



概要

このリファレンスマニュアルは、アプリケーション開発者を対象としています。STM32F412 マイクロコントローラのメモリやペリフェラルを使用する方法について、詳しく説明しています。

STM32F412 は、さまざまなメモリサイズ、パッケージ、およびペリフェラルを持つマイクロコントローラファミリ製品になります。

注文情報、機械的および電気的特性については、データシートを参照してください。

ARM[®] FPU 搭載 Cortex[®]-M4 コアについては、FPU 搭載 Cortex[®]-M4 テクニカルリファレンスマニュアルを参照してください。

関連ドキュメント

STMicroelectronics のウェブサイト (www.st.com) では、以下のドキュメントが入手可能です。

- STM32F412xE/xG データシート
- PM0214 『STM32F3 および STM32F4 シリーズ FPU 搭載 Cortex[®]-M4-M4 プログラミングマニュアル』(ARM[®] FPU 搭載 Cortex[®]-M4 について)

目次

1	このマニュアルにおける表記の規則	45
1.1	レジスタに関する略記	45
1.2	用語	45
1.3	使用可能なペリフェラル	45
2	システムおよびメモリの概要	46
2.1	システムアーキテクチャ	46
2.1.1	I-バス	47
2.1.2	D-バス	47
2.1.3	S-バス	47
2.1.4	DMA メモリバス	47
2.1.5	DMA ペリフェラルバス	47
2.1.6	バスマトリックス	47
2.1.7	AHB-APB ブリッジ (APB)	47
2.2	メモリ構成	48
2.2.1	概要	48
2.2.2	メモリマップとレジスタ境界アドレス	49
2.3	内蔵 SRAM	52
2.4	フラッシュメモリの概要	52
2.5	ビットバンディング	52
2.6	ブート設定	53
3	内蔵フラッシュメモリインタフェース	56
3.1	概要	56
3.2	主な特長	56
3.3	内蔵フラッシュメモリ	57
3.4	読み出しインタフェース	58
3.4.1	CPU クロック周波数とフラッシュメモリ読み出し時間との関係	58
3.4.2	適応型リアルタイムメモリアクセラレータ (ART Accelerator™)	59
3.5	消去操作とプログラム操作	61
3.5.1	フラッシュ制御レジスタのアンロック	61
3.5.2	プログラム／消去の並列処理	62
3.5.3	消去	62

3.5.4	プログラミング	63
3.5.5	割り込み	64
3.6	オプションバイト	64
3.6.1	ユーザオプションバイトの説明	64
3.6.2	ユーザオプションバイトのプログラミング	66
3.6.3	読み出し保護 (RDP)	66
3.6.4	書き込み保護	69
3.6.5	独自仕様コード読み出し保護 (PCROP)	69
3.7	OTP (One-time programmable) バイト	71
3.8	フラッシュインタフェースレジスタ	72
3.8.1	フラッシュアクセス制御レジスタ (FLASH_ACR)	72
3.8.2	フラッシュキーレジスタ (FLASH_KEYR)	73
3.8.3	フラッシュオプションキーレジスタ (FLASH_OPTKEYR)	73
3.8.4	フラッシュステータスレジスタ (FLASH_SR)	74
3.8.5	フラッシュ制御レジスタ (FLASH_CR)	75
3.8.6	フラッシュオプション制御レジスタ (FLASH_OPTCR)	76
3.8.7	フラッシュインタフェースレジスタマップ	79
4	CRC 計算ユニット	80
4.1	CRC の概要	80
4.2	CRC の主な機能	80
4.3	CRC の機能説明	81
4.4	CRC レジスタ	81
4.4.1	データレジスタ (CRC_DR)	81
4.4.2	独立型データレジスタ (CRC_IDR)	82
4.4.3	制御レジスタ (CRC_CR)	82
4.4.4	CRC レジスタマップ	83
5	電源コントローラ (PWR)	84
5.1	電源	84
5.1.1	独立した A/D コンバータ用電源と基準電圧	85
5.1.2	バッテリーバックアップドメイン	86
5.1.3	電圧レギュレータ	87
5.2	電源供給スーパバイザ	88
5.2.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	88
5.2.2	ブラウンアウトリセット (BOR)	88

5.2.3	プログラム可能な電圧検出器 (PVD)	89
5.3	低電力モード	90
5.3.1	システムクロックの低速化	92
5.3.2	ペリフェラルクロックゲーティング	92
5.3.3	SLEEP モード	93
5.3.4	Batch Acquisition Mode	94
5.3.5	STOP モード	95
5.3.6	STANDBY モード	98
5.3.7	デバイスをウェイクアップさせるための RTC 代替機能のプログラミング (STOP および STANDBY モードから)	99
5.4	電源制御レジスタ	102
5.4.1	PWR 電源制御レジスタ (PWR_CR)	102
5.4.2	PWR 電源制御/ステータスレジスタ (PWR_PWR)	104
5.5	PWR レジスタマップ	106
6	STM32F412xx のリセットおよびクロック制御 (RCC)	107
6.1	リセット	107
6.1.1	システムリセット	107
6.1.2	電源リセット	108
6.1.3	バックアップドメインリセット	109
6.2	クロック	109
6.2.1	HSE クロック	111
6.2.2	HSI クロック	112
6.2.3	PLL の設定	113
6.2.4	LSE クロック	113
6.2.5	LSI クロック	114
6.2.6	システムクロック (SYSCLK) の選択	114
6.2.7	クロックセキュリティシステム (CSS)	114
6.2.8	RTC/AWU クロック	115
6.2.9	ウォッチドッグクロック	116
6.2.10	クロック信号出力	116
6.2.11	TIM5/TIM11 を使用した内部/外部クロックの測定	116
6.3	RCC レジスタ	119
6.3.1	RCC クロック制御レジスタ (RCC_CR)	119
6.3.2	RCC PLL 設定レジスタ (RCC_PLLCFGR)	121
6.3.3	RCC クロック設定レジスタ (RCC_CFGR)	123
6.3.4	RCC クロック割り込みレジスタ (RCC_CIR)	126

6.3.5	RCC AHB1 ペリフェラルリセットレジスタ (RCC_AHB1RSTR)	128
6.3.6	RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR)	130
6.3.7	RCC AHB3 ペリフェラルリセットレジスタ (RCC_AHB3RSTR)	131
6.3.8	RCC APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR)	131
6.3.9	RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)	134
6.3.10	RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1ENR)	136
6.3.11	RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR)	137
6.3.12	RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3ENR)	138
6.3.13	RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR)	138
6.3.14	RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR)	141
6.3.15	低電力モードにおける RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1LPENR)	143
6.3.16	低電力モードにおける RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2LPENR)	144
6.3.17	低電力モードにおける RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3LPENR)	145
6.3.18	低電力モードにおける RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1LPENR)	146
6.3.19	低電力モードにおける RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2LPENR)	149
6.3.20	RCC バックアップドメイン制御レジスタ (RCC_BDCR)	151
6.3.21	RCC クロック制御およびステータスレジスタ (RCC_CSR)	152
6.3.22	RCC スペクトル拡散クロック生成レジスタ (RCC_SSCGR)	154
6.3.23	RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR)	155
6.3.24	RCC 専用クロック設定レジスタ (RCC_DCKCFGR)	157
6.3.25	RCC クロックゲート有効レジスタ (CKGATENR)	158
6.3.26	RCC 専用クロック設定レジスタ (RCC_DCKCFGR2)	159
6.3.27	RCC レジスタマップ	160

7	汎用 I/O (GPIO)	163
7.1	GPIO の概要	163
7.2	GPIO の主な機能	163
7.3	GPIO の機能説明	163
7.3.1	汎用 I/O (GPIO)	165
7.3.2	I/O ピンマルチプレクサとマッピング	165
7.3.3	I/O ポート制御レジスタ	168
7.3.4	I/O ポートデータレジスタ	168
7.3.5	I/O データのビット単位の操作	168

7.3.6	GPIO ロック機構	168
7.3.7	I/O オルタネート機能の入力/出力	169
7.3.8	外部割り込み/ウェイクアップライン	169
7.3.9	入力設定	169
7.3.10	出力設定	170
7.3.11	オルタネート機能設定	171
7.3.12	アナログ設定	172
7.3.13	OSC32_IN/OSC32_OUT ピンの GPIO PC14/PC15 ポートピンとしての使用	172
7.3.14	OSC_IN/OSC_OUT ピンの GPIO PH0/PH1 ポートピンとしての使用	172
7.3.15	RTC 追加機能の選択	173
7.4	GPIO レジスタ	174
7.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ H)	174
7.4.2	GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ H)	174
7.4.3	GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A ~ H)	175
7.4.4	GPIO ポートプルアップ/プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ H)	175
7.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ H)	176
7.4.6	GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ H)	176
7.4.7	GPIO ポートビットセット/リセットレジスタ (GPIOx_BSRR) (x = A ~ H)	176
7.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ H)	177
7.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFR1) (x = A ~ H)	178
7.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFR2) (x = A ~ H)	179
7.4.11	GPIO レジスタマップ	179
8	システム設定コントローラ (SYSCFG)	182
8.1	I/O 補正セル	182
8.2	SYSCFG レジスタ	182
8.2.1	SYSCFG メモリ再配置レジスタ (SYSCFG_MEMRMP)	182
8.2.2	SYSCFG ペリフェラルモード設定レジスタ (SYSCFG_PMC)	183
8.2.3	SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)	184
8.2.4	SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)	184
8.2.5	SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3)	185
8.2.6	SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)	185
8.2.7	SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2)	186
8.2.8	補正セル制御レジスタ (SYSCFG_CMPCR)	186
8.2.9	SYSCFG 設定レジスタ (SYSCFG_CFGR)	187

8.2.10	SYSCFG レジスタマップ	188
9	ダイレクトメモリアクセスコントローラ (DMA)	189
9.1	DMA の概要	189
9.2	DMA の主な機能	189
9.3	DMA の機能説明	191
9.3.1	概要	191
9.3.2	DMA トランザクション	192
9.3.3	チャンネル選択	192
9.3.4	アービタ	193
9.3.5	DMA ストリーム	194
9.3.6	転送元、転送先、および転送モード	194
9.3.7	ポインタのインクリメント	197
9.3.8	サーキュラモード	198
9.3.9	ダブルバッファモード	198
9.3.10	プログラム可能なデータ幅、パッキング／アンパッキング、 エンディアン形式	199
9.3.11	シングル転送とバースト転送	201
9.3.12	FIFO	201
9.3.13	DMA 転送の完了	204
9.3.14	DMA 転送の中断	205
9.3.15	フローコントローラ	205
9.3.16	実現可能な DMA 設定の概要	206
9.3.17	ストリーム設定手順	207
9.3.18	エラー管理	208
9.4	DMA 割り込み	209
9.5	DMA レジスタ	209
9.5.1	DMA ロー割り込みステータスレジスタ (DMA_LISR)	209
9.5.2	DMA ハイ割り込みステータスレジスタ (DMA_HISR)	210
9.5.3	DMA ロー割り込みフラグクリアレジスタ (DMA_LIFCR)	211
9.5.4	DMA ハイ割り込みフラグクリアレジスタ (DMA_HIFCR)	212
9.5.5	DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0 ~ 7)	213
9.5.6	DMA ストリーム x データ数レジスタ (DMA_HIFCR) (x=0 ~ 7)	216
9.5.7	DMA ストリーム x ペリフェラルアドレスレジスタ (DMA_SxPAR) (x=0 ~ 7)	217
9.5.8	DMA ストリーム x メモリ 0 アドレスレジスタ (DMA_SxM0AR) (x=0 ~ 7)	217

9.5.9	DMA ストリーム x メモリ 1 アドレスレジスタ (DMA_SxM1AR) (x=0 ~ 7)	217
9.5.10	DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0 ~ 7)	218
9.5.11	DMA レジスタマップ	220
10	割り込みとイベント	224
10.1	ネスト化されたベクタ割り込みコントローラ (NVIC)	224
10.1.1	NVIC の機能	224
10.1.2	SysTick 較正值レジスタ	224
10.1.3	割り込みベクタと例外ベクタ	224
10.2	外部割り込み/イベントコントローラ (EXTI)	224
10.2.1	EXTI の主な機能	228
10.2.2	EXTI ブロック図	228
10.2.3	ウェイクアップイベント管理	229
10.2.4	機能詳細	229
10.2.5	外部割り込み/イベントラインの配置	230
10.3	EXTI レジスタ	231
10.3.1	割り込みマスクレジスタ (EXTI_IMR)	231
10.3.2	イベントマスクレジスタ (EXTI_EMR)	231
10.3.3	立ち上がりトリガ選択レジスタ (EXTI_RTSTR)	232
10.3.4	立ち下がりトリガ選択レジスタ (EXTI_FTSTR)	233
10.3.5	ソフトウェア割り込みイベントレジスタ (EXTI_SWIER)	234
10.3.6	ペンディングレジスタ (EXTI_PR)	235
10.3.7	EXTI レジスタマップ	236
11	フレキシブルスタティックメモリコントローラ (FSMC)	237
11.1	FSMC の主な特長	237
11.2	ブロック図	238
11.3	AHB インタフェース	239
11.3.1	サポートされるメモリおよびトランザクション	239
11.4	外部デバイスアドレスマッピング	240
11.4.1	NOR/PSRAM アドレスマッピング	240
11.5	NOR 型フラッシュ/PSRAM コントローラ	241
11.5.1	外部メモリインタフェース信号	242
11.5.2	サポートされるメモリおよびトランザクション	244
11.5.3	一般的なタイミング規則	245
11.5.4	NOR フラッシュ /PSRAM コントローラ非同期トランザクション	246

11.5.5	同期トランザクション	263
11.5.6	NOR/PSRAM コントローラレジスタ	269
11.6	FSMC レジスタマップ	277
12	Quad SPI インタフェース (QUADSPI)	279
12.1	概要	279
12.2	QUADSPI の主な機能	279
12.3	QUADSPI の機能説明	279
12.3.1	QUADSPI ブロック図	279
12.3.2	QUADSPI コマンドシーケンス	280
12.3.3	QUADSPI シングルインタフェースプロトコルモード	283
12.3.4	QUADSPI インダイレクトモード	285
12.3.5	QUADSPI ステータスフラグポーリングモード	287
12.3.6	QUADSPI メモリマップドモード	287
12.3.7	QUADSPI フラッシュメモリの設定	288
12.3.8	QUADSPI 遅延データサンプリング	288
12.3.9	QUADSPI の設定	288
12.3.10	QUADSPI の使用	289
12.3.11	1 回限りの命令の送信	291
12.3.12	QUADSPI エラー管理	291
12.3.13	QUADSPI の BUSY ビットおよびアポート機能	291
12.3.14	nCS の動作	292
12.4	QUADSPI 割り込み	293
12.5	QUADSPI レジスタ	294
12.5.1	QUADSPI 制御レジスタ (QUADSPI_CR)	294
12.5.2	QUADSPI デバイス設定レジスタ (QUADSPI_DCR)	297
12.5.3	QUADSPI ステータスレジスタ (QUADSPI_SR)	298
12.5.4	QUADSPI フラグクリアレジスタ (QUADSPI_FCR)	299
12.5.5	QUADSPI データ長レジスタ (QUADSPI_DLR)	299
12.5.6	QUADSPI 通信設定レジスタ (QUADSPI_CCR)	300
12.5.7	QUADSPI アドレスレジスタ (QUADSPI_AR)	302
12.5.8	QUADSPI オルタネットバイトレジスタ (QUADSPI_ABR)	303
12.5.9	QUADSPI データレジスタ (QUADSPI_DR)	303
12.5.10	QUADSPI ポーリングステータスマスクレジスタ (QUADSPI_PSMKR)	304
12.5.11	QUADSPI ポーリングステータス一致レジスタ (QUADSPI_PSMAR)	304
12.5.12	QUADSPI ポーリングインターバルレジスタ (QUADSPI_PIR)	305

12.5.13	QUADSPI 低電力タイムアウトレジスタ (QUADSPI_LPTR)	305
12.5.14	QUADSPI レジスタマップ	306
13	アナログデジタルコンバータ (ADC)	307
13.1	ADC の概要	307
13.2	ADC の主な機能	307
13.3	ADC の機能詳細	307
13.3.1	ADC のオン/オフ制御	309
13.3.2	ADC クロック	309
13.3.3	チャネル選択	309
13.3.4	シングル変換モード	310
13.3.5	連続変換モード	310
13.3.6	タイミング図	311
13.3.7	アナログウォッチドッグ	311
13.3.8	スキャンモード	312
13.3.9	インジェクトチャネルの管理	312
13.3.10	不連続モード	314
13.4	データの配置	315
13.5	チャネル単位でプログラム可能なサンプリング時間	316
13.6	外部トリガによる変換およびトリガ極性	316
13.7	高速変換モード	318
13.8	データ管理	318
13.8.1	DMA の使用	318
13.8.2	DMA を使用しない変換シーケンスの管理	319
13.8.3	DMA およびオーバーラン検出を使用しない変換	319
13.9	温度センサ	319
13.10	バッテリー充電監視	321
13.11	ADC 割り込み	321
13.12	ADC レジスタ	322
13.12.1	ADC ステータスレジスタ (ADC_SR)	322
13.12.2	ADC 制御レジスタ 1 (ADC_CR1)	324
13.12.3	ADC 制御レジスタ 2 (ADC_CR2)	326
13.12.4	ADC サンプル時間レジスタ 1 (ADC_SMPR1)	329
13.12.5	ADC サンプル時間レジスタ 2 (ADC_SMPR2)	329
13.12.6	ADC インジェクトチャネルデータオフセットレジスタ x (ADC_JOFRx) (x=1..4)	330

13.12.7	ADC ウォッチドッグ高閾値レジスタ (ADC_HTR)	330
13.12.8	ADC ウォッチドッグ低閾値レジスタ (ADC_LTR)	331
13.12.9	ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)	331
13.12.10	ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)	332
13.12.11	ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)	332
13.12.12	ADC インジェクトシーケンスレジスタ (ADC_JSQR)	333
13.12.13	ADC インジェクトデータレジスタ x (ADC_JDRx) (x= 1..4)	333
13.12.14	ADC レギュラデータレジスタ (ADC_DR)	334
13.12.15	ADC 共通ステータスレジスタ (ADC_CSR)	334
13.12.16	ADC 共通制御レジスタ (ADC_CCR)	335
13.12.17	ADC レジスタマップ	336
14	デルタシグマモジュレータのデジタルフィルタ (DFSDM)	338
14.1	概要	338
14.2	DFSDM の主な機能	339
14.3	DFSDM の機能詳細	340
14.3.1	DFSDM ブロック図	340
14.3.2	DFSDM ピンおよび内部信号	341
14.3.3	DFSDM のリセットおよびクロック	342
14.3.4	シリアルチャネルトランシーバ	342
14.3.5	入力シリアルインタフェースの設定	352
14.3.6	並列データ入力	352
14.3.7	チャネル選択	354
14.3.8	デジタルフィルタ設定	354
14.3.9	積分器ユニット	356
14.3.10	アナログウォッチドッグ	356
14.3.11	短絡検出回路	359
14.3.12	極値検出回路	359
14.3.13	データユニットブロック	360
14.3.14	符号付きデータフォーマット	361
14.3.15	変換の開始	361
14.3.16	連続モードおよび高速連続モード	362
14.3.17	リクエスト優先度	362
14.3.18	RUN モードでの省電力	363

14.4	DFSDM 割り込み	363
14.5	DFSDM の DMA 転送	365
14.6	DFSDM チャネル y レジスタ (y=0 ~ 3)	365
14.6.1	DFSDM チャネル設定 y レジスタ (DFSDM_CHyCFGR1) (y=0 ~ 3)	365
14.6.2	DFSDM チャネル設定 y レジスタ (DFSDM_CHyCFGR2) (y=0 ~ 3)	367
14.6.3	DFSDM チャネルアナログウォッチドックおよび短絡検出回路レジスタ (DFSDM_CHyAWSCDR) (y=0 ~ 3)	368
14.6.4	DFSDM チャネルウォッチドックフィルタデータレジスタ (DFSDM_CHyWDATR) (y=0 ~ 3)	369
14.6.5	DFSDM チャネルデータ入力レジスタ (DFSDM_CHyDATINR) (y=0 ~ 3)	370
14.7	DFSDM フィルタ x モジュールレジスタ (x=0 ~ 1)	371
14.7.1	DFSDM 制御レジスタ 1 (DFSDM_FLTxCR1)	371
14.7.2	DFSDM 制御レジスタ 2 (DFSDM_FLTxCR2)	373
14.7.3	DFSDM 割り込みおよびステータスレジスタ (DFSDM_FLTxISR)	375
14.7.4	DFSDM 割り込みフラグクリアレジスタ (DFSDM_FLTxICR)	377
14.7.5	DFSDM インジェクトチャネルグループ選択レジスタ (DFSDM_FLTxJCHGR)	378
14.7.6	DFSDM フィルタ制御レジスタ (DFSDM_FLTxFCR)	379
14.7.7	インジェクトグループの DFSDM データレジスタ (DFSDM_FLTxJDATAR)	380
14.7.8	レギュラチャネルの DFSDM データレジスタ (DFSDM_FLTxRDATAR)	381
14.7.9	DFSDM アナログウォッチドッグ高閾値レジスタ (DFSDM_FLTxAWHTR)	381
14.7.10	DFSDM アナログウォッチドッグ低閾値レジスタ (DFSDM_FLTxAWLTR)	382
14.7.11	DFSDM アナログウォッチドックスステータスレジスタ (DFSDM_FLTxAWSR)	383
14.7.12	DFSDM アナログウォッチドッククリアフラグレジスタ (DFSDM_FLTxAWCFR)	383
14.7.13	DFSDM 極値検出回路最大値レジスタ (DFSDM_FLTxEXMAX)	384
14.7.14	DFSDM 極値検出回路最小値レジスタ (DFSDM_FLTxEXMIN)	384
14.7.15	DFSDM 変換タイマレジスタ (DFSDM_FLTxCNVTIMR)	385
14.8	DFSDM レジスタマップ	386

15	乱数発生器 (RNG)	391
15.1	概要	391
15.2	RNG の主な機能	391
15.3	RNG の機能説明	391
15.3.1	動作	392
15.3.2	エラー管理	392
15.4	RNG レジスタ	393
15.4.1	RNG 制御レジスタ (RNG_CR)	393
15.4.2	RNG ステータスレジスタ (RNG_SR)	393
15.4.3	RNG データレジスタ (RNG_DR)	395
15.4.4	RNG レジスタマップ	395
16	高機能制御タイマ (TIM1 および TIM8)	396
16.1	TIM1 および TIM8 の概要	396
16.2	TIM1 および TIM8 の主な機能	396
16.3	TIM1 および TIM8 の機能詳細	398
16.3.1	タイムベースユニット	398
16.3.2	カウンタモード	400
16.3.3	繰り返しカウンタ	409
16.3.4	クロック選択	411
16.3.5	キャプチャ/比較チャネル	414
16.3.6	入力キャプチャモード	417
16.3.7	PWM 入力モード	418
16.3.8	強制出力モード	419
16.3.9	出力比較モード	419
16.3.10	PWM モード	420
16.3.11	相補出力とデッドタイム挿入	423
16.3.12	ブレーク機能の使用	425
16.3.13	外部イベントによる OCxREF 信号のクリア	428
16.3.14	6 ステップ PWM 生成	429
16.3.15	ワンパルスモード	430
16.3.16	エンコーダインタフェースモード	431
16.3.17	タイマ入力 XOR 機能	434
16.3.18	ホールセンサとのインタフェース	434
16.3.19	TIMx と外部トリガの同期	436
16.3.20	タイマの同期	439

16.3.21	デバッグモード	439
16.4	TIM1 および TIM8 レジスタ	440
16.4.1	TIM1 および TIM8 制御レジスタ 1 (TIMx_CR1)	440
16.4.2	TIM1 および TIM8 制御レジスタ 2 (TIMx_CR2)	441
16.4.3	TIM1 および TIM8 のスレーブモード制御レジスタ (TIMx_SMCR)	443
16.4.4	TIM1 および TIM8 DMA / 割り込み有効レジスタ (TIMx_DIER)	445
16.4.5	TIM1 および TIM8 のステータスレジスタ (TIMx_SR)	447
16.4.6	TIM1 および TIM8 のイベント生成レジスタ (TIMx_EGR)	448
16.4.7	TIM1 および TIM8 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	450
16.4.8	TIM1 および TIM8 のキャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	453
16.4.9	TIM1 および TIM8 のキャプチャ/比較有効レジスタ (TIMx_CCER)	455
16.4.10	TIM1 および TIM8 のカウンタ (TIMx_CNT)	458
16.4.11	TIM1 および TIM8 プリスケアラ (TIMx_PSC)	458
16.4.12	TIM1 および TIM8 自動再ロードレジスタ (TIMx_ARR)	458
16.4.13	TIM1 および TIM8 繰り返しカウンタレジスタ (TIMx_RCR)	459
16.4.14	TIM1 および TIM8 キャプチャ/比較レジスタ 1 (TIMx_CCR1)	459
16.4.15	TIM1 および TIM8 キャプチャ/比較レジスタ 2 (TIMx_CCR2)	460
16.4.16	TIM1 および TIM8 キャプチャ/比較レジスタ 3 (TIMx_CCR3)	460
16.4.17	TIM1 および TIM8 キャプチャ/比較レジスタ 4 (TIMx_CCR4)	461
16.4.18	TIM1 および TIM8 ブレークおよびデッドタイムレジスタ (TIMx_BDTR)	461
16.4.19	TIM1 および TIM8 DMA 制御レジスタ (TIMx_DCR)	463
16.4.20	フル転送のための TIM1 および TIM8 DMA アドレス (TIMx_DMAR)	464
16.4.21	TIM1 および TIM8 レジスタマップ	465
17	汎用タイマ (TIM2 から TIM5)	467
17.1	TIM2 から TIM5 の概要	467
17.2	TIM2 から TIM5 の主な機能	467
17.3	TIM2 から TIM5 の機能説明	468
17.3.1	タイムベースユニット	468
17.3.2	カウンタモード	470
17.3.3	クロック選択	479
17.3.4	キャプチャ/比較チャネル	482
17.3.5	入力キャプチャモード	484
17.3.6	PWM 入力モード	485
17.3.7	強制出力モード	486

17.3.8	出力比較モード	486
17.3.9	PWM モード	487
17.3.10	ワンパルスモード	491
17.3.11	外部イベントによる OCxREF 信号のクリア	492
17.3.12	エンコーダインタフェースモード	493
17.3.13	タイマ入力 XOR 機能	495
17.3.14	タイマと外部トリガの同期	495
17.3.15	タイマの同期	499
17.3.16	デバッグモード	504
17.4	TIM2 から TIM5 のレジスタ	505
17.4.1	TIMx 制御レジスタ 1 (TIMx_CR1)	505
17.4.2	TIMx 制御レジスタ 2 (TIMx_CR2)	507
17.4.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR)	508
17.4.4	TIMx DMA / 割り込み有効レジスタ (TIMx_DIER)	510
17.4.5	TIMx ステータスレジスタ (TIMx_SR)	511
17.4.6	TIMx イベント生成レジスタ (TIMx_EGR)	513
17.4.7	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	514
17.4.8	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	517
17.4.9	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER)	518
17.4.10	TIMx カウンタ (TIMx_CNT)	520
17.4.11	TIMx プリスケアラ (TIMx_PSC)	520
17.4.12	TIMx 自動再ロードレジスタ (TIMx_ARR)	520
17.4.13	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	521
17.4.14	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCR2)	521
17.4.15	TIMx キャプチャ/比較モードレジスタ 3 (TIMx_CCR3)	522
17.4.16	TIMx キャプチャ/比較モードレジスタ 4 (TIMx_CCR4)	522
17.4.17	TIMx DMA 制御レジスタ (TIMx_DCR)	523
17.4.18	完全転送の TIMx DMA アドレス (TIMx_DMAR)	523
17.4.19	TIM2 オプションレジスタ (TIM2_OR)	524
17.4.20	TIM5 オプションレジスタ (TIM5_OR)	525
17.4.21	TIMx レジスタマップ	526
18	汎用タイマ (TIM9 から TIM14)	528
18.1	TIM9 から TIM14 の概要	528
18.2	TIM9 から TIM14 の主な機能	528
18.2.1	TIM9/TIM12 の主な機能	528
18.2.2	TIM10/TIM11 および TIM13/TIM14 の主な機能	529

18.3	TIM9 から TIM14 の機能詳細	531
18.3.1	タイムベースユニット	531
18.3.2	カウンタモード	533
18.3.3	クロック選択	536
18.3.4	キャプチャ/比較チャネル	538
18.3.5	入力キャプチャモード	539
18.3.6	PWM 入力モード (TIM9/12 の場合のみ)	540
18.3.7	強制出力モード	541
18.3.8	出力比較モード	542
18.3.9	PWM モード	543
18.3.10	ワンパルスモード	544
18.3.11	TIM9/12 外部トリガ同期	546
18.3.12	タイマ同期 (TIM9/12)	549
18.3.13	デバッグモード	549
18.4	TIM9 および TIM12 レジスタ	549
18.4.1	TIM9/12 制御レジスタ 1 (TIMx_CR1)	549
18.4.2	TIM9/12 のスレーブモード制御レジスタ (TIMx_SMCR)	551
18.4.3	TIM9/12 割り込み有効レジスタ (TIMx_DIER)	552
18.4.4	TIM9/12 のステータスレジスタ (TIMx_SR)	553
18.4.5	TIM9/12 のイベント生成レジスタ (TIMx_EGR)	555
18.4.6	TIM9/12 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	556
18.4.7	TIM9/12 のキャプチャ/比較有効レジスタ (TIMx_CCER)	559
18.4.8	TIM9/12 のカウンタ (TIMx_CNT)	560
18.4.9	TIM9/12 のプリスケアラ (TIMx_PSC)	560
18.4.10	TIM9/12 の自動再ロードレジスタ (TIMx_ARR)	560
18.4.11	TIM9/12 のキャプチャ/比較レジスタ 1 (TIMx_CCR1)	561
18.4.12	TIM9/12 のキャプチャ/比較レジスタ 2 (TIMx_CCR2)	561
18.4.13	TIM9/12 のレジスタマップ	562
18.5	TIM10/11/13/14 レジスタ	564
18.5.1	TIM10/11/13/14 制御レジスタ 1 (TIMx_CR1)	564
18.5.2	TIM10/11/13/14 割り込み有効レジスタ (TIMx_DIER)	565
18.5.3	TIM10/11/13/14 ステータスレジスタ (TIMx_SR)	565
18.5.4	TIM10/11/13/14 のイベント生成レジスタ (TIMx_EGR)	566
18.5.5	TIM10/11/13/14 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	567
18.5.6	TIM10/11/13/14 のキャプチャ/比較有効レジスタ (TIMx_CCER)	570
18.5.7	TIM10/11/13/14 のカウンタ (TIMx_CNT)	571
18.5.8	TIM10/11/13/14 プリスケアラ (TIMx_PSC)	571

18.5.9	TIM10/11/13/14 自動再ロードレジスタ (TIMx_ARR)	571
18.5.10	TIM10/11/13/14 キャプチャ／比較レジスタ 1 (TIMx_CCR1)	572
18.5.11	TIM11 オプションレジスタ 1 (TIM11_OR)	572
18.5.12	TIM10/11/13/14 レジスタマップ	573
19	基本タイマ (TIM6/7)	575
19.1	概要	575
19.2	TIM6/7 の主な機能	575
19.3	TIM6/7 の機能説明	576
19.3.1	タイムベースユニット	576
19.3.2	カウントモード	578
19.3.3	クロックソース	581
19.3.4	デバッグモード	582
19.4	TIM6/7 レジスタ	583
19.4.1	TIM6/7 制御レジスタ 1 (TIMx_CR1)	583
19.4.2	TIM6/7 制御レジスタ 2 (TIMx_CR2)	585
19.4.3	TIM6/7 DMA / 割り込み有効レジスタ (TIMx_DIER)	585
19.4.4	TIM6/7 のステータスレジスタ (TIMx_SR)	586
19.4.5	TIM6/7 のイベント生成レジスタ (TIMx_EGR)	586
19.4.6	TIM6/7 のカウンタ (TIMx_CNT)	586
19.4.7	TIM6/7 のプリスケアラ (TIMx_PSC)	587
19.4.8	TIM6/7 の自動再ロードレジスタ (TIMx_ARR)	587
19.4.9	TIM6/7 のレジスタマップ	588
20	独立型ウォッチドッグ (IWDG)	589
20.1	IWDG の概要	589
20.2	IWDG の主な機能	589
20.3	IWDG の機能説明	589
20.3.1	ハードウェアウォッチドッグ	589
20.3.2	レジスタのアクセス保護	589
20.3.3	デバッグモード	590
20.4	IWDG レジスタ	591
20.4.1	キーレジスタ (IWDG_KR)	591
20.4.2	プリスケアラレジスタ (IWDG_PR)	592
20.4.3	再ロードレジスタ (IWDG_RLR)	593
20.4.4	ステータスレジスタ (IWDG_SR)	593

20.4.5	IWDG レジスタマップ	594
21	ウィンドウ型ウォッチドッグ (WWDG)	595
21.1	WWDG の概要	595
21.2	WWDG の主な機能	595
21.3	WWDG の機能説明	595
21.4	ウォッチドッグタイムアウトをプログラムする方法	597
21.5	デバッグモード	598
21.6	WWDG レジスタ	599
21.6.1	制御レジスタ (WWDG_CR)	599
21.6.2	設定レジスタ (WWDG_CFR)	600
21.6.3	ステータスレジスタ (WWDG_SR)	600
21.6.4	WWDG レジスタマップ	601
22	リアルタイムクロック (RTC)	602
22.1	概要	602
22.2	RTC の主な機能	602
22.3	RTC の機能説明	604
22.3.1	クロックとプリスケアラ	604
22.3.2	リアルタイムクロックとカレンダー	604
22.3.3	プログラム可能なアラーム	605
22.3.4	周期的自動ウェイクアップ	605
22.3.5	RTC の初期化と設定	606
22.3.6	カレンダーの読み出し	608
22.3.7	RTC のリセット	609
22.3.8	RTC の同期	609
22.3.9	RTC リファレンスクロック検出	610
22.3.10	RTC の簡易デジタル較正	610
22.3.11	RTC の高精度デジタル較正	611
22.3.12	タイムスタンプ機能	613
22.3.13	タンパ検出	614
22.3.14	較正クロック出力	615
22.3.15	アラーム出力	616

22.4	RTC と低電力モード	616
22.5	RTC 割り込み	617
22.6	RTC レジスタ	618
22.6.1	RTC 時刻レジスタ (RTC_TR)	618
22.6.2	RTC 日付レジスタ (RTC_DR)	619
22.6.3	RTC 制御レジスタ (RTC_CR)	620
22.6.4	RTC 初期化とステータスレジスタ (RTC_ISR)	622
22.6.5	RTC プリスケアラレジスタ (RTC_PRER)	624
22.6.6	RTC ウェイクアップタイムレジスタ (RTC_WUTR)	625
22.6.7	RTC 較正レジスタ (RTC_CALIBR)	626
22.6.8	RTC アラーム A レジスタ (RTC_ALRMAR)	627
22.6.9	RTC アラーム B レジスタ (RTC_ALRMBR)	628
22.6.10	RTC 書き込み保護レジスタ (RTC_WPR)	629
22.6.11	RTC サブセカンドレジスタ (RTC_SSR)	629
22.6.12	RTC シフト制御レジスタ (RTC_SHIFTR)	630
22.6.13	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	631
22.6.14	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	631
22.6.15	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)	632
22.6.16	RTC 較正レジスタ (RTC_CALR)	632
22.6.17	RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)	633
22.6.18	RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)	635
22.6.19	RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)	636
22.6.20	RTC バックアップレジスタ (RTC_BKPxR)	637
22.6.21	RTC レジスタマップ	638
23	FMPI2C (高速モードプラス Inter-Integrated Circuit) インタフェース	640
23.1	概要	640
23.2	FMPI2C の主な機能	640
23.3	FMPI2C の実装	641
23.4	FMPI2C 機能の詳細	641
23.4.1	FMPI2C ブロック図	642
23.4.2	FMPI2C クロックの要件	643
23.4.3	モード選択	643
23.4.4	FMPI2C の初期化	645
23.4.5	ソフトウェアリセット	649
23.4.6	データ転送	650

23.4.7	FMPI2C スレーブモード	652
23.4.8	FMPI2C マスタモード	661
23.4.9	FMPI2C_TIMINGR レジスタの設定例	673
23.4.10	SMBus 固有の機能	673
23.4.11	SMBus 初期化	676
23.4.12	SMBus : FMPI2C_TIMEOCTR レジスタの設定例	678
23.4.13	SMBus スレーブモード	679
23.4.14	エラー条件	685
23.4.15	DMA リクエスト	687
23.4.16	デバッグモード	688
23.5	FMPI2C 低電力モード	688
23.6	FMPI2C 割り込み	689
23.7	FMPI2C レジスタ	690
23.7.1	制御レジスタ 1 (FMPI2C_CR1)	690
23.7.2	制御レジスタ 2 (FMPI2C_CR2)	693
23.7.3	Own Address 1 レジスタ (FMPI2C_OAR1)	696
23.7.4	Own Address 2 レジスタ (FMPI2C_OAR2)	697
23.7.5	タイミングレジスタ (FMPI2C_TIMINGR)	698
23.7.6	タイムアウトレジスタ (FMPI2C_TIMEOCTR)	699
23.7.7	割り込みおよびステータスレジスタ (FMPI2C_ISR)	700
23.7.8	割り込みクリアレジスタ (FMPI2C_ICR)	702
23.7.9	PEC レジスタ (FMPI2C_PECR)	703
23.7.10	受信データレジスタ (FMPI2C_RXDR)	704
23.7.11	送信データレジスタ (FMPI2C_TXDR)	704
23.7.12	FMPI2C レジスタマップ	705
24	I²C (Inter-integrated circuit) インタフェース	707
24.1	I ² C の概要	707
24.2	I ² C の主な特長	708
24.3	I ² C の機能説明	709
24.3.1	モード選択	709
24.3.2	I ² C スレーブモード	710
24.3.3	I ² C マスタモード	713
24.3.4	エラー条件	719
24.3.5	プログラム可能なノイズフィルタ	720
24.3.6	SDA/SCL ライン制御	721

