の GPDMA2 チャンネル 4 停止

- 0 : 通常動作。GPDMA2 チャンネル 4 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 4 は、CPU がデバッグモードの間は停止します。

ビット 19 **DBG_GPDMA2_3_STOP** : デバッグ時の GPDMA2 チャンネル 3 停止

- 0 : 通常動作。GPDMA2 チャンネル 3 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 3 は、CPU がデバッグモードの間は停止します。

ビット 18 **DBG_GPDMA2_2_STOP** : デバッグ時の GPDMA2 チャンネル 2 停止

- 0 : 通常動作。GPDMA2 チャンネル 2 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 2 は、CPU がデバッグモードの間は停止します。

ビット 17 **DBG_GPDMA2_1_STOP** : デバッグ時の GPDMA2 チャンネル 1 停止

- 0 : 通常動作。GPDMA2 チャンネル 1 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 1 は、CPU がデバッグモードの間は停止します。

ビット 16 **DBG_GPDMA2_0_STOP** : デバッグ時の GPDMA2 チャンネル 0 停止

- 0 : 通常動作。GPDMA2 チャンネル 0 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA2 チャンネル 0 は、CPU がデバッグモードの間は停止します。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **DBG_GPDMA1_7_STOP** : デバッグ時の GPDMA1 チャンネル 7 停止

- 0 : 通常動作。GPDMA1 チャンネル 7 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 7 は、CPU がデバッグモードの間は停止します。

ビット 6 **DBG_GPDMA1_6_STOP** : デバッグ時の GPDMA1 チャンネル 6 停止

- 0 : 通常動作。GPDMA1 チャンネル 6 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 6 は、CPU がデバッグモードの間は停止します。

ビット 5 **DBG_GPDMA1_5_STOP** : デバッグ時の GPDMA1 チャンネル 5 停止

- 0 : 通常動作。GPDMA1 チャンネル 5 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 5 は、CPU がデバッグモードの間は停止します。

ビット 4 **DBG_GPDMA1_4_STOP** : デバッグ時の GPDMA1 チャンネル 4 停止

- 0 : 通常動作。GPDMA1 チャンネル 4 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 4 は、CPU がデバッグモードの間は停止します。

ビット 3 **DBG_GPDMA1_3_STOP** : デバッグ時の GPDMA1 チャンネル 3 停止

- 0 : 通常動作。GPDMA1 チャンネル 3 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 3 は、CPU がデバッグモードの間は停止します。

ビット 2 **DBG_GPDMA1_2_STOP** : デバッグ時の GPDMA1 チャンネル 2 停止

- 0 : 通常動作。GPDMA1 チャンネル 2 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 2 は、CPU がデバッグモードの間は停止します。

ビット 1 **DBG_GPDMA1_1_STOP** : デバッグ時の GPDMA1 チャンネル 1 停止

- 0 : 通常動作。GPDMA1 チャンネル 1 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 1 は、CPU がデバッグモードの間は停止します。

ビット 0 **DBG_GPDMA1_0_STOP** : デバッグ時の GPDMA1 チャンネル 0 停止

- 0 : 通常動作。GPDMA1 チャンネル 0 は、CPU がデバッグモードの間は動作し続けます。
- 1 : デバッグ時の停止。GPDMA1 チャンネル 0 は、CPU がデバッグモードの間は停止します。

DBGMCU ステータスレジスタ (DBGMCU_SR)

アドレスオフセット : 0xFC

リセット値 : 0x0001 XX03

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AP_ENABLED[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AP_PRESENT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **AP_ENABLED[15:0]** : ビット n は、アクセスポート AP n がオープンしている（デバッグポートからアクセス可能）かロックされている（AP へのデバッグアクセスがブロックされる）かを示します。

Bit n = 0 : APn がロック

Bit n = 1 : APn が有効

ビット 15:0 **AP_PRESENT[15:0]** : ビット n は、アクセスポート AP n がデバイスにあるかどうかを示します。

Bit n = 0 : APn がありません

Bit n = 1 : APn があります

DBGMCU デバッグ認証メールボックスホストレジスタ (DBGMCU_DBG_AUTH_HOST)

アドレスオフセット : 0x100

リセット値 : 0xFFFF XXXX

このレジスタは、CPU によってアクセスされた場合のみ読み出され、書込みは無効です。

このレジスタは、システムリセットがアサートされるか、認証プロセス中にアクセスが許可される場合のみ、外部デバッグによる書込み/読出しができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MESSAGE[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGE[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MESSAGE[31:0]** : デバッグホストからデバイスへのメールボックスメッセージ。

デバッグホストは、認証をリクエストするために、システムリセットをリリースする前にこのレジスタに値を書き込みます。デバッグ認証中、デバッグホストはこのレジスタを通じてデバイス（CPU）と通信します。

DBGMCU デバッグ認証メールボックスデバイスレジスタ (DBGMCU_DBG_AUTH_DEVICE)

アドレスオフセット : 0x104

リセット値 : 0XXXXX XXXX

このレジスタは、デバッグポートを通じてアクセスされた場合のみ読み出され、書込みは無効です。

このレジスタの書込み/読出しは、CPUで行うことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MESSAGE[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGE[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **MESSAGE[31:0]** : デバイスからデバッグホストへのメールボックスメッセージ
デバッグ認証中、デバイス (CPU) はこのレジスタを通じてデバッグホストと通信します。

DBGMCU デバッグ認証メールボックス確認応答レジスタ (DBGMCU_DBG_AUTH_ACK)

アドレスオフセット : 0x108

リセット値 : 0x0000 0000

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV_ACK	HOST_ACK
														r	r

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DEV_ACK** : デバイスからホストへの確認応答

このビットは、デバイス (CPU) が DBGMCU_DBG_AUTH_DEVICE レジスタにメッセージを書き込んだときに、ハードウェアによってセットされます。ホスト (デバッガ) がメッセージを読み出すと、自動的にリセットされます。

0 : DBGMCU_DBG_AUTH_DEVICE デバイスが空です

1. DBGMCU_DBG_AUTH_DEVICE レジスタに読み出されていないメッセージがあります

ビット 0 **HOST_ACK** : ホストからデバイスへの確認応答。

このビットは、ホスト (デバッガ) DBGMCU_DBG_AUTH_HOST レジスタにメッセージを書き込んだときに、ハードウェアによってセットされます。デバイス (CPU) がメッセージを読み出すと、自動的にリセットされます。

0 : DBGMCU_DBG_AUTH_HOST デバイスが空です

1. DBGMCU_DBG_AUTH_HOST レジスタに読み出されていないメッセージがあります

DBGMCU CoreSight ペリフェラル ID レジスタ 4 (DBGMCU_PIDR4)

アドレスオフセット : 0xFD0

リセット値 : 0x0000 0000

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE[3:0]				JEP106CON [3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **SIZE[3:0]** : レジスタファイルサイズ

0x0 : レジスタファイルは 4 KB の領域 1 つ分を占有します。

ビット 3:0 **JEP106CON[3:0]** : JEP106 継続コード

0x0 : STMicroelectronics JEDEC コード

DBGMCU CoreSight ペリフェラル ID レジスタ 0 (DBGMCU_PIDR0)

アドレスオフセット : 0xFE0

リセット値 : 0x0000 0000

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PARTNUM[7:0]** : 部品番号のビット [7:0]

0x00 : DBGMCU 部品番号

DBGMCU CoreSight ペリフェラル ID レジスタ 1 (DBGMCU_PIDR1)