24.3.7	SMBus	721
24.3.8	DMA リクエスト	724
24.3.9	パケットエラーチェック	725
24.4	I ² C 割り込み	726
24.5	I ² C デバッグモード	728
24.6	I ² C レジスタ	728
24.6.1	I ² C 制御レジスタ 1 (I2C_CR1)	728
24.6.2	I ² C 制御レジスタ 2 (I2C_CR2)	730
24.6.3	I ² C 自己アドレスレジスタ 1 (I2C_OAR1)	732
24.6.4	I ² C 自己アドレスレジスタ 2 (I2C_OAR2)	732
24.6.5	I ² C データレジスタ (I2C_DR)	733
24.6.6	I ² C ステータスレジスタ 1 (I2C_SR1)	733
24.6.7	I ² C ステータスレジスタ 2 (I2C_SR2)	737
24.6.8	I ² C クロック制御レジスタ (I2C_CCR)	738
24.6.9	I ² C TRISE レジスタ (I2C_TRISE)	739
24.6.10	I ² C FLTR レジスタ (I2C_FLTR)	740
24.6.11	I2C のレジスタマップ	741
25	USART	
	(Universal synchronous asynchronous receiver transmitter)	742
25.1	USART の概要	742
25.2	USART の主な機能	743
25.3	USART の実装	744
25.4	USART の機能詳細	744
25.4.1	USART キャラクタの説明	747
25.4.2	トランスミッタ	748
25.4.3	レシーバ	751
25.4.4	フラクショナルボーレート生成	756
25.4.5	クロック偏差に対する USART レシーバの許容誤差	765
25.4.6	マルチプロセッサ通信	766
25.4.7	パリティ制御	768
25.4.8	LIN (Local Interconnection Network) モード	769
25.4.9	USART 同期モード	771
25.4.10	単線半二重通信	773
25.4.11	スマートカード	774
25.4.12	Ir SIR ENDEC ブロック	776
25.4.13	DMA を使用した連続通信	778



25.4.14	ハードウェアフロー制御	780
25.5	USART 割り込み	782
25.6	USART レジスタ	783
25.6.1	ステータスレジスタ (USART_SR)	783
25.6.2	データレジスタ (USART_DR)	786
25.6.3	ボーレートレジスタ (USART_BRR)	786
25.6.4	制御レジスタ 1 (USART_CR1)	787
25.6.5	制御レジスタ 2 (USART_CR2)	789
25.6.6	制御レジスタ 3 (USART_CR3)	790
25.6.7	ガード時間およびプリスケアラレジスタ (USART_GTPR)	792
25.6.8	USART レジスタマップ	793
26	シリアルペリフェラルインタフェース / I2S (SPI/I2S)	794
26.1	概要	794
26.1.1	SPI の主な機能	795
26.1.2	SPI の拡張機能	796
26.1.3	I2S の機能	796
26.2	SPI/I2S の実装	796
26.3	SPI の機能説明	797
26.3.1	概要	797
26.3.2	マスタとスレーブの 1 対 1 の通信	798
26.3.3	標準マルチスレーブ通信	801
26.3.4	マルチマスタ通信	802
26.3.5	スレーブ選択 (NSS) ピンの管理	802
26.3.6	通信フォーマット	804
26.3.7	SPI の設定	806
26.3.8	SPI を有効にする手順	806
26.3.9	データの送受信手順	807
26.3.10	SPI を無効にする手順	809
26.3.11	DMA (ダイレクトメモリアクセス) を使用する通信	810
26.3.12	SPI ステータスフラグ	812
26.3.13	SPI エラーフラグ	813
26.4	SPI の特殊機能	814
26.4.1	TI モード	814
26.4.2	CRC 計算	815

26.5	SPI 割り込み	817
26.6	I ² S の機能説明	818
26.6.1	I ² S の概要	818
26.6.2	I2S 全二重	819
26.6.3	サポートされるオーディオプロトコル	820
26.6.4	クロックジェネレータ	826
26.6.5	I ² S マスタモード	829
26.6.6	I ² S スレーブモード	831
26.6.7	I ² S ステータスフラグ	832
26.6.8	I ² S エラーフラグ	833
26.6.9	I ² S 割り込み	834
26.6.10	DMA の機能	834
26.7	SPI および I ² S レジスタ	835
26.7.1	SPI 制御レジスタ 1 (SPI_CR1) (I ² S モードでは使用しません)	835
26.7.2	SPI 制御レジスタ 2 (SPI_CR2)	837
26.7.3	SPI ステータスレジスタ (SPI_SR)	838
26.7.4	SPI データレジスタ (SPI_DR)	840
26.7.5	SPI CRC 多項式レジスタ (SPI_CRCPR) (I ² S モードでは使用しません)	840
26.7.6	SPI RX CRC レジスタ (SPI_RXCRCR) (I ² S モードでは使用しません)	841
26.7.7	SPI TX CRC レジスタ (SPI_TXCRCR) (I ² S では使用しません)	841
26.7.8	SPI_I ² S 設定レジスタ (SPI_I2SCFGR)	842
26.7.9	SPI_I ² S プリスケーラレジスタ (SPI_I2SPR)	844
26.7.10	SPI レジスタマップ	845
27	セキュアデジタル入力/出力インタフェース (SDIO)	846
27.1	SDIO の主な特長	846
27.2	SDIO バストポロジ	846
27.3	SDIO 機能詳細	848
27.3.1	SDIO アダプタ	850
27.3.2	SDIO APB2 インタフェース	861
27.4	カード機能詳細	862
27.4.1	カード識別モード	862
27.4.2	カードリセット	863
27.4.3	動作電圧範囲の検証	863
27.4.4	カード識別プロセス	863
27.4.5	ブロック書き込み	864

27.4.6	ブロック読み出し	865
27.4.7	ストリームアクセス、ストリーム書き込み、およびストリーム読み出し (MultiMediaCard のみ)	865
27.4.8	消去: グループ消去とセクタ消去	867
27.4.9	ワイドバス選択または選択解除	867
27.4.10	保護管理	867
27.4.11	カードステータスレジスタ	871
27.4.12	SD ステータスレジスタ	874
27.4.13	SD I/O モード	878
27.4.14	コマンドおよびレスポンス	879
27.5	レスポンスフォーマット	882
27.5.1	R1 (ノーマルレスポンスコマンド)	882
27.5.2	R1b	883
27.5.3	R2 (CID、CSD レジスタ)	883
27.5.4	R3 (OCR レジスタ)	883
27.5.5	R4 (高速 I/O)	884
27.5.6	R4b	884
27.5.7	R5 (割り込みリクエスト)	885
27.5.8	R6	885
27.6	SDIO I/O カード固有の操作	886
27.6.1	SDIO_D2 信号による SDIO I/O 読み出しウェイト操作	886
27.6.2	SDIO_CK の停止による SDIO 読み出しウェイト操作	886
27.6.3	SDIO サスペンド/レジャーム操作	886
27.6.4	SDIO 割込み	887
27.7	HW フロー制御	887
27.8	SDIO レジスタ	888
27.8.1	SDIO 電源制御レジスタ (SDIO_POWER)	888
27.8.2	SDIO クロック制御レジスタ (SDIO_CLKCR)	888
27.8.3	SDIO 引数レジスタ (SDIO_ARG)	890
27.8.4	SDIO コマンドレジスタ (SDIO_CMD)	890
27.8.5	SDIO コマンドレスポンスレジスタ (SDIO_RESPCMD)	891
27.8.6	SDIO レスポンス 1 ~ 4 レジスタ (SDIO_RESPx)	892
27.8.7	SDIO データタイマレジスタ (SDIO_DTIMER)	892
27.8.8	SDIO データ長レジスタ (SDIO_DLEN)	893
27.8.9	SDIO データ制御レジスタ (SDIO_DCTRL)	893
27.8.10	SDIO データカウンタレジスタ (SDIO_DCOUNT)	895
27.8.11	SDIO ステータスレジスタ (SDIO_STA)	895

27.8.12	SDIO 割り込みクリアレジスタ (SDIO_ICR)	896
27.8.13	SDIO マスクレジスタ (SDIO_MASK)	898
27.8.14	SDIO FIFO カウンタレジスタ (SDIO_FIFOCNT)	901
27.8.15	SDIO データ FIFO レジスタ (SDIO_FIFO)	901
27.8.16	SDIO レジスタマップ	902
28	コントローラエリアネットワーク (bxCAN)	904
28.1	概要	904
28.2	bxCAN の主な機能	904
28.3	bxCAN の概要	905
28.3.1	CAN 2.0B アクティブコア	905
28.3.2	制御レジスタ、ステータスレジスタ、設定レジスタ	905
28.3.3	送信メールボックス	905
28.3.4	受信フィルタ	906
28.4	bxCAN 動作モード	907
28.4.1	初期化モード	907
28.4.2	通常モード	907
28.4.3	スリープモード (低電力)	908
28.5	テストモード	909
28.5.1	サイレントモード	909
28.5.2	ループバックモード	909
28.5.3	サイレントモードとループバックモードの組み合わせ	910
28.6	デバッグモードでの挙動	910
28.7	bxCAN 機能の説明	910
28.7.1	送信処理	910
28.7.2	タイムトリガ通信モード	912
28.7.3	受信処理	912
28.7.4	識別子フィルタリング	914
28.7.5	メッセージストレージ	918
28.7.6	エラー管理	920
28.7.7	ビットタイミング	920
28.8	bxCAN の割込み	923
28.9	CAN レジスタ	924
28.9.1	レジスタのアクセス保護	924
28.9.2	CAN 制御/ステータスレジスタ	924
28.9.3	CAN メールボックスレジスタ	935

28.9.4	CAN フィルタレジスタ	941
28.9.5	bxCAN レジスタマップ	945
29	USB On-The-Go フルスピード (OTG_FS)	949
29.1	概要	949
29.2	USB_OTG の主な機能	950
29.2.1	標準機能	950
29.2.2	ホストモード機能	951
29.2.3	ペリフェラルモード機能	951
29.2.4	USB の分割ルール	951
29.3	USB_OTG の実装	952
29.4	USB OTG の機能詳細	952
29.4.1	USB OTG ブロック図	952
29.4.2	OTG コア	953
29.4.3	フルスピード OTG PHY	953
29.5	OTG デュアルロールデバイス (DRD)	954
29.5.1	ID ラインの検出	954
29.5.2	HNP デュアルロールデバイス	954
29.5.3	SRP デュアルロールデバイス	955
29.6	USB ペリフェラル	955
29.6.1	SRP 対応ペリフェラル	956
29.6.2	ペリフェラルの状態	956
29.6.3	ペリフェラルエンドポイント	957
29.7	USB ホスト	959
29.7.1	SRP 対応ホスト	960
29.7.2	USB ホストの状態	960
29.7.3	ホストチャネル	962
29.7.4	ホストスケジューラ	963
29.8	SOF トリガ	964
29.8.1	ホストの SOF	964
29.8.2	ペリフェラルの SOF	964
29.9	電力オプション	965
29.10	OTG_HFIR レジスタの動的更新	966
29.11	USB データ FIFO	966
29.11.1	ペリフェラル FIFO アーキテクチャ	967
29.11.2	ホスト FIFO アーキテクチャ	968

29.11.3	FIFO RAM の割り当て	969
29.12	OTG_FS システムのパフォーマンス	971
29.13	OTG_FS 割り込み	971
29.14	OTG_FS 制御／ステータスレジスタ	973
29.14.1	CSR メモリマップ	973
29.15	OTG_FS レジスタ	978
29.15.1	OTG 制御およびステータスレジスタ (OTG_GOTGCTL)	978
29.15.2	OTG 割り込みレジスタ (OTG_GOTGINT)	981
29.15.3	OTG AHB 設定レジスタ (OTG_GAHBCFG)	982
29.15.4	OTG USB 設定レジスタ (OTG_GUSBCFG)	983
29.15.5	OTG リセットレジスタ (OTG_GRSTCTL)	985
29.15.6	OTG コア割り込みレジスタ (OTG_GINTSTS)	987
29.15.7	OTG 割り込みマスクレジスタ (OTG_GINTMSK)	991
29.15.8	OTG_FS 受信ステータスデバッグ読み出し／OTG ステータス読み出し およびポップレジスタ (OTG_GRXSTSR/OTG_GRXSTSP)	994
29.15.9	OTG 受信 FIFO サイズレジスタ (OTG_GRXFSIZ)	995
29.15.10	OTG ホスト非周期的送信 FIFO サイズレジスタ (OTG_HNPTXFSIZ) / エンドポイント 0 送信 FIFO サイズ (OTG_DIEPTXF0)	996
29.15.11	OTG 非周期的送信 FIFO / キューステータスレジスタ (OTG_HNPTXSTS)	997
29.15.12	OTG 一般コア設定レジスタ (OTG_GCCFG)	998
29.15.13	OTG コア ID レジスタ (OTG_CID)	999
29.15.14	OTG コア LPM 設定レジスタ (OTG_GLPMCFG)	1000
29.15.15	OTG ホスト周期的送信 FIFO サイズレジスタ (OTG_HPTXFSIZ)	1004
29.15.16	OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTXFx) (x = 1 ~ 5, ここで x は FIFO_number)	1004
29.15.17	ホストモードレジスタ	1005
29.15.18	OTG ホスト設定レジスタ (OTG_HCFG)	1005
29.15.19	OTG ホストフレームインターバルレジスタ (OTG_HFIR)	1006
29.15.20	OTG ホストフレーム番号／残りフレーム時間 レジスタ (OTG_HFNUM)	1007
29.15.21	OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG_HPTXSTS)	1008
29.15.22	OTG ホスト全チャネル割り込みレジスタ (OTG_HAINT)	1009
29.15.23	OTG ホスト全チャネル割り込みマスクレジスタ (OTG_HAINTMSK)	1009
29.15.24	OTG ホストポート制御およびステータスレジスタ (OTG_HPRT)	1010
29.15.25	OTG ホストチャネル x 特性レジスタ (OTG_HCCHARx) (x = 0..11, ここで x = Channel_number)	1012

29.15.26	OTG ホストチャネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..11, ここで x = Channel_number)	1014
29.15.27	OTG ホストチャネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..11, ここで x = Channel_number)	1015
29.15.28	OTG ホストチャネル x 転送サイズレジスタ (OTG_HCTSIZx) (x = 0..11, ここで x = Channel_number)	1016
29.15.29	デバイスモードレジスタ	1017
29.15.30	OTG デバイス設定レジスタ (OTG_DCFG)	1017
29.15.31	OTG デバイス制御レジスタ (OTG_DCTL)	1018
29.15.32	OTG デバイスステータスレジスタ (OTG_DSTS)	1020
29.15.33	OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK)	1021
29.15.34	OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK)	1022
29.15.35	OTG デバイス全エンドポイント割り込みレジスタ (OTG_DAIN)	1022
29.15.36	OTG 全エンドポイント割り込みマスクレジスタ (OTG_DAINMSK)	1023
29.15.37	OTG デバイス V _{BUS} 放電時間レジスタ (OTG_DVBUSDIS)	1024
29.15.38	OTG デバイス V _{BUS} パルシング時間レジスタ (OTG_DVBUSPULSE)	1024
29.15.39	OTG デバイス IN エンドポイント FIFO エンプティ割り込み マスクレジスタ (OTG_DIEPEMPMSK)	1025
29.15.40	OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_DIEPCTL0)	1025
29.15.41	OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1 ~ 5、ここで x = Endpoint_number)	1027
29.15.42	OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG_DOEPCTL0)	1029
29.15.43	OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPCTLx) (x = 1 ~ 5、ここで x = Endpoint_number)	1031
29.15.44	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx) (x = 0 ~ 5、ここで x = Endpoint_number)	1033
29.15.45	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPINTx) (x = 0 ~ 5、ここで x = Endpoint_number)	1034
29.15.46	OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_DIEPTSIZ0)	1035
29.15.47	OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_DOEPTSIZ0)	1036
29.15.48	OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSIZx) (x = 1 ~ 5、ここで x = Endpoint_number)	1037
29.15.49	OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) (x = 0 ~ 5、ここで x = Endpoint_number)	1038
29.15.50	OTG デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_DOEPTSIZx) (x = 1 ~ 5、ここで x = Endpoint_number)	1038

29.15.51	OTG パワーおよびクロックゲーティング制御レジスタ (OTG_PCGCCTL)	1039
29.15.52	OTG_FS レジスタマップ	1040
29.16	OTG_FS プログラミングモデル	1048
29.16.1	コアの初期化	1048
29.16.2	ホストの初期化	1049
29.16.3	デバイスの初期化	1049
29.16.4	ホストプログラミングモデル	1050
29.16.5	デバイスプログラミングモデル	1070
29.16.6	最悪ケースの応答時間	1090
29.16.7	OTG プログラミングモデル	1091
30	デバッグサポート (DBG)	1097
30.1	概要	1097
30.2	ARM® リファレンス資料	1098
30.3	SWJ デバッグポート (シリアルワイヤと JTAG)	1098
30.3.1	JTAG-DP または SW-DP の選択メカニズム	1099
30.4	ピン名とデバッグポートピン	1099
30.4.1	SWJ デバッグポートピン	1100
30.4.2	柔軟性の高い SWJ-DP ピンの割り当て	1100
30.4.3	JTAG ピンでの内部プルアップ／プルダウン	1101
30.4.4	シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法	1102
30.5	JTAG TAP 接続	1102
30.6	ID コードとロック機構	1104
30.6.1	MCU デバイス ID コード	1104
30.6.2	バウンダリスキャン TAP	1104
30.6.3	FPU 搭載 Cortex®-M4 TAP	1104
30.6.4	FPU 搭載 Cortex®-M4 JEDEC-106 ID コード	1105
30.7	JTAG デバッグポート	1105
30.8	SW デバッグポート	1107
30.8.1	SW プロトコルの概要	1107
30.8.2	SW プロトコルシーケンス	1107
30.8.3	SW-DP ステートマシン (リセット、アイドル状態、ID コード)	1108
30.8.4	DP と AP の読み出し／書き込みアクセス	1108
30.8.5	SW-DP レジスタ	1109
30.8.6	SW-AP レジスタ	1110

30.9	AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効	1110
30.10	コアデバッグ	1111
30.11	システムリセット中のデバッガホスト接続機能	1112
30.12	FPB (フラッシュパッチブレイクポイント)	1112
30.13	DWT (データウォッチポイントトリガ)	1113
30.14	ITM (計測トレースマクロセル)	1113
30.14.1	概要	1113
30.14.2	タイムスタンプパケット、同期およびオーバーフローパケット	1113
30.15	ETM (組み込みトレースマクロセル)	1115
30.15.1	概要	1115
30.15.2	信号プロトコル、パケットタイプ	1115
30.15.3	主な ETM レジスタ	1116
30.15.4	設定例	1116
30.16	MCU デバッグコンポーネント (DBGMCU)	1116
30.16.1	低電力モードのデバッグサポート	1117
30.16.2	タイマ、ウォッチドッグ、bxCAN、および I ² C のデバッグサポート	1117
30.16.3	デバッグ MCU 設定レジスタ	1117
30.16.4	デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)	1119
30.16.5	デバッグ MCU APB2 フリーズレジスタ (DBGMCU_APB2_FZ)	1121
30.17	TPIU (トレースポートインタフェースユニット)	1122
30.17.1	概要	1122
30.17.2	TRACE ピンの割当て	1123
30.17.3	TPUI フォーマット	1124
30.17.4	TPUI フレーム同期パケット	1125
30.17.5	同期フレームパケットの送信	1125
30.17.6	同期モード	1125
30.17.7	非同期モード	1126
30.17.8	TRACECLKIN 接続	1126
30.17.9	TPIU レジスタ	1126
30.17.10	設定例	1127
30.18	DBG レジスタマップ	1128

31	デバイス電子署名	1129
31.1	ユニークデバイス ID レジスタ (96 ビット)	1129
31.2	フラッシュサイズ	1130
31.3	パッケージデータレジスタ	1131
32	改版履歴	1136



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表の一覧

表 1.	レジスタ境界アドレス	49
表 2.	ブートモード	53
表 3.	内蔵ブートローダインタフェース	54
表 4.	メモリ割当てとブートモード／物理的な再割当て (STM32F412xx)	55
表 5.	フラッシュモジュールの構成	57
表 6.	CPU クロック (HCLK) 周波数によるウェイトステート数	58
表 7.	プログラム／消去の並列処理	62
表 8.	フラッシュ割り込みリクエスト	64
表 9.	オプションバイトの構成	64
表 10.	ユーザオプションバイトの説明	65
表 11.	アクセスと読み出し保護レベル	68
表 12.	OTP エリアの構成	71
表 13.	フラッシュレジスタマップとリセット値	79
表 14.	CRC 計算ユニットのレジスタマップとリセット値	83
表 15.	低電力モードの概要	92
表 16.	Sleep-now への移行と終了	93
表 17.	Sleep-on-exit への移行と終了	94
表 18.	BAM-now への移行と終了	95
表 19.	BAM-on-exit への移行と終了	95
表 20.	STOP モードの動作モード	96
表 21.	STOP モードへの移行と終了	97
表 22.	STANDBY モードへの移行と終了	99
表 23.	PWR - レジスタマップとリセット値	106
表 24.	STM32F412xx の RCC レジスタマップとリセット値	160
表 25.	ポートビット設定表	164
表 26.	柔軟性の高い SWJ-DP ピンの割り当て	166
表 27.	RTC 追加機能	173
表 28.	GPIO レジスタマップとリセット値	179
表 29.	SYSCFG レジスタマップとリセット値	188
表 30.	DMA1 リクエストマッピング	193
表 31.	DMA2 リクエストマッピング	193
表 32.	転送元および転送先アドレス	194
表 33.	ダブルバッファモード (DBM=1) における転送元および転送先アドレスレジスタ	199
表 34.	パッキング／アンパッキングおよびエンディアンの動作 (ビット PINC=MINC=1)	199
表 35.	PSIZE と MSIZE に対する NDT に関する制限事項	200
表 36.	FIFO 閾値設定	202
表 37.	実現可能な DMA 設定	206
表 38.	DMA 割り込みリクエスト	209
表 39.	DMA レジスタマップとリセット値	220
表 40.	STM32F412xx のベクタテーブル	225
表 41.	外部割り込み／イベントコントローラのレジスタマップとリセット値	236
表 42.	NOR/PSRAM バンク選択	240
表 43.	NOR/PSRAM 外部メモリアドレス	241
表 44.	プログラム可能な NOR/PSRAM のアクセスパラメータ	242
表 45.	非マルチプレクス I/O NOR フラッシュメモリ	242
表 46.	16 ビットマルチプレクス I/O NOR フラッシュメモリ	243
表 47.	非マルチプレクス I/O PSRAM/SRAM	243
表 48.	16 ビットマルチプレクス I/O PSRAM	244
表 49.	NOR フラッシュ／PSRAM : サポートされているメモリおよびトランザクションの例	244

表 50.	FSMC_BCRx ビットフィールド	247
表 51.	FSMC_BTRx ビットフィールド	248
表 52.	FSMC_BCRx ビットフィールド	249
表 53.	FSMC_BTRx ビットフィールド	250
表 54.	FSMC_BWTRx ビットフィールド	250
表 55.	FSMC_BCRx ビットフィールド	252
表 56.	FSMC_BTRx ビットフィールド	253
表 57.	FSMC_BWTRx ビットフィールド	253
表 58.	FSMC_BCRx ビットフィールド	255
表 59.	FSMC_BTRx ビットフィールド	255
表 60.	FSMC_BWTRx ビットフィールド	256
表 61.	FSMC_BCRx ビットフィールド	257
表 62.	FSMC_BTRx ビットフィールド	258
表 63.	FSMC_BWTRx ビットフィールド	258
表 64.	FSMC_BCRx ビットフィールド	260
表 65.	FSMC_BTRx ビットフィールド	260
表 66.	FSMC_BCRx ビットフィールド	265
表 67.	FSMC_BTRx ビットフィールド	266
表 68.	FSMC_BCRx ビットフィールド	267
表 69.	FSMC_BTRx ビットフィールド	268
表 70.	FSMC レジスタマップ	277
表 71.	QUADSPI 割り込みリクエスト	293
表 72.	QUADSPI レジスタマップとリセット値	306
表 73.	ADC ピン	309
表 74.	アナログウォッチドッグチャネル選択	312
表 75.	トリガ極性の設定	316
表 76.	レギュラチャネルの外部トリガ	317
表 77.	インジェクトチャネルの外部トリガ	317
表 78.	ADC 割り込み	321
表 79.	ADC グローバルレジスタマップ	336
表 80.	ADC レジスタマップとリセット値	336
表 81.	ADC レジスタマップとリセット値 (共通 ADC レジスタ)	337
表 82.	DFSDM 外部ピン	341
表 83.	DFSDM 内部信号	341
表 84.	DFSDM トリガ接続	341
表 85.	フィルタの最大出力分解能 (フィルタ出力のピークデータ値) 一部の FOSR 値	355
表 86.	積分器の最大出力分解能 (積分器出力のピークデータ値) (一部の IOSR 値、FOSR = 256 および Sinc3 フィルタタイプ (最大データ) の場合)	356
表 87.	DFSDM 割り込みリクエスト	364
表 88.	DFSDM レジスタマップとリセット値	386
表 89.	RNG レジスタマップとリセット値	395
表 90.	カウント方向とエンコーダ信号	432
表 91.	TIMx 内部トリガ接続	445
表 92.	ブレーク機能を持つ相補 OCx および OCxN チャネルの出力 制御ビット機能	457
表 93.	TIM1 および TIM8 レジスタマップとリセット値	465
表 94.	カウント方向とエンコーダ信号	494
表 95.	TIMx 内部トリガ接続	509
表 96.	標準 OCx チャネルの出力制御ビット	519
表 97.	TIM2 から TIM5 のレジスタマップとリセット値	526
表 98.	TIMx 内部トリガ接続	552
表 99.	標準 OCx チャネルの出力制御ビット	560
表 100.	TIM9/12 レジスタマップとリセット値	562

表 101.	標準 OCx チャネルの出力制御ビット	570
表 102.	TIM10/11/13/14 レジスタマップとリセット値	573
表 103.	TIM6/7 レジスタマップとリセット値	588
表 104.	最小／最大 IWDG タイムアウト期間 (32 kHz) (LSI)	590
表 105.	IWDG レジスタマップとリセット値	594
表 106.	WWDG レジスタマップとリセット値	601
表 107.	低電力モードが RTC に与える影響	616
表 108.	割り込み制御ビット	617
表 109.	RTC レジスタマップとリセット値	638
表 110.	STM32F412xx FMPI2C の実装	641
表 111.	I2C-SMBUS 仕様のデータのセットアップおよびホールド時間	648
表 112.	FMPI2C 設定表	652
表 113.	I2C-SMBUS 仕様のクロックタイミング	663
表 114.	SMBus タイムアウト仕様	675
表 115.	SMBUS の PEC 設定	677
表 116.	さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{TIMEOUT}} = 25 \text{ ms}$)	678
表 117.	さまざまな FMPI2CCLK 周波数での TIMEOUTB の設定例	678
表 118.	さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{IDLE}} = 50 \mu\text{s}$)	678
表 119.	低電力モード	688
表 120.	FMPI2C 割り込みリクエスト	689
表 121.	FMPI2C レジスタマップとリセット値	705
表 122.	Thd:dat (max) に準拠した最大 DNF[3:0] 値	720
表 123.	SMBus と I2C	721
表 124.	I2C 割り込みリクエスト	726
表 125.	I2C レジスタマップとリセット値	741
表 126.	USART の機能	744
表 127.	サンプリングされたデータからのノイズ検出	754
表 128.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 8 \text{ MHz}$ または $f_{\text{PCLK}} = 12 \text{ MHz}$)、 16 倍のオーバーサンプリング	757
表 129.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 8 \text{ MHz}$ または $f_{\text{PCLK}} = 12 \text{ MHz}$)、 8 倍のオーバーサンプリング	758
表 130.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 16 \text{ MHz}$ または $f_{\text{PCLK}} = 24 \text{ MHz}$)、 16 倍のオーバーサンプリング	759
表 131.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 16 \text{ MHz}$ または $f_{\text{PCLK}} = 24 \text{ MHz}$)、 8 倍のオーバーサンプリング	759
表 132.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 8 \text{ MHz}$ または $f_{\text{PCLK}} = 16 \text{ MHz}$)、 16 倍のオーバーサンプリング	760
表 133.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 8 \text{ MHz}$ または $f_{\text{PCLK}} = 16 \text{ MHz}$)、 8 倍のオーバーサンプリング	760
表 134.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 30 \text{ MHz}$ または $f_{\text{PCLK}} = 60 \text{ MHz}$)、 16 倍のオーバーサンプリング	761
表 135.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 30 \text{ MHz}$ または $f_{\text{PCLK}} = 60 \text{ MHz}$)、 8 倍のオーバーサンプリング	761
表 136.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 42 \text{ MHz}$ または $f_{\text{PCLK}} = 84 \text{ MHz}$)、 16 倍のオーバーサンプリング	762
表 137.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 42 \text{ MHz}$ または $f_{\text{PCLK}} = 84 \text{ MHz}$)、 8 倍のオーバーサンプリング	763
表 138.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 100 \text{ MHz}$ または $f_{\text{PCLK}} = 50 \text{ MHz}$)、 16 倍のオーバーサンプリング	764
表 139.	プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 100 \text{ MHz}$ または $f_{\text{PCLK}} = 50 \text{ MHz}$)、 8 倍のオーバーサンプリング	764
表 140.	USART レシーバ許容誤差 (DIV_Fraction が 0 の場合)	765

表 141.	USART レシーバ許容誤差 (DIV_Fraction が 0 でない場合)	766
表 142.	フレームフォーマット	768
表 143.	USART 割り込みリクエスト	782
表 144.	USART レジスタマップとリセット値	793
表 145.	STM32F412 SPI 実装	796
表 146.	SPI 割り込みリクエスト	817
表 147.	標準 8 MHz HSE を使用した場合のオーディオ周波数精度	828
表 148.	I ² S 割り込みリクエスト	834
表 149.	SPI レジスタマップとリセット値	845
表 150.	SDIO I/O 定義	849
表 151.	コマンドフォーマット	854
表 152.	ショートレスポンスフォーマット	855
表 153.	ロングレスポンスフォーマット	855
表 154.	コマンドパスステータスフラグ	855
表 155.	データトークンフォーマット	858
表 156.	DPSM フラグ	859
表 157.	送信 FIFO ステータスフラグ	860
表 158.	受信 FIFO ステータスフラグ	860
表 159.	カードステータス	871
表 160.	SD ステータス	874
表 161.	速度クラスコードフィールド	875
表 162.	移動パフォーマンスフィールド	876
表 163.	AU_SIZE フィールド	876
表 164.	最大 AU サイズ	876
表 165.	消去サイズフィールド	877
表 166.	消去タイムアウトフィールド	877
表 167.	消去オフセットフィールド	877
表 168.	ブロック指向書き込みコマンド	880
表 169.	ブロック指向書き込み保護コマンド	880
表 170.	消去コマンド	881
表 171.	I/O モードコマンド	881
表 172.	カードのロック	881
表 173.	アプリケーション固有コマンド	882
表 174.	R1 レスポンス	882
表 175.	R2 レスポンス	883
表 176.	R3 レスポンス	883
表 177.	R4 レスポンス	884
表 178.	R4b レスポンス	884
表 179.	R5 レスポンス	885
表 180.	R6 レスポンス	885
表 181.	レスポンスタイプと SDIO_RESPx レジスタ	892
表 182.	SDIO レジスタマップ	902
表 183.	送信メールボックスの配置	919
表 184.	受信メールボックスの配置	919
表 185.	bxCAN レジスタマップとリセット値	945
表 186.	サポートされる OTG_FS 速度	950
表 187.	TinyMantaLP1M の USB_OTG 実装	952
表 188.	コアグローバル制御およびステータスレジスタ (CSR)	973
表 189.	ホストモード制御およびステータスレジスタ (CSR)	974
表 190.	デバイスモード制御およびステータスレジスタ	975
表 191.	データ FIFO (DFIFO) アクセスレジスタマップ	977
表 192.	パワーおよびクロックゲーティング制御およびステータスレジスタ	977

表 193.	TRDT 値 (FS).....	984
表 194.	ソフト切断のための最短時間	1019
表 195.	OTG_FS レジスタマップとリセット値.....	1040
表 196.	SWJ デバッグポートピン	1100
表 197.	柔軟性の高い SWJ-DP ピンの割り当て	1100
表 198.	JTAG デバッグポートのデータレジスタ.....	1105
表 199.	シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ	1106
表 200.	パケットリクエスト (8 ビット).....	1107
表 201.	ACK 応答 (3 ビット)	1108
表 202.	データ転送 (33 ビット)	1108
表 203.	SW-DP レジスタ	1109
表 204.	FPU 搭載 Cortex®-M4AHB-AP レジスタ	1110
表 205.	コアデバッグレジスタ.....	1111
表 206.	主な ITM レジスタ	1114
表 207.	主な ETM レジスタ	1116
表 208.	非同期 TRACE ピンの割り当て	1123
表 209.	同期 TRACE ピンの割り当て	1123
表 210.	柔軟性の高い TRACE ピン割り当て	1124
表 211.	重要な TPIU レジスタ	1126
表 212.	DBG レジスタマップとリセット値.....	1128
表 213.	文書改版履歴.....	1136

図の一覧

図 1.	システムアーキテクチャ	46
図 2.	メモリマップ	48
図 3.	システムアーキテクチャ内のフラッシュメモリインタフェース接続	56
図 4.	連続 32 ビット命令の実行	60
図 5.	RDP レベル	68
図 6.	PCROP レベル	70
図 7.	CRC 計算ユニットのブロック図	80
図 8.	電源の概要	85
図 9.	パワーオンリセット／パワーダウンリセット波形	88
図 10.	BOR の閾値	89
図 11.	PVD の閾値	90
図 12.	簡略化されたリセット回路図	108
図 13.	クロックツリー	110
図 14.	HSE/LSE クロックソース	112
図 15.	入力キャプチャモードにおける TIM5 を使用した周波数測定	117
図 16.	入力キャプチャモードにおける TIM11 を使用した周波数測定	118
図 17.	5 V トレラント I/O ポートビットの基本構成	164
図 18.	STM32F412xx におけるオルタネート機能の選択	167
図 19.	入力フローティング／プルアップ／プルダウン設定	170
図 20.	出力設定	171
図 21.	オルタネート機能設定	171
図 22.	高 インピーダンスアナログ設定	172
図 23.	DMA ブロック図	191
図 24.	チャンネルの選択	192
図 25.	“ペリフェラルからメモリ”モード	195
図 26.	“メモリからペリフェラル”モード	196
図 27.	メモリ間モード	197
図 28.	FIFO の構造	202
図 29.	外部割り込み／イベントコントローラのブロック図	228
図 30.	外部割り込み／イベント GPIO の配置	230
図 31.	FSMC ブロック図	238
図 32.	FSMC メモリバンク	240
図 33.	モード 1 読み出しアクセス波形	246
図 34.	モード 1 書き込みアクセス波形	247
図 35.	モード A 読み出しアクセス波形	248
図 36.	モード A 書き込みアクセス波形	249
図 37.	モード 2 およびモード B 読み出しアクセス波形	251
図 38.	モード 2 書き込みアクセス波形	251
図 39.	モード B 書き込みアクセス波形	252
図 40.	モード C 読み出しアクセス波形	254
図 41.	モード C 書き込みアクセス波形	254
図 42.	モード D 読み出しアクセス波形	256
図 43.	モード D 書き込みアクセス波形	257
図 44.	マルチプレクス読み出しアクセス波形	259
図 45.	マルチプレクス書き込みアクセス波形	259
図 46.	読み出しアクセス中の非同期ウェイト波形	262
図 47.	書き込みアクセス中の非同期ウェイト波形	262
図 48.	待ち設定波形	264
図 49.	同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)	265



図 50.	同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)	267
図 51.	QUADSPI ブロック 図 (デュアルフラッシュモードが無効のとき)	279
図 52.	QUADSPI ブロック 図 (デュアルフラッシュモードが有効のとき)	280
図 53.	クワッドモードにおける読み出しコマンドの例	281
図 54.	クワッドモードにおける DDR コマンドの例	284
図 55.	nCS - CKMODE = 0 (T = CLK 周期) のとき	292
図 56.	nCS - SDR モード (T = CLK 周期) において CKMODE = 1 のとき	292
図 57.	nCS - DDR モード (T = CLK 周期) において CKMODE = 1 のとき	292
図 58.	nCS - アボートが発生し (T = CLK 周期)、CKMODE = 1 のとき	293
図 59.	1 つの ADC ブロック 図	308
図 60.	タイミング図	311
図 61.	アナログウォッチドッグによって保護される領域	311
図 62.	インジェクト変換の遅延時間	313
図 63.	12 ビットデータの右詰め	315
図 64.	12 ビットデータの左詰め	315
図 65.	6 ビットデータの左詰め	315
図 66.	温度センサおよび VREFINT チャンルのブロック 図	320
図 67.	1 つの DFSDM ブロック 図	340
図 68.	入力チャネルピンの変更	344
図 69.	チャネルトランシーバのタイミング図	346
図 70.	SPI のクロック無タイミング図	347
図 71.	マンチェスタ符号化のクロック無タイミング図	348
図 72.	マンチェスタ符号化の最初の変換 (マンチェスタ同期)	350
図 73.	DFSDM_CHyDATINR レジスタの動作モードおよび割り当て	353
図 74.	例: Sinc3 フィルタレスポンス	355
図 75.	ブロック 図	391
図 76.	高機能制御タイマのブロック 図	397
図 77.	プリスケラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	399
図 78.	プリスケラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	399
図 79.	内部クロック分周比が 1 の場合のカウンタのタイミング図	400
図 80.	内部クロック分周比が 2 の場合のカウンタのタイミング図	401
図 81.	内部クロック分周比が 4 の場合のカウンタのタイミング図	401
図 82.	内部クロック分周比が N の場合のカウンタのタイミング図	401
図 83.	ARPE=0 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	402
図 84.	ARPE=1 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	402
図 85.	内部クロック分周比が 1 の場合のカウンタのタイミング図	404
図 86.	内部クロック分周比が 2 の場合のカウンタのタイミング図	404
図 87.	内部クロック分周比が 4 の場合のカウンタのタイミング図	405
図 88.	内部クロック分周比が N の場合のカウンタのタイミング図	405
図 89.	繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図	406
図 90.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	407
図 91.	内部クロック分周比が 2 の場合のカウンタのタイミング図	407
図 92.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	408
図 93.	内部クロック分周比が N の場合のカウンタのタイミング図	408
図 94.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図	409
図 95.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	409
図 96.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	410
図 97.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	411
図 98.	TI2 外部クロックの接続例	412
図 99.	外部クロックモード 1 の制御回路	413