アドレスオフセット : 0xFE4

リセット値 : 0x0000 0000

このレジスタは常にアクセス可能です

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **JEP106ID[3:0]** : JEP106 ID コードビット [3:0]

0x0 : STMicroelectronics JEDEC コード

ビット 3:0 **PARTNUM[11:8]** : 部品番号のビット [11:8]

0x0 : DBGMCU 部品番号

DBGMCU CoreSight ペリフェラル ID レジスタ 2 (DBGMCU_PIDR2)

アドレスオフセット : 0xFE8

リセット値 : 0x0000 000A

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID [6:4]		
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVISION[3:0]** : コンポーネントのリビジョン番号

0x0 : r0p0

ビット 3 **JEDEC** : JEDEC 割り当て値

0x1 : JEDEC によって指定される設計者識別

ビット 2:0 **JEP106ID[6:4]** : JEP106 ID コードビット [6:4]

0x2 : STMicroelectronics JEDEC コード

DBGMCU CoreSight ペリフェラル ID レジスタ 3 (DBGMCU_PIDR3)

アドレスオフセット : 0xFEC

リセット値 : 0x0000 0000

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **REVAND[3:0]** : メタルフィックスバージョン

0x0 : メタルフィックスではありません。

ビット 3:0 **CMOD[3:0]** : 顧客による修正

0x0 : 顧客による修正はありません。

DBGMCU CoreSight コンポーネント ID レジスタ 0 (DBGMCU_CIDR0)

アドレスオフセット : 0xFF0

リセット値 : 0x0000 000D

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[7:0]** : コンポーネント識別ビット [7:0]

0x0D : 共通の識別値

DBGMCU CoreSight コンポーネント ID レジスタ 1 (DBGMCU_CIDR1)

アドレスオフセット : 0xFF4

リセット値 : 0x0000 00F0

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **CLASS[3:0]** : コンポーネント ID ビット [15:12] - コンポーネントクラス

0xF : CoreSight 以外のコンポーネント

ビット 3:0 **PREAMBLE[11:8]** : コンポーネント識別ビット [11:8]

0x0 : 共通の識別値

DBGMCU CoreSight コンポーネント ID レジスタ 2 (DBGMCU_CIDR2)

アドレスオフセット : 0xFF8

リセット値 : 0x0000 0005

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[19:12]** : コンポーネント識別ビット [23:16]

0x05 : 共通の識別値

DBGMCU CoreSight コンポーネント ID レジスタ 3 (DBGMCU_CIDR3)

アドレスオフセット : 0xFFC

リセット値 : 0x0000 00B1

このレジスタは常にアクセス可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **PREAMBLE[27:20]** : コンポーネント識別ビット [31:24]
0xB1 : 共通の識別値

58.12.5 DBGMCU レジスタマップ

表 715. DBGMCU レジスタのマップとリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x000	DBGMCU_IDCODE	REV_ID[15:0]																DEV_ID[11:0]																	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X		
0x004	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.		TRACE_MODE [1:0]	TRACE_EN	TRACE_IJEN	Res.	DBG_STANDBY	DBG_STOP	Res.			
	リセット値																0									0	0	0	0	0	0	0	0		
0x008	DBGMCU_APB1LFZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_I3C1_STOP	DBG_I2C2_STOP	DBG_I2C1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	Res.	Res.	Res.	Res.	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
	リセット値									0	0	0									0	0				0	0	0	0	0	0	0	0	0	
0x00C0	DBGMCU_APB1HFZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP								
	リセット値																										0								
0x010	DBGMCU_APB2FZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP	Res.	Res.	DBG_TIM8_STOP		Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値														0	0	0			0		0													
0x04	DBGMCU_APB3FZR	Res.	DBG_RTC_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM6_STOP	DBG_LPTIM5_STOP	DBG_LPTIM4_STOP	DBG_LPTIM3_STOP	DBG_LPTIM1_STOP	Res.	Res.	Res.	Res.	Res.	DBG_I2C4_STOP	DBG_I2C3_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0													0	0	0					0	0												
0x018 から 0x01C	予約済みです。	予約済みです。																																	
0x020	DBGMCU_AHB1FZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_GPDMA2_7_STOP	DBG_GPDMA2_6_STOP	DBG_GPDMA2_5_STOP	DBG_GPDMA2_4_STOP	DBG_GPDMA2_3_STOP	DBG_GPDMA2_2_STOP	DBG_GPDMA2_1_STOP	DBG_GPDMA2_0_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_GPDMA1_7_STOP	DBG_GPDMA1_6_STOP	DBG_GPDMA1_5_STOP	DBG_GPDMA1_4_STOP	DBG_GPDMA1_3_STOP	DBG_GPDMA1_2_STOP	DBG_GPDMA1_1_STOP	DBG_GPDMA1_0_STOP	
	リセット値										0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	
0x024 から 0x0F8	予約済みです。	予約済みです。																																	
0x0FC	DBGMCU_SR	AP_ENABLED[15:0]																AP_PRESENT[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	

表 715. DBGMCU レジスタのマップとリセット値 (続き)

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x100	DBGMCU_DBG_AUTH_HOST	MESSAGE[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x104	DBGMCU_DBG_AUTH_DEVICE	MESSAGE[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x108	DBGMCU_DBG_AUTH_ACK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV_ACK	HOST_ACK
	リセット値																														0	0	
0x10C から 0xFBC	予約済みです。	予約済みです。																															
0xFD0	DBGMCU_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SIZE [3:0]				JEP106CON [3:0]			
	リセット値																									0	0	0	0	0	0	0	0
0xFD4 から 0xFDC	予約済みです。	予約済みです。																															
0xFE0	DBGMCU_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	リセット値																									0	0	0	0	0	0	0	0
0xFE4	DBGMCU_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	リセット値																									0	0	0	0	0	0	0	0
0xFE8	DBGMCU_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	リセット値																									0	0	0	0	1	0	1	0
0xFEC	DBGMCU_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	リセット値																									0	0	0	0	0	0	0	0
0xFF0	DBGMCU_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	リセット値																									0	0	0	0	1	1	0	1
0xFF4	DBGMCU_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	リセット値																									1	1	1	1	0	0	0	0
0xFF8	DBGMCU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	リセット値																									0	0	0	0	0	1	0	1
0xFFC	DBGMCU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	リセット値																									1	0	1	1	0	0	0	1

レジスタ境界アドレスについては[セクション 2.3](#) を参照してください。

58.13 参照

1. IHI 0031C (ID080813) - Arm® Debug Interface Architecture Specification ADIv5.0 ~ ADIv5.2、C 版、2013 年 8 月 8 日
2. DDI 0480F (ID100313) - Arm® CoreSight™ SoC-400 r3p2 Technical Reference Manual、G 版、2015 年 3 月 16 日
3. DDI 0314H - Arm® CoreSight™ Components Technical Reference Manual、H 版、2009 年 7 月 10 日
4. DDI 0553A (ID092917) - Arm® v8-M Architecture Reference Manual、A.f 版、2017 年 9 月 29 日
5. 100230_0002_00_en - Arm® Cortex®-M33 Processor r0p2 Technical Reference Manual、0002-00 版、2017 年 5 月 10 日
6. 100232_0001_00_en - Arm® CoreSight™ ETM-M33 r0p1 Technical Reference Manual、0001-00 版、2017 年 2 月 3 日

59 デバイス電子署名

デバイス電子署名は、Flash メモリモジュールのシステムメモリ領域に格納され、デバッグインタフェースまたは CPU を使用して読み出すことができます。電子署名に含まれる出荷時にプログラムされた識別データおよび較正データを使用すれば、ユーザファームウェアやその他の外部デバイスは、デバイスの特性に自動的に整合させることができます。

59.1 ユニークデバイス ID レジスタ（96 ビット）

このユニークデバイス識別子は、以下の用途に最適です。

- シリアル番号（例：USB 文字列シリアル番号やその他のエンドアプリケーション）として使用
- 内部Flashメモリをプログラムする前に、このユニーク ID をソフトウェア暗号プリミティブやプロトコルと組み合わせて使用する際に、Flashメモリ内のコードのセキュリティを高めるためのセキュリティキーの一部として使用
- セキュアなブートプロセスの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

ベースアドレス：0x08FF : F800

アドレスオフセット：0x00

読出し専用 = 0xXXXX XXXX（X は出荷時にプログラムされます）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **UID[31:0]** : ウェハ上での X および Y 座標

アドレスオフセット : 0x04

読出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:8 **UID[63:40]** : LOT_NUM[23:0]

ロット番号 (ASCII で符号化)

ビット 7:0 **UID[39:32]** : WAF_NUM[7:0]

ウェハー番号 (8 ビットの符号なし番号)

アドレスオフセット : 0x08

読出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **UID[95:64]** : LOT_NUM[55:24]

ロット番号 (ASCII で符号化)

59.2 Flashサイズデータレジスタ

ベースアドレス : 0x08FF : F80C

アドレスオフセット : 0x00

読出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flash_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **Flash_SIZE[15:0]** : Flashメモリサイズ

このフィールドは、デバイスの Flash メモリサイズを KB 単位で示します。

たとえば、0x800 は 2048 KB に対応します。

59.3 パッケージデータレジスタ

ベースアドレス : 0x08FF : F80E

アドレスオフセット : 0x00

読出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKG[4:0]				
											r	r	r	r	r

ビット 15:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **PKG[4:0]** : パッケージタイプ

- 00000 : LQFP64
- 00001 : VFQFPN68
- 00010 : LQFP100
- 00100 : LQFP144
- 00111 : LQFP176
- 00011 : UFBGA176+25
- 00110 : UFBGA169
- 01010 : LQFP100 SMPS
- 01100 : LQFP144 SMPS
- 01101 : LQFP176 SMPS
- 01011 : UFBGA176 +25 SMPS
- 01110 : UFBGA169 SMPS
- その他 : 予約済み

60 セキュリティ上の重要通知

ST マイクロエレクトロニクス（ST）では製品のセキュリティを重視しているため、本書に示す ST の製品は各種セキュリティ認証機関の認証を受け、ここに記載する独自のセキュリティ対策を実施しています。ただし、どのセキュリティ認証も内部のセキュリティ対策も、あらゆる形態の攻撃に対する ST 製品の耐性を保証するものではありません。そのため、ST 製品を単体で使用する場合も、他のコンポーネントやソフトウェアと組み合わせてお客様の最終製品やアプリケーションを構築する場合も、ST 製品が提供するセキュリティのレベルがお客様のニーズを満たしているか、お客様の責任でご判断ください。特に、次の点に注意してください。

- ST 製品は、Platform Security Architecture (www.psacertified.org)、Security Evaluation standard for IoT Platforms (www.trustcb.com) をはじめとする 1 つまたは複数のセキュリティ認証機関の認証を受けている場合があります。ここに示す ST 製品がこのような認証のレベルや現在のステータスと一緒にセキュリティ認証を受けているかどうかについてお知らせになりたい場合は、関連する認証基準のウェブサイトまたは関連する製品ページ (www.st.com) にアクセスして、最新情報をご確認ください。ST 製品のセキュリティ認証のステータスやレベルは変更されることがあるため、必要に応じてセキュリティ認証のステータス/レベルをその都度ご確認ください。特定のセキュリティ基準への適合性が証明されない場合、認証を受けているとみなさないでください。
- 認証機関は ST 製品に関して、セキュリティ認証を評価し、付与し、取り消す権利を有しています。そのため、ST 製品に対するセキュリティ認証の付与または取り消しに関する責任はこのような認証機関のみにあり、ST は ST 製品に関して認証機関で生じた誤り、実施した評価、査定、テスト、その他の活動についていかなる責任も負いません。
- ST 製品と一緒に使用できる業界向けの暗号化アルゴリズム（AES、DES、MD5 など）やその他のオープンスタンダードのテクノロジーは、ST が開発したものではない基準に基づいています。ST は、このような暗号化アルゴリズムやオープンテクノロジーの欠陥について、またこのようなアルゴリズムやテクノロジーの迂回、復号、解読のために開発された、または開発される可能性のある技法について、責任を負いません。
- 堅固なセキュリティテストは実施されていると考えられますが、いかなるレベルの認証でも、まだテストがされていない高度な攻撃、新しく未定義の攻撃の形態、仕様や想定された使用法に従わずに ST 製品を使用した場合や、最終製品またはアプリケーションを構築するためにお客様が他のコンポーネントやソフトウェアと一緒に使用した場合の攻撃の形態など、あらゆる攻撃に対する保護について絶対的な保証をすることはできません。ST はこのような攻撃に対する耐性について、責任を負いません。そのため、組み込まれたセキュリティ機能、ST が提供する情報やサポートに関係なく、ST 製品を単体で使用する場合も、お客様の最終製品またはアプリケーションに組み込んで使用する場合も、テストする攻撃のレベルが実際のニーズに適合しているかどうかを判断するのは、お客様の責任です。
- ST 製品のあらゆるセキュリティ機能（ハードウェア、ソフトウェア、ドキュメントなどを含む）は、ST が付加する強化されたセキュリティ機能を含め（ただし、これに限定しない）、現状のまま提供されます。そのため、適用される署名付きの書面による契約条件で特に規定していない限り、適用される法律が許す範囲で、特定の目的に対する商品性または適格性の黙示の保証を含め（ただし、これに限定しない）、ST はすべての明示的または黙示的な保証を行わないものとします。

61 改版履歴

表 716. 文書改版履歴

日付	版	変更内容
2023 年 2 月 28 日	1	初版発行

表 717. 日本語版文書改版履歴

日付	版	変更内容
2023 年 9 月	1	日本語版 初版発行

索引

A

ADC_AWD2CR	1108
ADC_AWD3CR	1108
ADC_CALFACT	1109
ADC_CCR	1112
ADC_CDR	1115
ADC_CFGR	1090
ADC_CFGR2	1095
ADC_CR	1087
ADC_CSR	1111
ADC_DIFSEL	1109
ADC_DR	1104
ADC_HWCFGR0	1116
ADC_IER	1085
ADC_IPDR	1117
ADC_ISR	1083
ADC_JDRy	1107
ADC_JSQR	1104
ADC_OFRy	1106
ADC_OR	1110
ADC_SIDR	1117
ADC_SMPR1	1097
ADC_SMPR2	1098
ADC_SQR1	1100
ADC_SQR2	1101
ADC_SQR3	1102
ADC_SQR4	1103
ADC_TR1	1098
ADC_TR2	1099
ADC_TR3	1100
ADC_VERR	1116
AES_CR	1263
AES_DINR	1266
AES_DOUTR	1267
AES_ICR	1274
AES_IER	1272
AES_ISR	1273
AES_IVR0	1269
AES_IVR1	1269
AES_IVR2	1269
AES_IVR3	1270
AES_KEYR0	1267
AES_KEYR1	1268
AES_KEYR2	1268
AES_KEYR3	1268
AES_KEYR4	1270
AES_KEYR5	1270
AES_KEYR6	1271