図 100.	外部トリガ入力ブロック	413
図 101.	外部クロックモード 2 の制御回路	414
図 102.	キャプチャ／比較チャンネル（例：チャンネル 1 入力ステージ）	415
図 103.	キャプチャ／比較チャンネル 1 メイン回路	415
図 104.	キャプチャ／比較チャンネル（チャンネル 1 から 3）の出力ステージ	416
図 105.	キャプチャ／比較チャンネル（チャンネル 4）の出力ステージ	416
図 106.	PWM 入力モードタイミング	418
図 107.	出力比較モード、OC1 のトグル	420
図 108.	エッジアライン PWM 波形（ARR=8）	421
図 109.	センターアライン PWM 波形（ARR=8）	422
図 110.	デッドタイム挿入のある相補出力	424
図 111.	負のパルスより長い遅延があるときのデッドタイムの波形	424
図 112.	正のパルスより長い遅延があるときのデッドタイムの波形	424
図 113.	ブレークに対する出力の動作	427
図 114.	TIMx OCxREF のクリア	428
図 115.	6 ステップ生成 COM の例（OSSR=1）	429
図 116.	ワンパルスモードの例	430
図 117.	エンコーダインタフェースモードにおけるカウンタの動作例	433
図 118.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	433
図 119.	ホールセンサインタフェースの例	435
図 120.	リセットモードの制御回路	436
図 121.	ゲートモードの制御回路	437
図 122.	トリガモードの制御回路	438
図 123.	外部クロックモード 2 + トリガモードの制御回路	439
図 124.	汎用タイマのブロック図	468
図 125.	プリスケール分周比が 1 から 2 に変化したときのカウンタのタイミング図	469
図 126.	プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図	470
図 127.	内部クロック分周比が 1 の場合のカウンタのタイミング図	471
図 128.	内部クロック分周比が 2 の場合のカウンタのタイミング図	471
図 129.	内部クロック分周比が 4 の場合のカウンタのタイミング図	471
図 130.	内部クロック分周比が N の場合のカウンタのタイミング図	472
図 131.	ARPE=0（TIMx_ARR はプリロードされない）の場合の更新イベント時のカウンタのタイミング図	472
図 132.	ARPE=1（TIMx_ARR はプリロードされる）のときの更新イベント時のカウンタのタイミング図	473
図 133.	内部クロック分周比が 1 の場合のカウンタのタイミング図	474
図 134.	内部クロック分周比が 2 の場合のカウンタのタイミング図	474
図 135.	内部クロック分周比が 4 の場合のカウンタのタイミング図	474
図 136.	内部クロック分周比が N の場合のカウンタのタイミング図	475
図 137.	更新イベント時のカウンタのタイミング図	475
図 138.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	476
図 139.	内部クロック分周比が 2 の場合のカウンタのタイミング図	477
図 140.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	477
図 141.	内部クロック分周比が N の場合のカウンタのタイミング図	477
図 142.	ARPE=1（カウンタアンダーフロー）の場合の更新イベント時、カウンタタイミング図	478
図 143.	ARPE=1（カウンタオーバーフロー）の場合の更新イベント時のカウンタのタイミング図	478
図 144.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	479
図 145.	TI2 外部クロックの接続例	480
図 146.	外部クロックモード 1 の制御回路	481
図 147.	外部トリガ入力ブロック	481
図 148.	外部クロックモード 2 の制御回路	482
図 149.	キャプチャ／比較チャンネル（例：チャンネル 1 入力ステージ）	483

図 150.	キャプチャ／比較チャネル 1 メイン回路	483
図 151.	キャプチャ／比較チャネル (チャネル 1) の出力ステージ	484
図 152.	PWM 入力モードタイミング	486
図 153.	出力比較モード、OC1 の反転	487
図 154.	エッジアライン PWM 波形 (ARR=8)	488
図 155.	センターアライン PWM 波形 (ARR=8)	490
図 156.	ワンパルスモードの例	491
図 157.	TIMx OCxREF のクリア	493
図 158.	エンコーダインタフェースモードにおけるカウンタの動作例	494
図 159.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	495
図 160.	リセットモードの制御回路	496
図 161.	ゲートモードの制御回路	497
図 162.	トリガモードの制御回路	497
図 163.	外部クロックモード 2 + トリガモードの制御回路	498
図 164.	マスタ／スレーブタイマの例	499
図 165.	タイマ 1 の OC1REF によるタイマ 2 のゲート操作	500
図 166.	タイマ 1 の有効化によるタイマ 2 のゲート操作	501
図 167.	タイマ 1 の更新によるタイマ 2 のトリガ	502
図 168.	タイマ 1 の有効化によるタイマ 2 のトリガ	502
図 169.	タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ	504
図 170.	汎用 タイマのブロック図 (TIM9 と TIM12)	529
図 171.	汎用タイマのブロック図 (TIM10/11/13/14)	530
図 172.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	532
図 173.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	532
図 174.	内部クロック分周比が 1 の場合のカウンタのタイミング図	533
図 175.	内部クロック分周比が 2 の場合のカウンタのタイミング図	534
図 176.	内部クロック分周比が 4 の場合のカウンタのタイミング図	534
図 177.	内部クロック分周比が N の場合のカウンタのタイミング図	534
図 178.	ARPE=0 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	535
図 179.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	535
図 180.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	536
図 181.	TI2 外部クロックの接続例	537
図 182.	外部クロックモード 1 の制御回路	538
図 183.	キャプチャ／比較チャネル (例: チャネル 1 入力ステージ)	538
図 184.	キャプチャ／比較チャネル 1 メイン回路	539
図 185.	キャプチャ／比較チャネル (チャネル 1) の出力ステージ	539
図 186.	PWM 入力モードタイミング	541
図 187.	出力比較モード、OC1 のトグル	543
図 188.	エッジアライン PWM 波形 (ARR=8)	544
図 189.	ワンパルスモードの例	545
図 190.	リセットモードの制御回路	547
図 191.	ゲートモードの制御回路	548
図 192.	トリガモードの制御回路	548
図 193.	基本タイマブロック図	575
図 194.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	577
図 195.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	577
図 196.	内部クロック分周比が 1 の場合のカウンタのタイミング図	578
図 197.	内部クロック分周比が 2 の場合のカウンタのタイミング図	579
図 198.	内部クロック分周比が 4 の場合のカウンタのタイミング図	579
図 199.	内部クロック分周比が N の場合のカウンタのタイミング図	580

図 200.	ARPE=0 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない).....	580
図 201.	ARPE=1 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR がプリロードされる).....	581
図 202.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図.....	582
図 203.	独立型ウォッチドッグのブロック図.....	590
図 204.	ウォッチドッグのブロック図.....	595
図 205.	ウィンドウ型ウォッチドッグのタイミング図.....	597
図 206.	RTC ブロック図.....	603
図 207.	FMPI2C ブロック図.....	642
図 208.	I2C バスプロトコル.....	644
図 209.	セットアップおよびホールドタイミング.....	646
図 210.	FMPI2C 初期化フローチャート.....	649
図 211.	データ受信.....	650
図 212.	データ送信.....	651
図 213.	スレーブ初期化フローチャート.....	654
図 214.	FMPI2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0).....	656
図 215.	FMPI2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1).....	657
図 216.	FMPI2C スレーブトランスミッタの転送バス図.....	658
図 217.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0).....	659
図 218.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1).....	660
図 219.	FMPI2C スレーブレシーバの転送バス図.....	660
図 220.	マスタクロック生成.....	662
図 221.	マスタ初期化フローチャート.....	664
図 222.	HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス.....	664
図 223.	HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス.....	665
図 224.	N≤255 バイトの場合の FMPI2C マスタトランスミッタの転送シーケンスフローチャート.....	666
図 225.	N>255 バイトの場合の FMPI2C マスタトランスミッタの転送シーケンスフローチャート.....	667
図 226.	FMPI2C マスタトランスミッタの転送バス図.....	668
図 227.	N≤255 バイトの場合の FMPI2C マスタレシーバの転送シーケンスフローチャート.....	670
図 228.	N>255 バイトの場合の FMPI2C マスタレシーバの転送シーケンスフローチャート.....	671
図 229.	FMPI2C マスタレシーバの転送バス図.....	672
図 230.	t _{LOW:SEXT} 、t _{LOW:MEXT} のタイムアウト間隔.....	675
図 231.	N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート.....	679
図 232.	SMBus スレーブトランスミッタの転送バス図 (SBC=1).....	680
図 233.	N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート.....	681
図 234.	SMBus スレーブレシーバのバス転送図 (SBC=1).....	682
図 235.	SMBus マスタトランスミッタのバス転送図.....	683
図 236.	SMBus マスタレシーバのバス転送図.....	685
図 237.	FMPI2C 割り込みマッピング図.....	690
図 238.	I2C バスプロトコル.....	709
図 239.	I2C ブロック図.....	710
図 240.	スレーブトランスミッタの転送シーケンス.....	712
図 241.	スレーブ受信の転送シーケンス図.....	713
図 242.	マスタ送信の転送シーケンス図.....	716
図 243.	マスタ受信の転送シーケンス図.....	718
図 244.	I2C 割り込みマッピング図.....	727
図 245.	USART のブロック図.....	746
図 246.	ワード長のプログラミング.....	747
図 247.	設定可能なストップビット.....	749
図 248.	送信時の TC/TXE の動作.....	750
図 249.	16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出.....	751

図 250.	データサンプリング (16 倍のオーバーサンプリング).....	754
図 251.	データサンプリング (8 倍のオーバーサンプリング).....	754
図 252.	アイドルライン検出を使用したミュートモード.....	767
図 253.	アドレスマーク検出を使用したミュートモード.....	767
図 254.	LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1).....	770
図 255.	LIN モードでのブレーク検出とフレーミングエラー検出.....	771
図 256.	USART の同期送信の例.....	772
図 257.	USART データクロックタイミング図 (M=0).....	772
図 258.	USART データクロックタイミング図 (M=1).....	773
図 259.	RX データセットアップ/ホールド時間.....	773
図 260.	ISO 7816-3 非同期プロトコル.....	774
図 261.	ストップビット 1.5 個を使用したパリティエラー検出.....	775
図 262.	IrDA SIR ENDEC- ブロック図.....	777
図 263.	IrDA データ変調 (3/16) - 通常モード.....	777
図 264.	DMA を使用した送信.....	779
図 265.	DMA を使用した受信.....	780
図 266.	2 つの USART 間のハードウェアフロー制御.....	780
図 267.	RTS フロー制御.....	781
図 268.	CTS フロー制御.....	781
図 269.	USART 割り込みマッピング図.....	783
図 270.	SPI ブロック図.....	797
図 271.	全二重シングルマスタ/シングルスレーブアプリケーション.....	798
図 272.	半二重シングルマスタ/シングルスレーブアプリケーション.....	799
図 273.	単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ).....	800
図 274.	マスタと 3 つの独立したスレーブ.....	801
図 275.	マルチマスタアプリケーション.....	802
図 276.	ハードウェア/ソフトウェアスレーブ選択管理.....	803
図 277.	データクロックのタイミング図.....	805
図 278.	マスタ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合.....	808
図 279.	スレーブ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合.....	809
図 280.	DMA を使用した送信.....	811
図 281.	DMA を使用した受信.....	812
図 282.	TI モードでの転送.....	815
図 283.	I ² S ブロック図.....	818
図 284.	I ² S 全二重のブロック図.....	819
図 285.	フィリップス I ² S プロトコルの波形 (16/32 ビットフル精度、CPOL = 0).....	820
図 286.	フィリップス I ² S 規格の波形 (24 ビットフレーム、CPOL = 0).....	821
図 287.	0x8EAA33 の送信.....	821
図 288.	0x8EAA33 の受信.....	821
図 289.	I ² S フィリップス標準 (32 ビットパケットフレームに拡張された 16 ビット、CPOL = 0).....	822
図 290.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例.....	822
図 291.	MSB 詰め 16 ビットまたは 32 ビットフル精度長、CPOL = 0.....	822
図 292.	MSB 詰め 24 ビットフレーム長、CPOL = 0.....	823
図 293.	MSB 詰め 16 ビットを 32 ビットパケットフレームに拡張、CPOL = 0.....	823
図 294.	LSB 詰め 16 ビットまたは 32 ビットフル精度、CPOL = 0.....	823
図 295.	LSB 詰め 24 ビットフレーム長、CPOL = 0.....	824
図 296.	0x3478AE を送信するために必要な動作.....	824
図 297.	0x3478AE の受信に必要な動作.....	824
図 298.	LSB 詰め 16 ビットを 32 ビットパケットフレームに拡張、CPOL = 0.....	825

図 299.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例	825
図 300.	PCM 規格の波形 (16 ビット)	826
図 301.	PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)	826
図 302.	オーディオサンプリング周波数の定義	827
図 303.	I ² S クロックジェネレータのアーキテクチャ	827
図 304.	「no response」と「no data」操作	847
図 305.	(複数) ブロック読み出し操作	847
図 306.	(複数) ブロック書き込み操作	847
図 307.	連続読み出し操作	848
図 308.	連続書き込み操作	848
図 309.	SDIO ブロック図	848
図 310.	SDIO アダプタ	850
図 311.	制御ユニット	851
図 312.	SDIO_CK クロックの位相反転 (BYPASS = 0)	851
図 313.	SDIO アダプタコマンドパス	852
図 314.	コマンドパスステートマシン (SDIO)	853
図 315.	SDIO コマンド転送	854
図 316.	データパス	856
図 317.	データパスステートマシン (DPSM)	857
図 318.	CAN ネットワークのトポロジ	905
図 319.	デュアル CAN ブロック図	906
図 320.	bxCAN 動作モード	908
図 321.	bxCAN (サイレントモード)	909
図 322.	bxCAN (ループバックモード)	909
図 323.	bxCAN (組み合わせモード)	910
図 324.	送信メールボックスの状態	912
図 325.	受信 FIFO の状態	913
図 326.	フィルタバンクのスケール設定 - レジスタ構造	916
図 327.	フィルタ番号の例	917
図 328.	フィルタのメカニズム - 例	918
図 329.	CAN エラー状態図	919
図 330.	ビットタイミング	921
図 331.	CAN フレーム	922
図 332.	イベントフラグと割り込み生成	923
図 333.	CAN メールボックスレジスタ	935
図 334.	OTG フルスピードブロック図	952
図 335.	OTG_FS A-B デバイス接続	954
図 336.	USB_FS ペリフェラル専用接続	955
図 337.	USB_FS ホスト専用接続	960
図 338.	SOF 接続性 (TIM および ITR1 接続への SOF トリガ出力)	964
図 339.	OTG_HFIR の動的更新	966
図 340.	デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	967
図 341.	ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	968
図 342.	割り込みの階層	972
図 343.	送信 FIFO 書き込みタスク	1052
図 344.	受信 FIFO 読み出しタスク	1053
図 345.	通常のバルク/コントロール OUT/SETUP	1054
図 346.	バルク/コントロール IN トランザクション	1058
図 347.	通常インタラプト OUT	1060
図 348.	通常インタラプト IN	1064
図 349.	アイソクロナス OUT トランザクション	1066
図 350.	アイソクロナス IN トランザクション	1069

図 351.	受信 FIFO パケット読み出し.....	1074
図 352.	SETUP パケットの処理.....	1076
図 353.	バルク OUT トランザクション.....	1082
図 354.	TRDT 最大タイミングのケース.....	1091
図 355.	A デバイスの SRP.....	1092
図 356.	B デバイスの SRP.....	1093
図 357.	A デバイスの HNP.....	1094
図 358.	B デバイスの HNP.....	1095
図 359.	ブロック図 - STM32 MCU および FPU 搭載 Cortex®-M4 レベルのデバッグサポート.....	1097
図 360.	SWJ デバッグポート.....	1099
図 361.	JTAG TAP 接続.....	1103
図 362.	TPIU ブロック図.....	1122

1 このマニュアルにおける表記の規則

1.1 レジスタに関する略記

レジスタの説明では、次の略記が使用されます。

読み出し／ 書き込み (rw)	このビットは、ソフトウェアによる読み出しと書き込みができます。
読み出し専用 (r)	このビットは、ソフトウェアによる読み出しのみが可能です。
書き込み専用 (w)	このビットは、ソフトウェアによる書き込みのみが可能です。ビットを読み出すと、リセット値が返されます。
読み出し／ クリア (rc_w1)	このビットは、ソフトウェアによって読み出すことができ、“1”を書き込むことによってクリアできます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／ クリア (rc_w0)	このビットは、ソフトウェアによって読み出すことができ、“0”を書き込むことによってクリアできます。“1”を書き込んでも、ビットの値は変化しません。
読み出し／ 読み出しによる クリア (rc_r)	このビットは、ソフトウェアによって読み出すことができます。このビットを読み出すと、自動的に“0”にクリアされます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／ セット (rs)	このビットは、ソフトウェアによって読み出しとセットができます。“0”を書き込んでも、ビットの値は変化しません。
予約済み (Res.)	予約済みビットであり、リセット値に保持する必要があります。

1.2 用語

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

- **ワード**：32 ビット長のデータ。
- **ハーフワード**：16 ビット長のデータ。
- **バイト**：8 ビット長のデータ。
- **IAP (アプリケーション内プログラミング)**：IAP は、ユーザプログラム実行中にマイクロコントローラのフラッシュメモリを再プログラムする機能です。
- **ICP (インサーキットプログラミング)**：ICP は、ユーザアプリケーションボードにデバイスが搭載された状態で、JTAG プロトコルや SWD プロトコルまたはブートローダを用いて、マイクロコントローラのフラッシュメモリをプログラムする機能です。
- **オプションバイト**：フラッシュメモリに格納された製品設定ビット。
- **AHB**：アドバンスドハイパフォーマンスパス。

1.3 使用可能なペリフェラル

すべての販売タイプで使用可能なペリフェラルとその型番については、該当するデバイスのデータシートを参照してください。



2 システムおよびメモリの概要

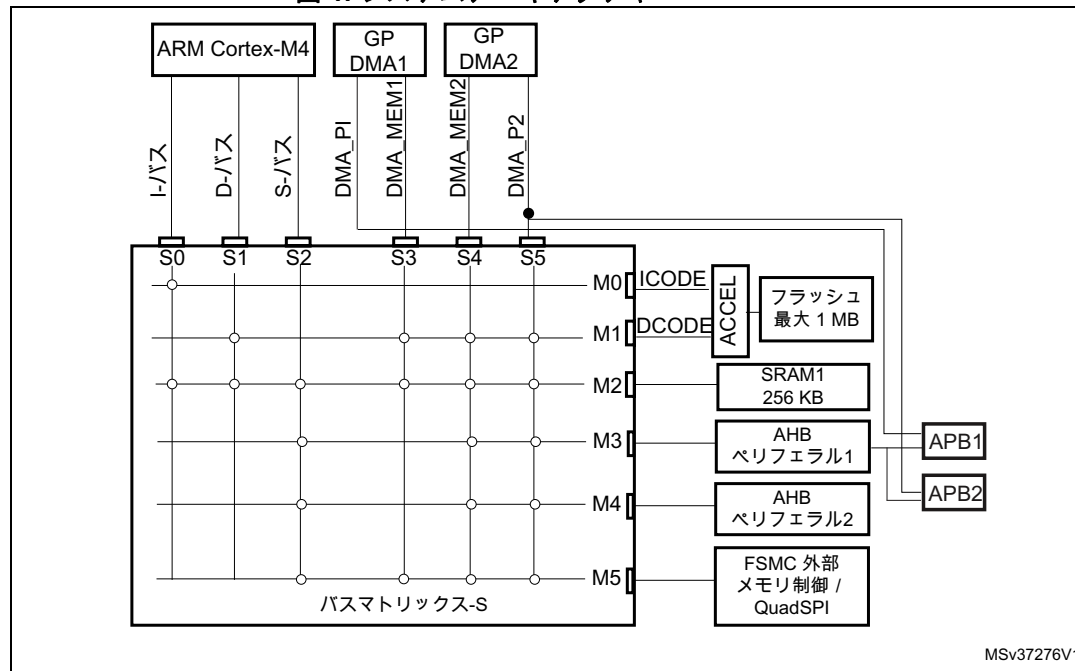
2.1 システムアーキテクチャ

STM32F412xxでは、メインシステムは、以下のものを相互接続する 32 ビットマルチレイヤ AHB バスマトリックスで構成されています。

- 6つのマスタ：
 - FPU 搭載 Cortex®-M4 コアの I-バス、D-バス、S-バス
 - DMA1 メモリバス
 - DMA2 メモリバス
 - DMA2 ペリフェラルバス
- 6つのスレーブ：
 - 内蔵フラッシュメモリ ICode バス
 - 内蔵フラッシュメモリ DCode バス
 - メイン内蔵 SRAM
 - AHB-APBブリッジおよび APB ペリフェラルを含む AHB1 ペリフェラル
 - AHB2 ペリフェラル
 - FSMC および QuadSPI

複数の高速ペリフェラルが同時動作中であっても、バスマトリックスによって、マスタからスレーブへのアクセスが可能になるとともに、同時アクセスや効率的な動作を行うことが可能になっています。このアーキテクチャを図 1 に示します。

図 1. システムアーキテクチャ



2.1.1 I-バス

このバスによって、FPU 搭載 Cortex®-M4 コアの命令バスがバスマトリックスに接続されます。コアは、このバスを用いて命令をフェッチします。このバスの対象は、コードを保持するメモリです（内蔵フラッシュメモリ／SRAM1）。

2.1.2 D-バス

このバスによって、FPU 搭載 Cortex®-M4 コアのデータバスがバスマトリックスに接続されます。このバスは、コアによってリテラルロードとデバッグアクセスに使用されます。このバスの対象は、コードまたはデータを保持するメモリです（内蔵フラッシュメモリ／SRAM1）。

2.1.3 S-バス

このバスによって、FPU 搭載 Cortex®-M4 コアのシステムバスがバスマトリックスに接続されます。このバスを使用してペリフェラルまたは SRAM1 内のデータにアクセスします。このバス上で、命令をフェッチすることもできます（Icode より低効率です）。このバスの対象となるのは、内蔵 SRAM1、APB ペリフェラルを含む AHB1 ペリフェラル、AHB2 ペリフェラル、FSMC および QUADSPI を経由した外部メモリです。

2.1.4 DMA メモリバス

このバスによって、DMA メモリバスマスタインタフェースがバスマトリックスに接続されます。このバスは、DMA がメモリとの間の転送に使用します。このバスの対象となるのはデータメモリで、内蔵フラッシュメモリ、内蔵 SRAM1、さらに S4 の場合、APB ペリフェラルを含む AHB1/AHB2 ペリフェラルです。

2.1.5 DMA ペリフェラルバス

このバスによって、DMA ペリフェラルマスタバスインタフェースがバスマトリックスに接続されます。このバスは、DMA が AHB ペリフェラルへのアクセス、またはメモリ間転送を行うために使用します。このバスの対象は、AHB および APB ペリフェラルおよびデータメモリ（フラッシュメモリおよび内蔵 SRAM1）です。

2.1.6 バスマトリックス

このバスマトリックスによって、マスタ間のアクセス調停を管理します。調停には、ラウンドロビン方式を使用します。

2.1.7 AHB-APB ブリッジ（APB）

2 つの AHB/APB ブリッジである APB1 と APB2 は、AHB および 2 つの APB バス間に完全同期接続を提供し、ペリフェラル周波数の柔軟な選択を可能にしています。

APB1 および APB2 の最大周波数に関する詳細はデバイスのデータシートを、AHB および APB ペリフェラルのアドレス割り当てに関する詳細は [表 1](#) を参照してください。

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

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



2.2 メモリ構成

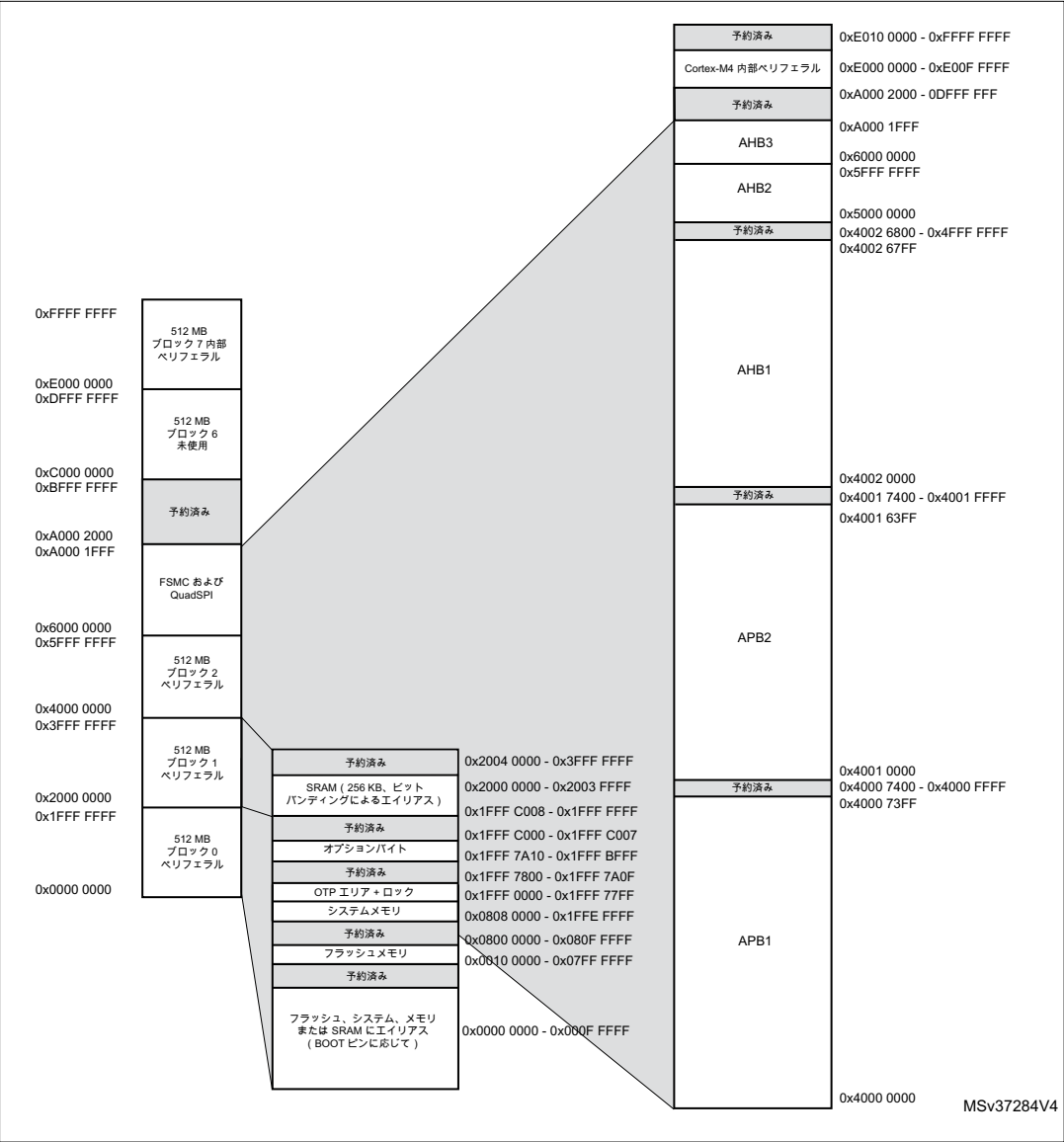
2.2.1 概要

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

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

アドレス指定可能なメモリ空間は、それぞれ 512 MB の 8 つのメインブロックに分割されています。

図 2. メモリマップ



オンチップメモリとペリフェラルに割り当てられていないメモリ領域はすべて、「予約済み」とみなされます。使用可能なメモリ領域とレジスタ領域の詳細なマッピングについては、[メモリマップとレジスタ境界アドレス](#)およびペリフェラルのセクションを参照してください。

2.2.2 メモリマップとレジスタ境界アドレス

包括的なメモリマッピング図については、お使いのデバイスに対応するデータシートを参照してください。

次の表に、デバイスで使用可能なペリフェラルの境界アドレスを示します。

表 1. レジスタ境界アドレス

バス	境界アドレス	ペリフェラル
	0xE010 0000 - 0xFFFF FFFF	予約済み
Cortex [®] -M4	0xE000 0000 - 0xE00F FFFF	Cortex-M4 内部ペリフェラル
AHB3	0xA000 2000 - 0xDFFF FFFF	予約済み
	0xA000 1000 - 0xA000 1FFF	Quad SPI 制御レジスタ
	0xA000 0000 - 0xA000 0FFF	FSMC 制御レジスタ
	0x9000 0000 - 0x9FFF FFFF	QUADSPI
	0x7000 0000 - 0x8FFF FFFF	予約済み
	0x6000 0000 - 0x6FFF FFFF	FSMC
AHB2	0x5006 0C00 - 0x5FFF FFFF	予約済み
	0x5006 0800 - 0x5006 0BFF	RNG
	0x5004 0000 - 0x5006 07FF	予約済み
	0x5000 0000 - 0x5003 FFFF	USB OTG FS
AHB1	0x4002 6800 - 0x4FFF FFFF	予約済み
	0x4002 6400 - 0x4002 67FF	DMA2
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 5000 - 0x4002 4FFFF	予約済み
	0x4002 3C00 - 0x4002 3FFF	フラッシュインタフェースレジスタ
	0x4002 3800 - 0x4002 3BFF	RCC
	0x4002 3400 - 0x4002 37FF	予約済み
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 2FFF	予約済み
	0x4002 1C00 - 0x4002 1FFF	GPIOH
	0x4002 1800 - 0x4002 1BFF	GPIOG
	0x4002 1400 - 0x4002 17FF	GPIOF
	0x4002 1000 - 0x4002 13FF	GPIOE
	0x4002 0C00 - 0x4002 0FFF	GIOD
	0x4002 0800 - 0x4002 0BFF	GPIOC
	0x4002 0400 - 0x4002 07FF	GPIOB
	0x4002 0000 - 0x4002 03FF	GPIOA

表 1. レジスタ境界アドレス（続き）

バス	境界アドレス	ペリフェラル
APB2	0x4001 6400 - 0x4001 FFFF	予約済み
	0x4001 6000 - 0x4001 63FF	DFSDM1
	0x4001 5400 - 0x4001 5FFF	予約済み
	0x4001 5000 - 0x4001 53FF	SPI5/I2S5
	0x4001 4800 - 0x4001 4BFF	TIM11
	0x4001 4400 - 0x4001 47FF	TIM10
	0x4001 4000 - 0x4001 43FF	TIM9
	0x4001 3C00 - 0x4001 3FFF	EXTI
	0x4001 3800 - 0x4001 3BFF	SYSCFG
	0x4001 3400 - 0x4001 37FF	SPI4/I2S4
	0x4001 3000 - 0x4001 33FF	SPI1/I2S1
	0x4001 2C00 - 0x4001 2FFF	SDIO
	0x4001 2400 - 0x4001 2BFF	予約済み
	0x4001 2000 - 0x4001 23FF	ADC1
	0x4001 1800 - 0x4001 1FFF	予約済み
	0x4001 1400 - 0x4001 17FF	USART6
	0x4001 1000 - 0x4001 13FF	USART1
	0x4001 0800 - 0x4001 0FFF	予約済み
	0x4001 0400 - 0x4001 07FF	TIM8
	0x4001 0000 - 0x4001 03FF	TIM1
	0x4000 7400 - 0x4000 FFFF	予約済み



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 1. レジスタ境界アドレス（続き）

バス	境界アドレス	ペリフェラル
APB1	0x4000 7000 - 0x4000 73FF	PWR
	0x4000 6C00 - 0x4000 6FFF	予約済み
	0x4000 6800 - 0x4000 6BFF	CAN2
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	I2CFMP1
	0x4000 5C00 - 0x4000 5FFF	I2C3
	0x4000 5800 - 0x4000 5BFF	I2C2
	0x4000 5400 - 0x4000 57FF	I2C1
	0x4000 4C00 - 0x4000 53FF	予約済み
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 3FFF	I2S3ext
	0x4000 3C00 - 0x4000 3FFF	SPI3/I2S3
	0x4000 3800 - 0x4000 3BFF	SPI2/I2S2
	0x4000 3400 - 0x4000 37FF	I2S2ext
	0x4000 3000 - 0x4000 33FF	IWDG
	0x4000 2C00 - 0x4000 2FFF	WWDG
	0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers
	0x4000 2400 - 0x4000 27FF	予約済み
	0x4000 2000 - 0x4000 23FF	TIM14
	0x4000 1C00 - 0x4000 1FFF	TIM13
	0x4000 1800 - 0x4000 1BFF	TIM12
	0x4000 1400 - 0x4000 17FF	TIM7
	0x4000 1000 - 0x4000 13FF	TIM6
	0x4000 0C00 - 0x4000 0FFF	TIM5
	0x4000 0800 - 0x4000 0BFF	TIM4
	0x4000 0400 - 0x4000 07FF	TIM3
	0x4000 0000 - 0x4000 03FF	TIM2

2.3 内蔵 SRAM

STM32F412xx デバイスには、256 KB のシステム SRAM を搭載しています。

内蔵 SRAM は、バイト、ハーフワード（16 ビット）、またはフルワード（32 ビット）によるアクセスが可能です。読み出し／書き込み操作は、ウェイトステート 0 個の CPU 速度で実施されます。

CPU は、SRAM からのブートが選択されている場合、または物理的な再割当てが選択されている場合に、システムバスまたは I-Code／D-Code バスを通じて内蔵 SRAM1 にアクセスできます（SYSCFG コントローラの [セクション 8.2.1: SYSCFG メモリ再配置レジスタ \(SYSCFG_MEMRMP\)](#)）。SRAM 実行のパフォーマンスを最大にするには、物理的な再割当てを選択します（ブートまたはソフトウェア選択）。

2.4 フラッシュメモリの概要

フラッシュメモリインタフェースは、フラッシュメモリへの CPU AHB I-Code および D-Code アクセスを管理します。消去を実行し、フラッシュメモリ操作そして読み出し／書き込み保護メカニズムをプログラムします。また、命令プリフェッチおよびキャッシュラインでコードの実行を加速します。

フラッシュメモリは、次のように構成されています。

- セクタに分割されたメインメモリブロック
- システムメモリブートモードでデバイスがブートするシステムメモリ
- ユーザデータ用の 512 バイト OTP（one-time programmable）
- デバイスが STANDBY または STOP モードのときに読み出し／書き込み保護、BOR レベル、ウォッチドッグソフトウェア／ハードウェアおよびリセットを設定するオプションバイト

詳細については、[セクション 3: 内蔵フラッシュメモリインタフェース](#)を参照してください。

2.5 ビットバンディング

FPU 搭載 Cortex®-M4 のメモリマップには、2 つのビットバンド領域が含まれています。これらの領域は、メモリのエイリアス領域の各ワードをメモリのビットバンド領域のビットに割り当てます。エイリアス領域のワードへの書き込みは、ビットバンド領域の対象ビットへの読み出し／変更／書き込み操作と同じ効果を持ちます。

STM32F412xx デバイスでは、ペリフェラルレジスタも SRAM1 もビットバンド領域に割り当てられますので、1 回のビットバンドの書き込みと読み出し操作が許可されます。操作は FPU 搭載 Cortex®-M4 アクセスでのみ可能であり、他のバスマスタ（DMA など）からはできません。

マッピングの計算式は、エイリアス領域の各ワードからビットバンド領域の対応するビットを参照する方法を示します。マッピング式は、次のとおりです。

$$bit_word_addr = bit_band_base + (byte_offset \times 32) + (bit_number \times 4)$$

ここで、

- *bit_word_addr* は、対象ビットに割り当てられるエイリアスメモリ領域内のワードのアドレスです。
- *bit_band_base* は、エイリアス領域の開始アドレスです。
- *byte_offset* は、対象ビットを含むビットバンド領域のバイト数です。
- *bit_number* は、対象ビットのビット位置（0-7）です。

例

次の例は、SRAM1 アドレス 0x20000300 にあるバイトのビット 2 がどのようにエイリアス領域に割り当てられるかを示します。

$$0x22006008 = 0x22000000 + (0x300 \times 32) + (2 \times 4)$$

アドレス 0x22006008 への書き込みは、SRAM1 アドレス 0x20000300 のバイトのビット 2 に対して読み出し／変更／書き込み操作を実行するのと同じ効果があります。

アドレス 0x22006008 を読み出すと、SRAM1 アドレス 0x20000300 のバイトのビット 2 の値 (0x01 または 0x00) が返されます (ビットがセットされていれば 0x01、リセットされていれば 0x00)。

ビットバンディングの詳細については、FPU 搭載 Cortex®-M4 プログラミングマニュアル ([関連ドキュメント \(1 ページ\)](#)) を参照してください。

2.6 ブート設定

固定メモリマップにより、コード領域はアドレス 0x0000 0000 から始まり (ICode/DCode バス経由でアクセス)、データ領域 (SRAM) はアドレス 0x2000 0000 から始まります (システムバス経由でアクセス)。FPU 搭載 Cortex®-M4 CPU は、常に ICode バスのリセットベクタをフェッチします。これは、ブート領域はコード領域 (一般にはフラッシュメモリ) でのみ使用可能なことを意味します。STM32F4xx マイクロコントローラは、内蔵 SRAM などの他のメモリからブートできる特殊なメカニズムを実装しています。

STM32F412xx では、[表 2](#) に示すように、BOOT[1:0] ピンによって 3 種類のブートモードを選択できます。

表 2. ブートモード

ブートモード選択ピン		ブートモード	エイリアシング
BOOT1	BOOT0		
x	0	メインフラッシュメモリ	メインフラッシュメモリがブート領域として選択されます。
0	1	システムメモリ	システムメモリがブート領域として選択されます。
1	1	内蔵 SRAM	内蔵 SRAM がブート領域として選択されます。

BOOT ピンの値は、リセット後、SYSCLK の 4 番目の立ち上りエッジでラッチされます。ユーザは、リセット後に BOOT1 と BOOT0 ピンを設定して必要なブートモードを指定します。

BOOT0 は専用ピンですが、BOOT1 は GPIO ピンと共有となっています。BOOT1 がサンプリングされると該当する GPIO ピンが解放され、別の目的に使用できるようになります。

BOOT ピンは、デバイスが STANDBY モードを終了するときにも再度サンプリングされます。したがって、これらのピンはデバイスが STANDBY モードのときでも必要なブートモード設定に保たれる必要があります。このスタートアップ遅延が終了すると、CPU はアドレス 0x0000 0000 からスタック最上位の値をフェッチし、0x0000 0004 から始まるブートメモリからコード実行を開始します。

注： **アプリケーションの初期化コードでデバイスが SRAM からブートするときには、NVIC 例外テーブルとオフセットレジスタを使用して、SRAM のベクタテーブルを再配置する必要があります。**

内蔵ブートローダ

表 3 に示すインタフェースのいずれかを使用してフラッシュメモリを再プログラムするには、内蔵ブートローダモードを使用します。使用できるインタフェースはパッケージによって異なります。

表 3. 内蔵ブートローダインタフェース

パッケージ	USART1 PA9/ PA10	USART2 PD6/ PD5	USART3 PB11/ PB10	I2C1 PB6/ PB7	I2C2 PF0/ PF1	I2C3 PA8/ PB4	I2C FMP1 PB14/ PB15	SPI1 PA4/P A5/PA 6/PA7	SPI3 PA15/ PC10/ PC11/ PC12	SPI4 PE11/ PE12/ PE13/ PE14	CAN2 PB5/ PB13	USB PA11/ P12
UFQFPN48	Y	-	-	Y	-	Y	Y	Y	-	-	Y	Y
WLCSP64	Y	-	-	Y	-	Y	Y	Y	Y	-	Y	Y
LQFP64	Y	-	-	Y	-	Y	Y	Y	Y	-	Y	Y
LQFP100	Y	Y	-	Y	-	Y	Y	Y	Y	Y	Y	Y
LQFP144	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
UFBGA100	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y
UFBGA144	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