AES_KEYR7	1271
AES_SR	1265
AES_SUSPxR	1272
AP0_CSWR	2926
AP1_CSWR	2927
APx_BASER	2929
APx_BDnR	2929
APx_DRWR	2928
APx_IDR	2930
APx_TAR	2928

B

BPU_CIDR0	2987
BPU_CIDR1	2987
BPU_CIDR2	2988
BPU_CIDR3	2988
BPU_COMPxR	2983
BPU_CTRLR	2982
BPU_DEVARCHR	2983
BPU_DEVTYPER	2984
BPU_PIDR0	2985
BPU_PIDR1	2985
BPU_PIDR2	2986
BPU_PIDR3	2986
BPU_PIDR4	2984

C

C1ROM_CIDR3	2940, 2946, 2952
CORDIC_CSR	758
CORDIC_RDATA	761
CORDIC_WDATA	760
CPUROM_CIDR0	2950
CPUROM_CIDR1	2950
CPUROM_CIDR2	2951
CPUROM_CIDR3	2951
CPUROM_MEMTYPER	2947
CPUROM_PIDR0	2948
CPUROM_PIDR1	2948
CPUROM_PIDR2	2949
CPUROM_PIDR3	2949
CPUROM_PIDR4	2947
CRC_CR	741
CRC_DR	740
CRC_IDR	740
CRC_INIT	742
CRC_POL	742
CRS_CFGR	553

CRS_CR	552	DBGMCU_APB2FZR	3055
CRS_ICR	556	DBGMCU_APB3FZR	3056
CRS_ISR	554	DBGMCU_CIDR0	3063
CTI_APPCLEAR	3037	DBGMCU_CIDR1	3064
CTI_APPPULSER	3037	DBGMCU_CIDR2	3064
CTI_APPSETR	3036	DBGMCU_CIDR3	3065
CTI_CHINSTSR	3040	DBGMCU_CR	3051
CTI_CHOUTSTSR	3040	DBGMCU_DBG_AUTH_ACK	3060
CTI_CIDR0	3045	DBGMCU_DBG_AUTH_DEVICE	3060
CTI_CIDR1	3045	DBGMCU_DBG_AUTH_HOST	3059
CTI_CIDR2	3046	DBGMCU_IDCODE	3051
CTI_CIDR3	3046	DBGMCU_PIDR0	3061
CTI_CONTROLR	3035	DBGMCU_PIDR1	3062
CTI_DEVIDR	3041	DBGMCU_PIDR2	3062
CTI_DEVTYPER	3042	DBGMCU_PIDR3	3063
CTI_GATER	3041	DBGMCU_PIDR4	3061
CTI_INENxR	3038	DBGMCU_SR	3059
CTI_INTACKR	3035	DCACHE_CMDREADDRR	387
CTI_OUTENxR	3038	DCACHE_CMDRSADDR	386
CTI_PIDR0	3043	DCACHE_CR	381
CTI_PIDR1	3043	DCACHE_FCR	384
CTI_PIDR2	3044	DCACHE_IER	384
CTI_PIDR3	3044	DCACHE_RHMONR	385
CTI_PIDR4	3042	DCACHE_RMMONR	385
CTI_TRGISTSR	3039	DCACHE_SR	383
CTI_TRGOSTSR	3039	DCACHE_WHMONR	386
		DCACHE_WMMONR	386
D		DCMI_CR	1190
DAC_CCR	1169	DCMI_CWSIZE	1199
DAC_CR	1158	DCMI_CWSTRT	1199
DAC_DHR12L1	1162	DCMI_DR	1200
DAC_DHR12L2	1164	DCMI_ESCR	1197
DAC_DHR12LD	1165	DCMI_ESUR	1198
DAC_DHR12R1	1162	DCMI_ICR	1196
DAC_DHR12R2	1163	DCMI_IER	1194
DAC_DHR12RD	1165	DCMI_MIS	1195
DAC_DHR8R1	1163	DCMI_RIS	1193
DAC_DHR8R2	1164	DCMI_SR	1192
DAC_DHR8RD	1166	DLYB_CFGR	1004
DAC_DOR1	1166	DLYB_CR	1003
DAC_DOR2	1167	DP_ABORTR	2918
DAC_MCR	1169	DP_CTRLSTATR	2919
DAC_SHHR	1172	DP_DLCR	2920
DAC_SHRR	1172	DP_DLPIDR	2921
DAC_SHSR1	1171	DP_DPIDR	2917
DAC_SHSR2	1171	DP_RDBUFFR	2923
DAC_SR	1167	DP_RESENDER	2922
DAC_SWTRGR	1161	DP_SELECTR	2922
DBGMCU_AHB1FZR	3057	DP_TARGETIDR	2921
DBGMCU_APB1HFZR	3054	DTS_CFGR1	1129
DBGMCU_APB1LFZR	3053	DTS_DR	1132
		DTS_ICIFR	1135

DTS_ITENR	1134	ETH_DMAMR	2773
DTS_ITR1	1132	ETH_DMASBMR	2775
DTS_OR	1136	ETH_MAC1USTCR	2837
DTS_RAMPVALR	1131	ETH_MACA0HR	2851
DTS_SR	1133	ETH_MACACR	2881
DTS_TOVALR1	1131	ETH_MACARPAR	2850
DWT_CIDR0	2966	ETH_MACATSNR	2882
DWT_CIDR1	2967	ETH_MACATSSR	2883
DWT_CIDR2	2967	ETH_MACAxHR	2852
DWT_CIDR3	2968	ETH_MACAxLR	2851
DWT_COMPxR	2958	ETH_MACCR	2809
DWT_CPICNTR	2956	ETH_MACCSRSWCR	2850
DWT_CTRLR	2954	ETH_MACDR	2838
DWT_CYCCNTR	2955	ETH_MACECR	2814
DWT_DEVARCHR	2963	ETH_MACHT0R	2818
DWT_DEVTYPER	2963	ETH_MACHT1R	2819
DWT_EXCCNTR	2956	ETH_MACHWF0R	2839
DWT_FOLDCNTR	2958	ETH_MACHWF1R	2841
DWT_FUNCTR0	2959	ETH_MACHWF2R	2844
DWT_FUNCTR1	2960	ETH_MACHWF3R	2846
DWT_FUNCTR2	2961	ETH_MACIER	2830
DWT_FUNCTR3	2962	ETH_MACISR	2828
DWT_LSUCNTR	2957	ETH_MACIVIR	2824
DWT_PCSR	2958	ETH_MACL3A00R	2866
DWT_PIDR0	2964	ETH_MACL3A01R	2871
DWT_PIDR1	2965	ETH_MACL3A10R	2866
DWT_PIDR2	2965	ETH_MACL3A11R	2871
DWT_PIDR3	2966	ETH_MACL3A20R	2867
DWT_PIDR4	2964	ETH_MACL3A21R	2872
DWT_SLPCNTR	2957	ETH_MACL3A30R	2867
		ETH_MACL3A31R	2872
		ETH_MACL3L4C0R	2863
		ETH_MACL3L4C1R	2868
		ETH_MACL4A0R	2865
		ETH_MACL4A1R	2870
		ETH_MACLCSR	2834
		ETH_MACLETR	2837
		ETH_MACLMIR	2892
		ETH_MACLTCR	2836
		ETH_MACMDIOAR	2847
		ETH_MACMDIODR	2849
		ETH_MACPCSR	2832
		ETH_MACPFR	2815
		ETH_MACPOCR	2889
		ETH_MACPPSCR	2885, 2886
		ETH_MACPPSIR	2888
		ETH_MACPPSTTNR	2888
		ETH_MACPPSTTSR	2887
		ETH_MACPPSWR	2889
		ETH_MACQTXFCR	2825
		ETH_MACRWKPFR	2834
		ETH_MACRXFCR	2827
E			
ETH_DMACCARXBR	2791		
ETH_DMACCARXDR	2790		
ETH_DMACCATXBR	2791		
ETH_DMACCATXDR	2790		
ETH_DMACCR	2778		
ETH_DMACIER	2786		
ETH_DMACMFCR	2794		
ETH_DMACRXCR	2781		
ETH_DMACRXDLAR	2784		
ETH_DMACRXDTPR	2785		
ETH_DMACRXIWTR	2789		
ETH_DMACRXRLR	2786		
ETH_DMACSR	2792		
ETH_DMACTXCR	2779		
ETH_DMACTXDLAR	2783		
ETH_DMACTXDTPR	2784		
ETH_DMACTXRLR	2785		
ETH_DMADSR	2777		
ETH_DMAISR	2776		

ETH_MACRXTXSR	2831	ETM_CIDR0	3014
ETH_MACSPI0R	2890	ETM_CIDR1	3015
ETH_MACSPI1R	2891	ETM_CIDR2	3015
ETH_MACSPI2R	2891	ETM_CIDR3	3016
ETH_MACSSIR	2875	ETM_CLAIMCLR	3010
ETH_MACSTNR	2876	ETM_CLAIMSETR	3009
ETH_MACSTNUR	2877	ETM_CNTRL DVR0	2997
ETH_MACSTSR	2876	ETM_CONFIGR	2991
ETH_MACSTSUR	2877	ETM_DEVARCHR	3011
ETH_MACTSAR	2878	ETM_DEVTYPER	3011
ETH_MACTSCR	2873	ETM_EVENTCTL0R	2992
ETH_MACTSEACR	2884	ETM_EVENTCTL1R	2993
ETH_MACTSECNR	2885	ETM_IDR0	3000
ETH_MACTSIACR	2883	ETM_IDR1	3001
ETH_MACTSICNR	2884	ETM_IDR10	2998
ETH_MACTSSR	2879	ETM_IDR11	2998
ETH_MACTXTSSNR	2880	ETM_IDR12	2999
ETH_MACTXTSSSR	2881	ETM_IDR13	2999
ETH_MACVHTR	2822	ETM_IDR2	3001
ETH_MACVIR	2823	ETM_IDR3	3002
ETH_MACVR	2838	ETM_IDR4	3003
ETH_MACVTR	2820	ETM_IDR5	3004
ETH_MACWTR	2817	ETM_IDR8	2997
ETH_MMC_CONTROL	2853	ETM_IDR9	2998
ETH_MMC_RX_INTERRUPT	2854	ETM_IMSPECR0	2999
ETH_MMC_RX_INTERRUPT_MASK	2856	ETM_PDCR	3008
ETH_MMC_TX_INTERRUPT	2855	ETM_PDSR	3009
ETH_MMC_TX_INTERRUPT_MASK	2857	ETM_PIDR0	3012
ETH_MTLISR	2799	ETM_PIDR1	3013
ETH_MTLQICSR	2802	ETM_PIDR2	3013
ETH_MTLRXQDR	2806	ETM_PIDR3	3014
ETH_MTLRXQMPOCR	2805	ETM_PIDR4	3012
ETH_MTLRXQOMR	2803	ETM_PRGCTLR	2990
ETH_MTLTXQDR	2801	ETM_RSCTLR2	3005
ETH_MTLTXQOMR	2799	ETM_RSCTLR3	3006
ETH_MTLTXQUR	2801	ETM_SSCCR0	3006
ETH_RX_ALIGNMENT_ERROR_		ETM_SSCSR0	3007
PACKETS	2860	ETM_SSPCICR0	3008
ETH_RX_CRC_ERROR_PACKETS	2859	ETM_STALLCTLR	2994
ETH_RX_LPI_TRAN_CNTR	2862	ETM_STATR	2991
ETH_RX_LPI_USEC_CNTR	2862	ETM_SYNCPR	2995
ETH_RX_UNICAST_PACKETS_GOOD	2860	ETM_TRACEIDR	2996
ETH_TX_LPI_TRAN_CNTR	2861	ETM_VICTLR	2996
ETH_TX_LPI_USEC_CNTR	2861	EXTI_EMR1	732
ETH_TX_MULTIPLE_COLLISION_		EXTI_EMR2	734
GOOD_PACKETS	2858	EXTI_EXTICR1	723
ETH_TX_PACKET_COUNT_GOOD	2859	EXTI_EXTICR2	725
ETH_TX_SINGLE_COLLISION_GOOD_		EXTI_EXTICR3	727
PACKETS	2858	EXTI_EXTICR4	729
ETM_AUTHSTATR	3010	EXTI_FPR1	714
ETM_CCCTLR	2995	EXTI_FPR2	720
		EXTI_FTSR1	712

EXTI_FTSR2	716
EXTI_IMR1	732
EXTI_IMR2	733
EXTI_LOCKR	731
EXTI_PRIVCFGR1	715
EXTI_PRIVCFGR2	722
EXTI_RPR1	713
EXTI_RPR2	719
EXTI_RTSR1	711
EXTI_RTSR2	715
EXTI_SECCFGR1	714
EXTI_SECCFGR2	721
EXTI_SWIER1	712
EXTI_SWIER2	717

F

FDCAN_CCCR	2517
FDCAN_CKDIV	2544
FDCAN_CREL	2514
FDCAN_DBTP	2515
FDCAN_ECR	2523
FDCAN_ENDN	2514
FDCAN_HPMS	2534
FDCAN_IE	2529
FDCAN_ILE	2532
FDCAN_ILS	2531
FDCAN_IR	2526
FDCAN_NBTP	2520
FDCAN_PSR	2524
FDCAN_RWD	2517
FDCAN_RXF0A	2536
FDCAN_RXF0S	2535
FDCAN_RXF1A	2537
FDCAN_RXF1S	2536
FDCAN_RXGFC	2532
FDCAN_TDCR	2526
FDCAN_TEST	2516
FDCAN_TOCC	2522
FDCAN_TOCV	2523
FDCAN_TSCC	2521
FDCAN_TSCV	2521
FDCAN_TXBAR	2539
FDCAN_TXBC	2537
FDCAN_TXBCF	2541
FDCAN_TXBCIE	2542
FDCAN_TXBCR	2540
FDCAN_TXBRP	2538
FDCAN_TXBTIE	2541
FDCAN_TXBTO	2540
FDCAN_TXEFA	2543
FDCAN_TXEFS	2542
FDCAN_TXFQS	2538
FDCAN_XIDAM	2534
Flash_ACR	303
FLASH_BOOTR_PRG	333
FLASH_ECCCORR	340
FLASH_ECCDETR	341
FLASH_ECCDR	342
FLASH_EDATA1R_CUR	337
FLASH_EDATA1R_PRG	338
FLASH_EDATA2R_CUR	345
FLASH_EDATA2R_PRG	346
FLASH_HDP1R_CUR	339
FLASH_HDP1R_PRG	339
FLASH_HDP2R_CUR	347
FLASH_HDP2R_PRG	347
FLASH_HDPEXTR	324
FLASH_NSBOOTR_CUR	331
FLASH_NSBOOTR_PRG	332
FLASH_NSCCR	319
Flash_NSCR	313
FLASH_NSEPOCHR_CUR	328
Flash_NSKEYR	304
FLASH_NSOKKCFGR	321
FLASH_NSOKKKEYR	305
Flash_NSSR	308
FLASH_OPSR	306
Flash_OPTCR	307
Flash_OPTKEYR	305
Flash_OPTSR_CUR	324
Flash_OPTSR_PRG	326
FLASH_OPTSR2_CUR	329
FLASH_OPTSR2_PRG	330
FLASH_OTPBLR_CUR	333
FLASH_OTPBLR_PRG	334
FLASH_PRIVBB1Rx	335
FLASH_PRIVBB2Rx	343
Flash_PRIVCFGR	321
FLASH_SECBB1Rx	334
FLASH_SECBB2Rx	342
FLASH_SECBOOTR_CUR	332
FLASH_SECCCR	319
Flash_SECCR	316
FLASH_SECEPOCHR_CUR	328
Flash_SECKEYR	304
FLASH_SECOBKCFGR	323
FLASH_SECOBKKEYR	306
Flash_SECSR	311
FLASH_SECWM1R_CUR	335
FLASH_SECWM1R_PRG	336
FLASH_SECWM2R_CUR	343
FLASH_SECWM2R_PRG	344
FLASH_WRP1R_CUR	336