USART ペリフェラルは内部 16 MHz オシレータ（HSI）周波数で動作しますが、CAN および USB OTG FS は、1 MHz の倍数である外部クロック（HSE）（4 から 26 MHz）が必要です。

内蔵ブートローダコードは、システムメモリに配置されます。これは生産時に ST によってプログラムされています。詳細については、アプリケーションノート AN2606 を参照してください。

STM32F412xxの物理的な再割当て

ブートピンを選択したら、アプリケーションソフトウェアはコード領域でアクセス可能なメモリを変更できます（このようにするとシステムバスの代わりに ICode バスを使ってコードを実行できます）。この変更は、SYSCFG コントローラで [セクション 8.2.1 : SYSCFG メモリ再配置レジスタ \(SYSCFG_MEMRMP\)](#) をプログラミングすることによって実行できます。

その後、次のメモリが再割当て可能となります。

- メインフラッシュメモリ
- システムメモリ
- 内蔵 SRAM

表 4. メモリ割当てとブートモード／物理的な再割当て（STM32F412xx）

アドレス	メインフラッシュメモリ でのブート／再割当て	内蔵 SRAM での ブート／再割当て	システムメモリでの ブート／再割当て
0x2000 0000 - 0x2003 FFFF	SRAM（256 KB）	SRAM（256 KB）	SRAM（256 KB）
0x1FFF 0000 - 0x1FFF 77FF	システムメモリ	システムメモリ	システムメモリ
0x0802 0000 - 0x1FFE FFFF	予約済み	予約済み	予約済み
0x0800 0000 - 0x080F FFFF	フラッシュメモリ	フラッシュメモリ	フラッシュメモリ
0x0400 000 - 0x07FF FFFF	予約済み	予約済み	予約済み
0x0000 0000 - 0x0003 FFFF ⁽¹⁾	フラッシュ（1 MB）、 エイリアス	SRAM1（256 KB）、 エイリアス	システムメモリ （30 KB）、エイリアス

1. ブートメモリ空間に割り当てられている関連メモリは、同時にメモリとしての通常のアドレスからもアクセスが可能です。

3 内蔵フラッシュメモリインタフェース

3.1 概要

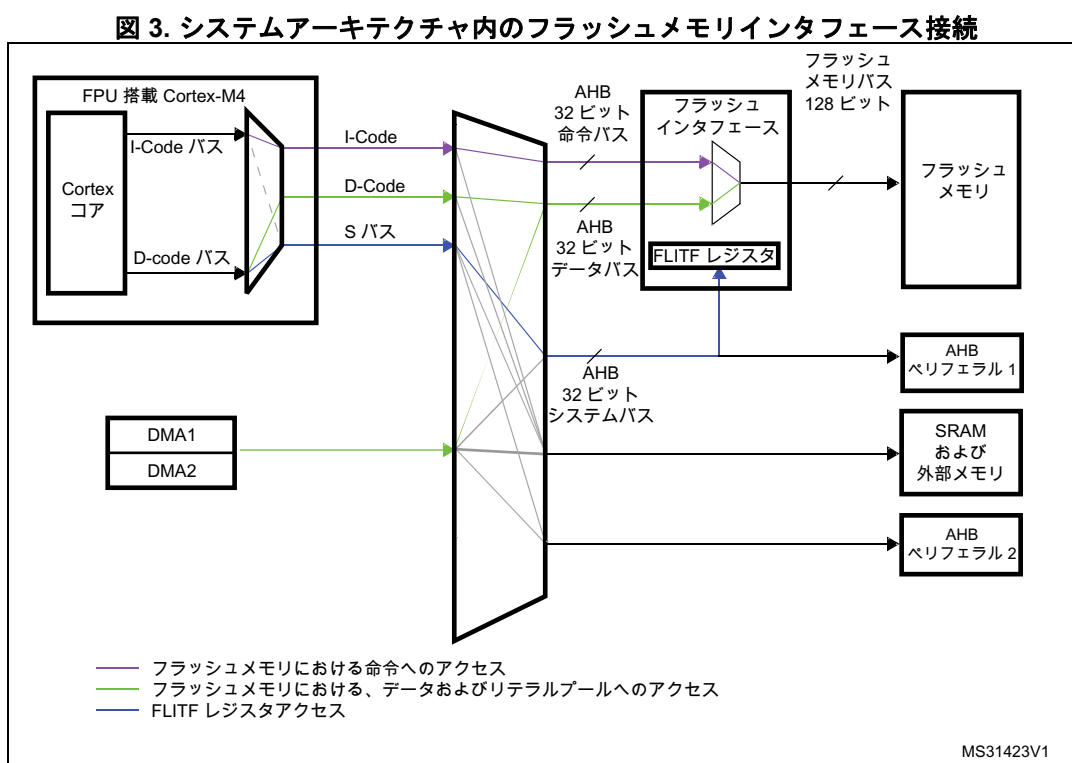
フラッシュメモリインタフェースは、フラッシュメモリへの CPU AHB I-Code および D-Code アクセスを管理します。消去を実行し、フラッシュメモリ操作そして読み出し／書き込み保護メカニズムをプログラムします。

また、命令プリフェッチおよびキャッシュラインでコードの実行を加速します。

3.2 主な特長

- フラッシュメモリ読み出し操作
- フラッシュメモリプログラム／消去操作
- 読み出し／書き込み保護
- I-Code でプリフェッチ
- I-Code で 128 ビットのキャッシュライン 64 本
- D-Code で 128 ビットのキャッシュライン 8 本

図 3 は、システムアーキテクチャ内のフラッシュメモリインタフェース接続を示しています。



3.3 内蔵フラッシュメモリ

フラッシュメモリは、以下のような主な特徴を備えています。

- 最大容量 1 MB
- 128 ビット幅のデータ読み出し
- バイト、ハーフワード、ワード、ダブルワードでの書き込み
- セクタ消去と全体消去
- メモリ構成

フラッシュメモリは、次のように構成されています。

- 16 KB のセクタ 4 つ、64 KB のセクタ 1 つ、128 KB のセクタ 7 つに分割されたメインメモリブロック
- システムメモリブートモードでデバイスがブートするシステムメモリ
- ユーザデータ用の 512 バイト OTP (one-time programmable)
1 度だけプログラム可能なエリア (OTP エリア) には、該当する OTP データブロックをロックする付加的な 32 ビットがあります。
- 読み出し/書き込み保護、BOR レベル、ウォッチドッグソフトウェア/ハードウェアおよびデバイスが STANDBY または STOP モードのときのリセットを設定するオプションバイト
- 低電力モード (詳細についてはリファレンスマニュアルの電源管理 (PWR) のセクションを参照)

表 5. フラッシュモジュールの構成

ブロック	名前	ブロックのベースアドレス	サイズ
メインメモリ	セクタ 0	0x0800 0000 - 0x0800 3FFF	16 KB
	セクタ 1	0x0800 4000 - 0x0800 7FFF	16 KB
	セクタ 2	0x0800 8000 - 0x0800 BFFF	16 KB
	セクタ 3	0x0800 C000 - 0x0800 FFFF	16 KB
	セクタ 4	0x0801 0000 - 0x0801 FFFF	64 KB
	セクタ 5	0x0802 0000 - 0x0803 FFFF	128 KB
	セクタ 6	0x0804 0000 - 0x0805 FFFF	128 KB
	セクタ 7	0x0806 0000 - 0x0807 FFFF	128 KB
	セクタ 8	0x0808 0000 - 0x0809 FFFF	128 KB
	セクタ 9	0x080A 0000 - 0x080B FFFF	128 KB
	セクタ 10	0x080C 0000 - 0x080D FFFF	128 KB
	セクタ 11	0x080E 0000 - 0x080F FFFF	128 KB
システムメモリ		0x1FFF 0000 - 0x1FFF 77FF	30 KB
OTP エリア		0x1FFF 7800 - 0x1FFF 7A0F	528 バイト
オプションバイト		0x1FFF C000 - 0x1FFF C00F	16 バイト

3.4 読み出しインタフェース

3.4.1 CPU クロック周波数とフラッシュメモリ読み出し時間との関係

データをフラッシュメモリから正しく読み出すには、CPU クロック（HCLK）の周波数およびデバイスの供給電圧に従ってフラッシュアクセス制御レジスタ（FLASH_ACR）でウェイトステート（LATENCY）の数を正しくプログラムする必要があります。

供給電圧が 2.1 V 未満のときは、プリフェッチバッファを無効にする必要があります。ウェイトステートと CPU クロック周波数との対応は、表 6 に記載されています。

- VOS[1:0] = 0x01 のとき、f_{HCLK} の最大値 = 64 MHz
- VOS[1:0] = 0x10 のとき、f_{HCLK} の最大値 = 84 MHz
- VOS[1:0] = 0x11 のとき、f_{HCLK} の最大値 = 100 MHz

表 6. CPU クロック（HCLK）周波数によるウェイトステート数

ウェイトステート (WS) (LATENCY)	HCLK (MHz)			
	電圧範囲 2.7 V - 3.6 V	電圧範囲 2.4 V - 2.7 V	電圧範囲 2.1 V - 2.4 V	電圧範囲 1.7 V - 2.1 V
0 WS (1 CPU サイクル)	0 < HCLK ≤ 30	0 < HCLK ≤ 24	0 < HCLK ≤ 18	0 < HCLK ≤ 16
1 WS (2 CPU サイクル)	30 < HCLK ≤ 64	24 < HCLK ≤ 48	18 < HCLK ≤ 36	16 < HCLK ≤ 32
2 WS (3 CPU サイクル)	64 < HCLK ≤ 90	48 < HCLK ≤ 72	36 < HCLK ≤ 54	32 < HCLK ≤ 48
3 WS (4 CPU サイクル)	90 < HCLK ≤ 100	72 < HCLK ≤ 96	54 < HCLK ≤ 72	48 < HCLK ≤ 64
4 WS (5 CPU サイクル)	-	96 < HCLK ≤ 100	72 < HCLK ≤ 90	64 < HCLK ≤ 80
5 WS (6 CPU サイクル)	-	-	90 < HCLK ≤ 100	80 < HCLK ≤ 96
6 WS (7 CPU サイクル)	-	-	-	96 < HCLK ≤ 100

リセット後、CPU クロック周波数は 16 MHz であり、FLASH_ACR レジスタでは 0 ウェイトステート（WS）が設定されます。

CPU 周波数でフラッシュメモリにアクセスするために必要なウェイトステート数の調整には、以下のソフトウェアシーケンスの使用が大いに推奨されます。

CPU 周波数の増加

1. FLASH_ACR レジスタの LATENCY ビットに新しいウェイトステート数をプログラムします。
2. FLASH_ACR レジスタを読み出してフラッシュメモリへのアクセスに新しいウェイトステート数が考慮されていることを確認します。
3. RCC_CFGR レジスタに SW ビットを書き込んで CPU クロックソースを変更します。
4. 必要であれば、RCC_CFGR レジスタに HPRE ビットを書き込んで CPU クロックプリスケアラを変更します。
5. RCC_CFGR レジスタでクロックソースステータス（SWS ビット）や AHB プリスケアラの値（HPRE ビット）を読み出して新しい CPU クロックソースや新しい CPU クロックプリスケアラの値がそれぞれ考慮されていることを確認します。



CPU 周波数の減少

1. RCC_CFGR レジスタに SW ビットを書き込んで CPU クロックソースを変更します。
2. 必要であれば、RCC_CFGR レジスタに HPRE ビットを書き込んで CPU クロックプリスケアラを変更します。
3. RCC_CFGR レジスタでクロックソースステータス (SWS ビット) や AHB プリスケアラの値 (HPRE ビット) を読み出して新しい CPU クロックソースや新しい CPU クロックプリスケアラの値がそれぞれ考慮されていることを確認します。
4. FLASH_ACR レジスタの LATENCY ビットに新しいウェイトステート数をプログラムします。
5. FLASH_ACR レジスタを読み出してフラッシュメモリへのアクセスに新しいウェイトステート数が使用されていることを確認します。

注： CPU クロック設定やウェイトステート (WS) 設定の変更は、すぐには有効にならないことがあります。AHB プリスケアラ分周比とクロックソースステータス値を確認すると、現在の CPU クロック周波数が設定した周波数であることが確認できます。FLASH_ACR レジスタを読み出すと、プログラムした WS 数が有効になっているかどうかを確認できます。

3.4.2 適応型リアルタイムメモリアクセラレータ (ART Accelerator™)

独自仕様の適応型リアルタイム (ART) メモリアクセラレータは、STM32 の業界標準 ARM® FPU 搭載 Cortex®-M4 プロセッサ用に最適化されています。これはフラッシュメモリ技術における FPU 搭載 ARM® FPU 搭載 Cortex®-M4 固有のパフォーマンス上の利点 (通常、動作周波数が高くなると、プロセッサはフラッシュメモリを待つ必要性が発生します) を調整します。

プロセッサのパフォーマンスを最大限に引き出すため、アクセラレータは 128 ビットのフラッシュメモリからのプログラム実行速度を上昇させる命令プリフェッチキューと、ブランチキャッシュを搭載しています。CoreMark ベンチマークによると、ART アクセラレータによって達成されるパフォーマンスは、最大 CPU 周波数 100 MHz におけるフラッシュメモリからの 0 ウェイトステートプログラム実行と同等となります。

命令プリフェッチ

各フラッシュメモリ読み出し操作では、起動されるプログラムによって、32 ビットの命令 4 個または 16 ビットの命令 8 個によって 128 ビットが提供されます。したがって連続コードの場合は、その前の読み出し命令ラインの実行に 4 個以上の CPU サイクルが必要となります。I-Code バスでのプリフェッチを使用すると、CPU によって現在の命令ラインが要求されている間にフラッシュメモリから次の連続命令ラインを読み出すことができます。プリフェッチは、FLASH_ACR レジスタの PRFTEN ビットをセットすると有効になります。この機能は、フラッシュメモリのアクセスに 1 つ以上のウェイトステートが必要な場合に有効です。

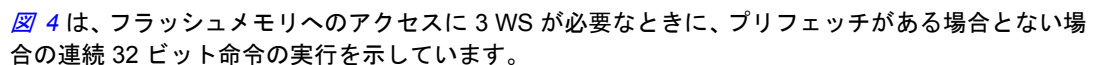
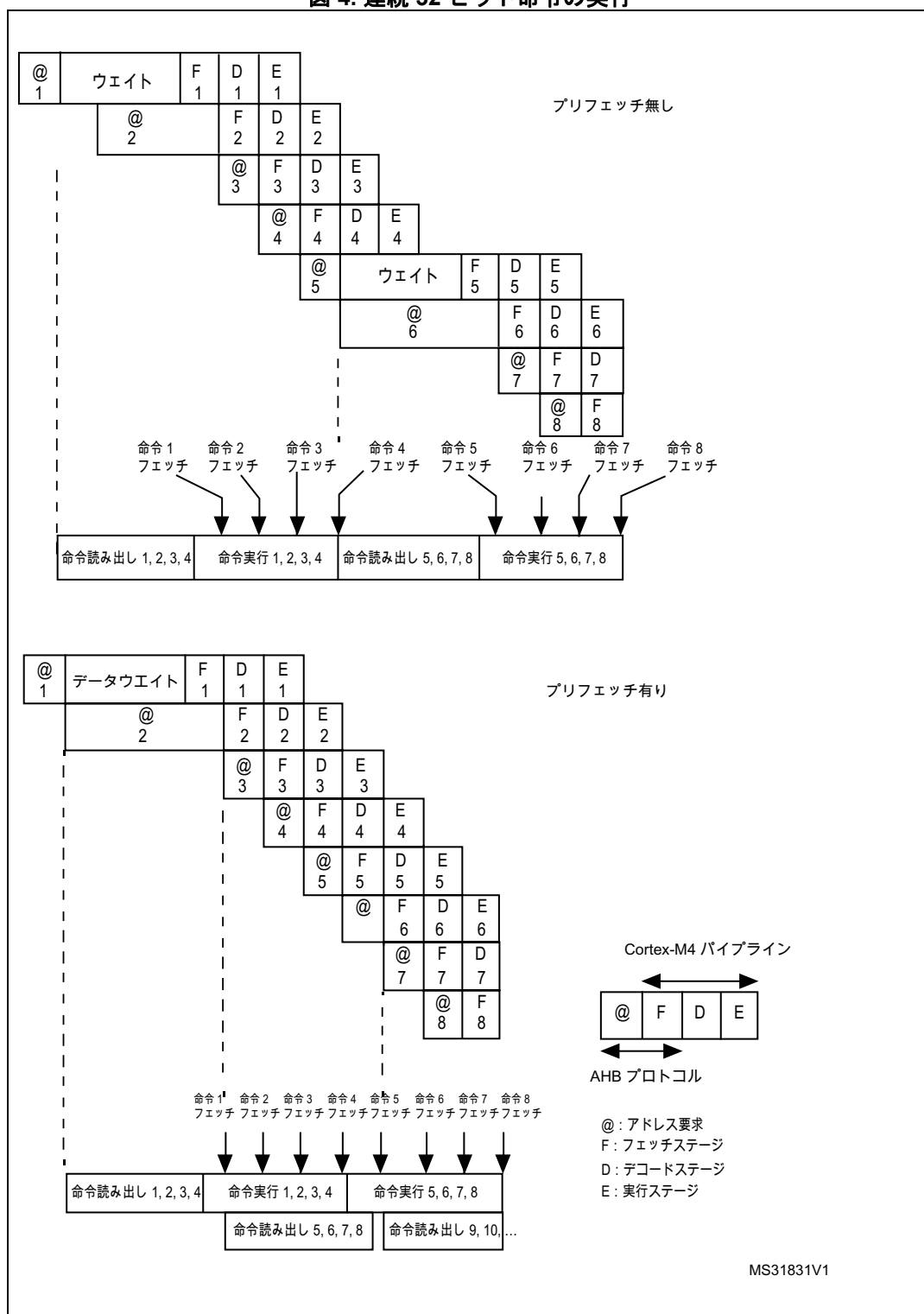
 図 4 は、フラッシュメモリへのアクセスに 3 WS が必要なときに、プリフェッチがある場合とない場合の連続 32 ビット命令の実行を示しています。

図 4. 連続 32 ビット命令の実行



コードが連続でない場合（ブランチ）、現在使用されている命令ラインやプリフェッチされた命令ラインには命令がないことがあります。この場合、サイクル数によるペナルティはウェイトステート数以上となります。

命令キャッシュメモリ

ジャンプによる時間のロスを制限するため、命令キャッシュメモリ内に 128 ビット 64 ラインを維持することができます。この機能は、FLASH_ACR レジスタで命令キャッシュ有効 (ICEN) ビットをセットすると有効にできます。失敗が発生 (現在使用している命令ライン、プリフェッチされた命令ラインまたは命令キャッシュメモリに要求されたデータがない) するたびに、ラインの読み出しが命令キャッシュメモリにコピーされます。命令キャッシュメモリに含まれるデータの中に CPU が要求するデータがある場合には、全く遅延なしにそのデータが提供されます。すべての命令キャッシュメモリラインが満たされると、LRU (最も長い時間使われていない) ポリシーを使用して命令メモリキャッシュの中で置換するラインを決定します。この機能は、ループを含むコードの場合に特に有用です。

データ管理

リテラルプールは、CPU パイプラインの実行ステージにおいて D-Code バスを通じてフラッシュメモリからフェッチされます。その結果として、CPU パイプラインは要求されるリテラルプールが提供されるまでストールされます。リテラルプールによる時間のロスを制限するため、AHB データバス DCodeからのアクセスは、AHB命令バス I-Codeからのアクセスよりも優先されます。

頻繁に使用されるリテラルプールがある場合は、FLASH_ACR レジスタのデータキャッシュ有効 (DCEN) ビットをセットすると、データキャッシュメモリを有効にできます。この機能は、命令キャッシュメモリのように動作しますが、保持されるデータのサイズは 128 ビット 8 ラインに制限されます。

注： ユーザー設定セクタのデータは、キャッシュできません。

3.5 消去操作とプログラム操作

すべてのフラッシュメモリプログラムの動作 (消去またはプログラム) では、CPU クロック周波数 (HCLK) が 1 MHz 以上である必要があります。フラッシュメモリ操作中にデバイスのリセットが発生すると、フラッシュメモリの内容は保証されません。

STM32F4xx でフラッシュメモリの書き込みまたは消去操作中にフラッシュメモリを読み出そうとすると、バスがストールされます。読み出し操作は、プログラム操作が完了すると正しく処理されます。これは、書き込み/消去操作中はコードやデータをフェッチできないことを意味します。

3.5.1 フラッシュ制御レジスタのアンロック

リセット後は、たとえば電気妨害などによって考えられる不要な操作からフラッシュメモリを保護するため、フラッシュ制御レジスタ (FLASH_CR) には書き込みません。このレジスタのアンロックには、次のシーケンスを使用します。

1. フラッシュキーレジスタ (FLASH_KEYR) に KEY1 = 0x45670123 を書き込みます
2. フラッシュキーレジスタ (FLASH_KEYR) に KEY2 = 0xCDEF89AB を書き込みます

シーケンスを誤るとバスエラーが返り、次のリセットまで FLASH_CR レジスタがロックされます。

FLASH_CR レジスタは、ソフトウェアで FLASH_CR レジスタの LOCK ビットをセットすると再びロックできます。

注： FLASH_SR レジスタの BSY ビットがセットされていると、書き込みモードでは FLASH_CR レジスタにアクセスできません。BSY ビットがセットされている状態で書き込もうとすると、BSY ビットがクリアされるまで AHB バスはストールします。



3.5.2 プログラム／消去の並列処理

並列処理サイズは、FLASH_CR レジスタの PSIZE フィールドで設定します。これはフラッシュメモリに書き込み操作が発生するたびにプログラムされるバイト数を表します。PSIZE は、供給電圧や、外部 V_{PP} 電源が使用されているかどうかによって制限されます。したがって、プログラム／消去操作の前に FLASH_CR レジスタで正しく設定しておく必要があります。

フラッシュメモリの消去操作は、セクタ単位で、またはフラッシュメモリ全体に対して実行されます（全体消去）。消去時間は、PSIZE にプログラムされている値に依存します。消去時間の詳細については、デバイスデータシートの電気的特性のセクションを参照してください。

表 7 は、正しい PSIZE 値を記載しています。

表 7. プログラム／消去の並列処理

	電圧範囲 2.7~3.6 V、外部 V _{PP}	電圧範囲 2.7 - 3.6 V	電圧範囲 2.4 - 2.7 V	電圧範囲 2.1 - 2.4 V	電圧範囲 1.7 V - 2.1 V
並列処理サイズ	x64	x32	x16		x8
PSIZE (1:0)	11	10	01		00

注： 整合しないプログラムの並列処理／電圧範囲設定を使ってプログラムまたは消去操作を開始すると、予想しない結果を招くことがあります。次の読み出し操作で実際に論理的な値がメモリに書き込まれたことがわかっていても、この値は保持されないことがあります。

V_{PP}を使用するには、外部高電圧電源（8 から 9 V の間）を V_{PP} パッドに印加する必要があります。外部電源は、DC 消費が 10 mA を超えてもこの電圧範囲を持続できる必要があります。V_{PP} の使用は、生産ラインの初期設定に制限することをお勧めします。V_{PP} 電源は、1 時間以上印加しないようにしてください。フラッシュメモリが損傷することがあります。

3.5.3 消去

フラッシュメモリの消去操作は、セクタ単位で、またはフラッシュメモリ全体に対して実行できます（全体消去）。全体消去は、OTP セクタや設定セクタには影響しません。

セクタ消去

セクタを消去するには、次の手順に従います。

- FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
- FLASH_CR レジスタで SER ビットをセットし、メインメモリブロックで 12 個あるセクタの中から消去するセクタ（SNB）を選択します。
- FLASH_CR レジスタの STRT ビットをセットします。
- BSY ビットがクリアされるのを待ちます。

全体消去

全体消去の実施には、次のシーケンスを推奨します。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの MER ビットをセットします。
3. FLASH_CR レジスタの STRT ビットをセットします。
4. BSY ビットがクリアされるのを待ちます。

注： *FLASH_CR レジスタの MERx および SER ビットがどちらもセットされていれば、全体消去が実施されます。*

MERx および SER ビットがどちらもリセットされ、STRT ビットがセットされると、エラーフラグを生成せずに予測不可能な動作を実行することがあります。このような状態は禁止する必要があります。

3.5.4 プログラミング

標準プログラミング

フラッシュメモリのプログラミングシーケンスは、次のようになっています。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のメインフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの PG ビットをセットします。
3. 指定したメモリアドレス（メインメモリブロックまたは OTP エリア内）にデータ書き込み操作を実施します。
 - x8 並列処理の場合はバイトアクセス
 - x16 並列処理の場合はハーフワードアクセス
 - x32 並列処理の場合はワードアクセス
 - x64 並列処理の場合はダブルワードアクセス
4. BSY ビットがクリアされるのを待ちます。

注： *ビットを“1”から“0”に変更するときは、消去操作を実施しなくても続けて書き込み操作を実行できます。“1”を書き込むには、フラッシュメモリ消去操作が必要です。*

消去およびプログラミング操作が同時に要求される場合には、まず消去操作が実施されます。

プログラミングエラー

128 ビットの列境界を超えるようなデータをフラッシュメモリにプログラムすることはできません。そのような場合、書き込み操作は実施されず、FLASH_SR レジスタでプログラムアライメントエラーフラグ (PGAERR) がセットされます。

書き込みアクセスタイプ（バイト、ハーフワード、ワード、ダブルワード）は、選択した並列処理タイプ（x8、x16、x32、x64）に対応している必要があります。対応していない場合、書き込み操作は実施されず、FLASH_SR レジスタでプログラム並列処理エラーフラグ (PGPERR) がセットされます。

標準プログラミングシーケンスが尊重されない場合（PG ビットがセットされていない時にフラッシュメモリアドレスに書き込もうとした場合など）、操作は中止となり、FLASH_SR レジスタでプログラムシーケンスエラーフラグ (PGSERR) がセットされます。

プログラミングとキャッシュ

フラッシュメモリ書き込みアクセスがデータキャッシュのデータに関連している場合、フラッシュ書き込みアクセスはフラッシュメモリ内のデータ、そしてキャッシュ内のデータを変更します。

フラッシュメモリ内の消去操作もデータまたは命令キャッシュ内のデータに関連している場合は、コード実行でこのデータにアクセスする前に、このデータが再度書き込まれることを確認する必要があります。これが安全に実施できない場合には、FLASH_CR レジスタで DCRST および ICRST ビットをセットしてキャッシュを一掃することを推奨します。

注： I/D キャッシュは、無効になっている場合にのみ一掃するようにします (I/DCEN = 0)。

3.5.5 割り込み

FLASH_CR レジスタで操作終了割り込みイネーブルビット (EOPIE) をセットすると、消去またはプログラム操作の終了時、つまり FLASH_SR レジスタのビジービット (BSY) がクリアされるとき (正しいかどうかにかかわらず操作が完了したことを示す) に割り込みを発生させることができます。この場合、FLASH_SR レジスタの操作終了 (EOP) ビットがセットされます。

プログラム、消去、または読み出し操作リクエスト中にエラーが発生すると、FLASH_SR レジスタで次のいずれかのエラーフラグがセットされます。

- PGAERR、PGPERR、PGSERR (プログラムエラーフラグ)
- WRPERR (保護エラーフラグ)

この場合、FLASH_SR レジスタのエラー割り込みイネーブルビット (ERRIE) がセットされると、割り込みが生成され、FLASH_SR レジスタの操作エラービット (OPERR) がセットされます。

注： 複数のエラー (フラッシュメモリへの DMA 転送の場合など) が連続して検出されると、エラーフラグは連続した書き込みリクエストが終了するまでクリアできません。

表 8. フラッシュ割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
操作終了	EOP	EOPIE
書き込み保護エラー	WRPERR	ERRIE
プログラミングエラー	PGAERR, PGPERR, PGSERR	ERRIE

3.6 オプションバイト

3.6.1 ユーザオプションバイトの説明

オプションバイトは、アプリケーション要件によってエンドユーザが設定します。表 9 は、ユーザ設定セクタ内のこのバイトの構成を示しています。

表 9. オプションバイトの構成

アドレス	[63:16]	[15:0]
0x1FFF C0000	予約済み	ROP およびユーザオプションバイト (RDP&USER)
0x1FFF C008	予約済み	書き込み保護 nWRP ビット (セクタ 0 から 11)

表 10. ユーザオプションバイトの説明

オプションバイト（ワード、アドレス 0x1FFF C000）	
RDP : 読み出し保護オプションバイトです。 読み出し保護は、フラッシュメモリに格納されているソフトウェアコードの保護に使用します。	
ビット 15:8	0xAA : レベル 0、保護なし 0xCC : レベル 2、チップ保護（デバッグと RAM からのブート機能は無効） その他 : レベル 1、メモリの読み出し保護（デバッグ機能は制限付き）
USER : ユーザオプションバイト このバイトは次の機能の設定に使用します。 – ウォッチドッグイベントの選択 : ハードウェアまたはソフトウェア – STOP モードに入るときのリセットイベント – STANDBY モードに入るときのリセットイベント	
ビット 7	nRST_STDBY 0 : STANDBY モードに入るときにリセットを生成します。 1 : リセットは生成されません。
ビット 6	nRST_STOP 0 : STOP モードに入るときにリセットを生成します。 1 : リセットは生成されません。
ビット 5	WDG_SW 0 : ハードウェアに依存しないウォッチドッグ 1 : ソフトウェアに依存しないウォッチドッグ
ビット 4	0x1 : 未使用
ビット 3:2	BOR_LEV : BOR リセットレベル このビットには、リセットをアクティブにしたりリセットを解除したりする供給レベル閾値が含まれています。フラッシュメモリに新しい BOR レベルをプログラムするように書き込むことができます。 00 : BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3 01 : BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2 10 : BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1 11 : BOR オフ、POR/PDR リセット閾値レベルが適用されます。 注 : <i>BOR 特性の詳細については、製品データシートの「電気的特性」のセクションを参照してください。</i>
ビット 1:0	0x1 : 未使用
オプションバイト（ワード、アドレス 0x1FFF C008）	
ビット 15	SPRMOD : nWPRI ビットの保護モードの選択 0 : nWPRI ビットをセクタ i の書き込み保護に使用します（デフォルト）。 1 : nWPRI ビットをセクタ i の PCROP 保護に使用します（セクタ）。
ビット 14:12	予約済み

表 10. ユーザオプションバイトの説明

nWRP：フラッシュメモリ書き込み保護オプションバイト セクタ 0 から 11 までが書き込み保護可能です。	
ビット 11:0	nWRPi SPRMOD がリセットされている場合（デフォルト値）： 0：セクタ i で書き込み保護がアクティブになっています。 1：セクタ i で書き込み保護がアクティブになっていません。 SPRMOD がセットされている場合（アクティブ）： 0：セクタ i で PCROP 保護がアクティブになっていません。 1：セクタ i で PCROP 保護がアクティブになっています。

3.6.2 ユーザオプションバイトのプログラミング

このセクタで操作を実行するには、どんな操作であってもフラッシュオプション制御レジスタ (FLASH_OPTCR) でオプションロックビット (OPTLOCK) がクリアされている必要があります。このビットがクリアできるようにするには、次のシーケンスを実施する必要があります。

1. フラッシュオプションキーレジスタ (FLASH_OPTKEYR) で OPTKEY1 = 0x0819 2A3B を書き込みます。
2. フラッシュオプションキーレジスタ (FLASH_OPTKEYR) で OPTKEY2 = 0x4C5D 6E7F を書き込みます。

ソフトウェアによって OPTLOCK ビットをセットすると、ユーザオプションバイトを不要な消去／プログラム操作から保護することができます。

ユーザオプションバイトの変更

ユーザオプション値を変更するには、次のシーケンスに従います。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. オプション値を FLASH_OPTCR レジスタに書き込みます。
3. FLASH_OPTCR レジスタでオプションスタートビット (OPTSTRT) をセットします。
4. BSY ビットがクリアされるのを待ちます。

注： *まずユーザ設定セクタを消去し、すべてのオプションバイトを FLASH_OPTCR レジスタに含まれている値でプログラムすると、オプションの値が自動的に変更されます。*

3.6.3 読み出し保護 (RDP)

フラッシュメモリのユーザーエリアは、信頼できないコードによる読み出し操作から保護できます。読み出し保護レベルは、3 種類定義されています。

- レベル 0：読み出し保護なし
読み出し保護オプションバイト (RDP) に 0xAA を書き込んで読み出し保護レベルをレベル 0 にセットすると、フラッシュメモリに対するすべての読み出し／書き込み操作（書き込み保護がセットされていない場合）があらゆるブート設定（フラッシュユーザブート、デバッグ、RAM からのブート）で可能となります。

- レベル 1：読み出し保護有効

オプションバイト消去後のデフォルトの読み出し保護レベルです。読み出し保護レベル 1 は、RDP オプションバイトに何らかの値（それぞれレベル 0 およびレベル 2 をセットする 0xAA および 0xCC を除く）を書き込むとアクティブになります。読み出し保護レベル 1 をセットすると次のようになります。

- デバッグ機能が接続されている場合や RAM またはシステムメモリブートローダからのブート中は、フラッシュメモリへの一切のアクセス（読み出し、消去、プログラム）は実施できなくなります。読み出しリクエストの場合には、バスエラーが生成されます。
- フラッシュメモリからブートする場合は、ユーザコードからのフラッシュメモリへのアクセス（読み出し、消去、プログラム）が許可されます。

レベル 1 がアクティブのときに保護オプションバイト（RDP）をレベル 0 にセットすると、フラッシュメモリが全体消去されます。その結果、読み出し保護が解除される前にユーザコードエリアがクリアされます。全体消去は、ユーザコードエリアのみを消去します。書き込み保護などのその他のオプションバイトは、全体消去操作前と変わりません。OTP エリアは全体消去に影響されませんので変化しません。全体消去は、レベル 1 がアクティブのときにレベル 0 がリクエストされたときにのみ実施されます。保護レベルが引き上げられた場合には（0 から 1、1 から 2、0 から 2）、全体消去は実施されません。

- レベル 2：デバッグ／チップ読み出し無効

RDP オプションバイトに 0xCC を書き込むと、読み出し保護レベル 2 がアクティブになります。読み出し保護レベル 2 をセットすると次のようになります。

- レベル 1 によるすべての保護がアクティブになります。
- RAM やシステムメモリブートローダからのブートも許可されなくなります。
- JTAG、SWV（シリアルワイヤビュア）、ETM、バウンダリスキャンが無効になります。
- ユーザオプションバイトが変更できなくなります。
- フラッシュメモリからブートする場合は、ユーザコードからのフラッシュメモリへのアクセス（読み出し、消去、プログラム）が許可されます。

メモリ読み出し保護レベル 2 は、不可逆操作です。レベル 2 がアクティブのときは、保護レベルをレベル 0 やレベル 1 に下げることはできません。

注： **レベル 2 がアクティブのときは、JTAG ポートは恒久的に無効になります（JTAG ヒューズとして作用）。結果として、バウンダリスキャンは実施できません。ST マイクロエレクトロニクスは、レベル 2 保護がセットされている不良部品には分析を実施できません。**

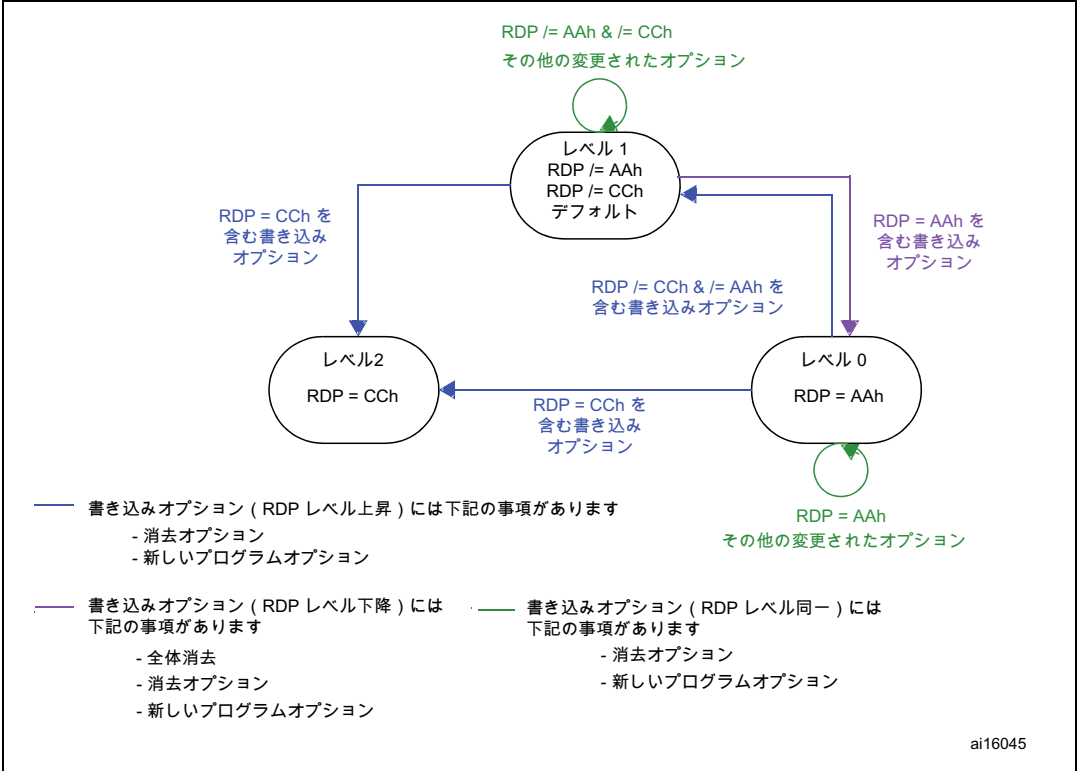
表 11. アクセスと読み出し保護レベル

メモリ領域	保護レベル	デバッグ機能、RAM またはシステム メモリブートローダからのブート			フラッシュメモリからのブート		
		読み出し	書き込み	消去	読み出し	書き込み	消去
メインフラッシュメモリ	レベル1	なし		なし ⁽¹⁾	あり		
	レベル2	なし			あり		
オプションバイト	レベル1	あり			あり		
	レベル2	なし			なし		
OTP	レベル1	なし		NA	あり		NA
	レベル2	なし		NA	あり		NA

1. メインフラッシュメモリは、RDP がレベル 1 から 0 に変化するときのみ消去されます。OTP エリアは変化しません。

図 5 は、ある RDP レベルから別の RDP レベルへの遷移を示しています。

図 5. RDP レベル



3.6.4 書き込み保護

フラッシュメモリでは、最大 12 個までのユーザセクタをプログラムカウンタコンテキストの損失による不要な書き込み操作から保護できます。FLASH_OPTCR レジスタの書き込み保護 $nWRPi$ ビット ($0 \leq i \leq 11$) がローのときは、該当するセクタを消去したりプログラムしたりすることができません。結果として、セクタが 1 つ書き込み保護されていると全体消去が実施できません。