FLASH_WRP1R_PRG	337
FLASH_WRP2R_CUR	344
FLASH_WRP2R_PRG	345
FMAC_CR	785
FMAC_PARAM	784
FMAC_RDATA	788
FMAC_SR	786
FMAC_WDATA	788
FMAC_X1BUFCFG	782
FMAC_X2BUFCFG	782
FMAC_YBUFCFG	783
FMC_BCRx	827
FMC_BTRx	830
FMC_BWTRx	833
FMC_ECCR	845
FMC_PATT	844
FMC_PCR	841
FMC_PCSCNTR	834
FMC_PMEM	843
FMC_SDCMR	859
FMC_SDCR1, 2	856
FMC_SDRTR	860
FMC_SDSR	862
FMC_SDTR1, 2	857
FMC_SR	842

G

GPDMA_CxBR1	683, 684
GPDMA_CxBR2	690
GPDMA_CxCR	673
GPDMA_CxDAR	688
GPDMA_CxFCR	671
GPDMA_CxLBAR	670
GPDMA_CxLLR	691, 692
GPDMA_CxSAR	687
GPDMA_CxSR	672
GPDMA_CxTR1	676
GPDMA_CxTR2	679
GPDMA_CxTR3	689
GPDMA_MISR	668
GPDMA_PRIVCFGR	667
GPDMA_RCFGLOCKR	667
GPDMA_SECCFGR	666
GPDMA_SMISR	669
GPIOx_AFRH	574
GPIOx_AFRL	573
GPIOx_BRR	575
GPIOx_BSRR	571
GPIOx_HSLVR	575
GPIOx_IDR	570
GPIOx_LCKR	572

GPIOx_MODER	568
GPIOx_ODR	571
GPIOx_OSPEEDR	569
GPIOx_OTYPER	569
GPIOx_PUPDR	570
GPIOx_SECCFGR	576
GTZC1_MPCBBz_CFGLOCK1	219
GTZC1_MPCBBz_CR	218
GTZC1_MPCBBz_PRIVCFGRx	220
GTZC1_MPCBBz_SECCFGRx	219
GTZC1_TZIC_FCR1	207
GTZC1_TZIC_FCR2	209
GTZC1_TZIC_FCR3	211
GTZC1_TZIC_FCR4	213
GTZC1_TZIC_IER1	190
GTZC1_TZIC_IER2	192
GTZC1_TZIC_IER3	194
GTZC1_TZIC_IER4	196
GTZC1_TZIC_SR1	198
GTZC1_TZIC_SR2	201
GTZC1_TZIC_SR3	203
GTZC1_TZIC_SR4	205
GTZC1_TZSC_CR	171
GTZC1_TZSC_MPCWMxAR	186
GTZC1_TZSC_MPCWMxBR	187
GTZC1_TZSC_MPCWMxzCFGR	185
GTZC1_TZSC_PRIVCFGR1	178
GTZC1_TZSC_PRIVCFGR2	181
GTZC1_TZSC_PRIVCFGR3	183
GTZC1_TZSC_SECCFGR1	171
GTZC1_TZSC_SECCFGR2	174
GTZC1_TZSC_SECCFGR3	176

H

HASH_CR	1345
HASH_CSRx	1353
HASH_DIN	1347
HASH_HRAx	1349
HASH_HRx	1350
HASH_IMR	1351
HASH_SR	1351
HASH_STR	1348

I

I2C_CR1	2075
I2C_CR2	2078
I2C_ICR	2086
I2C_ISR	2084
I2C_OAR1	2080
I2C_OAR2	2081
I2C_PECR	2087

I2C_RXDR	2088
I2C_TIMEOUTR	2083
I2C_TIMINGR	2082
I2C_TXDR	2088
I3C_BCR	2189
I3C_CEVr	2177
I3C_CFGR	2156
I3C_CR	2152, 2154
I3C_CRCAPR	2192
I3C_DCR	2190
I3C_DEVR0	2179
I3C_DEVRx	2181
I3C_EPIDR	2195
I3C_EVR	2171
I3C_GETCAPR	2191
I3C_GETMXDSR	2193
I3C_IBIDR	2165
I3C_IER	2175
I3C_MAXRLR	2183
I3C_MAXWLR	2184
I3C_RDR	2161
I3C_RDWR	2161
I3C_RMR	2170
I3C_SER	2168
I3C_SR	2167
I3C_TDR	2162
I3C_TDWR	2163
I3C_TGTTDR	2166
I3C_TIMINGR0	2185
I3C_TIMINGR1	2186
I3C_TIMINGR2	2188
ICACHE_CR	364
ICACHE_CRRx	368
ICACHE_FCR	366
ICACHE_HMONR	367
ICACHE_IER	366
ICACHE_MMONR	367
ICACHE_SR	365
ITM_CIDR0	2979
ITM_CIDR1	2979
ITM_CIDR2	2980
ITM_CIDR3	2980
ITM_DEVARCHR	2975
ITM_DEVTYPER	2976
ITM_PIDR0	2977
ITM_PIDR1	2977
ITM_PIDR2	2978
ITM_PIDR3	2978
ITM_PIDR4	2976
ITM_STIMRx	2972
ITM_TCR	2974
ITM_TER	2973

ITM_TPR	2973
IWDG_EWCR	1913
IWDG_KR	1909
IWDG_PR	1910
IWDG_RLR	1911
IWDG_SR	1911
IWDG_WINR	1913

L

LPTIM_ARR	1891
LPTIM_CCMR1	1893
LPTIM_CCR1	1890
LPTIM_CCR2	1896
LPTIM_CFGR	1886
LPTIM_CFGR2	1892
LPTIM_CNT	1891
LPTIM_CR	1889
LPTIM_RCR	1893
LPTIM4_DIER	1881
LPTIM4_ICR	1877
LPTIM4_ISR	1872
LPTIMx_DIER	1882, 1884
LPTIMx_ICR	1878, 1880
LPTIMx_ISR	1873, 1875
LPUART_BRR	2326
LPUART_CR1	2315, 2318
LPUART_CR2	2321
LPUART_CR3	2323
LPUART_ICR	2335
LPUART_ISR	2327, 2331
LPUART_PRESC	2337
LPUART_RDR	2336
LPUART_RQR	2326
LPUART_TDR	2336

M

MCUROM_CIDR0	2944
MCUROM_CIDR1	2944
MCUROM_CIDR2	2945
MCUROM_CIDR3	2945
MCUROM_MEMTYPER	2941
MCUROM_PIDR0	2942
MCUROM_PIDR1	2942
MCUROM_PIDR2	2943
MCUROM_PIDR3	2943
MCUROM_PIDR4	2941

O

OCTOSPI_ABR	907
OCTOSPI_AR	901

OCTOSPI_CCR	904
OCTOSPI_CR	893
OCTOSPI_DCR1	896
OCTOSPI_DCR2	897
OCTOSPI_DCR3	898
OCTOSPI_DCR4	899
OCTOSPI_DLR	901
OCTOSPI_DR	902
OCTOSPI_FCR	900
OCTOSPI_HLCR	915
OCTOSPI_IR	907
OCTOSPI_LPTR	908
OCTOSPI_PIR	904
OCTOSPI_PSMAR	903
OCTOSPI_PSMKR	903
OCTOSPI_SR	899
OCTOSPI_TCR	906
OCTOSPI_WABR	915
OCTOSPI_WCCR	912
OCTOSPI_WIR	914
OCTOSPI_WPABR	911
OCTOSPI_WPCCR	908
OCTOSPI_WPIR	911
OCTOSPI_WPTCR	910
OCTOSPI_WTCR	914
OTFDEC_CR	1395
OTFDEC_ICR	1403
OTFDEC_IER	1404
OTFDEC_ISR	1402
OTFDEC_PRIVCFGR	1396
OTFDEC_RxCFGR	1396
OTFDEC_RxENDADDR	1399
OTFDEC_RxKEYR0	1400
OTFDEC_RxKEYR1	1401
OTFDEC_RxKEYR2	1401
OTFDEC_RxKEYR3	1402
OTFDEC_RxNONCER0	1399
OTFDEC_RxNONCER1	1400
OTFDEC_RxSTARTADDR	1398

P

PKA_CLRFR	1383
PKA_CR	1380
PKA_SR	1382
PSSI_CR	1210
PSSI_DR	1215
PSSI_ICR	1214
PSSI_IER	1213
PSSI_MIS	1213
PSSI_RIS	1212
PSSI_SR	1212

PWR_BDCR	424
PWR_BDSR	425
PWR_DBPCR	425
PWR_IORETR	432
PWR_PMCR	420
PWR_PMSR	422
PWR_PRIVCFGR	434
PWR_SCCR	427
PWR_SECCFGR	433
PWR_UCPDR	426
PWR_USBSCR	429
PWR_VMCR	428
PWR_VMSR	429
PWR_VOSCR	422
PWR_VOSSR	423
PWR_WUCR	431
PWR_WUSCR	430
PWR_WUSR	431

R

RAMCFG_M2WPR1	228
RAMCFG_M2WPR2	229
RAMCFG_MxCR	225
RAMCFG_MxDEAR	227
RAMCFG_MxECCKEYR	229
RAMCFG_MxERKEYR	230
RAMCFG_MxICR	228
RAMCFG_MxIER	226
RAMCFG_MxISR	226
RAMCFG_MxSEAR	227
RCC_AHB1ENR	502
RCC_AHB1LPENR	514
RCC_AHB1RSTR	491
RCC_AHB2ENR	504
RCC_AHB2LPENR	516
RCC_AHB2RSTR	492
RCC_AHB4ENR	506
RCC_AHB4LPENR	518
RCC_AHB4RSTR	494
RCC_APB1HENR	510
RCC_APB1HLPENR	522
RCC_APB1HRSTR	498
RCC_APB1LENR	507
RCC_APB1LLPENR	519
RCC_APB1LRSTR	495
RCC_APB2ENR	511
RCC_APB2LPENR	523
RCC_APB2RSTR	499
RCC_APB3ENR	512
RCC_APB3LPENR	524
RCC_APB3RSTR	500

RCC_BDCR	535	RTC_TSDR	1964
RCC_CCIPR1	526	RTC_TSSSR	1965
RCC_CCIPR2	528	RTC_TSTR	1963
RCC_CCIPR3	530	RTC_WPR	1960
RCC_CCIPR4	532	RTC_WUTR	1954
RCC_CCIPR5	533		
RCC_CFGR1	471	S	
RCC_CFGR2	474	SAES_CR	1316
RCC_CICR	490	SAES_DINR	1320
RCC_CIER	487	SAES_DOUTR	1321
RCC_CIFR	488	SAES_ICR	1329
RCC_CR	466	SAES_IER	1326
RCC_CRRCR	470	SAES_ISR	1327
RCC_CSICFGR	470	SAES_IVR0	1323
RCC_HSICFGR	469	SAES_IVR1	1323
RCC_PLL1CFGR	476	SAES_IVR2	1323
RCC_PLL1DIVR	481	SAES_IVR3	1324
RCC_PLL1FRACR	482	SAES_KEYR0	1321
RCC_PLL2CFGR	478	SAES_KEYR1	1322
RCC_PLL2DIVR	483	SAES_KEYR2	1322
RCC_PLL2FRACR	484	SAES_KEYR3	1322
RCC_PLL3CFGR	479	SAES_KEYR4	1324
RCC_PLL3DIVR	485	SAES_KEYR5	1324
RCC_PLL3FRACR	486	SAES_KEYR6	1325
RCC_PRIVCFGR	541	SAES_KEYR7	1325
RCC_RSR	537	SAES_SR	1319
RCC_SECCFGR	539	SAES_SUSPxR	1326
RNG_CR	1228	SAI_ACLRFR	2476
RNG_DR	1231	SAI_ACR1	2455
RNG_HTCR	1231	SAI_ACR2	2461
RNG_SR	1230	SAI_ADR	2478
RTC_ALRABINR	1974	SAI_AFRCCR	2465
RTC_ALRBBINR	1975	SAI_AIM	2469
RTC_ALRMAR	1965	SAI_ASLOTR	2467
RTC_ALRMASSR	1967	SAI_ASR	2472
RTC_ALRMBR	1968	SAI_BCLRFR	2477
RTC_ALRMBSSR	1969	SAI_BCR1	2458
RTC_CALR	1961	SAI_BCR2	2463
RTC_CR	1955	SAI_BDR	2479
RTC_DR	1949	SAI_BFRCCR	2466
RTC_ICSR	1951	SAI_BIM	2471
RTC_MISR	1971	SAI_BSLOTR	2468
RTC_OR	1974	SAI_BSR	2474
RTC_PRER	1953	SAI_GCR	2454
RTC_PRIVCFGR	1958	SAI_PDMCR	2479
RTC_SCR	1973	SAI_PDMDLY	2480
RTC_SECCFGR	1959	SBS_CCCSR	597
RTC_SHIFTR	1962	SBS_CCSWCR	599
RTC_SMISR	1972	SBS_CCVALR	598
RTC_SR	1970	SBS_CFGR2	600
RTC_SSR	1950	SBS_CNSLCKR	601
RTC_TR	1948		