フラッシュメモリの書き込み保護されている部分（書き込み保護ビットで保護されているセクタ、OTP のロックされている部分や ICP のように書き込むことが絶対にできないフラッシュメモリの部分）に消去／プログラム操作を実施しようとする、FLASH_SR レジスタで書き込み保護エラーフラグ (WRPERR) がセットされます。

注： *メモリ読み出し保護レベルが選択されると (RDP レベル = 1)、CPU デバッグ機能が接続されている場合 (JTAG やシリアルワイヤ) やブートコードが RAM から実行されている場合には、 $nWRPi = 1$ のときでもフラッシュメモリセクタ i をプログラムしたり消去したりすることはできません。*

書き込み保護エラーフラグ

フラッシュメモリの書き込み保護されているエリアに消去／プログラム操作を実施すると、FLASH_SR レジスタで書き込み保護エラーフラグ (WRPERR) がセットされます。

消去操作がリクエストされると、次のような場合に WRPERR ビットがセットされます。

- 全体消去、セクタ消去が設定されている場合 (MER または MER/MER1 および SER = 1)
- セクタ消去がリクエストされ、セクタ番号 SNB フィールドが有効でない場合
- オプションビット (FLASH_OPTCRx レジスタの MER または MER/MER1 = 1 および $nWRPi = 0$, $0 \leq i \leq 11$ ビット) によって 1 つ以上のユーザセクタが書き込み保護されているときに全体消去がリクエストされた場合
- 書き込み保護されているセクタでセクタ消去がリクエストされた場合 (FLASH_OPTCRx レジスタで SER = 1、SNB = i および $nWRPi = 0$, $0 \leq i \leq 11$ ビット)
- フラッシュメモリが読み出し保護されており、侵入が検出された場合

プログラム操作がリクエストされると、次のような場合に WRPERR ビットがセットされます。

- システムメモリやユーザ指定のセクタの予約済みの部分に書き込み操作が実施された場合
- ユーザ設定セクタに書き込み操作を実施した場合
- オプションビットで書き込み保護されているセクタに書き込み操作を実施した場合
- 既にロックされている OTP エリアに書き込み操作がリクエストされた場合
- フラッシュメモリが読み出し保護されており、侵入が検出された場合

3.6.5 独自仕様コード読み出し保護 (PCROP)

フラッシュメモリユーザセクタ (0 から 11) は、独自仕様の読み出し保護 (PCROP) を使って D-バス読み出しアクセスから保護することができます。

PCROP 保護は、FLASH_CR レジスタの SPRMOD オプションビットを使って次のように選択されます。

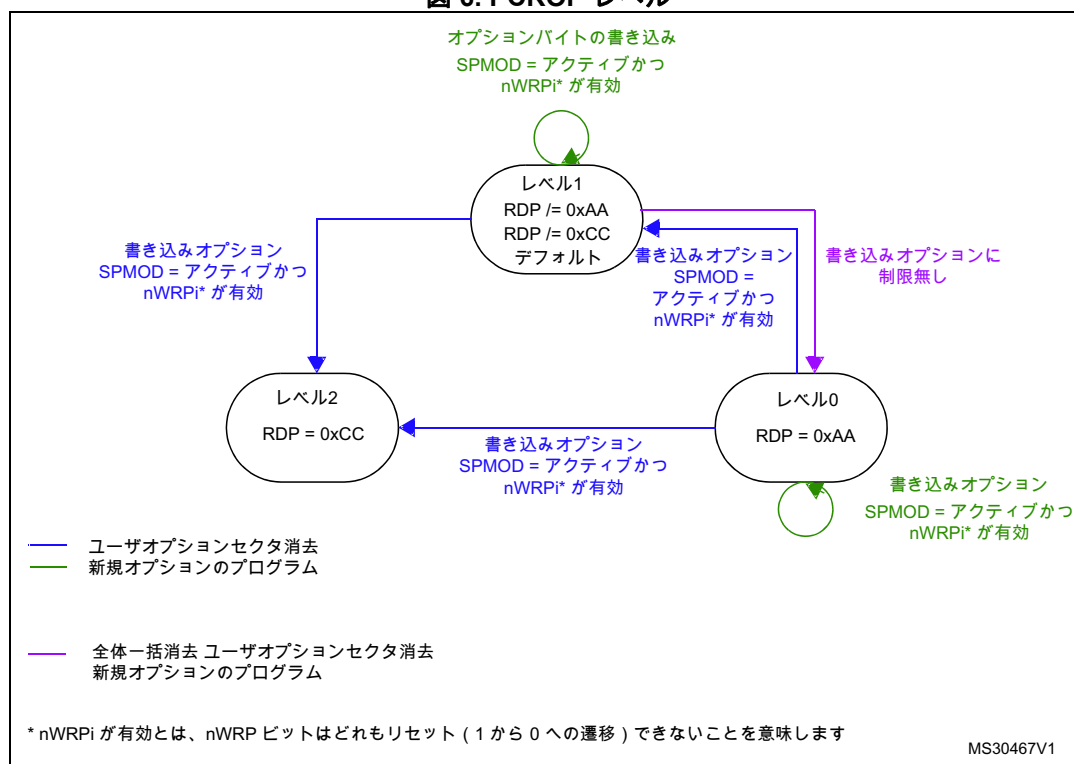
- SPRMOD = 0 : $nWRPi$ が各ユーザセクタの書き込み保護を制御します。
- SPRMOD = 1 : $nWRPi$ が各ユーザセクタの読み出し／書き込み保護 (PCROP) を制御します。

セクタが読み出し保護 (PCROP モードがアクティブ) されているときは、フラッシュインタフェースで ICODE バスを使ってコードフェッチする場合にのみアクセスできます。

- D-バスからの読み出しアクセスは、すべて RDERR フラグエラーとなります。
- PCROP で保護されているセクタでのプログラム／消去操作は、すべて WRPERR フラグエラーとなります。



図 6. PCROP レベル



SPRMOD の非アクティブ化や PCROP で保護されているユーザインタフェースの保護解除は、RDP が 1 から 0 に変化した場合にのみ可能です。この条件が順守されない場合、ユーザオプションバイトの変更は取り消しになり、書き込みエラー WRPERR フラグがセットされます。アクティブな nWRPi ビットが 1 つもリセットされておらず、SPRMOD がアクティブなままであるため、ユーザオプションバイト (BOR_LEV、RST_STDBY など) は変更できません。

注： **PCROP モードがアクティブ (SPRMOD = 1) のときは、nWRPi ビットのアクティブな値が反転します。**

3.7 OTP（One-time programmable）バイト

表 12 は、OTP エリアの構成を示しています。

表 12. OTP エリアの構成

ブロック	[128:96]	[95:64]	[63:32]	[31:0]	アドレスバイト 0
0	OTP0	OTP0	OTP0	OTP0	0x1FFF 7800
	OTP0	OTP0	OTP0	OTP0	0x1FFF 7810
1	OTP1	OTP1	OTP1	OTP1	0x1FFF 7820
	OTP1	OTP1	OTP1	OTP1	0x1FFF 7830
.	.				.
.	.				.
.	.				.
15	OTP15	OTP15	OTP15	OTP15	0x1FFF 79E0
	OTP15	OTP15	OTP15	OTP15	0x1FFF 79F0
ロック ブロック	LOCKB15 ...LOCKB12	LOCKB11 ...LOCKB8	LOCKB7 ...LOCKB4	LOCKB3 ...LOCKB0	0x1FFF 7A00

OTP エリアは、32 バイトの OTP データブロック 16 個と、16 バイトのロック OTP ブロック 1 個に分割されています。OTP データブロックおよびロックブロックは、消去できません。ロックブロックには、該当する OTP データブロック（ブロック 0 から 15）をロックする、16 バイトの LOCKBi（ $0 \leq i \leq 15$ ）が含まれています。各 OTP データブロックは、該当する OTP ロックバイトで値 0x00 がプログラムされるまでプログラムできます。ロックバイトには、0x00 および 0xFF の値が含まれている必要があります。これらの値が含まれていない場合、OTP バイトが正しく考慮されないことがあります。

3.8 フラッシュインタフェースレジスタ

3.8.1 フラッシュアクセス制御レジスタ (FLASH_ACR)

フラッシュアクセス制御レジスタは、加速機能を有効／無効にし、CPU 周波数に従って フラッシュメモリアクセス時間を制御するために使用されます。

アドレスオフセット : 0x00
リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DCRST	ICRST	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY			
			rW	W	rW	rW	rW					rW	rW	rW	rW

ビット 31:13 予約済み、クリア状態を保つ必要があります。

ビット 12 **DCRST** : データキャッシュのリセット

- 0 : データキャッシュはリセットされません。
 - 1 : データキャッシュはリセットされます。
- このビットに書き込めるのは、D キャッシュが無効のときだけです。

ビット 11 **ICRST** : 命令キャッシュのリセット

- 0 : 命令キャッシュはリセットされません。
 - 1 : 命令キャッシュはリセットされます。
- このビットに書き込めるのは、I キャッシュが無効のときだけです。

ビット 10 **DCEN** : データキャッシュ有効

- 0 : データキャッシュは無効です。
- 1 : データキャッシュは有効です。

ビット 9 **ICEN** : 命令キャッシュ有効

- 0 : 命令キャッシュは無効です。
- 1 : 命令キャッシュは有効です。

ビット 8 **PRFTEN** : プリフェッチ有効化

- 0 : プリフェッチは無効です。
- 1 : プリフェッチは有効です。

ビット 7:4 予約済み、クリア状態を保つ必要があります。

ビット 3:0 **LATENCY** : 遅延

- これらのビットは、フラッシュメモリアクセス時間に対する CPU クロック周期の割合を表します。
- 0000 : ウェイトステート 0 個
 - 0001 : ウェイトステート 1 個
 - 0010 : ウェイトステート 2 個
 -
 -
 -
 - 1110 : ウェイトステート 14 個
 - 1111 : ウェイトステート 15 個

3.8.2 フラッシュキーレジスタ (FLASH_KEYR)

フラッシュキーレジスタは、フラッシュ制御レジスタへのアクセスを許可し、それによってプログラムおよび消去操作を許可します。

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードアクセス

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

ビット 31:0 **FKEYR** : FPEC キー

FLASH_CR レジスタのロックを解除し、プログラミングや消去を許可するには、次の値を連続してプログラムする必要があります。

- a) KEY1 = 0x45670123
- b) KEY2 = 0xCDEF89AB

3.8.3 フラッシュオプションキーレジスタ (FLASH_OPTKEYR)

フラッシュオプションキーレジスタは、ユーザ設定セクタでのプログラムおよび消去操作を許可します。

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードアクセス

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

ビット 31:0 **OPTKEYR** : オプションバイトキー

FLASH_OPTCR レジスタをアンロックし、プログラミングを許可するには、次の値を連続してプログラムする必要があります。

- a) OPTKEY1 = 0x08192A3B
- b) OPTKEY2 = 0x4C5D6E7F

3.8.4 フラッシュステータスレジスタ (FLASH_SR)

フラッシュステータスレジスタは、現在実行しているプログラムおよび消去操作に関する情報を提供します。

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDERR	PGSEERR	PGPERR	PGAERR	WRPERR	Res.	Res.	OPERR	EOP
							rw	rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	rc_w1

ビット 31:17 予約済み、クリア状態を保つ必要があります。

ビット 16 **BSY** : ビジー

このビットは、フラッシュメモリ操作が進行中であることを示します。フラッシュメモリ操作の開始時にセットされ、操作が終了するかエラーが発生するとクリアされます。

0 : 進行中のフラッシュメモリ操作はありません。

1 : 進行中のフラッシュメモリ操作があります。

ビット 15:9 予約済み、クリア状態を保つ必要があります。

ビット 8 **RDERR** : 読み出し保護エラー (PCROP)

D バスを通じて読み出すアドレスがフラッシュの読み出し保護された部分のアドレスである場合、ハードウェアによってセットされます。

1 を書き込むとリセットされます。

ビット 7 **PGSEERR** : プログラミングシーケンスエラー

制御レジスタが正しく設定されていない時にコードによってフラッシュメモリへの書き込みアクセスが実施されると、ハードウェアによってセットされます。

1 を書き込むとクリアされます。

ビット 6 **PGPERR** : プログラミング並列処理エラー

プログラムシーケンスにおけるアクセスのサイズ (バイト、ハーフワード、ワード、ダブルワード) が並列処理設定 PSIZE (x8, x16, x32, x64) と一致しないときに、ハードウェアによってセットされます。

1 を書き込むとクリアされます。

ビット 5 **PGAERR** : プログラミング配置エラー

プログラムするデータを同じ 128 ビットのフラッシュメモリ列に含むことができないときに、ハードウェアによってセットされます。

1 を書き込むとクリアされます。

ビット 4 **WRPERR** : 書き込み保護エラー

消去/プログラムするアドレスがフラッシュメモリの書き込み保護された部分のアドレスである場合、ハードウェアによってセットされます。

1 を書き込むとクリアされます。

ビット 3:2 予約済み、クリア状態を保つ必要があります。

ビット 1 **OPERR** : 操作エラー

フラッシュ操作（プログラミング／消去／読み出し）リクエストが検出され、並列処理、配置、または書き込み保護エラーによってフラッシュ操作を実行できない場合に、ハードウェアによってセットされます。このビットは、エラー割り込みが有効になっている場合（ERRIE = 1）にのみセットされます。

ビット 0 **EOP** : 操作終了

1 つ以上のフラッシュメモリ操作（プログラム／消去）が正常終了するとハードウェアによってセットされます。操作終了割り込みが有効になっている場合（EOPIE = 1）にのみセットされます。
1 を書き込むとクリアされます。

3.8.5 フラッシュ制御レジスタ（FLASH_CR）

フラッシュ制御レジスタは、フラッシュメモリ操作を設定し、開始するのに使用します。

アドレスオフセット : 0x10

リセット値 : 0x8000 0000

アクセス : ノーウェイトステート（進行中のフラッシュメモリ操作がない場合）、ワード、ハーフワード、バイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.	Res.	Res.	Res.	Res.	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STRT
rs						rw	rw								rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	PSIZE[1:0]		Res.	SNB[3:0]				MER	SER	PG
						rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31 **LOCK** : ロック状態

1 のみを書き込みます。このビットがセットされていると、FLASH_CR レジスタがロックされていることを示します。アンロックシーケンスが検出されると、ハードウェアによってクリアされます。
アンロック操作が成功しない場合には、このビットは次のリセットまでセットされたままとなります。

ビット 30:26 予約済み、クリア状態を保つ必要があります。

ビット 25 **ERRIE** : エラー割り込みイネーブル

このビットは、FLASH_SR レジスタの OPERR ビットが 1 にセットされると割り込み生成を有効にします。
0 : エラー割り込み生成は無効です。
1 : エラー割り込み生成は有効です。

ビット 24 **EOPIE** : 操作終了割り込み有効化

このビットは、FLASH_SR レジスタの EOP ビットが 1 になると割り込み生成を有効にします。
0 : 割り込み生成は無効です。
1 : 割り込み生成は有効です。

ビット 23:17 予約済み、クリア状態を保つ必要があります。

ビット 16 **STRT** : 開始

このビットがセットされると、消去操作がトリガされます。ソフトウェアによってのみセットされ、BSY ビットがクリアされるとクリアされます。

ビット 15:10 予約済み、クリア状態を保つ必要があります。



ビット 9:8 **PSIZE** : プログラムサイズ
このビットは、プログラムの並列処理を選択します。
00 プログラム x8
01 プログラム x16
10 プログラム x32
11 プログラム x64

ビット 7 予約済み、クリア状態を保つ必要があります。

ビット 6:3 **SNB** : セクタ番号
このビットは、消去するセクタを選択します。
0000 セクタ 0
0001 セクタ 1
...
1010 セクタ 10
1011 セクタ 11
1100 ユーザ固有セクタ
1101 ユーザ設定セクタ
1110 設定禁止
1111 設定禁止

ビット 2 **MER** : 全体消去
すべてのユーザセクタに対して消去がアクティブになります。

ビット 1 **SER** : セクタ消去
セクタ消去がアクティブになります。

ビット 0 **PG** : プログラミング
フラッシュプログラミングがアクティブになります。

3.8.6 フラッシュオプション制御レジスタ (FLASH_OPTCR)

FLASH_OPTCR レジスタは、ユーザオプションバイトの変更に使用します。

アドレスオフセット : 0x14

リセット値 : 0x0FFF FFED。オプションビットには、リセット解除時にフラッシュメモリからの値が入力されます。

アクセス : ノーウェイトステート (進行中のフラッシュメモリ操作がない場合)、ワード、ハーフワード、バイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPR MOD	Res.	Res.	Res.	nWRP[11:0]											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP[7:0]								nRST_ STDBY	nRST_ STOP	WDG_ SW	Res.	BOR_LEV		OPT STRT	OPT LOCK
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rs	rs

ビット 31 **SPRMOD** : nWPRI ビットの保護モードの選択

- 0 : PCROP は無効です。セクタ i の書き込み保護に nWPRI ビットを使用します。
- 1 : PCROP は有効です。セクタ i の PCROP 保護に nWPRI ビットを使用します。

ビット 30:28 予約済み、クリア状態を保つ必要があります。

ビット 27:16 **nWRP[11:0]** : 書き込み保護なし

このビットには、リセット後のセクタの書き込み保護オプションバイトの値が含まれています。フラッシュメモリに新しい書き込み保護値をプログラムするように書き込むことができます。

- 0 : 選択したセクタで書き込み保護がアクティブになっています。
- 1 : 選択したセクタで書き込み保護がアクティブになっていません。

このビットには、リセット後のセクタ 0 から 11 の書き込み保護および読み出し保護 (PCROP) オプションバイトの値が含まれています。フラッシュメモリに新しい書き込み保護または PCROP 値をプログラムするように書き込むことができます。

SPRMOD がリセットされている場合

- 0 : セクタ i で書き込み保護がアクティブになっています。
- 1 : セクタ i で書き込み保護がアクティブになっていません。

SPRMOD がセットされている場合

- 0 : セクタ i で PCROP 保護がアクティブになっていません。
- 1 : セクタ i で PCROP 保護がアクティブになっています。

ビット 15:8 **RDP** : 読み出し保護

このビットには、リセット後の読み出し保護オプションレベルの値が含まれています。フラッシュメモリに新しい読み出し保護値をプログラムするように書き込むことができます。

- 0xAA : レベル 0、読み出し保護はアクティブではありません。
- 0xCC : レベル 2、チップ読み出し保護がアクティブです。
- その他 : レベル 1、メモリの読み出し保護がアクティブです。

ビット 7:5 **USER** : ユーザオプションバイト

このビットには、リセット後のユーザオプションバイトの値が含まれています。フラッシュメモリに新しいユーザオプションバイト値をプログラムするように書き込むことができます。

- ビット 7 : nRST_STDBY
- ビット 6 : nRST_STOP
- ビット 5 : WDG_SW

注 : **ハードウェアからソフトウェアまたはソフトウェアからハードウェアに WDG モードを変更するときに変更を有効にするには、システムリセットが必要です。**

ビット 4 予約済み、クリア状態を保つ必要があります。常に「0」が読み出されます。

ビット 3:2 BOR_LEV : BOR リセットレベル

このビットには、リセットをアクティブにしたリリセットを解除したりする供給レベル閾値が含まれています。新しい BOR レベルをプログラムするように書き込むことができます。デフォルトでは、BOR はオフです。電源電圧 (V_{DD}) が選択された BOR レベルを下回ると、デバイスリセットが生成されます。

00 : BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3

01 : BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2

10 : BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1

11 : BOR オフ、POR/PDR リセット閾値レベルが適用されます。

注： BOR 特性の詳細については、デバイスデータシートの「電気的特性」のセクションを参照してください。

ビット 1 OPTSTRT : オプション開始

このビットがセットされると、ユーザオプション操作がトリガされます。ソフトウェアによってのみセットされ、BSY ビットがクリアされるとクリアされます。

ビット 0 OPTLOCK : オプションロック

1 のみを書き込みます。このビットがセットされていると、FLASH_OPTCR レジスタがロックされていることを示します。このビットは、アンロックシーケンスが検出されると、ハードウェアによってクリアされます。

アンロック操作が成功しない場合には、このビットは次のリセットまでセットされたままとなります。

3.8.7 フラッシュインタフェースレジスタマップ

表 13. フラッシュレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DORST	ICRST	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY			
	リセット値																				0	0	0	0	0					0	0	0	0
0x04	FLASH_KEYR	KEY[31:16]																KEY[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	FLASH_OPTKEYR	OPTKEYR[31:16]																OPTKEYR[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDERR	PGSERR	PGPERR	PGAERR	WRPERR	Res.	Res.	OPERR	EOP
	リセット値																0								0	0	0	0	0			0	0
0x10	FLASH_CR	LOCK	Res.	Res.	Res.	Res.	Res.	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STRT	Res.	Res.	Res.	Res.	Res.	Res.	PSIZE[1:0]		Res.	SNB[3:0]			MER	SER	PG	
	リセット値	1						0	0								0							0	0		0	0	0	0	0	0	0
0x14	FLASH_OPTCR	SPRMOD	Res.	Res.	Res.	nWRP[11:0]												RDP[7:0]					nRST_STDBY	nRST_STOP	WDG_SW	Res.	BOR_LEV		OPTSTRT	OPTLOCK			
	リセット値	0				1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1		1	1	0	1

4 CRC 計算ユニット

4.1 CRC の概要

CRC (Cyclic Redundancy Check) 計算ユニットは、32 ビットデータワードと、ある一定の生成多項式から、CRC コードを得るために使用されます。

他のアプリケーションの中でも、CRC ベースのテクニックは、データ転送やストレージの整合性を確認するために使用されます。EN/IEC 60335-1 規格の範囲内では、CRC ベースのテクニックがフラッシュメモリの整合性を確認するひとつの手段となっています。CRC 計算ユニットは、実行時にソフトウェアのシグネチャ計算を支援します。リンク時に生成されて、特定のメモリ領域に保存されたりファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

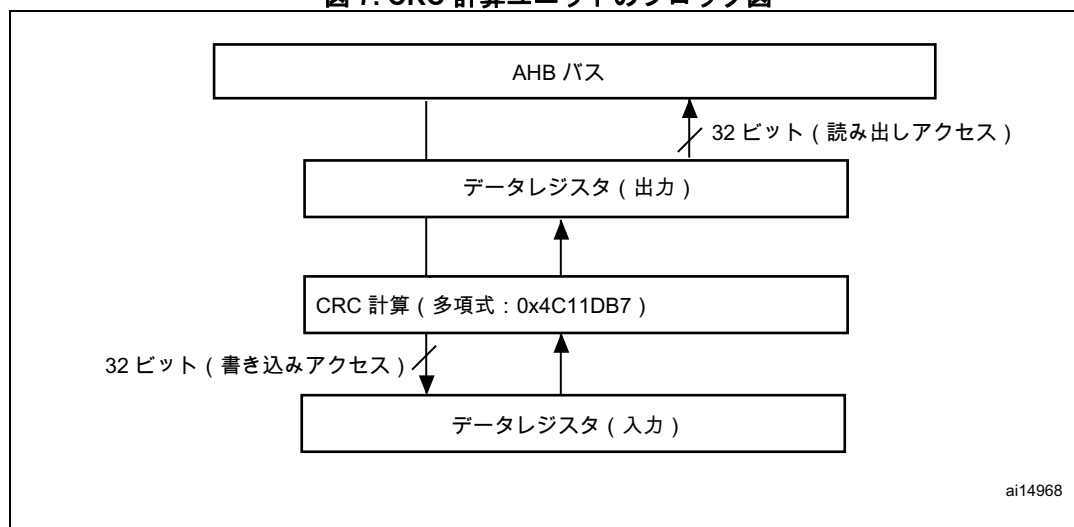
4.2 CRC の主な機能

- 以下のCRC-32 (イーサネット) 多項式を使用します。0x4C11DB7

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$
- シングル入力／出力 32 ビットデータレジスタ
- CRC 計算は 4 AHB クロックサイクル (HCLK) で完了
- 汎用 8 ビットレジスタ (一時ストレージとして使用可能)

図 7 にブロック図を示します。

図 7. CRC 計算ユニットのブロック図



4.3 CRC の機能説明

CRC 計算ユニットは、主に次のような 1 つの 32 ビットデータレジスタから構成されています。

- CRC 計算機に新しいデータを入力する入力レジスタとして使用されます（レジスタ書き込み時）。
- 前回の CRC 計算結果を保持します（レジスタ読み出し時）。

データレジスタへの書き込み操作のたびに、前の CRC 値と新しい値の組み合わせが作成されます（CRC 計算は 32 ビットデータワード単位で行われ、バイト単位ではありません）。

書き込み操作は、CRC 計算の終了まで一時停止されるので、連続書き込みアクセスや連続読み書きアクセスが可能です。

CRC 計算機は、CRC_CR レジスタの RESET 制御ビットで、0xFFFF FFFF にリセットできます。この操作は、CRC_IDR レジスタの内容に影響しません。

4.4 CRC レジスタ

CRC 計算ユニットは、2 つのデータレジスタと 1 つの制御レジスタから構成されています。CRC レジスタには、ワード（32 ビット）単位でアクセスする必要があります。

4.4.1 データレジスタ（CRC_DR）

アドレスオフセット：0x00
リセット値：0xFFFF FFFF

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

ビット 31:0 **データレジスタビット**
新しいデータを CRC 計算機への書き込み時に入力レジスタとして使用されます。
読み出し時には、前回の CRC 計算結果を保持します。

4.4.2 独立型データレジスタ (CRC_IDR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 汎用 8 ビットデータレジスタビット

1 バイトの一時的なストレージとして使用できます。

このレジスタは、CRC_CR レジスタの RESET ビットによって生成される CRC リセットの影響を受けません。

4.4.3 制御レジスタ (CRC_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 RESET ビット

CRC 計算ユニットをリセットして、データレジスタを 0xFFFF FFFF にセットします。

このビットはセットのみができ、ハードウェアによって自動的にクリアされます。

4.4.4 CRC レジスタマップ

表 14. CRC 計算ユニットのレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	CRC_DR	データレジスタ																																	
	リセット値	0xFFFF FFFF																																	
0x04	CRC_IDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	独立型データレジスタ									
	リセット値																										0x0000								
0x08	CRC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RESET		
	リセット値																																0		

5 電源コントローラ (PWR)

5.1 電源

主電源の供給方法は 2 種類あります。

- $V_{DD} = 1.7 \sim 3.6 \text{ V}$: 無効化された内蔵レギュレータと I/O 用外部電源供給で、 V_{DD} ピンを介した外部供給です。 V_{DD} および PDR_ON ピンに接続された 外部電源供給スーパーバイザを使用する必要があります。
- $V_{DD} = 1.8 \sim 3.6 \text{ V}$: 内蔵レギュレータ (有効の場合) と I/O 用外部電源供給で、 V_{DD} ピンを介した外部供給です。
- $V_{DD_USB} = 3.0 \sim 3.6 \text{ V}$
 V_{DD_USB} は、フルスピードトランシーバ専用の独立した USB 電源です。

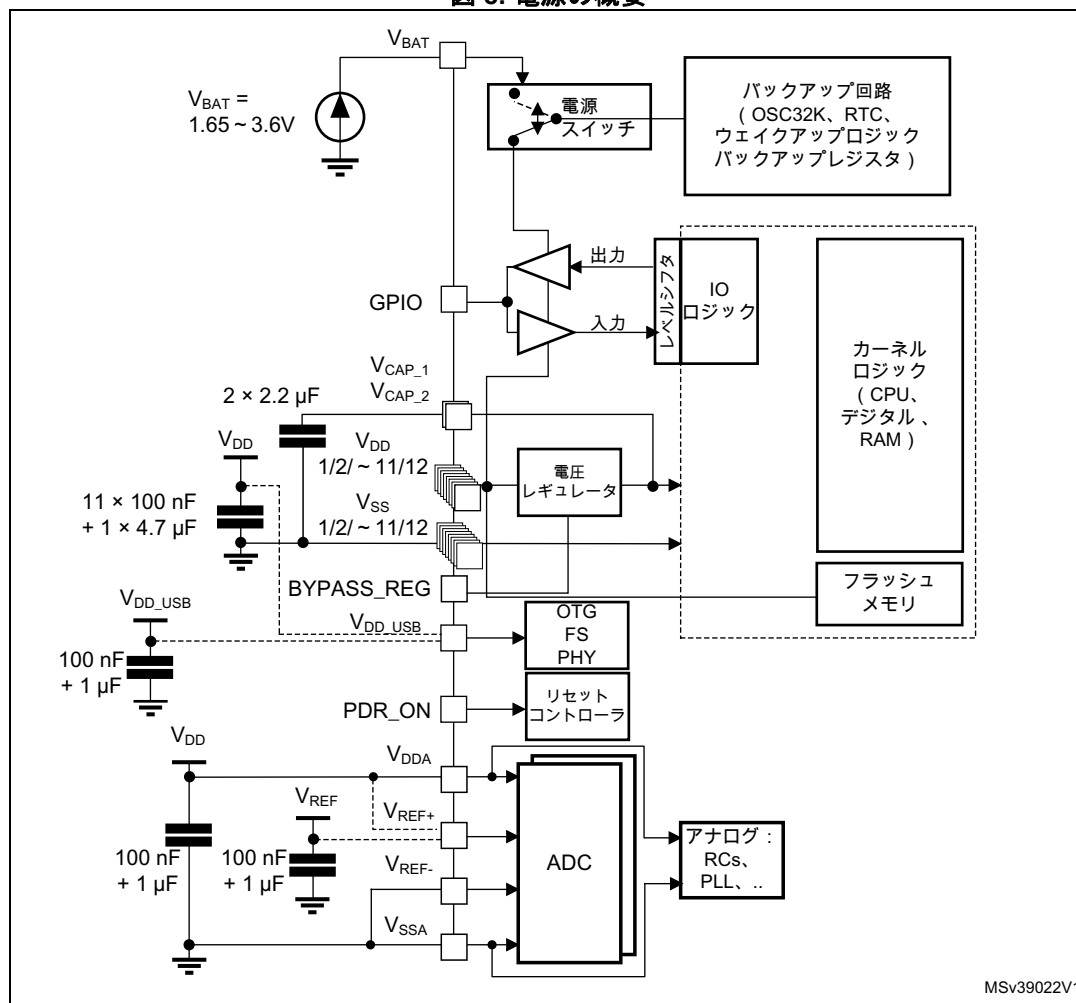
注 : V_{DD_USB} 値は、 V_{DD} および V_{DDA} に依存しません。ただし、 V_{DD_USB} は、最後にデバイスに供給され、かつ最初に切断される電源である必要があります。3 つの電源がシャットダウンしても、 V_{DD_USB} が短時間アクティブなままであり、 V_{DDA}/V_{DDIO} が機能範囲を下回る場合は、デバイスは損傷を受けません。

V_{DD_USB} が切断されてもデバイスは機能します。

リアルタイムクロック (RTC) および RTC バックアップレジスタは、主電源 V_{DD} がオフの場合、 V_{BAT} から供給することができます。

注 : 動作電源範囲に応じて、一部のペリフェラルは限定された機能と性能で 사용됩니다。詳細については、データシートの「一般動作条件」のセクションを参照してください。

図 8. 電源の概要



1. V_{DDA} と V_{SSA} はそれぞれ V_{DD} と V_{SS} に接続する必要があります。

5.1.1 独立した A/D コンバータ用電源と基準電圧

AD 変換の精度を向上させるため、ADC には独立した電源が供給されます。この電源回路は、フィルタやシールドを内蔵しているため、PCB のノイズを防ぐことができます。

- ADC 用の電源入力、独立した V_{DDA} ピンから供給されます。
- ADC 電源のグラウンドは、 V_{SSA} ピンとして用意されています。

低電圧入力時の精度を確保するため、独立した外部基準電圧を ADC の V_{REF} 入力に接続できます。 V_{REF} の電圧は、1.7 V から V_{DDA} の範囲になります。

5.1.2 バッテリバックアップドメイン

バックアップドメインの説明

V_{DD} がオフになった場合に、RTC バックアップレジスタの内容を保持し、RTC への電源供給を維持するために、 V_{BAT} ピンをバッテリーやその他の電源から供給されるオプションの予備電源に接続することができます。

メインのデジタル電源 (V_{DD}) がオフになったときでも RTC が動作できるよう、 V_{BAT} ピンから次に挙げるブロックに電源を供給します。

- RTC
- LSE オシレータ
- PC13 から PC15 の I/O

V_{BAT} 電源への切り替えは、リセットブロックに組み込まれているパワーダウンリセット回路によって制御されます。

警告 : $t_{RSTTEMPO}$ (V_{DD} 起動時の過渡期間) 中や PDR の検出後、 V_{BAT} と V_{DD} の間の電源スイッチは V_{BAT} に接続されたままになります。
 起動フェーズ中、 V_{DD} が $t_{RSTTEMPO}$ 以内に規定値に達し ($t_{RSTTEMPO}$ の値については、データシートを参照)、かつ $V_{DD} > V_{BAT} + 0.6\text{ V}$ である場合、電流は V_{BAT} まで V_{DD} と電源スイッチ (V_{BAT}) の間に接続された内部ダイオードを通して注入されます。
 V_{BAT} ピンに接続された電源/バッテリーがこの電流注入に対応できない場合は、この電源と V_{BAT} ピンの間に外部低電圧降下ダイオードを接続することを強く推奨します。

外部バッテリーを使用しないアプリケーションでは、 V_{BAT} ピンを V_{DD} に接続し、外部にデカップリング用の 100 nF のセラミックコンデンサを並列接続することを推奨します。

バックアップドメインが V_{DD} から供給を受けている場合 (アナログスイッチが V_{DD} に接続された状態)、次の機能が使用できます。

- PC14 と PC15 は、GPIO または LSE ピンとして使用できます。
- PC13 を GPIO として使用したり、追加機能を設定することができます (このピン設定の詳細は [表 27 : RTC 追加機能](#) を参照してください)。

注 : スイッチは限られた電流 (3 mA) しか流せないため、出力モードでの PC13 から PC15 までの GPIO の使用には制限があります。最大負荷 30 pF で最大速度 2 MHz に制限する必要があり、これらの I/O を電流ソースとして使用することはできません (たとえば、LED を駆動するなど)。

バックアップドメインが V_{BAT} から電源供給を受けている場合 (V_{DD} が印加されないため、アナログスイッチが V_{BAT} に接続された状態)、次の機能が使用できます。

- PC14 と PC15 は、LSE ピンとしてのみ使用できます。
- PC13 を RTC 追加機能ピンとして使用できます (このピン設定の詳細は [表 27 : RTC 追加機能](#) を参照してください)。

バックアップドメインアクセス

リセット後、バックアップドメイン (RTC レジスタおよび RTC バックアップレジスタ) は、予期しない書き込みアクセスから保護されます。バックアップドメインへのアクセスを有効にするには、次の手順に従います。

- RTC および RTC バックアップレジスタへのアクセス
- 1. RCC_APB1ENR レジスタの PWREN ビットをセットして、電源インタフェースクロックを有効にします ([セクション 6.3.12 : RCC_AHB3 ペリフェラルクロック有効レジスタ \(RCC_AHB3ENR\)](#) を参照)。
- 2. [セクション 5.4.1](#) の DBP ビットをセットして、バックアップドメインへのアクセスを有効にします。
- 3. RTC クロックソースを選択します: [セクション 6.2.8 : RTC/AWU クロック](#) を参照。
- 4. [セクション 6.3.20 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN [15] ビットをプログラムすることで、RTC クロックを有効にします。

RTC および RTC バックアップレジスタ

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。この RTC には、時刻クロック/カレンダー、プログラム可能なアラーム割り込みを 2 本、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグを搭載しています。この RTC には、20 個のバックアップデータレジスタ (80 バイト) を搭載しており、タンパ検出イベントが発生するとリセットされます。詳細については、[セクション 22 : リアルタイムクロック \(RTC\)](#) を参照してください。

5.1.3 電圧レギュレータ

バックアップドメインと STANDBY 回路以外のすべてのデジタル回路に電圧を供給するリニア電圧レギュレータが組み込まれています。このレギュレータの出力電圧は、約 1.2 V です。

この電圧レギュレータでは、1 つか 2 つの外付けコンデンサを専用ピン、V_{CAP_1} および一部のパッケージでは V_{CAP_2} に接続する必要があります。電圧レギュレータを有効にしたり無効にしたりするには、ある特定のピンを V_{SS} または V_{DD} に接続する必要があります。これらのピンはパッケージによって異なります。

電圧レギュレータがソフトウェアによってアクティブにされると、リセット後は常に使用可能となります。レギュレータは、アプリケーションのモードに応じて、3 種類のモードで動作します。

- **RUN モード**では、レギュレータは全電力を 1.2 V ドメイン (コア、メモリ、デジタルペリフェラル) に供給します。このモードでは、レギュレータの出力電圧 (約 1.2 V) はソフトウェアによって、いくつかの異なる電圧値にスケールリングできます (スケール 1、スケール 2、スケール 3 が PWR_CR レジスタの VOS[1:0] ビットを介して設定できます)。リセット後、VOS レジスタはスケール 2 にセットされます。PLL がオフの場合、電圧レギュレータは、VOS レジスタの内容に関係なく、スケール 3 にセットされます。VOS レジスタの内容は、PLL が有効化され、HSI または HSE がクロックソースとして選択されている場合のみ考慮されます。

この電圧スケールリングによって、デバイスが最大システム周波数より低速のクロックで動作しているときの消費電力を最適化することができます。

- **STOP モード**では、メインレギュレータまたは低電力レギュレータが 1.2 V ドメインに低電力を供給して、レジスタと内蔵 SRAM の内容を保持します。電圧レギュレータは、メインレギュレータモード (MR) か、低電力モード (LPR) のいずれかに設定できます。プログラムされた電圧スケールは STOP モードのときも維持されます。

電圧スケール 3 は、マイクロコントローラが STOP モードに入るときに自動的に選択されます ([セクション 5.4.1 : PWR 電源制御レジスタ \(PWR_CR\)](#) を参照)。

- **STANDBY モード**では、レギュレータは停止状態になります。STANDBY 回路とバックアップドメインを除き、レジスタと SRAM の内容は失われます。