SBS_CSLCKR	602	SYSROM_PIDR0	2936
SBS_DBGCR	592	SYSROM_PIDR1	2936
SBS_DBGLOCKR	592	SYSROM_PIDR2	2937
SBS_ECCNMIR	603	SYSROM_PIDR3	2937
SBS_EPOCHSELCR	593	SYSROM_PIDR4	2935
SBS_FPUIMR	596		
SBS_HDPLCR	590	T	
SBS_HDPLSR	591	TAMP_ATCR1	2001
SBS_MESR	597	TAMP_ATCR2	2005
SBS_NEXTHDPLCR	591	TAMP_ATOR	2004
SBS_PMCRR	595	TAMP_ATSEEDR	2003
SBS_RSSCMDR	593	TAMP_BKPxR	2022
SBS_SECCFGR	594	TAMP_COUNT1R	2020
SDMMC_ACKTIMER	992	TAMP_CR1	1993
SDMMC_ARGR	977	TAMP_CR2	1995
SDMMC_CLKCR	975	TAMP_CR3	1998
SDMMC_CMDR	977	TAMP_FLTCR	1999
SDMMC_DCNTR	984	TAMP_IER	2011
SDMMC_DCTRL	981	TAMP_MISR	2015
SDMMC_DLENR	981	TAMP_OR	2020
SDMMC_DTIMER	980	TAMP_PRIVCFGR	2010
SDMMC_FIFORx	992	TAMP_RPCFGR	2021
SDMMC_ICR	987	TAMP_SCR	2018
SDMMC_IDMABAR	996	TAMP_SECCFGR	2008
SDMMC_IDMABASER	994	TAMP_SMISR	2016
SDMMC_IDMABSIZER	994	TAMP_SR	2013
SDMMC_IDMACTRLR	993	TIM12_ARR	1727
SDMMC_IDMALAR	995	TIM12_CCER	1724
SDMMC_MASKR	989	TIM12_CCMR1	1721, 1722
SDMMC_POWER	974	TIM12_CCR1	1728
SDMMC_RESPCMDR	979	TIM12_CCR2	1729
SDMMC_RESPxR	979	TIM12_CNT	1726
SDMMC_STAR	984	TIM12_CR1	1715
SPI_CFG1	2401	TIM12_CR2	1716
SPI_CFG2	2404	TIM12_DIER	1718
SPI_CR1	2399	TIM12_EGR	1720
SPI_CR2	2401	TIM12_PSC	1726
SPI_CRCPOLY	2412	TIM12_SMCR	1717
SPI_I2SCFGR	2414	TIM12_SR	1719
SPI_IER	2406	TIM12_TISEL	1730
SPI_IFCR	2410	TIM15_AF1	1818
SPI_RXCRC	2413	TIM15_AF2	1820
SPI_RXDR	2411	TIM15_ARR	1810
SPI_SR	2407	TIM15_BDTR	1813
SPI_TXCRC	2412	TIM15_CCER	1806
SPI_TXDR	2411	TIM15_CCMR1	1802, 1803
SPI_UDRDR	2413	TIM15_CCR1	1811
SYSROM_CIDR0	2938	TIM15_CCR2	1812
SYSROM_CIDR1	2938	TIM15_CNT	1809
SYSROM_CIDR2	2939	TIM15_CR1	1793
SYSROM_CIDR3	2939	TIM15_CR2	1794
SYSROM_MEMTYPER	2935		

TIM15_DCR 1821
 TIM15_DIER 1798
 TIM15_DMAR 1822
 TIM15_DTR2 1816
 TIM15_EGR 1801
 TIM15_PSC 1809
 TIM15_RCR 1810
 TIM15_SMCR 1796
 TIM15_SR 1799
 TIM15_TISEL 1817
 TIMx_AF1 1539, 1656, 1842
 TIMx_AF2 1542, 1657, 1845
 TIMx_ARR
 1524, 1644, 1645, 1680, 1741, 1836
 TIMx_BDTR 1529, 1838
 TIMx_CCER 1519, 1641, 1739, 1833
 TIMx_CCMR1 .. 1511, 1513, 1635, 1637, 1736,
 1738, 1830, 1831
 TIMx_CCMR2 1515, 1516, 1639, 1640
 TIMx_CCMR3 1535
 TIMx_CCR1 1525, 1646, 1647, 1742, 1837
 TIMx_CCR2 1526, 1648, 1649
 TIMx_CCR3 1527, 1650, 1651
 TIMx_CCR4 1528, 1652, 1653
 TIMx_CCR5 1533
 TIMx_CCR6 1534
 TIMx_CNT 1523, 1643, 1679, 1740, 1835
 TIMx_CR1 1497, 1625, 1675, 1733, 1825
 TIMx_CR2 1499, 1627, 1677, 1826
 TIMx_DCR 1545, 1658, 1845
 TIMx_DIER 1506, 1631, 1677, 1734, 1827
 TIMx_DMAR 1546, 1659, 1846
 TIMx_DTR2 1536, 1841
 TIMx_ECR 1537, 1654
 TIMx_EGR 1510, 1634, 1678, 1736, 1829
 TIMx_PSC 1523, 1644, 1679, 1741, 1835
 TIMx_RCR 1524, 1836
 TIMx_SMCR 1502, 1628
 TIMx_SR 1507, 1632, 1678, 1735, 1828
 TIMx_TISEL 1538, 1655, 1743, 1842
 TPIU_ACPR 3022
 TPIU_CIDR0 3030
 TPIU_CIDR1 3030
 TPIU_CIDR2 3031
 TPIU_CIDR3 3031
 TPIU_CLAIMCLR 3026
 TPIU_CLAIMSETR 3025
 TPIU_CSPSR 3021
 TPIU_DEVIDR 3026
 TPIU_DEVTYPER 3027
 TPIU_FFCR 3024
 TPIU_FFSR 3023

TPIU_PIDR0 3028
 TPIU_PIDR1 3028
 TPIU_PIDR2 3029
 TPIU_PIDR3 3029
 TPIU_PIDR4 3027
 TPIU_PSCR 3025
 TPIU_SPPR 3022
 TPIU_SSPSR 3021

U

UCPD_CFGR1 2615
 UCPD_CFGR2 2617
 UCPD_CFGR3 2618
 UCPD_CR 2618
 UCPD_ICR 2625
 UCPD_IMR 2621
 UCPD_RX_ORDEXTR1 2630
 UCPD_RX_ORDEXTR2 2630
 UCPD_RX_ORDSETR 2628
 UCPD_RX_PAYSZR 2629
 UCPD_RXDR 2629
 UCPD_SR 2622
 UCPD_TX_ORDSETR 2626
 UCPD_TX_PAYSZR 2627
 UCPD_TXDR 2627
 USART_BRR 2264
 USART_CR1 2249, 2253
 USART_CR2 2256
 USART_CR3 2260
 USART_GTPR 2265
 USART_ICR 2279
 USART_ISR 2268, 2274
 USART_PRESC 2282
 USART_RDR 2281
 USART_RQR 2267
 USART_RTOR 2266
 USART_TDR 2281
 USB_BCDR 2578
 USB_CHEP_RXTXBD_n 2591
 USB_CHEP_TXRXBD_n 2588, 2590
 USB_CHEPnR 2579
 USB_CNTR 2569
 USB_DADDR 2577
 USB_FNR 2576
 USB_ISTR 2572
 USB_LPMCSR 2577

V

VREFBUF_CCR 1178
 VREFBUF_CSR 1177

W

WWDG_CFR1922

WWDG_CR1921

WWDG_SR1922



重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST製品は、注文請書発行時点で有効なSTの販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してSTは一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

ST およびST ロゴはSTMicroelectronics の商標です。STの登録商標についてはSTウェブサイトをご覧ください。www.st.com/trademarks
その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

この資料は、STMicroelectronics NV 並びにその子会社(以下ST)が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くためにSTマイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST及びSTマイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

© 2023 STMicroelectronics - All rights reserved