注: 詳細については、STM32F412xx データシートの電圧レギュレータのセクションを参照してください。

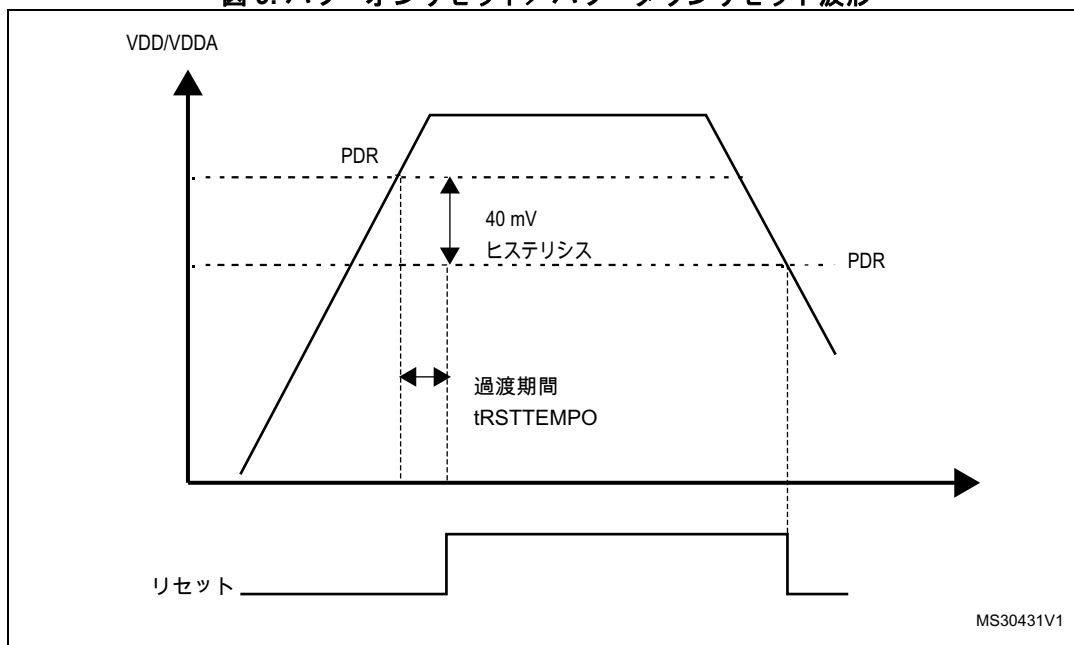
5.2 電源供給スーパバイザ

5.2.1 パワーオンリセット (POR) / パワーダウンリセット (PDR)

このデバイスには、POR/PDR 統合回路が搭載されています。
これにより、1.8 V 以上での正常な動作が可能になります。

1.8 V 未満のデバイスを使用するには、PDR_ON ピンを使用して内部電源供給スーパバイザをオフにする必要があります (STM32F412xx データシートの電源供給スーパバイザのセクションを参照)。デバイスは、 V_{DD}/V_{DDA} が規定の閾値 $V_{POR/PDR}$ を下回っている間は、外部のリセット回路を必要とせずに、リセットモードを維持します。POR/PDR 閾値の詳細については、データシートの電気的特性の項を参照してください。

図 9. パワーオンリセット/パワーダウンリセット波形



5.2.2 ブラウンアウトリセット (BOR)

パワーオン時、電源電圧が規定の V_{BOR} の閾値に達するまで、ブラウンアウトリセット (BOR) によってデバイスはリセット状態に保持されます。

V_{BOR} は、デバイスオプションバイトを介して設定されます。デフォルトでは、BORはオフです。プログラム可能な V_{BOR} 閾値レベルを 3 種類選択できます。

- BOR レベル 3 (VBOR3)。ブラウンアウト閾値レベル 3。
- BOR レベル 2 (VBOR2)。ブラウンアウト閾値レベル 2。
- BOR レベル 1 (VBOR1)。ブラウンアウト閾値レベル 1。

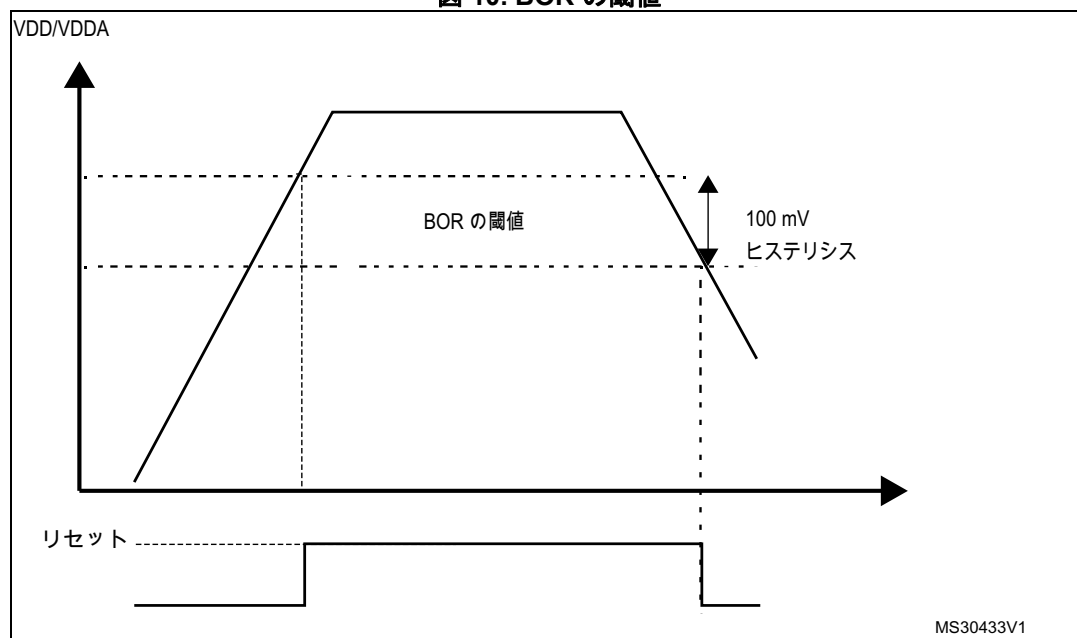
注: BOR 特性の詳細については、デバイスデータシートの「電気的特性」のセクションを参照してください。

電源電圧 (V_{DD}) が、選択された V_{BOR} の閾値を下回ると、デバイスリセットが生成されます。

BOR は、デバイスのオプションバイトをプログラミングすることで無効にすることができます。この場合、パワーオンとパワーダウンは POR/PDR または PDR が PDR_ON ピンを介してオフになっている場合は外部電源供給スーパーバイザによって監視されます ([セクション 5.2.1: パワーオンリセット \(POR\)](#) / [パワーダウリセット \(PDR\)](#) を参照)。

BOR 閾値のヒステリシスは 約100 mV です (電源電圧の立ち上がりエッジと立ち下がりエッジの間)。

図 10. BOR の閾値



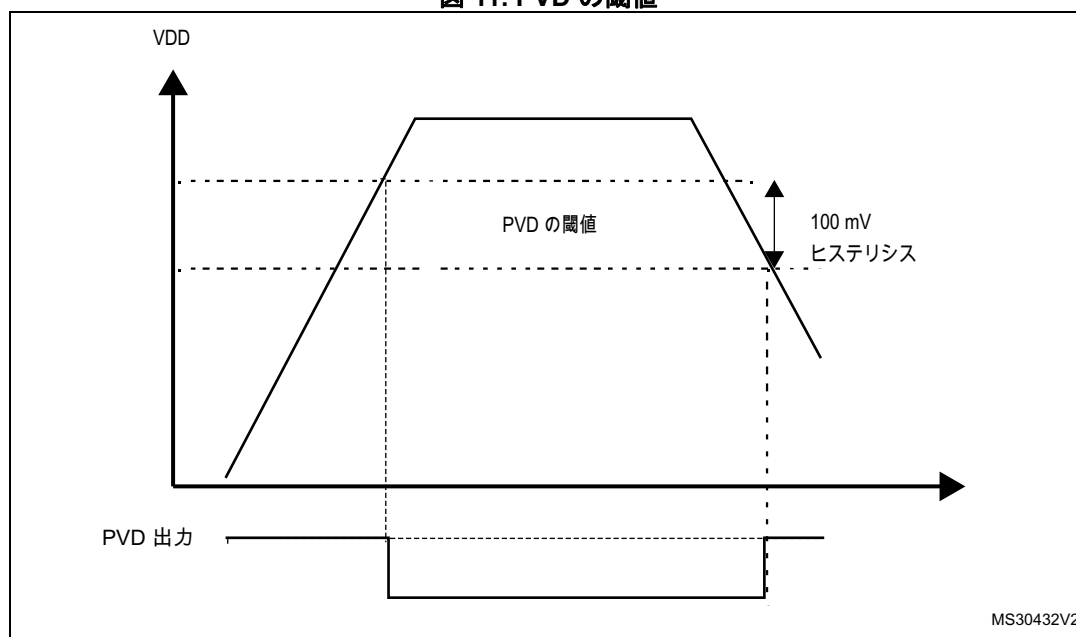
5.2.3 プログラム可能な電圧検出器 (PVD)

PVD を使用した V_{DD} 電源の監視が可能です。[PWR 電源制御レジスタ \(PWR_CR\)](#) の PLS[2:0] ビットで選択した閾値と比較することで監視します。

PVD は PVDE ビットをセットすることで有効になります。

[PWR 電源制御/ステータスレジスタ \(PWR_PWR\)](#) の PVDO フラグを使用して、 V_{DD} が PVD の閾値よりも高いか低いかに示すことができます。このイベントは EXTI のライン 16 に内部接続され、EXTI レジスタで有効な場合は割り込みリクエストを生成させることができます。EXTI ライン 16 の立ち上がり/立ち下がりエッジの設定に応じて、 V_{DD} が PVD の閾値を下回るか、上回ったとき、あるいはその両方で、PVD 出力割り込みを生成させることができます。たとえば、サービスルーチンで、緊急停止処理を実行することなどが可能です。

図 11. PVD の閾値



5.3 低電力モード

デフォルトでは、マイクロコントローラは、システムリセットまたはパワーオンリセット後は RUN モードです。RUN モードでは、CPU クロックとして HCLK が使用され、プログラムコードが実行されます。外部イベント待ちなど、CPU の連続実行が不要なときの節電のために、いくつかの低電力モードが用意されています。消費電力の節減、スタートアップ時間の短縮、使用可能なウェイクアップソースを考慮した最適なモード選択はユーザに委ねられています。

このデバイスは、次の 4 つの低電力モードを備えています。

- SLEEP モード (FPU 搭載 Cortex[®]-M4 コアは停止、ペリフェラルは動作状態を維持)
- STOP モード (すべてのクロックが停止)
- STANDBY モード (1.2 V ドメインの電源オフ)
- Batch Acquisition Mode (BAM) : デバイスが SLEEP モード、フラッシュメモリがオフ、必要なペリフェラルが動作継続中、DMA を介したデータ転送可能

さらに、次の手段のいずれかによって RUN モードの消費電力を節減できます。

- システムクロックの低速化
- 使用しない APBx および AHBx ペリフェラルへのクロックのゲーティング

低電力モードへの移行

低電力モードへの移行は、MCU が WFI (Wait For Interrupt) または WFE (Wait For Event) 命令を実行することにより、または FPU 搭載 Cortex[®]-M4 システム制御レジスタの SLEEPONEXIT ビットが割り込みサービスルーチン (ISR) 終了時にセットされている時点で実行されます。

WFI または WFE による低電力モードへの移行は、ペンディング状態の割り込みやイベントがない場合にのみ実行されます。

低電力モードの終了

MCU は、低電力モードへの移行方法に応じて、低電力の SLEEP および STOP モードを終了します。

- WFI 命令または 割り込みサービスルーチン (ISR) 終了によって低電力モードに移行した場合、NVIC によって認識されたペリフェラル割り込みであれば何でもデバイスをウェイクアップすることができます。
- WFE 命令によって低電力モードに移行した場合、MCU はイベントの発生直後に低電力モードを終了します。ウェイクアップイベントは、次のいずれかによって生成できます。

- NVIC IRQ 割り込み：

FPU 搭載 Cortex®-M4 システム制御レジスタの SEVONPEND = 0 の場合：ペリフェラル制御レジスタおよび NVIC にて割り込みを有効にすることによって行います。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) NVIC ペリフェラル IRQ チャンネルペンディングビットをクリアする必要があります。十分な優先度の NVIC 割り込みのみが MCU をウェイクアップし、割り込みを生成します。

FPU 搭載 Cortex®-M4 システム制御レジスタの SEVONPEND = 1 の場合：ペリフェラル制御レジスタ (および任意で NVIC の割り込み) にて割り込みを有効にすることによって行います。MCU が WFE からリスタートするときには、(NVIC 割り込みクリアペンディングレジスタの) ペリフェラル割り込みペンディングビットと、有効であれば NVIC ペリフェラル IRQ チャンネルペンディングビットをクリアする必要があります。すべての NVIC 割り込み (無効化されているものも含む) が MCU をウェイクアップします。十分な優先度の有効化された NVIC 割り込みのみが MCU をウェイクアップし、割り込みを生成します。

- イベント

EXTI ラインをイベントモードに設定することによって行います。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、EXIT ペリフェラル割り込みペンディングビットや NVIC IRQ チャンネルペンディングビットをクリアする必要はありません。ペリフェラルの割り込みフラグをクリアする必要があるかもしれません。

MCU は、外部リセット (NRST ピン)、IWDG リセット、有効な WKUPx ピンのうちの 1 つの立ち上がりエッジ、または RTC イベントのいずれかが発生すると、低電力の STANDBY モードを終了します (図 206 : RTC ブロック図を参照)。

STANDBY モードからのウェイクアップ後、プログラム実行はリセット後と同様にリスタートされます (ブートピン信号のサンプリング、オプションバイトローディング、リセットベクタのフェッチなど)。

十分な優先度の有効化された NVIC 割り込みのみが MCU をウェイクアップし、割り込みを生成します。

表 15. 低電力モードの概要

モード名	エントリ	ウェイクアップ	1.2 V ドメインクロックへの影響	V _{DD} ドメインクロックへの影響	電圧レギュレータ
SLEEP および BAM ⁽¹⁾ (Sleep-now または Sleep-on-exit)	WFI または ISR 終了	割り込み	CPU クロックオフ 他のクロック およびアナログ クロックソース への影響なし	なし	オン
	WFE	ウェイクアップイベント			
STOP	SLEEPDEEP ビット + WFI、ISR 終了、または WFE	任意の EXTI ライン (EXTI レジスタで設定、内部ラ インおよび外部ライン)	すべての 1.2 V ドメインクロック はオフ	HSI および HSE オシレータは オフ	メインレギュレータ または低電力 レギュレータ (PWR 電源制御レジスタ (PWR_CR) に 応じて)
STANDBY	PDDS ビット + SLEEPDEEP ビット + WFI、ISR 終了、または WFE	WKUP ピンの立ち上がり エッジ、RTC アラーム (アラーム A またはアラーム B)、 RTC ウェイクアップイベント、 RTC タンパイイベント、 RTC タイムスタンプイベント、 NRST ピンによる外部リセット、 IWDG リセット			オフ

1. 特定の BAM への移行と終了の要件については、[セクション 5.3.4 : Batch Acquisition Mode](#)を参照してください。

5.3.1 システムクロックの低速化

RUN モードでは、プリスケアラジスタをプログラミングすることによって、システムクロック (SYSCLK、HCLK、PCLK1、PCLK2) の速度を下げるできます。SLEEP モードに移行する前にペリフェラルの速度を下げるため、これらのプリスケアラを使用することもできます。

詳細については、[セクション 6.3.3: RCC クロック設定レジスタ \(RCC_CFGR\)](#) を参照してください。

5.3.2 ペリフェラルクロックゲーティング

RUN モードでは、消費電力を低減するため、個々のペリフェラルとメモリへの HCLKx および PCLKx をいつでも停止することができます。

SLEEP モードで消費電力をさらに低減するため、WFI または WFE 命令を実行する前に、ペリフェラルクロックを停止することができます。

ペリフェラルへのクロックゲーティングは、AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1ENR)、AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR) によって制御されます ([セクション 6.3.10 : RCC AHB1 ペリフェラルクロック有効レジスタ \(RCC_AHB1ENR\)](#)、[セクション 6.3.11 : RCC AHB2 ペリフェラルクロック有効レジスタ \(RCC_AHB2ENR\)](#)、および [セクション 6.3.12 : RCC AHB3 ペリフェラルクロック有効レジスタ \(RCC_AHB3ENR\)](#) を参照)。

RCC_AHBxLPENR レジスタと RCC_APBxLPENR レジスタの対応ビットをリセットすることで、SLEEP モード時のペリフェラルクロックを自動的に停止させることができます。



5.3.3 SLEEP モード

SLEEP モードへの移行

SLEEP モードへの移行は、FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットがクリアされている場合に、[セクション：低電力モードへの移行](#)に従って実行されます。

SLEEP モードへの移行方法の詳細については、[表 16](#) および [表 17](#) を参照してください。

注： **SLEEP モードに移行する前に、すべての割り込みペンディングビットがクリアされる必要があります。**

SLEEP モードの終了

SLEEP モードは、[セクション：低電力モードの終了](#)に従って終了します。

SLEEP モードの終了方法の詳細については、[表 16](#) および [表 17](#) を参照してください。

表 16. Sleep-now への移行と終了

Sleep-now モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0、 – ペンディング状態の割り込み (WFI) やイベント (WFE) なし FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) 終了時 – SLEEPDEEP = 0 および – SLEEPONEXIT = 1、 – ペンディング状態の割り込みなし FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	WFI または Return from ISR を使用して移行した場合 割り込み： 表 40：STM32F412xx のベクタテーブル を参照してください。 WFE 命令を使用して移行し、SEVONPEND = 0 の場合： ウェイクアップイベント： セクション 10.2.3：ウェイクアップイベント管理 を参照してください。 WFE 命令を使用して移行し、SEVONPEND = 1 の場合： 割り込みイベント (NVIC で無効化されている場合も含む)： 表 40：STM32F412xx のベクタテーブル またはウェイクアップイベント (セクション 10.2.3：ウェイクアップイベント管理 を参照) を参照してください。
ウェイクアップ遅延時間	なし

表 17. Sleep-on-exit への移行と終了

Sleep-on-exit	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0、 – ペンディング状態の割り込み (WFI) やイベント (WFE) なし FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) 終了時 – SLEEPDEEP = 0、 – SLEEPONEXIT = 1、 – ペンディング状態の割り込みなし FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	割り込み: 表 40: STM32F412xx のベクタテーブルを参照。
ウェイクアップ遅延時間	なし

5.3.4 Batch Acquisition Mode

BAM への移行

BAM への移行は、FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットがクリアされている場合に、[セクション: 低電力モードへの移行](#)に従って実行されます。

SLEEP モードへの移行方法の詳細については、[表 18](#) および [表 19](#) を参照してください。

SLEEP モードに移行する前に、フラッシュメモリが必要な低電力モードで動作するようにソフトウェアによって設定される必要があります。BAM 時にペリフェラルから RAM にデータを転送する必要がある場合、SLEEP モードに移行する前に DMA を有効にする必要があります。

BAM の終了

BAM は、[セクション: 低電力モードの終了](#)に従って終了します。

SLEEP モードの終了方法の詳細については、[表 18](#) および [表 19](#) を参照してください。

BAM からウェイクアップした後、コード実行がフラッシュメモリから再開される場合、まずフラッシュメモリをウェイクアップさせる必要があります。

このウェイクアップ時間は、内部 SRAM から実行するソフトウェアで管理する必要があります。

表 18. BAM-now への移行と終了

Sleep-now モード	説明
モードへの移行	フラッシュメモリを低電力モードにセットします。 – PWR_CR レジスタの FISSR/FMSSR および FPDS ビット 次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0 および – SLEEPONEXIT = 0 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	WFI 命令を使用して移行した場合： 割り込み：表 40 : STM32F412xx のベクタテーブルを参照してください。 WFE 命令を使用して移行した場合： ウェイクアップイベント：セクション 10.2.3: ウェイクアップイベント管理を参照してください。 フラッシュメモリのウェイクアップ時間が必要である場合、PWR_CR レジスタの FISSR/FMSSR ビットをセットする必要があります。
ウェイクアップ遅延時間	なし (RAM からコードを実行する場合) フラッシュメモリからコード実行を再開する前の低電力モードのフラッシュメモリのウェイクアップ時間 (データシートの電気的特性のセクションにあるフラッシュメモリのウェイクアップ時間を参照)。

表 19. BAM-on-exit への移行と終了

Sleep-on-exit	説明
モードへの移行	フラッシュメモリを低電力モードにセットします。 – PWR_CR レジスタの FISSR/FMSSR および FPDS ビット 次の条件下での WFI (Wait for Interrupt) – SLEEPDEEP = 0 および – SLEEPONEXIT = 1 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	割り込み：表 40 : STM32F412xx のベクタテーブルを参照。 フラッシュメモリのウェイクアップ時間が必要である場合、PWR_CR レジスタの FISSR/FMSSR ビットをセットする必要があります。
ウェイクアップ遅延時間	なし (内部 SRAM からコードを実行する場合) フラッシュメモリからコード実行を再開する前の低電力モードのフラッシュメモリのウェイクアップ時間 (データシートの電気的特性のセクションにあるフラッシュメモリのウェイクアップ時間を参照)。

5.3.5 STOP モード

STOP モードは、ペリフェラルクロックゲーティングと組み合わせられた FPU 搭載 Cortex®-M4 のディープスリープ (deepsleep) モードに準拠しています。電圧レギュレータは、通常モードまたは低電力モードのいずれかに設定できます。STOP モードでは、1.2 V ドメインのすべてのクロックが停止し、PLL、HSI、HSE RC オシレータが停止します。内蔵 SRAM とレジスタの内容は保持されます。

PWR_CR レジスタの一部の設定によって、電力消費をさらに低減できます。フラッシュメモリが低電力モードで動作している場合、STOP モードからのウェイクアップの際に、さらなる起動時間が必要になります (表 20 : STOP モードの動作モードおよびセクション 5.4.1 : PWR 電源制御レジスタ (PWR_CR) を参照)。



表 20. STOPモードの動作モード

	STOP モード	MRLV ビット	LPLV ビット	FPDS ビット	LPDS ビット	ウェイクアップ遅延時間
システム 通常モード	STOP MR	0	-	0	0	HSI RC 起動時間
	STOP MRFPD	0	-	1	0	HSI RC 起動時間 + ディープパワーダウンモードからの フラッシュのウェイクアップ時間
	STOP LP	0	0	0	1	HSI RC 起動時間 + LP モードからのレギュレータの ウェイクアップ時間
	STOP LPFPD	-	0	1	1	HSI RC 起動時間 + ディープパワーダウンモードからの フラッシュのウェイクアップ時間 + LP モードからのレギュレータの ウェイクアップ時間
	STOP MRLV	1	-	-	0	HSI RC 起動時間 + ディープパワーダウンモードからの フラッシュのウェイクアップ時間 + 低電圧モードからのメインレギュレータ
	STOP LPLV	-	1	-	1	HSI RC 起動時間 + ディープパワーダウンモードからの フラッシュのウェイクアップ時間 + 低電圧 LP モードからのレギュレータの ウェイクアップ時間

STOP モードへの移行

STOP モードへの移行は、FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットがセットされている場合に、[セクション : 低電力モードへの移行](#)に従って実行されます。

STOP モードへの移行方法の詳細については、[表 21](#)を参照してください。

STOP モードで消費電力をさらに低減するには、内蔵電圧レギュレータを低電力モードにします。この設定は[PWR 電源制御レジスタ \(PWR_CR\)](#)の LPDS ビットで行います。

フラッシュメモリがプログラミング中の場合、メモリアクセスが終了してから、STOP モードに移行します。

APB ドメインにアクセス中の場合、APB アクセスが終了してから、STOP モードに移行します。

STOP モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照: [セクション 21.3](#)、[セクション 21 : ウィンドウ型ウォッチドッグ \(WWDG\)](#)
- リアルタイムクロック (RTC) : この設定は [セクション 6.3.20 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#)の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は [セクション 6.3.21 : RCC クロック制御およびステータスレジスタ \(RCC_CSR\)](#)の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は [セクション 6.3.20 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#)の LSEON ビットで行います。

ADC は、STOP モードに移行する前に停止させない限り、STOP モード中にも電力を消費します。ADC を停止するには、ADC_CR2 レジスタの ADON ビットに 0 を書き込む必要があります。

注 : **STOP モードに移行する前にアプリケーションが外部クロックを無効にする必要がある場合、最初に HSEON ビットを無効にし、次いでシステムクロックを HSI に切り替える必要があります。**

さもなければ、STOP モードに移行する前に外部クロック (外部オシレータ) が取り外される可能性があるにもかかわらず HSEON ビットが有効のままであった場合、STOP モードに入るときに外部オシレータ障害を検出して誤動作を回避するべく、クロックセキュリティシステム (CSS) 機能が作動してしまいます。

STOP モードの終了

STOP モードは、[セクション : 低電力モードの終了](#)に従って終了します。

STOP モードの終了方法の詳細については、[表 21](#) を参照してください。

割り込みまたはウェイクアップイベントの発行によって STOP モードを終了する場合、システムクロックとして HSI RC オシレータが選択されます。

電圧レギュレータが低電力モードで動作している場合、STOP モードからのウェイクアップの際、更なる起動時間が必要になります。STOP モードの間も内蔵レギュレータをオン状態に保つことによって、消費電力は増加しますが、起動時間は短縮されます。

表 21. STOP モードへの移行と終了

STOP モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) <ul style="list-style-type: none">– ペンディング状態の割り込み (WFI) やイベント (WFE) なし、– FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット– 電源制御レジスタ (PWR_CR) の PDDS ビットをクリア– PWR_CR の LPDS ビットを設定することによって電圧レギュレータのモードを選択
	割り込みサービスルーチン (ISR) 終了時 : <ul style="list-style-type: none">– ペンディング状態の割り込みなし、– FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット– SLEEPONEXIT = 1、– 電源制御レジスタ (PWR_CR) の PDDS ビットをクリア
	注 : STOP モードに移行するには、すべての EXTI ラインのペンディングビット (セクション 10.3.6 : ペンディングレジスタ (EXTI_PR) 内)、すべてのペリフェラル割り込みのペンディングビット、RTC アラーム (アラーム A およびアラーム B)、RTC ウェイクアップ、RTC タンパ、RTC タイムスタンプフラグをリセットする必要があります。そうしないと、STOP モード移行手順が無視され、プログラムが実行され続けます。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 21. STOP モードへの移行と終了

STOP モード	説明
モードの終了	<p>WFI または Return from ISR を使用して移行した場合 割り込みモードに設定されている任意の EXTI ライン（対応する EXTI 割り込みベクタが NVIC で有効になっている必要があります）。割り込みソースは、外部割り込みまたはウェイクアップ機能を備えたペリフェラルになることがあります。 表 40 : STM32F412xx のベクタテーブルを参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 0 の場合 イベントモードに設定されている任意の EXTI ライン セクション 10.2.3 : ウェイクアップイベント管理を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 1 の場合：</p> <ul style="list-style-type: none"> 割り込みモードに設定されている任意の EXTI ライン（対応する EXTI 割り込みベクタが NVIC で無効になっている場合も含む）。割り込みソースは、外部割り込みまたはウェイクアップ機能を備えたペリフェラルになることがあります。 表 40 : STM32F412xx のベクタテーブルを参照してください。 ウェイクアップイベント： セクション 10.2.3 : ウェイクアップイベント管理を参照。
ウェイクアップ遅延時間	表 20 : STOP モードの動作モード を参照してください。

5.3.6 STANDBY モード

STANDBY モードでは、消費電力を最も少なくできます。このモードは、電圧レギュレータを無効にした状態の FPU 搭載 Cortex®-M4 のディープスリープ (deepsleep) モードに基づきます。結果として、1.2 V ドメインの電源がオフになります。PLL、HSI オシレータ、HSE オシレータもオフになります。バックアップドメインのレジスタ (RTC レジスタ、RTC バックアップレジスタ) と STANDBY 回路を除いて、SRAM とレジスタの内容は失われます ([図 8](#) を参照)。

STANDBY モードへの移行

STANDBY モードへの移行は、FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットがセットされている場合に、[セクション : 低電力モードへの移行](#)に従って実行されます。

STANDBY モードへの移行方法の詳細については、[表 22](#) を参照してください。

STANDBY モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照：[セクション 20.3](#)、[セクション 20 : 独立型ウォッチドッグ \(IWDG\)](#)
- リアルタイムクロック (RTC) : この設定は、バックアップドメイン制御レジスタ (RCC_BDCR) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は、制御/ステータスレジスタ (RCC_CSR) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は、バックアップドメイン制御レジスタ (RCC_BDCR) の LSEON ビットで行います。

STANDBY モードの終了

STANDBY モードは、[セクション : 低電力モードの終了](#)に従って終了します。PWR_CR の SBF ステータスフラグ ([セクション 5.4.2 : PWR 電源制御/ステータスレジスタ \(PWR_PWR\)](#) を参照) は MCU が STANDBY モードにあったことを示します。PWR_CR を除くすべてのレジスタは、

STANDBY モードからのウェイクアップ後にリセットされます。

STANDBY モードの終了方法の詳細については、[表 22](#) を参照してください。

表 22. STANDBY モードへの移行と終了

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) <ul style="list-style-type: none">– FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット、– 電源制御レジスタ (PWR_CR) の PDDS ビットをセット、– ペンディング状態の割り込み (WFI) やイベント (WFE) なし、– 電源制御レジスタ (PWR_CR) の WUF ビットをクリア、– 選択されたウェイクアップソース (RTC アラーム A、RTC アラーム B、RTC ウェイクアップ、タンパ、タイムスタンプフラグ) に対応する RTC フラグをクリア
	次の条件下での割り込みサービスルーチン (ISR) 終了時 <ul style="list-style-type: none">– FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット、– SLEEPONEXIT = 1、– 電源制御レジスタ (PWR_CR) の PDDS ビットをセット、– ペンディング状態の割り込みなし、– 電源制御/ステータスレジスタ (PWR_SR) の WUF ビットをクリア、– 選択されたウェイクアップソース (RTC アラーム A、RTC アラーム B、RTC ウェイクアップ、タンパ、タイムスタンプフラグ) に対応する RTC フラグをクリア
モードの終了	WKUP ピンの立ち上がりエッジ、RTC アラーム (アラーム A またはアラーム B)、RTC ウェイクアップイベント、タンパイベント、タイムスタンプイベント、NRST ピンによる外部リセット、IWDG リセット
ウェイクアップ遅延時間	リセットフェーズ

STANDBY モードにおける I/O の状態

STANDBY モードでは、以下のピンを除き、すべての I/O ピンがハイインピーダンス状態となります。

- リセットパッド (有効)
- タンパ、タイムスタンプ、RTC アラーム出力、RTC クロック較正出力として設定されている場合の RTC_AF1 ピン (PC13)
- WKUP ピン (PA0/PC0/PC1) (有効な場合)

デバッグモード

デフォルトでは、デバッグ機能が使用されているときにアプリケーションが MCU を STOP または STANDBY モードにすると、デバッグ接続は失われます。これは、FPU 搭載 Cortex®-M4 コアにクロックが供給されなくなるためです。

ただし、DBGMCU_CR レジスタの設定ビットをセットすることによって、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。詳細については、[セクション 30.16.1: 低電力モードのデバッグサポート](#)を参照してください。

5.3.7 デバイスをウェイクアップさせるための RTC 代替機能のプログラミング (STOP および STANDBY モードから)

MCU は、RTC 代替機能により、低電力モードからウェイクアップさせることができます。

RTC 代替機能には、RTC アラーム (アラーム A およびアラーム B)、RTC ウェイクアップ、RTC タンパイベント検出、RTC タイムスタンプイベント検出があります。

これらの RTC 代替機能によって、システムを低電力の STOP モードや STANDBY モードからウェイクアップさせることができます。

システムはまた、外部割り込み (オートウェイクアップモード) に頼ることなく、RTC アラームや RTC ウェイクアップイベントを使用して低電力モードからウェイクアップすることもできます。

RTC は、一定の時間間隔で STOP モードや STANDBY モードからウェイクアップさせるためのプログラム可能なタイムベースとすることができます。

この目的のため、[セクション 6.3.20 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1 : 0] ビットをプログラムすることによって、次の 3 つの代替 RTC クロックソースのうちから 2 つを選択できます。

- 低電力 32.768 kHz 外部クリスタルオシレータ (LSE OSC)
このクロックソースは、非常に少ない消費電力 (標準的な条件下で追加消費電力 1 μ A 未満) で高精度のタイムベースとなります。
- 低電力内部 RC オシレータ (LSI RC)
このクロックソースには、32.768 kHz クリスタルのコストを節約できるという利点があります。この内部 RC オシレータは、最小限の消費電力で動作するように設計されています。

STOP モードからデバイスをウェイクアップさせるための RTC 代替機能

- RTC アラームイベントによって STOP モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) EXTI ライン 17 を立ち上がりエッジを検知するように設定します (割り込みモードまたはイベントモード)。
 - b) RTC_CR レジスタにて RTC アラーム割り込みを有効にします。
 - c) RTC が RTC アラームを生成するように設定します。
- RTC タンパまたはタイムスタンプイベントによって STOP モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) EXTI ライン 21 を立ち上がりエッジを検知するように設定します (割り込みモードまたはイベントモード)。
 - b) RTC_CR レジスタの RTC タイムスタンプ割り込みまたは RTC_TAFRCR レジスタの RTC タンパ割り込みを有効にします。
 - c) タンパまたはタイムスタンプイベントを検出するよう RTC を設定します。
- RTC ウェイクアップイベントによって STOP モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) EXTI ライン 22 を立ち上がりエッジを検知するように設定します (割り込みモードまたはイベントモード)。
 - b) RTC_CR レジスタ内の RTC ウェイクアップ割り込みを有効にします。
 - c) RTC ウェイクアップイベントを生成するように RTC を設定します。

STANDBY モードからデバイスをウェイクアップさせるための RTC 代替機能

- RTC アラームイベントによって STANDBY モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) RTC_CR レジスタにて RTC アラーム割り込みを有効にします。
 - b) RTC が RTC アラームを生成するように設定します。

- RTC タンパまたはタイムスタンプイベントによって STANDBY モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) RTC_CR レジスタの RTC タイムスタンプ割り込みまたは RTC_TAFCR レジスタの RTC タンパ割り込みを有効にします。
 - b) タンパまたはタイムスタンプイベントを検出するよう RTC を設定します。
- RTC ウェイクアップイベントによって STANDBY モードからデバイスをウェイクアップさせるには、次の設定が必要です。
 - a) RTC_CR レジスタ内の RTC ウェイクアップ割り込みを有効にします。
 - b) RTC ウェイクアップイベントを生成するように RTC を設定します。

RTC 代替機能の安全なウェイクアップフラグクリアシーケンス

PWR ウェイクアップフラグ (WUTF) がクリアされる前に選択された RTC 代替機能がセットされた場合、検出は立ち上がりエッジで一度だけ行われるため、次のイベントで検出されることはありません。

RTC 代替機能がマッピングされているピンでの検出ミス为避免、STOP モードおよび STANDBY モードを正しく終了するには、STANDBY モードに入る前に以下の手順に従うことを推奨します。

- RTC アラームを使用してデバイスを低電力モードからウェイクアップさせる場合は、次の設定が必要です。
 - a) RTC アラーム割り込みを無効にします (RTC_CR レジスタの ALRAIE または ALRBIE ビット)。
 - b) RTC アラーム (ALRAF/ALRBF) フラグをクリアします。
 - c) PWR ウェイクアップ (WUF) フラグをクリアします。
 - d) RTC アラーム割り込みを有効にします。
 - e) 再び低電力モードに移行します。
- RTC ウェイクアップを使用してデバイスを低電力モードからウェイクアップさせる場合は、次の設定が必要です。
 - a) RTC ウェイクアップ割り込みを無効にします (RTC_CR レジスタの WUTIE ビット)。
 - b) RTC ウェイクアップ (WUTF) フラグをクリアします。
 - c) PWR ウェイクアップ (WUF) フラグをクリアします。
 - d) RTC ウェイクアップ割り込みを有効にします。
 - e) 再び低電力モードに移行します。
- RTC タンパを使用してデバイスを低電力モードからウェイクアップさせる場合は、次の設定が必要です。
 - a) RTC タンパ割り込みを無効にします (RTC_TAFCR レジスタの TAMPIE ビット)。
 - b) タンパ (TAMP1F/TSF) フラグをクリアします。
 - c) PWR ウェイクアップ (WUF) フラグをクリアします。
 - d) RTC タンパ割り込みを有効にします。
 - e) 再び低電力モードに移行します。
- RTC タイムスタンプを使用してデバイスを低電力モードからウェイクアップさせる場合は、次の設定が必要です。
 - a) RTC タイムスタンプ割り込みを無効にします (RTC_CR レジスタの TSIE ビット)。
 - b) RTC タイムスタンプ (TSF) フラグをクリアします。
 - c) PWR ウェイクアップ (WUF) フラグをクリアします。
 - d) RTC タイムスタンプ割り込みを有効にします。
 - e) 再び低電力モードに移行します。

5.4 電源制御レジスタ

5.4.1 PWR 電源制御レジスタ (PWR_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 8000 (STANDBY モードからのウェイクアップによってリセット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FISSR	FMSSR	Res.	Res.	Res.	Res.
										rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOS		ADCD1	Res.	MRLV DS	LPLV DS	FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	w	w	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **FISSR** : システム実行中のフラッシュインタフェース停止

0 : フラッシュインタフェースクロック実行 (デフォルト値)

1 : フラッシュインタフェースクロックオフ

注 : このビットは、フラッシュ自体で実行している間はセットできません。RAM から実行される特定のルーチンで実行する必要があります。

ビット 20 **FMSSR** : システム実行中のフラッシュメモリのスリープ

0 : フラッシュ標準モード (デフォルト値)

1 : ハードウェアにより、フラッシュは強制的に STOP モードまたはディープパワーダウンモード (FPDS 値ビットによる) へ移行

注 : このビットは、フラッシュ自体で実行している間はセットできません。RAM から実行される特定のルーチンで実行する必要があります。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:14 **VOS[1:0]** : レギュレータ電圧スケーリング出力選択

このビットによって、メイン内蔵電圧レギュレータの出力電圧を制御し、デバイスが最大周波数で動作していないときに、パフォーマンスと消費電力のトレードオフを実現します (詳細は、対応するデータシートを参照)。

これらのビットを変更できるのは、PLL がオフのときだけです。新しくプログラムされた電圧値は、PLL がオンの場合にのみアクティブになります。PLL がオフの場合、電圧レギュレータは、VOS レジスタの内容に関係なく、スケール 3 にセットされます。

00 : 予約済み (スケール 3 モードが選択)

01 : スケール 3 モード <= 64 MHz

10 : スケール 2 モード (リセット値) <= 84 MHz

11 : スケール 1 モード <= 100 MHz

ビット 13 **ADCD1** :

0 : 影響はありません。

1 : このビットの使い方の詳細は、AN4073 を参照してください。

注 : 電源電圧範囲 2.7~3.6 V で動作し、かつプリフェッチがオフのときにのみ、このビットをセットすることができます。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **MRLVDS** : ディープスリープでのメインレギュレータの低電圧

- 0 : デバイスが STOP モードのとき、メインレギュレータは電圧スケール 3 になります。
- 1 : デバイスが STOP モードのとき、メインレギュレータは低電圧、フラッシュメモリはディープスリープモードになります。

ビット 10 **LPLVDS** : ディープスリープでの低電力レギュレータの低電圧

- 0 : デバイスが STOP モードのとき LPDS ビットがセットされていれば低電力レギュレータはオンになります。
- 1 : デバイスが STOP モードのとき、LPDS ビットがセットされていれば低電力レギュレータは低電圧、フラッシュメモリはディープスリープモードになります。

ビット 9 **FPDS** : STOP モードにおけるフラッシュパワーダウン

- セットされている場合、デバイスが STOP モードに入るとフラッシュメモリがパワーダウンモードに移行します。これにより、STOP モードの消費電力を削減できますが、再起動時間は長くなります。
- 0 : デバイスが STOP モードのときフラッシュメモリはパワーダウンにはならない
- 1 : デバイスが STOP モードのときフラッシュメモリはパワーダウン

ビット 8 **DBP** : バックアップドメイン書き込み保護の無効化

- リセット状態において、RCC_BDCR レジスタ、RTC レジスタ (バックアップレジスタを含む)、PWR_CSR レジスタの BRE ビットは、不要な書き込みアクセスから保護されています。これらのレジスタへの書き込みアクセスを可能にするには、このビットをセットする必要があります。
- 0 : RTC および RTC バックアップレジスタへのアクセス
- 1 : RTC および RTC バックアップレジスタへのアクセス

ビット 7:5 **PLS[2:0]** : PVD レベル選択

- これらのビットは、電源電圧検出器によって検出される電圧閾値を選択するため、ソフトウェアで書き込みます。
- 000 : 2.2 V
- 001 : 2.3 V
- 010 : 2.4 V
- 011 : 2.5 V
- 100 : 2.6 V
- 101 : 2.7 V
- 110 : 2.8 V
- 111 : 2.9 V

注 : 詳細については、データシートの電氣的特性を参照してください。

ビット 4 **PVDE** : 電源電圧検出器有効化

- このビットは、ソフトウェアによってセット/クリアされます。
- 0 : PVD は無効です。
- 1 : PVD は有効です。

ビット 3 **CSBF** : STANDBY フラグのクリア

- このビットは常に 0 として読み出されます。
- 0 : 影響はありません。
- 1 : SBF STANDBY フラグのクリア (書き込み)

ビット 2 **CWUF** : ウェイクアップフラグのクリア

- このビットは常に 0 として読み出されます。
- 0 : 影響はありません。
- 1 : WUF ウェイクアップフラグを 2 システムクロックサイクル後にクリアします。

- ビット 1 **PDDS** : パワーダウンディープスリープ
このビットは、ソフトウェアによってセット／クリアされます。LPDS ビットとの組み合わせで動作します。
0 : CPU がディープスリープに移行すると、STOP モードに移行します。レギュレータの状態は LPDS ビットによって変化します。
1 : CPU がディープスリープに移行すると、STANDBY モードに移行します。
- ビット 0 **LPDS** : 低電力ディープスリープ
このビットは、ソフトウェアによってセット／クリアされます。このビットは PDDS ビットとの組み合わせで動作します
0 : 電圧レギュレータは STOP モードの間もオン状態を保ちます。
1 : 低電力電圧レギュレータは STOP モードの間もオン状態を保ちます。

5.4.2 PWR 電源制御／ステータスレジスタ (PWR_PWR)

アドレスオフセット : 0x04
リセット値 : 0x0000 0000 (STANDBY モードからのウェイクアップによってリセットされない)
このレジスタを読み出すには、通常の APB 読み出しに対して、追加の APB サイクルが必要です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VOS RDY	Res.	Res.	Res.	Res.	BRE	EWUP 1	EWUP 2	EWUP 3	Res.	Res.	BRR	PVDO	SBF	WUF
	r					rw	rw	rw	rw			r	r	r	r

- ビット 31:15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **VOSRDY** : レギュレータ電圧スケーリング出力選択レディビット
0 : レディでない状態
1 : レディ状態
- ビット 13:10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **BRE** : バックアップレギュレータ有効化
このビットがセットされると、バックアップレギュレータ (バックアップドメインの内容を保持するために使用) が有効になります。BRE がリセットされるとバックアップレギュレータはオフになります。いったんセットされると、アプリケーションは、バックアップレギュレータレディフラグ (BRR) が設定され、バックアップレジスタに書き込まれたデータが STANDBY モードおよび V_{BAT} モードにおいて保持されることが示されるのを待つ必要があります。
0 : バックアップレギュレータは無効
1 : バックアップレギュレータは有効
- 注 :** システムリセットや電源リセットによって、デバイスが STANDBY モードからウェイクアップするとき、このビットはリセットされません。
- ビット 8 **EWUP1** : WKUP1 ピン (PA0) 有効化
このビットは、ソフトウェアによってセット／クリアされます。
0 : WKUP1 ピンは汎用 I/O として使用されます。この WKUP1 ピンでイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。
1 : WKUP1 ピンは STANDBY モードからのウェイクアップに使用され、強制的に入力プルダウンモードに設定されます (WKUP1 ピンの立ち上がりエッジで、システムは STANDBY モードからウェイクアップします)。
- 注 :** このビットは、システムリセットによってリセットされます。

ビット 7 EWUP2 : WKUP2 ピン (PC0) 有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : WKUP2 ピンは汎用 I/O として使用されます。この WKUP2 ピンでイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : WKUP2 ピンは STANDBY モードからのウェイクアップに使用され、強制的にプルダウン入力モードに設定されます (WKUP2 ピンの立ち上がりエッジで、システムは STANDBY モードからウェイクアップします)。

注： このビットは、システムリセットによってリセットされます。

ビット 6 EWUP3 : WKUP3 ピン (PC1) 有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : WKUP3 ピンは汎用 I/O として使用されます。この WKUP3 ピンでイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : WKUP3 ピンは STANDBY モードからのウェイクアップに使用され、強制的にプルダウン入力モードに設定されます (WKUP3 ピンの立ち上がりエッジで、システムは STANDBY モードからウェイクアップします)。

注： このビットは、システムリセットによってリセットされます。

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 BRR : バックアップレギュレータレディ

ハードウェアによってセットされ、バックアップレギュレータが動作可能状態であることを示します。

0 : バックアップレギュレータは動作可能状態ではありません。

1 : バックアップレギュレータは動作可能状態です。

注： システムリセットや電源リセットによって、デバイスが STANDBY モードからウェイクアップするとき、このビットはリセットされません。

ビット 2 PVDO : PVD 出力

このビットは、ハードウェアによってセット/クリアされます。PVDE ビットによって PVD が有効化されている場合のみ有効です。

0 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より高いです。

1 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より低いです。

注： PVD は STANDBY モードによって停止します。したがって、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。

ビット 1 SBF : STANDBY フラグ

このビットはハードウェアによってセットされ、POR/PDR (パワーオンリセット/パワーダウンリセット)、または PWR_CR レジスタの CSBF ビットをセットすることによってのみクリアされます。

0 : デバイスは STANDBY モードではありません。

1 : デバイスは STANDBY モードです。

ビット 0 WUF : ウェイクアップフラグ

このビットはハードウェアによってセットされ、システムリセットまたは PWR_CR レジスタの CWUF ビットのセットのいずれかによってクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP ピン、RTC アラーム (アラーム A またはアラーム B)、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップのいずれかよりウェイクアップイベントを受信しています。

注： WKUP ピンのレベルがすでにハイであっても、(EWUP ビットをセットすることによって)WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。

5.5 PWR レジスタマップ

次の表に PWR レジスタの一覧を示します。

表 23. PWR - レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PWR_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FISSR	FMSRR	Res.	Res.	Res.	Res.	Res.	VOS[1:0]	ADCD1	Res.	MRLVDS	LPLVDS	FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	リセット値											0	0					1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	PWR_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOSRDY	Res.	Res.	Res.	Res.	BRE	EWUP1	EWUP2	EWUP3	Res.	Res.	BRR	PVDO	SBF	WUF
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。



6 STM32F412xx のリセットおよびクロック制御 (RCC)

6.1 リセット

リセットには、システムリセット、電源リセット、バックアップドメインリセットの 3 種類があります。

6.1.1 システムリセット

システムリセットは、クロックコントローラ CSR レジスタのリセットフラグとバックアップドメインのレジスタを除き、すべてのレジスタをリセット値に設定します。

システムリセットは、次のイベントのいずれかの発生時に生成されます。

1. NRST ピンのローレベル (外部リセット)
2. ウィンドウ型ウォッチドッグのカウント終了条件 (WWDG リセット)
3. 独立型ウォッチドッグのカウント終了条件 (IWDG リセット)
4. ソフトウェアリセット (SW リセット) ([ソフトウェアリセット](#)を参照)
5. 低電力管理リセット ([低電力管理リセット](#)を参照)

ソフトウェアリセット

リセットソースは、[RCC クロック制御およびステータスレジスタ \(RCC_CSR\)](#) のリセットフラグを確認することによって識別できます。

デバイス上でソフトウェアリセットを実行するには、FPU 搭載 Cortex®-M4 のアプリケーション割り込みおよびリセット制御レジスタの SYSRESETREQ ビットをセットする必要があります。詳細については、FPU 搭載 Cortex®-M4 Technical Reference Manual を参照してください。

低電力管理リセット

低電力管理リセットを生成するには、2つの方法があります。

1. STANDBY モードに入るときに実行されるリセット :
このリセットは、ユーザオプションバイトの nRST_STDBY ビットをリセットすることによって有効になります。この場合、STANDBY モードへの遷移シーケンスが正常に実行されるたびに、STANDBY モードに入る代わりにデバイスがリセットされます。
2. STOP モードに入るときのリセット :
このリセットは、ユーザオプションバイトの nRST_STOP ビットをリセットすることによって有効になります。この場合、STOP モード遷移シーケンスが正常に実行されるたびに、STOP モードに入る代わりにデバイスがリセットされます。

ユーザオプションバイトの詳細については、ST のウェブサイトから入手可能な STM32F412xx フラッシュプログラミングマニュアルを参照してください。

6.1.2 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

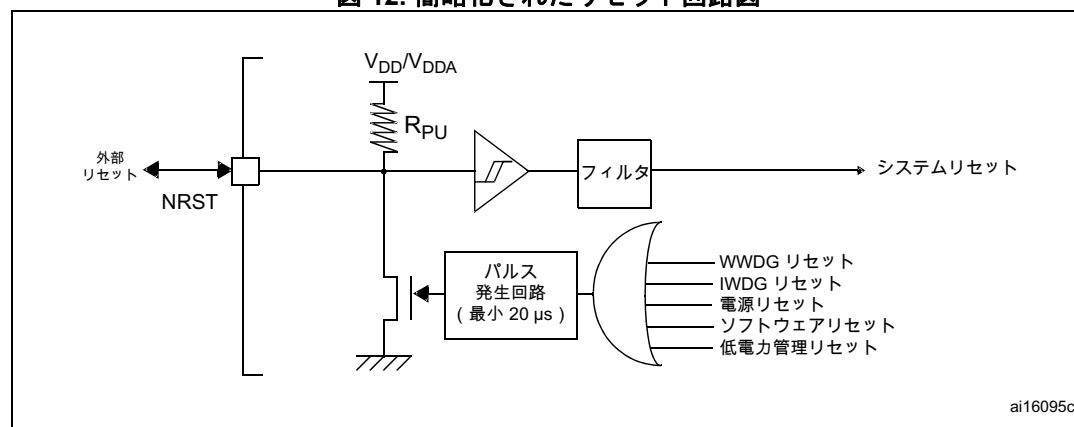
1. パワーオン/パワーダウンリセット (POR/PDR リセット) またはブラウンアウト (BOR) リセット
2. STANDBY モードを終了するとき

電源リセットは、バックアップドメインを除くすべてのレジスタをリセット値に設定します。

これらのソースは NRST ピンに作用し、遅延フェーズの間、ピンをローに保ちます。リセットサービスルーチンベクタは、メモリマップのアドレス 0x0000_0004 に固定されています。

デバイスに与えられるシステムリセット信号は、NRST ピンに出力されます。パルス発生回路は、各内部リセットソースについて 20 μ s の最小リセットパルス期間を保証します。外部リセットの場合、リセットパルスは NRST ピンがローにアサートされているときに生成されます。

図 12. 簡略化されたリセット回路図



6.1.3 バックアップドメインリセット

バックアップドメインリセットは、すべての RTC レジスタと RCC_BDCR レジスタをリセット値にセットします。

バックアップドメインのリセットは、次のいずれかのイベントが発生したときに生成されます。

1. [RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の BDRST ビットをセットすることでトリガされるソフトウェアリセット
2. V_{DD} または V_{BAT} パワーオン。ただし、両方の電源供給がともにオフ状態であった場合。

6.2 クロック

次の 3 つのクロックソースを使用して、システムクロック (SYSCLK) を駆動できます。

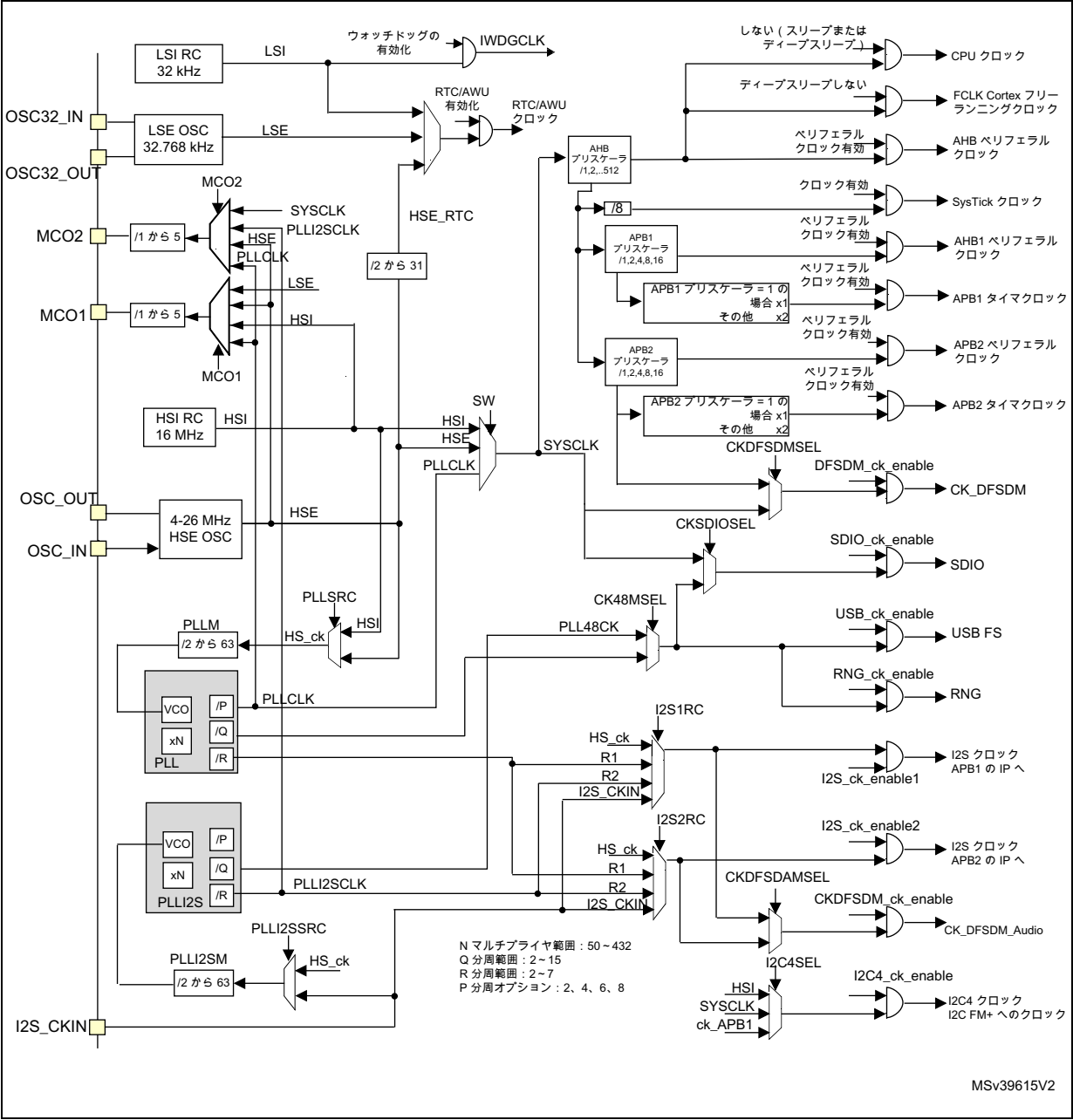
- HSI オシレータクロック
- HSE オシレータクロック
- メイン PLL (PLL) クロック

デバイスには、次の 2 つの 2 次クロックソースがあります。

- 32 kHz 低速内蔵 RC (LSI RC)。独立型ウォッチドッグを駆動し、オプションで、STOP/STANDBY モードからの自動ウェイクアップに使用される RTC を駆動します。
- 32.768 kHz 低速外部クリスタル (LSE クリスタル)。オプションで、RTC クロック (RTCCLK) を駆動します。

それぞれのクロックソースは、使用しないときに個別にオン/オフを切り替えて、電力消費を最適化可能です。

図 13. クロックツリー



1. 内部および外部クロックソース特性の詳細については、デバイスのデータシートの「電気的特性」のセクションを参照してください。

クロックコントローラは、外部クリスタルまたはオシレータの選択においてアプリケーションに高度な柔軟性を提供しており、コアおよびペリフェラルを最大周波数で実行し、USB OTG FS、I2S、SDIO などの特定のクロックを必要とするペリフェラルに適した周波数を保証します。

複数のプリスケアラを使用して AHB 周波数、高速 APB (APB2) および低速 APB (APB1) ドメインの周波数を設定します。AHB ドメインの最大周波数は 100 MHz です。高速 APB2 ドメインの最大許容周波数は 100 MHz です。低速 APB1 ドメインの最大許容周波数は 50 MHz です。

すべてのペリフェラルクロックは、システムクロック (SYSCLK) から生成されます。ただし、次のものを除きます。

- USB OTG FS クロック (48 MHz)、SDIO クロック (≤ 48 MHz)。これらのクロックは、特定の PLL (PLL48CLK) の出力から供給されます。
- I2S クロック。
高品質のオーディオ性能を実現するため、I2S クロックを特定の PLL (PLL12S) または I2S_CKIN ピン上に配置された外部クロックから得ることができます。I2S クロック周波数と精度に関する詳細については、[セクション 26.6.4 : クロックジェネレータ](#)を参照してください。
- I2CFMP1 クロック。HSI、SYSCLK または APB1 クロックから生成することもできます。

RCC は Cortex システムタイマ (SysTick) の外部クロックに、8 分周した AHB クロック (HCLK) を供給します。SysTick は、この分周されたクロックか Cortex クロック (HCLK) のいずれかで動作でき、SysTick 制御およびステータスレジスタで設定可能です。

タイマクロック周波数は、ハードウェアによって自動的に設定されます。RCC_DCKCFGR レジスタの TIMPRE ビットの値により、2 つのケースがあります。

- TIMPRE ビットがリセットされている場合：
APB プリスケアラが分周比 1 に設定されると、タイマクロック周波数 (TIMxCLK) は HCLK に設定されます。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます。 $TIMxCLK = 2 * PCLKx$
- TIMPRE ビットがセットされている場合：
APB プリスケアラが分周比 1 または 2 に設定されると、タイマクロック周波数 (TIMxCLK) は HCLK に設定されます。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 4 倍に設定されます。 $TIMxCLK = 4 * PCLKx$

FCLK は FPU 搭載 Cortex[®]-M4 のフリーランニングクロックとして動作します。詳細については、FPU 搭載 Cortex[®]-M4 Technical Reference Manual を参照してください。

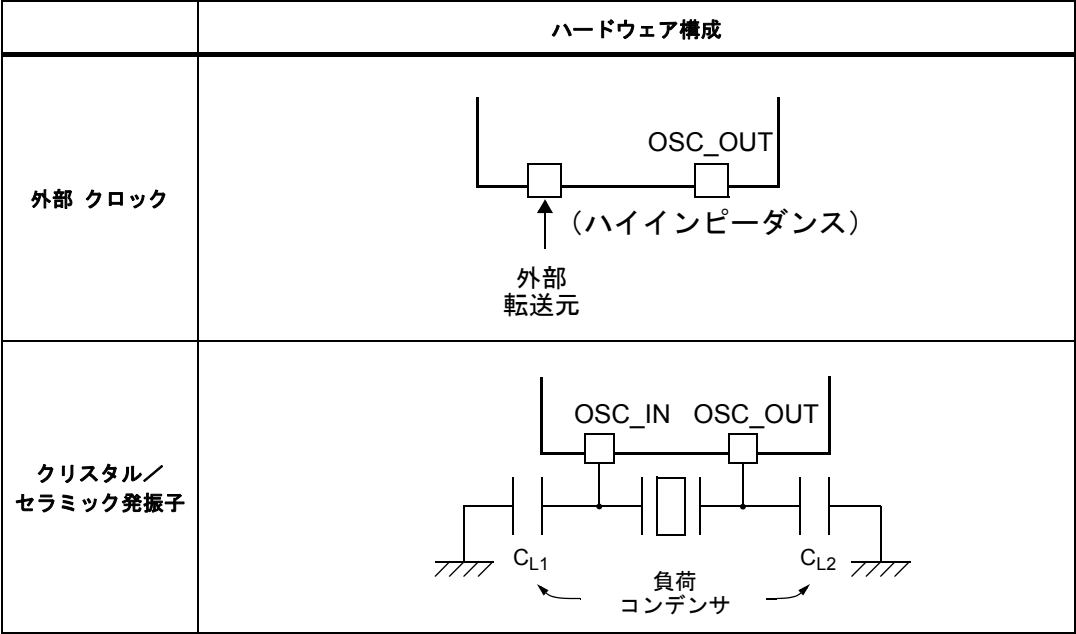
6.2.1 HSE クロック

高速外部クロック信号 (HSE) は、次のどちらかのクロックソースから生成できます。

- HSE 外部クリスタル/セラミック発振子
- HSE 外部ユーザクロック

波形ひずみと発振開始時の安定化までの時間を少なくするために、発振子と負荷コンデンサはオシレータのピンのできるだけ近くに配置する必要があります。負荷コンデンサの値は、選択したオシレータに応じて調整する必要があります。

図 14. HSE/LSE クロックソース



外部ソース (HSE バイパス)

このモードでは、外部クロックソースが必要です。このモードを選択するには、[RCC クロック制御レジスタ \(RCC_CR\)](#) HSEBYP および HSEON の <317 ビットをセットします。約 50 % のデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC_IN ピンを駆動する必要があります。その間、OSC_OUT ピンはハインピーダンスに保つ必要があります ([図 14](#) を参照)。

外部クリスタル／セラミック発振子 (HSE クリスタル)

HSE には、メインクロックの周波数を非常に高い精度で生成できる利点があります。

関連するハードウェア構成を [図 14](#) に示します。詳細については、データシートの電気的特性のセクションを参照してください。

[RCC クロック制御レジスタ \(RCC_CR\)](#) の HSERDY フラグは、外部高速オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、クロックは出力されません。[RCC クロック割り込みレジスタ \(RCC_CIR\)](#) で有効になっていれば、割り込みを生成することができます。

HSE クリスタルは、[RCC クロック制御レジスタ \(RCC_CR\)](#) の HSEON ビットを使用してオン／オフできます。

6.2.2 HSI クロック

HSI クロック信号は、内蔵 16 MHz RC オシレータから生成され、システムクロックとして直接、または PLL 入力として使用できます。

HSI RC オシレータには、低コスト (外付部品なし) でクロックソースを供給できるという利点があります。同時に、HSE クリスタルオシレータよりも起動時間を短縮できますが、較正を実施しても、外部クリスタルオシレータやセラミック発振子よりも周波数の精度は劣ります。

較正

RC オシレータの周波数は、製造工程でのばらつきのため、チップごとに異なります。このため、 $T_A = 25^\circ\text{C}$ で 1 % の精度を確保するよう、製造時に ST 社においてデバイスごとの較正が行われています。

リセット後、工場較正値が [RCC クロック制御レジスタ \(RCC_CR\)](#) の HSICAL[7:0] ビットにロードされます。

アプリケーションが電圧または温度の変動を受ける場合は、これが RC オシレータの速度に影響を与えることがあります。[RCC クロック制御レジスタ \(RCC_CR\)](#) の HSITRIM[4:0] ビットを使用して、アプリケーションで HSI 周波数をトリミングできます。

[RCC クロック制御レジスタ \(RCC_CR\)](#) の HSIRDY フラグは、HSI RC が安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、HSI RC 出力クロックは出力されません。

HSI RC は [RCC クロック制御レジスタ \(RCC_CR\)](#) の HSION ビットを使用してオン/オフの切り替えができます。

HSI 信号は、HSE クリスタルオシレータに障害がある場合のバックアップソース（補助クロック）としても使用できます。[セクション 6.2.7 : クロックセキュリティシステム \(CSS\) \(114 ページ\)](#) を参照してください。

6.2.3 PLL の設定

STM32F412xx デバイスには、次の 2 つの PLL が搭載されています。

- HSE または HSI オシレータで駆動されるメイン PLL (PLL) で、2 種類の異なる出力クロックを備えています。
 - 最初の出力は、高速システムクロック（最大 100 MHz）の生成に使用します。
 - 2 つ目の出力は、USB OTG FS (48 MHz)、RNG、SDIO (≤ 50 MHz) に使用されるクロックの生成に使用します。
- I2S インタフェースで高品質オーディオ性能を実現するための正確なクロックの生成に使用する専用の PLL (PLLI2S)。

メイン PLL 設定パラメータは、いったん PLL を有効にすると変更できないため、PLL を有効にする前に設定することをお勧めします (PLL クロックソースとしての HSI または HSE オシレータの選択および分周比 M、P、Q および逡倍数 N の設定)。

PLLI2S は、メイン PLL と同じ入力クロックを使用します (HSI または HSE)。しかしながら、PLLI2S には有効/無効および分周比を設定する専用のビットがあります。[セクション 6.3.1 : RCC クロック制御レジスタ \(RCC_CR\)](#)、[セクション 6.3.2 : RCC PLL 設定レジスタ \(RCC_PLLCFGR\)](#)、および [セクション 6.3.23 : RCC PLLI2S 設定レジスタ \(RCC_PLLI2SCFGR\)](#) を参照してください。PLLI2S が有効になると、設定パラメータは変更できません。

STOP モードおよび STANDBY モードに入るとき、HSE または PLL (HSE で駆動される) がシステムクロックとして使用されているときに HSE で障害が発生した場合、2 つの PLL はハードウェアによって無効化されます。[RCC PLL 設定レジスタ \(RCC_PLLCFGR\)](#)、および [RCC クロック設定レジスタ \(RCC_CFGR\)](#) を使用して、PLL、PLLI2S をそれぞれ設定できます。

6.2.4 LSE クロック

LSE クロックは、32.768kHz の低速外部クリスタルまたはセラミック発振子で生成されます。時計/カレンダー、その他のタイミング機能のためのリアルタイムクロックペリフェラル (RTC) に、低電力ながら高精度のクロックソースを供給できるという利点があります。



LSE オシレータは、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEON ビットを使用してオン/オフの切り替えができます。

[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の LSERDY フラグは、LSE クリスタルが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、LSE クリスタル出力クロック信号はリリースされません。[RCC クロック割り込みレジスタ \(RCC_CIR\)](#) で有効になっていれば、割り込みを生成することができます。

外部ソース (LSE バイパス)

このモードでは、外部クロックソースが必要です。最大 1 MHz までの周波数を使用する必要があります。このモードを選択するには、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEBYP および LSEON ビットをセットします。約50%のデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC32_IN ピンを駆動する必要があり、その間、OSC32_OUT ピンはハイインピーダンスに保つ必要があります (図 14 を参照)。

6.2.5 LSI クロック

LSI RC は、独立型ウォッチドッグ (IWDG) や自動ウェイクアップユニット (AWU) のために STOP モードおよび STANDBY モードで動作する低電力のクロックソースとして動作します。クロック周波数は約 32 kHz です。詳細については、データシートの電気的特性のセクションを参照してください。

LSI RC は、[RCC クロック制御およびステータスレジスタ \(RCC_CSR\)](#) の LSION ビットを使用してオン/オフの切り替えができます。

[RCC クロック制御およびステータスレジスタ \(RCC_CSR\)](#) の LSIRDY フラグは、内部低速オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、クロックは出力されません。[RCC クロック割り込みレジスタ \(RCC_CIR\)](#) で有効になっていれば、割り込みを生成することができます。

6.2.6 システムクロック (SYSCLK) の選択

システムリセット後は、HSI オシレータがシステムクロックとして選択されています。クロックソースが直接、または PLL を経由してシステムクロックとして使用されているときには、このクロック信号を停止することはできません。

あるクロックソースから別のクロックソースへの切り替えは、切り替え後に使用するクロックソースの準備ができていない場合 (起動遅延時間を経てクロックが安定している状態、または、PLL がロックされている状態) にのみ行われます。準備ができていないクロックソースが選択された場合は、クロックソースの準備ができたときに切り替えが行われます。[RCC クロック制御レジスタ \(RCC_CR\)](#) のステータスビットは、どのクロックの準備ができていないか、およびどのクロックがシステムクロックとして使用されているかを示します。

6.2.7 クロックセキュリティシステム (CSS)

クロックセキュリティシステムはソフトウェアで有効にできます。この場合、HSE オシレータのスタートアップ遅延時間の後にクロック検出回路が有効になり、オシレータが停止すると検出回路も無効になります。

HSE クロックに異常が検出された場合、このオシレータは自動的に無効になり、クロック異常イベントが高機能制御タイマTIM1のブレーク入力に送られます。また、ソフトウェアに異常を通知する割り込み (クロックセキュリティシステム割り込み (CSSI)) が生成されるので、MCU は対応処理を行うことができます。CSSI は、FPU 搭載 Cortex[®]-M4 NMI (ノンマスカブル割り込み) 例外ベクタにリンクされています。

注： CSS を有効にすると、HSE クロックに異常が発生した場合、CSS が割り込みを生成し、結果として NMI の自動生成を行うこととなります。NMI は、CSS 割り込みペンディングビットがクリアされない限り、無限に実行されます。つまり、アプリケーションで、クロック割り込みレジスタ (RCC_CIR) の CSSC ビットをセットすることにより NMI 割り込みサービスルーチンで CSS 割り込みをクリアする必要があります。

HSE オシレータが直接または間接的にシステムクロックとして使用されている場合（間接的とは、PLL 入力クロックとして直接使用されていて、その PLL クロックがシステムクロックであることを意味します）、異常が検出されると、システムクロックが HSI オシレータに切り替えられ、HSE オシレータは無効になります。

異常発生時に HSE オシレータクロックがシステムクロックとして使用されている PLL のクロックソースであった場合には、PLL も無効になります。この場合、PLLI2S が有効になっていても HSE 異常が発生すると PLLI2S も無効になります。

6.2.8 RTC/AWU クロック

いったん RTCCLK クロックソースが選択されると、電源ドメインをリセットしない限りその選択を変更することはできません。

RTCCLK クロックソースは、HSE 1 MHz（HSE をプログラマブルプリスケアラで分周）、LSE または LSI クロックのいずれかとなります。これは、[RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1:0] ビットおよび [RCC クロック設定レジスタ \(RCC_CFGR\)](#) の RTCPRE[4:0] ビットをプログラムすることによって、選択できます。この選択は、バックアップドメインをリセットしない限り変更できません。

RTC クロックとして LSE が選択されている場合、バックアップまたはシステム電源が無くなっても RTC は正常に動作します。AWU クロックとして LSI が選択されている場合、システム電源がなくなると AWU の状態は保証されません。RTC クロックとして 2 周分から 31 周分の間で分周された HSE オシレータクロックを選択している場合、バックアップまたはシステム電源が無くなると RTC の状態は保証されません。

LSE クロックはバックアップドメインに属しますが、HSE と LSI クロックはそうではありません。その結果、以下ようになります。

- LSE が RTC クロックとして使用されている場合：
 - － V_{DD} 供給がオフになった場合でも、 V_{BAT} の供給が保たれている限り、RTC は動作を続けます。
 - － システムリセット中、RTC はクロック供給および機能を保持します。
- LSI が自動ウェイクアップユニット (AWU) クロックとして選択されている場合：
 - － V_{DD} 供給がオフになった場合、AWU の状態は保証されません。LSI 較正の詳細については、[セクション 6.2.5：LSI クロック](#) を参照してください。
- HSE クロックが RTC クロックとして使用されている場合：
 - － V_{DD} 供給がオフになった場合や、内蔵されている電圧レギュレータがオフになった場合（1.2 V 電源ドメインへの電力供給が停止にされた場合）、RTC の状態は保証されません。

注： APB1 クロック周波数が RTC クロック周波数の 7 倍未満である場合 ($f_{APB1} < 7 \times f_{RTCCLK}$) に RTC カレンダーレジスタを読み込むには、ソフトウェアはカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR への 2 回目の読み出しアクセスの結果が最初の読み出しアクセスの結果と同一であれば、データは正しいということになります。同じでない場合は、3 回目の読み出しアクセスを実行する必要があります。



6.2.9 ウォッチドッグクロック

独立型ウォッチドッグ (IWDG) がハードウェアのオプションまたはソフトウェアアクセスによって開始された場合、LSI オシレータは強制的にオンになり、オフにすることはできません。LSI オシレータの起動時の過渡期間が終わると、このクロックが IWDG に供給されます。

6.2.10 クロック信号出力

2 つのマイクロコントローラクロック出力 (MCO) ピンが利用できます。

● MCO1

設定可能なプリスケアラ (1 から 5) を使用して、MCO1 ピン (PA8) から 4 種類のクロックソースを出力できます。

- HSI クロック
- LSE クロック
- HSE クロック
- PLL クロック

[RCC クロック設定レジスタ \(RCC_CFGR\)](#) の MCO1PRE[2:0] および MCO1[1:0] ビットを使用して、希望するクロックソースが選択されます。

● MCO2

設定可能なプリスケアラ (1 から 5) を使用して、MCO2 ピン (PC9) から 4 種類のクロックソースを出力できます。

- HSE クロック
- PLL クロック
- システムクロック (SYSCLK)
- PLLI2S クロック

[RCC クロック設定レジスタ \(RCC_CFGR\)](#) の MCO2PRE[2:0] および MCO2 ビットを使用して、希望するクロックソースが選択されます。

異なる MCO ピンに対し、対応する GPIO ポートをオルタネート機能モードにプログラムする必要があります。

選択されたクロックの MCO への出力が 100 MHz (最大 I/O 速度) を超えないようにしてください。

6.2.11 TIM5/TIM11 を使用した内部／外部クロックの測定

[図 15](#) および [図 16](#) に示すように、TIM5 チャンネル 4 および TIM11 チャンネル 1 の入力キャプチャを使用することにより、ボード上のすべてのクロックソース発生回路の周波数を間接的に測定することができます。

TIM5 チャンネル 4 を使用した内部／外部クロックの測定

TIM5 には、入力キャプチャが I/O によってトリガされるのか内部クロックによってトリガされるのかを選択できる入力マルチプレクサがあります。この選択は、TIM5_OR レジスタの TI4_RMP [1:0] ビットで行います。

LSE を チャンネル 4 入力キャプチャに接続しておく主な目的は、HSI を正確に測定できるようにすることです (このとき HSI をシステムクロックソースとして使用する必要があります)。LSE 信号の連続したエッジ間の HSI クロックカウント数により、内部クロック周期が測定されます。LSE クリスタルの高い精度 (通常は数十 ppm) を利用することにより、同じ分解能で内部クロック周波数を判断でき、製造工程、温度、電圧に関連した周波数偏差に対してクロックソースを調整して補正することができます。

HSI オシレータには、上記の目的のためにユーザがアクセスできる専用の較正ビットがあります。

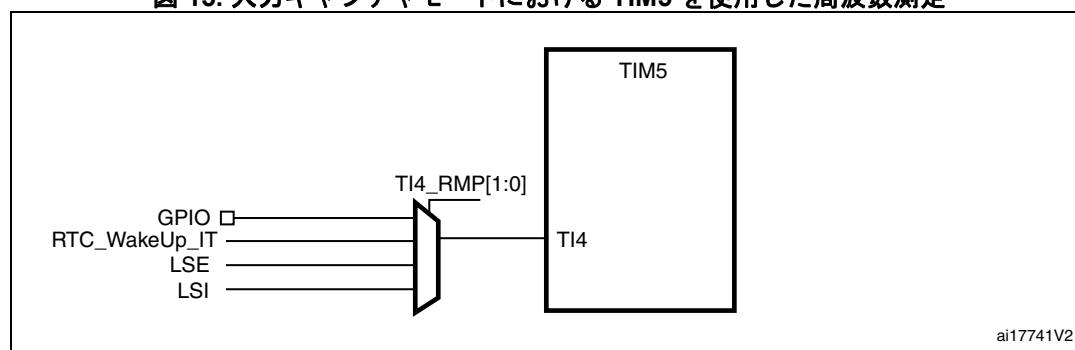
この基本概念は、相対測定 (HSI/LSE 比など) ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接につながっています。この比が大きければ大きいほど、測定精度は向上します。

LSI 周波数も測定できますので、クリスタルを使用しないアプリケーションで有用です。超低電力 LSI オシレータには大きな製造工程偏差があります。これを HSI クロックソースに対して測定することにより、HSI の精度で LSI の周波数を確定することができます。測定値を使用して、より正確な RTC 時間基準のタイムアウト (LSI が RTC クロックソースとして使用されているとき) や許容精度の IWDG タイムアウトを得ることができます。

LSI 周波数を測定するには、次の手順に従ってください。

1. TIM5 タイマを有効にして、チャンネル 4 を入力キャプチャモードに設定します。
2. TIM5_OR レジスタの TI4_RMP ビットを 0x01 にセットして、較正目的で、LSI クロックを TIM5 チャンネル 4 入力キャプチャに内部接続します。
3. TIM5 キャプチャ/比較 4 イベントまたは割り込みを使用して、LSI クロックの周波数を計測します。
4. 計測した LSI 周波数を使用して、必要な時間基準に従って RTC のプリスケアラを更新したり、IWDG タイムアウトを計算したりします。

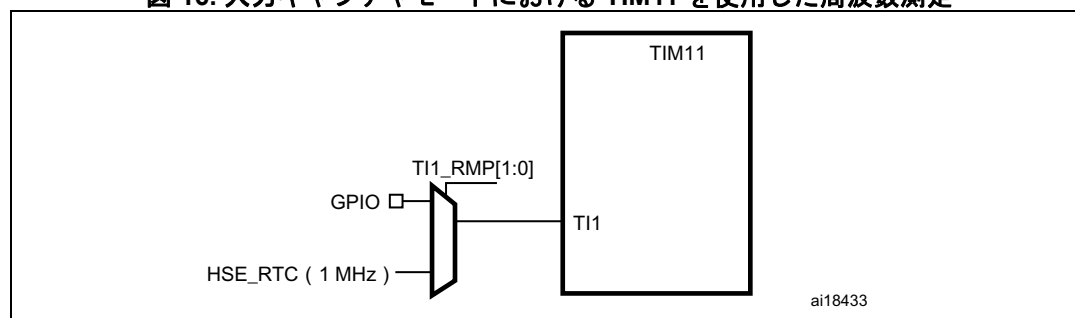
図 15. 入力キャプチャモードにおける TIM5 を使用した周波数測定



TIM11 チャンネル 1 を使用した内部／外部クロックの測定

TIM11 には、入力キャプチャが I/O によってトリガされるのか内部クロックによってトリガされるのかを選択できる入力マルチプレクサがあります。この選択は、TIM11_OR レジスタの TI1_RMP [1:0] ビットで行います。HSE_RTC クロック (HSE をプログラマブルプリスケーラで分周) は、チャンネル 1 の入力キャプチャに接続されており、およその外部クリスタル周波数を知ることができます。このとき、HSI がシステムクロックソースである必要があります。これは、たとえば高調波周波数または低調波周波数 (−50/+100 % の偏差) が判断できることを求めている IEC 60730/IEC 61335 規格に準拠していることを保証するのに有用です。

図 16. 入力キャプチャモードにおける TIM11 を使用した周波数測定



6.3 RCC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1: レジスタに関する略記](#)を参照してください。

6.3.1 RCC クロック制御レジスタ (RCC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 XX81 (X は未定義)。

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PLLI2S RDY	PLLI2S ON	PLLRDY	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
				r	rw	r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **PLLI2SRDY** : PLLI2S クロックレディフラグ

ハードウェアによってセットされ、PLLI2S がロック状態であることを示します。

0 : PLLI2S アンロック

1 : PLLI2S ロック

ビット 26 **PLLI2SON** : PLLI2S イネーブル

PLLI2S を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。

0 : PLLI2S オフ

1 : PLLI2S オン

ビット 25 **PLLRDY** : メイン PLL (PLL) クロックレディフラグ

ハードウェアによってセットされ、PLL がロック状態であることを示します。

0 : PLL アンロック

1 : PLL ロック

ビット 24 **PLLON** : メイン PLL (PLL) 有効化

PLL を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。PLL クロックがシステムクロックとして使用されている場合、このビットをリセットできません。

0 : PLL オフ

1 : PLL オン

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CSSON** : クロックセキュリティシステム有効化

クロックセキュリティシステムを有効にするために、ソフトウェアによってセット/クリアされます。CSSON がセットされているとき、HSE オシレータがレディになるとクロック検出回路がハードウェアによって有効にされ、オシレータの障害が検出された場合ハードウェアによって無効にされます。

0 : クロックセキュリティシステムオフ (クロック検出回路オフ)

1 : クロックセキュリティシステムオン (HSE オシレータが安定していればクロック検出回路オン、そうでなければオフ)



ビット 18 HSEBYP : HSE クロックバイパス

外部クロックでオシレータをバイパスするために、ソフトウェアによってセット／クリアされます。デバイスで使用するには、HSEON ビットで外部クロックを有効にする必要があります。

HSEBYP ビットは、HSE オシレータが無効のときのみ、書き込みができます。

0 : HSE オシレータはバイパスされません。

1 : HSE オシレータはバイパスされ、外部クロックが使用されます。

ビット 17 HSERDY : HSE クロックレディフラグ

HSE オシレータが安定していることを示すために、ハードウェアによってセットされます。HSEON ビットがクリアされた後、HSERDY は、HSE オシレータクロックの 6 サイクル後にローになります。

0 : HSE オシレータはレディ状態ではありません。

1 : HSE オシレータはレディ状態です。

ビット 16 HSEON : HSE クロック有効化

ソフトウェアでセット／クリアされます。

STOP または STANDBY モードに入るときに、HSE オシレータを停止するためにハードウェアによってクリアされます。HSE オシレータが直接的または間接的にシステムクロックとして使用されている場合は、このビットをリセットできません。

0 : HSE オシレータオフ

1 : HSE オシレータオン

ビット 15:8 HSICAL[7:0] : 内部高速クロック較正

これらのビットは、スタートアップ時に自動的に初期化されます。

ビット 7:3 HSITRIM[4:0] : 内部高速クロックトリミング

これらのビットにより、ユーザプログラミング可能なトリミング値が使用でき、この値は HSICAL[7:0] ビットに加算されます。内蔵 HSI RC の周波数に影響する電圧や温度の変化に対応できるようにプログラミングできます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 HSIRDY : 内部高速クロックレディフラグ

HSI オシレータが安定していることを示すために、ハードウェアによってセットされます。HSION ビットがクリアされた後、HSIRDY は、HSI オシレータクロックの 6 サイクル後にローになります。

0 : HSI オシレータはレディ状態ではありません。

1 : HSI オシレータはレディ状態です。

ビット 0 HSION : 内部高速クロック有効化

ソフトウェアでセット／クリアされます。

STOP または STANDBY モードを終了するとき、または、直接または間接的にシステムクロックとして使用されている HSE オシレータに障害が発生した場合、HSI オシレータを強制的にオンにするために、ハードウェアによってセットされます。HSI オシレータが直接または間接的にシステムクロックとして使用されている場合は、このビットをクリアできません。

0 : HSI オシレータオフ

1 : HSI オシレータオン

6.3.2 RCC PLL 設定レジスタ (RCC_PLLCFGR)

アドレスオフセット : 0x04

リセット値 : 0x2400 3010

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

このレジスタは、次の式に従って PLL クロック出力を設定するために使用します。

- $f_{\text{(VCO クロック)}} = f_{\text{(PLL クロック入力)}} \times (\text{PLL N} / \text{PLL M})$
- $f_{\text{(PLL 汎用クロック出力)}} = f_{\text{(VCO クロック)}} / \text{PLL P}$
- $f_{\text{(USB OTG FS, SDIO, RNG クロック出力)}} = f_{\text{(VCO クロック)}} / \text{PLL Q}$
- $f_{\text{(I2S, DFSDM クロック出力)}} = f_{\text{(VCO クロック)}} / \text{PLL R}$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLL R[2:0]			PLL Q[3:0]				Res.	PLL SRC	Res.	Res.	Res.	Res.	PLL P[1:0]	
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w					r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLL N[8:0]								PLL M[5:0]						
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **PLL R[2:0]** : I2S、DFSDM クロックのためのメイン PLL (PLL) 分周比

クロックの周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。

クロック周波数 = VCO 周波数 / PLL R、ここで $2 \leq \text{PLL R} \leq 7$

000 : PLL R = 0、誤った設定

001 : PLL R = 1、誤った設定

010 : PLL R = 2

011 : PLL R = 3

...

111 : PLL R = 7

ビット 27:24 **PLL Q[3:0]** : USB OTG FS、SDIO、および乱数発生回路 (RNG) クロックのためのメイン PLL (PLL) 分周比

USB OTG FS クロック、乱数発生回路クロック、SDIO クロックの周波数を制御するためにソフトウェアによってセット/クリアします。これらのビットに書き込めるのは、PLL が無効なときだけです。

注意 : **USB OTG FS が正しく動作するには、48 MHz のクロックが必要です。SDIO および乱数発生回路が正しく動作するには、48 MHz 以下の周波数が必要です。**

USB OTG FS クロック周波数 = VCO 周波数 / PLL Q。ここで $2 \leq \text{PLL Q} \leq 15$

0000 : PLL Q = 0、誤った設定

0001 : PLL Q = 1、誤った設定

0010 : PLL Q = 2

0011 : PLL Q = 3

0100 : PLL Q = 4

...

1111 : PLL Q = 15

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PLL SRC** : メイン PLL (PLL) およびオーディオ PLL (PLLI2S) 入力クロックソース

PLL および PLLI2S クロックソースを選択するために、ソフトウェアによってセット/クリアされます。このビットに書き込めるのは、PLL および PLLI2S が無効のときだけです。

0 : PLL および PLLI2S クロック入力として HSI クロックを選択

1 : PLL および PLLI2S クロック入力として HSE オシレータクロックを選択

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **PLL P[1:0]** : メインシステムクロックのためのメイン PLL (PLL) 分周比

汎用 PLL 出力クロックの周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。

注意 : **ソフトウェアは、このドメインで 100 MHz を超えないように、これらのビットを正しくセットする必要があります。**

PLL 出力クロック周波数 = VCO 周波数 / PLLP ここで PLLP = 2、4、6、または 8

00 : PLLP = 2

01 : PLLP = 4

10 : PLLP = 6

11 : PLLP = 8

ビット 14:6 **PLL N[8:0]** : メイン PLL (PLL) の VCO に対する通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

注意 : **これらのビットは、VCO 出力周波数が 100 ~ 432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。(セクション 6.3.23 : RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR) も参照)**

VCO 出力周波数 = VCO 入力周波数 × PLLN、ここで $50 \leq \text{PLLN} \leq 432$

000000000 : PLLN = 0、誤った設定

000000001 : PLLN = 1、誤った設定

...

000110010 : PLLN = 50

...

001100011 : PLLN = 99

001100100 : PLLN = 100

...

110110000 : PLLN = 432

110110001 : PLLN = 433、誤った設定

...

111111111 : PLLN = 511、誤った設定

注 : **1 MHz を超える VCO 入力周波数になりうる通倍数ですが、上記に指定された VCO 出力周波数範囲を満たすことに注意する必要があります。**

ビット 5:0 **PLLM[5:0]** : メイン PLL (PLL) 入力クロックのための分周比

PLL および PLLI2S 入力クロックを VCO の前で分周するために、ソフトウェアによってセット／クリアされます。これらのビットに書き込めるのは、PLL および PLLI2S が無効のときだけです。

注意 : **これらのビットは、VCO 入力周波数が 1~2 MHz の間になるように、ソフトウェアで正しく設定する必要があります。PLL ジッタを制限するため 2 MHz の周波数を選択することが推奨されます。**

VCO 入力周波数 = PLL 入力クロック周波数 / PLLM、ここで $2 \leq PLLM \leq 63$

000000 : PLLM = 0、誤った設定
000001 : PLLM = 1、誤った設定
000010 : PLLM = 2
000011 : PLLM = 3
000100 : PLLM = 4
...
111110 : PLLM = 62
111111 : PLLM = 63

6.3.3 RCC クロック設定レジスタ (RCC_CFGR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : $0 \leq \text{ウェイトステート} \leq 2$ 、ワード、ハーフワード、およびバイトアクセス

クロックソースの切り替え中にアクセスが発生した場合に限り、1 または 2 個のウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2[1:0]		MCO2 PRE[2:0]			MCO1 PRE[2:0]			Res.	MCO1[1:0]		RTCPRE[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRE2[2:0]			PPRE1[2:0]			Res.	Res.	HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	r	r	rw	rw

ビット 31:30 **MCO2[1:0]** : マイクロコントローラクロック出力 2

ソフトウェアでセット／クリアされます。クロックソースの選択によって、MCO2 でグリッチが発生することがあります。これらのビットは、リセット後に外部オシレータおよび PLL を有効にする前に設定することを強く推奨します。

00 : システムクロック (SYSCLK) の選択
01 : PLLI2S クロックの選択
10 : HSE オシレータクロックの選択
11 : PLL クロックの選択

ビット 29:27 MCO2PRE[1:0] : MCO2 プリスケアラ

MCO2 のプリスケアラを設定するために、ソフトウェアによってセット/クリアされます。このプリスケアラの変更によって、MCO2 でグリッチが発生することがあります。このプリスケアラは、リセット後に外部オシレータおよび PLL を有効にする前に変更することを強く推奨します。

0xx : 分周なし
100 : 分周比 2
101 : 分周比 3
110 : 分周比 4
111 : 分周比 5

ビット 26:24 MCO1PRE[1:0] : MCO1 プリスケアラ

MCO1 のプリスケアラを設定するために、ソフトウェアによってセット/クリアされます。このプリスケアラの変更によって、MCO1 でグリッチが発生することがあります。このプリスケアラは、リセット後に外部オシレータおよび PLL を有効にする前に変更することを強く推奨します。

0xx : 分周なし
100 : 分周比 2
101 : 分周比 3
110 : 分周比 4
111 : 分周比 5

ビット 23 予約済み、常に 0 として読み出されます。

ビット 22:21 MCO1[1:0] : マイクロコントローラクロック出力 1

ソフトウェアでセット/クリアされます。クロックソースの選択によって、MCO1 でグリッチが発生することがあります。これらのビットは、リセット後に外部オシレータおよび PLL を有効にする前に設定することを強く推奨します。

00 : HSI クロックの選択
01 : LSE オシレータの選択
10 : HSE オシレータクロックの選択
11 : PLL クロックの選択

ビット 20:16 RTCPRE[4:0] : RTC クロックのための HSE 分周比

HSE クロック入力クロックを分周し、RTC に 1 MHz クロックを生成するために、ソフトウェアによってセット/クリアされます。

注意： これらのビットは、RTC に供給されるクロックが 1 MHz となるように、ソフトウェアで正しく設定する必要があります。これらのビットは、必要に応じて RTC クロックソースが選択される前に設定する必要があります。

00000 : クロックなし
00001 : クロックなし
00010 : HSE/2
00011 : HSE/3
00100 : HSE/4
...
11110 : HSE/30
11111 : HSE/31

ビット 15:13 PPRE2[2:0] : APB 高速プリスケアラ (APB2)

APB 高速クロックの分周比を制御するために、ソフトウェアによってセット/クリアされます。

注意： ソフトウェアは、このドメインで 100 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE2 に書き込んでから 1~16 AHB の間の新しい分周比で分周されます。

0xx : AHB クロックは分周されません。
100 : AHB クロックは 2 分周されます。
101 : AHB クロックは 4 分周されます。
110 : AHB クロックは 8 分周されます。
111 : AHB クロックは 16 分周されます。

ビット 12:10 **PPRE1[2:0]** : APB低速プリスケアラ (APB1)

APB 低速クロックの分周比を制御するために、ソフトウェアによってセット/クリアされます。

注意: **ソフトウェアは、このドメインで 50 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE1 に書き込んだから 1~16 AHB の間の新しい分周比で分周されます。**

0xx : AHB クロックは分周されません。

100 : AHB クロックは 2 分周されます。

101 : AHB クロックは 4 分周されます。

110 : AHB クロックは 8 分周されます。

111 : AHB クロックは 16 分周されます。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **HPRE[3:0]** : AHB プリスケアラ

AHB クロックの分周比を制御するために、ソフトウェアによってセット/クリアされます。

注意: **クロックは、HPRE に書き込んだ後 1 から 16 の AHB サイクル後に新しい分周比で分周されます。**

注意: **イーサネットが使用されるときには、AHB クロック周波数は少なくとも 25 MHz でなければなりません。**

0xxx : システムクロックは分周されません。

1000 : システムクロックは 2 分周されます。

1001 : システムクロックは 4 分周されます。

1010 : システムクロックは 8 分周されます。

1011 : システムクロックは 16 分周されます。

1100 : システムクロックは 64 分周されます。

1101 : システムクロックは 128 分周されます。

1110 : システムクロックは 256 分周されます。

1111 : システムクロックは 512 分周されます。

ビット 3:2 **SWS[1:0]** : システムクロックスイッチステータス

どのクロックソースがシステムクロックとして使用されているかを示すために、ハードウェアによってセット/クリアされます。

00 : HSI オシレータがシステムクロックとして使用されています。

01 : HSE オシレータがシステムクロックとして使用されています。

10 : PLL がシステムクロックとして使用されています。

11 : 適用されません。

ビット 1:0 **SW[1:0]** : システムクロックスイッチ

システムクロックソースを選択するために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードを終了するとき、または、直接的または間接的にシステムクロックとして使用されている HSE オシレータに障害が発生した場合、強制的に HSI を選択するために、ハードウェアによってセットされます。

00 : HSI オシレータがシステムクロックとして選択されます。

01 : HSE オシレータがシステムクロックとして選択されます。

10 : PLL がシステムクロックとして選択されます。

11 : 設定禁止

6.3.4 RCC クロック割り込みレジスタ (RCC_CIR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	Res.	PLLI2S RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
								w		w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PLLI2S RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	Res.	PLLI2S RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **CSSC** : クロックセキュリティシステム割り込みクリア

このビットは、CSSF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : CSSF フラグをクリアします。

ビット 22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **PLLI2SRDYC** : PLLI2S レディ割り込みクリア

このビットは、PLLI2SRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : PLLI2SRDYF がクリアされます。

ビット 20 **PLLRDYC** : メイン PLL (PLL) レディ割り込みクリア

このビットは、PLLRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : PLLRDYF がクリアされます。

ビット 19 **HSERDYC** : HSE レディ割り込みクリア

このビットは、HSERDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSERDYF がクリアされます。

ビット 18 **HSIRDYC** : HSI レディ割り込みクリア

このビットは、HSIRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSIRDYF がクリアされます。

ビット 17 **LSERDYC** : LSE レディ割り込みクリア

このビットは、LSERDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : LSERDYF がクリアされます。

ビット 16 **LSIRDYC** : LSI レディ割り込みクリア

このビットは、LSIRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : LSIRDYF がクリアされます。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **PLLI2SRDYIE** : PLLI2S レディ割り込み有効化

PLLI2S ロックによって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : PLLI2S ロック割り込み無効

1 : PLLI2S ロック割り込み有効

ビット 12 **PLLRDYIE** : メイン PLL (PLL) レディ割り込み有効化

PLL ロックによって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : PLL ロック割り込みは無効です。

1 : PLL ロック割り込みは有効です。

ビット 11 **HSERDYIE** : HSE レディ割り込み有効化

HSE オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : HSE レディ割り込みは無効です。

1 : HSE レディ割り込みは有効です。

ビット 10 **HSIRDYIE** : HSI レディ割り込み有効化

HSI オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : HSI レディ割り込み無効

1 : HSI レディ割り込み有効

ビット 9 **LSE RDYIE** : LSE レディ割り込み有効化

LSE オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : LSE レディ割り込みは無効です。

1 : LSE レディ割り込みは有効です。

ビット 8 **LSIRDYIE** : LSI レディ割り込み有効化

LSI オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : LSI レディ割り込みは無効です。

1 : LSI レディ割り込みは有効です。

ビット 7 **CSSF** : クロックセキュリティシステム割り込みフラグ

HSE オシレータで障害が検出されたときに、ハードウェアによってセットされます。

CSSC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSE クロック障害によるクロックセキュリティ割り込みは発生していません。

1 : HSE クロック障害によってクロックセキュリティ割り込みが発生しました。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PLLI2SRDYF** : PLLI2S レディ割り込みフラグ

PLLI2S がロックされ、PLLI2SRDYIE がセットされているときに、ハードウェアによってセットされます。

PLLRISDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : PLLI2S ロックによるクロックレディ割り込みは発生していません。

1 : PLLI2S ロックによるクロックレディ割り込みが発生しました。

ビット 4 **PLLRDYF** : メイン PLL (PLL) レディ割り込みフラグ

PLL がロックされ、PLLRDYIE がセットされているときに、ハードウェアによってセットされます。

PLLRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : PLL ロックによるクロックレディ割り込みは発生していません。

1 : PLL ロックによるクロックレディ割り込みが発生しました。

ビット 3 **HSERDYF** : HSE レディ割り込みフラグ

外部高速クロックが安定し、HSERDYDIE ビットがセットされているとき、ハードウェアによってセットされます。

HSERDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSE オシレータによるクロックレディ割り込みは発生していません。

1 : HSE オシレータによるクロックレディ割り込みが発生しました。

ビット 2 **HSIRDYF** : HSI レディ割り込みフラグ

内部高速クロックが安定し、HSIRDYDIE ビットがセットされているとき、ハードウェアによってセットされます。

HSIRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSI オシレータによるクロックレディ割り込みは発生していません。

1 : HSI オシレータによるクロックレディ割り込みが発生しました。

ビット 1 **LSERDYF** : LSE レディ割り込みフラグ

外部低速クロックが安定し、LSERDYDIE ビットがセットされているとき、ハードウェアによってセットされます。

LSERDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : LSE オシレータによるクロックレディ割り込みは発生していません。

1 : LSE オシレータによるクロックレディ割り込みが発生しました。

ビット 0 **LSIRDYF** : LSI レディ割り込みフラグ

内部低速クロックが安定し、LSIRDYDIE ビットがセットされているとき、ハードウェアによってセットされます。

LSIRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : LSI オシレータによるクロックレディ割り込みは発生していません。

1 : LSI オシレータによるクロックレディ割り込みが発生しました。

6.3.5 RCC AHB1 ペリフェラルリセットレジスタ (RCC_AHB1RSTR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 RST	DMA1 RST	Res.	Res.	Res.	Res.	Res.
									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	Res.	GPIOH RST	GPIOG RST	GPIOF RST	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST
			rw					rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DMA2RST** : DMA2 リセット

ソフトウェアでセット／クリアされます。

0 : DMA2 をリセットしません。

1 : DMA2 をリセットします。

ビット 21 **DMA1RST** : DMA1 リセット

ソフトウェアでセット／クリアされます。

0 : DMA1 をリセットしません。

1 : DMA1 をリセットします。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCRST** : CRC リセット

ソフトウェアでセット／クリアされます。

0 : CRC をリセットしません。

1 : CRC をリセットします。

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHRST** : IO ポート H リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート H をリセットしません。

1 : IO ポート H をリセットします。

ビット 6 **GPIOGRST** : IO ポート G リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート G をリセットしません。

1 : IO ポート G をリセットします。

ビット 5 **GPIOFRST** : IO ポート F リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート F をリセットしません。

1 : IO ポート F をリセットします。

ビット 4 **GPIOERST** : IO ポート E リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート E をリセットしません。

1 : IO ポート E をリセットします。

ビット 3 **GPIODRST** : IO ポート D リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート D をリセットしません。

1 : IO ポート D をリセットします。

ビット 2 **GPIOCRST** : IO ポート C リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート C をリセットしません。

1 : IO ポート C をリセットします。

ビット 1 **GPIOBRST** : IO ポート B リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート B をリセットしません。

1 : IO ポート B をリセットします。

ビット 0 **GPIOARST** : IO ポート A リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート A をリセットしません。

1 : IO ポート A をリセットします。

6.3.6 RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSRST** : USB OTG FS モジュールリセット

ソフトウェアでセット／クリアされます。

0 : USB OTG FS モジュールをリセットしません。

1 : USB OTG FS モジュールをリセットします。

ビット 6 **RNGSRST** : RNG モジュールリセット

ソフトウェアでセット／クリアされます。

0 : RNG モジュールをリセットしません。

1 : RNG モジュールをリセットします。

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

6.3.7 RCC AHB3 ペリフェラルリセットレジスタ (RCC_AHB3RSTR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPIRST** : QUADSPI モジュールリセット

ソフトウェアでセット／クリアされます。

0 : QUADSPI モジュールをリセットしません。

1 : QUADSPI モジュールをリセットします。

ビット 0 **FSMCRST** : フレキシブルメモリコントローラモジュールリセット

ソフトウェアでセット／クリアされます。

0 : FSMC モジュールをリセットしません。

1 : FSMC モジュールをリセットします。

6.3.8 RCC APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	PWR RST	Res.	CAN2 RST	CAN1 RST	I2CFMP1 RST	I2C3 RST	I2C2 RST	I2C1 RST	Res.	Res.	USART3 RST	USART2 RST	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Res.	Res.	WWDG RST	Res.	Res.	TIM14 RST	TIM13 RST	TIM12 RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **PWRRST** : 電源インタフェースリセット

ソフトウェアでセット／クリアされます。

0 : 電源インタフェースをリセットしません。

1 : 電源インタフェースをリセットします。

ビット 27 予約済みであり、リセット値に保持する必要があります。



- ビット 26 **CAN2RST** : CAN2 リセット
ソフトウェアでセット／クリアされます。
0 : CAN2 をリセットしません。
1 : CAN2 をリセットします。
- ビット 25 **CAN1RST** : CAN1 リセット
ソフトウェアでセット／クリアされます。
0 : CAN1 をリセットしません。
1 : CAN1 をリセットします。
- ビット 24 **I2CFMP1RST** : I2CFMP1 リセット
ソフトウェアでセット／クリアされます。
0 : I2CFMP1 をリセットしません。
1 : I2CFMP1 をリセットします。
- ビット 23 **I2C3RST** : I2C3 リセット
ソフトウェアでセット／クリアされます。
0 : I2C3 をリセットしません。
1 : I2C3 をリセットします。
- ビット 22 **I2C2RST** : I2C2 リセット
ソフトウェアでセット／クリアされます。
0 : I2C2 をリセットしません。
1 : I2C2 をリセットします。
- ビット 21 **I2C1RST** : I2C1 リセット
ソフトウェアでセット／クリアされます。
0 : I2C1 をリセットしません。
1 : I2C1 をリセットします。
- ビット 20:19 予約済みであり、リセット値に保持する必要があります。
- ビット 18 **USART3RST** : USART3 リセット
ソフトウェアでセット／クリアされます。
0 : USART3 をリセットしません。
1 : USART3 をリセットします。
- ビット 17 **USART2RST** : USART2 リセット
ソフトウェアでセット／クリアされます。
0 : USART2 をリセットしません。
1 : USART2 をリセットします。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3RST** : SPI3 リセット
ソフトウェアでセット／クリアされます。
0 : SPI3 をリセットしません。
1 : SPI3 をリセットします。
- ビット 14 **SPI2RST** : SPI2 リセット
ソフトウェアでセット／クリアされます。
0 : SPI2 をリセットしません。
1 : SPI2 をリセットします。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGRST** : ウィンドウ型ウォッチドッグリセット
ソフトウェアでセット／クリアされます。
0 : ウィンドウ型ウォッチドッグをリセットしません。
1 : ウィンドウ型ウォッチドッグをリセットします。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14RST** : TIM14 リセット

ソフトウェアでセット／クリアされます。

0 : TIM14 をリセットしません。

1 : TIM14 をリセットします。

ビット 7 **TIM13RST** : TIM13 リセット

ソフトウェアでセット／クリアされます。

0 : TIM13 をリセットしません。

1 : TIM13 をリセットします。

ビット 6 **TIM12RST** : TIM12 リセット

ソフトウェアでセット／クリアされます。

0 : TIM12 をリセットしません。

1 : TIM12 をリセットします。

ビット 5 **TIM7RST** : TIM7 リセット

ソフトウェアでセット／クリアされます。

0 : TIM7 をリセットしません。

1 : TIM7 をリセットします。

ビット 4 **TIM6RST** : TIM6 リセット

ソフトウェアでセット／クリアされます。

0 : TIM6 をリセットしません。

1 : TIM6 をリセットします。

ビット 3 **TIM5RST** : TIM5 リセット

ソフトウェアでセット／クリアされます。

0 : TIM5 をリセットしません。

1 : TIM5 をリセットします。

ビット 2 **TIM4RST** : TIM4 リセット

ソフトウェアでセット／クリアされます。

0 : TIM4 をリセットしません。

1 : TIM4 をリセットします。

ビット 1 **TIM3RST** : TIM3 リセット

ソフトウェアでセット／クリアされます。

0 : TIM3 をリセットしません。

1 : TIM3 をリセットします。

ビット 0 **TIM2RST** : TIM2 リセット

ソフトウェアでセット／クリアされます。

0 : TIM2 をリセットしません。

1 : TIM2 をリセットします。

6.3.9 RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFSDM1 RST	Res.	Res.	Res.	SPI5 RST	Res.	TIM11 RST	TIM10 RST	TIM9 RST
							rw				rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	Res.	2	1	0
Res.	SYSCFG RST	SPI4 RST	SPI1 RST	SDIO RST	Res.	Res.	ADC1 RST	Res.	Res.	USART6 RST	USART1 RST	Res.	Res.	TIM8 RST	TIM1 RST
	rw	rw	rw	rw			rw			rw	rw			rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DFSDM1RST** : DFSDM1 リセット

ソフトウェアでセット／クリアされます。
0 : DFSDM1 をリセットしません。
1 : DFSDM1 をリセットします。

ビット 23:21 予約済み、常に 0 として読み出されます。

ビット 20 **SPI5RST** : SPI5RST

このビットは、ソフトウェアによってセット／クリアされます。
0 : SPI5 をリセットしません。
1 : SPI5 をリセットします。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11RST** : TIM11 リセット

ソフトウェアでセット／クリアされます。
0 : TIM11 をリセットしません。
1 : TIM11 をリセットします。

ビット 17 **TIM10RST** : TIM10 リセット

ソフトウェアでセット／クリアされます。
0 : TIM10 をリセットしません。
1 : TIM10 をリセットします。

ビット 16 **TIM9RST** : TIM9 リセット

ソフトウェアでセット／クリアされます。
0 : TIM9 をリセットしません。
1 : TIM9 をリセットします。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SYSCFGRST** : システム設定コントローラリセット

ソフトウェアでセット／クリアされます。
0 : システム設定コントローラをリセットしません。
1 : システム設定コントローラをリセットします。

ビット 13 **SPI4RST** : SPI4 リセット

ソフトウェアでセット／クリアされます。
0 : SPI4 をリセットしません。
1 : SPI4 をリセットします。

ビット 12 **SPI1RST** : SPI1 リセット

ソフトウェアでセット／クリアされます。

0 : SPI1 をリセットしません。

1 : SPI1 をリセットします。

ビット 11 **SDIORST** : SDIO リセット

ソフトウェアでセット／クリアされます。

0 : SDIO モジュールをリセットしません。

1 : SDIO モジュールをリセットします。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ADC1RST** : ADC インタフェースリセット

ソフトウェアでセット／クリアされます。

0 : ADC インタフェースをリセットしません。

1 : ADC インタフェースをリセットします。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **USART6RST** : USART6 リセット

ソフトウェアでセット／クリアされます。

0 : USART6 をリセットしません。

1 : USART6 をリセットします。

ビット 4 **USART1RST** : USART1 リセット

ソフトウェアでセット／クリアされます。

0 : USART1 をリセットしません。

1 : USART1 をリセットします。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM8RST** : TIM8 リセット

ソフトウェアでセット／クリアされます。

0 : TIM8 をリセットしません。

1 : TIM8 をリセットします。

ビット 0 **TIM1RST** : TIM1 リセット

ソフトウェアでセット／クリアされます。

0 : TIM1 をリセットしません。

1 : TIM1 をリセットします。

6.3.10 RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1ENR)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2EN	DMA1EN	Res.	Res.	Res.	Res.	Res.
									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	Res.	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	PIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DMA2EN** : DMA2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : DMA2 クロック無効
1 : DMA2 クロック有効

ビット 21 **DMA1EN** : DMA1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : DMA1 クロック無効
1 : DMA1 クロック有効

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCEN** : CRC クロック有効化
ソフトウェアでセット／クリアされます。
0 : CRC クロック無効
1 : CRC クロック有効

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHEN** : IO ポート H クロック有効化
ソフトウェアでセット／クリアされます。
0 : IO ポート H クロック無効
1 : IO ポート H クロック有効

ビット 6 **GPIOGEN** : IO ポート G クロック有効化
ソフトウェアでセット／クリアされます。
0 : IO ポート G クロック無効
1 : IO ポート G クロック有効

ビット 5 **GPIOFEN** : IO ポート F クロック有効化
ソフトウェアでセット／クリアされます。
0 : IO ポート F クロック無効
1 : IO ポート F クロック有効

ビット 4 **GPIOEEN** : IO ポート E クロック有効化
ソフトウェアでセット／クリアされます。
0 : IO ポート E クロック無効
1 : IO ポート E クロック有効

- ビット 3 **GPIODEN** : IO ポート D クロック有効化
ソフトウェアでセット/クリアされます。
0 : IO ポート D クロック無効
1 : IO ポート D クロック有効
- ビット 2 **GPIOCEN** : IO ポート C クロック有効化
ソフトウェアでセット/クリアされます。
0 : IO ポート C クロック無効
1 : IO ポート C クロック有効
- ビット 1 **GPIOBEN** : IO ポート B クロック有効化
ソフトウェアでセット/クリアされます。
0 : IO ポート B クロック無効
1 : IO ポート B クロック有効
- ビット 0 **GPIOAEN** : IO ポート A クロック有効化
ソフトウェアでセット/クリアされます。
0 : IO ポート A クロック無効
1 : IO ポート A クロック有効

6.3.11 RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR)

アドレスオフセット : 0x34
リセット値 : 0x0000 0000
アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

- ビット 31:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **OTGFSEN** : USB OTG FS クロック有効化
ソフトウェアでセット/クリアされます。
0 : USB OTG FS クロック無効
1 : USB OTG FS クロック有効
- ビット 6 **RNGEN** : RNG クロック有効化
ソフトウェアでセット/クリアされます。
0 : RNG クロック無効
1 : RNG クロック有効
- ビット 5:0 予約済み、常に 0 として読み出されます。

6.3.12 RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3ENR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPIEN** : QUADSPI メモリコントローラモジュールクロックイネーブル

ソフトウェアでセット／クリアされます。

0 : QUADSPI クロック無効

1 : QUADSPI クロック有効

ビット 0 **FSMCEN** : フレキシブルメモリコントローラモジュールクロックイネーブル

ソフトウェアでセット／クリアされます。

0 : FSMC モジュールクロック無効

1 : FSMC モジュールクロック有効

6.3.13 RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR)

アドレスオフセット : 0x40

リセット値 : 0x0000 0400

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	PWR EN	Res.	CAN2 EN	CAN1 EN	I2CFMP1 EN	I2C3 EN	I2C2 EN	I2C1 EN	Res.	Res.	USART3 EN	USART2 EN	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Res.	Res.	WWDG EN	RTCAPB	Res.	TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **PWREN** : 電源インタフェースクロック有効

ソフトウェアでセット／クリアされます。

0 : 電源インタフェースクロック無効

1 : 電源インタフェースクロック有効

ビット 27 予約済みであり、リセット値に保持する必要があります。

- ビット 26 **CAN2EN** : CAN 2 クロック有効化
このビットは、ソフトウェアによってセット／クリアされます。
0 : CAN 2 クロック無効
1 : CAN 2 クロック有効
- ビット 25 **CAN1EN** : CAN 1 クロック有効化
このビットは、ソフトウェアによってセット／クリアされます。
0 : CAN 1 クロック無効
1 : CAN 1 クロック有効
- ビット 24 **I2CFMP1EN** : I2CFMP1 クロック有効化
このビットは、ソフトウェアによってセット／クリアされます。
0 : I2CFMP1 クロック無効
1 : I2CFMP1 クロック有効
- ビット 23 **I2C3EN** : I2C3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C3 クロック無効
1 : I2C3 クロック有効
- ビット 22 **I2C2EN** : I2C2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C2 クロック無効
1 : I2C2 クロック有効
- ビット 21 **I2C1EN** : I2C1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C1 クロック無効
1 : I2C1 クロック有効
- ビット 20:17 予約済みであり、リセット値に保持する必要があります。
- ビット 18 **USART3EN** : USART3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : USART3 クロック無効
1 : USART3 クロック有効
- ビット 17 **USART2EN** : USART2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : USART2 クロック無効
1 : USART2 クロック有効
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3EN** : SPI3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI3 クロック無効
1 : SPI3 クロック有効
- ビット 14 **SPI2EN** : SPI2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI2 クロック無効
1 : SPI2 クロック有効
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGEN** : ウィンドウ型ウォッチドッグクロック有効化
ソフトウェアでセット／クリアされます。
0 : ウィンドウ型ウォッチドッグクロック無効
1 : ウィンドウ型ウォッチドッグクロック有効

ビット 10 **RTC APB** : クロック有効化

ソフトウェアでセット／クリアされます。

0 : RTC APB クロック無効

1 : RTC APB クロック有効 (デフォルト値)

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14EN** : TIM14 リセット

ソフトウェアでセット／クリアされます。

0 : TIM14 をリセットしません。

1 : TIM14 をリセットします。

ビット 7 **TIM13EN** : TIM13 リセット

ソフトウェアでセット／クリアされます。

0 : TIM13 をリセットしません。

1 : TIM13 をリセットします。

ビット 6 **TIM12EN** : TIM12 リセット

ソフトウェアでセット／クリアされます。

0 : TIM12 をリセットしません。

1 : TIM12 をリセットします。

ビット 5 **TIM7EN** : TIM7 リセット

ソフトウェアでセット／クリアされます。

0 : TIM7 をリセットしません。

1 : TIM7 をリセットします。

ビット 4 **TIM6EN** : TIM6 リセット

ソフトウェアでセット／クリアされます。

0 : TIM6 をリセットしません。

1 : TIM6 をリセットします。

ビット 3 **TIM5EN** : TIM5 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM5 クロック無効

1 : TIM5 クロック有効

ビット 2 **TIM4EN** : TIM4 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM4 クロック無効

1 : TIM4 クロック有効

ビット 1 **TIM3EN** : TIM3 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM3 クロック無効

1 : TIM3 クロック有効

ビット 0 **TIM2EN** : TIM2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM2 クロック無効

1 : TIM2 クロック有効

6.3.14 RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR)

アドレスオフセット : 0x44

リセット値 : 0x0000 8000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFSDM1 EN	Res.	Res.	Res.	SPI5EN	Res.	TIM11 EN	TIM10 EN	TIM9 EN
							rw				rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SYSCF G EN	SPI4EN	SPI1 EN	SDIO EN	Res.	Res.	ADC1 EN	Res.	Res.	USART6 EN	USART1 EN	Res.	Res.	TIM8 EN	TIM1 EN
	rw	rw	rw	rw			rw			rw	rw			rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DFSDM1EN** : DFSDM1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : DFSDM1 クロック無効
1 : DFSDM1 クロック有効

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **SPI5EN** : SPI5 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI5 クロック無効
1 : SPI5 クロック有効

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11EN** : TIM11 クロック有効化
ソフトウェアでセット／クリアされます。
0 : TIM11 クロック無効
1 : TIM11 クロック有効

ビット 17 **TIM10EN** : TIM10 クロック有効化
ソフトウェアでセット／クリアされます。
0 : TIM10 クロック無効
1 : TIM10 クロック有効

ビット 16 **TIM9EN** : TIM9 クロック有効化
ソフトウェアでセット／クリアされます。
0 : TIM9 クロック無効
1 : TIM9 クロック有効

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SYSCFGEN** : システム設定コントローラクロック有効化
ソフトウェアでセット／クリアされます。
0 : システム設定コントローラクロック無効
1 : システム設定コントローラクロック有効

ビット 13 **SPI4EN** : SPI4 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI4 クロック無効
1 : SPI4 クロック有効化



ビット 12 **SPI1EN** : SPI1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI1 クロック無効

1 : SPI1 クロック有効

ビット 11 **SDIOEN** : SDIO クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SDIO モジュールクロック無効

1 : SDIO モジュールクロック有効

ビット 8 **ADC1EN** : ADC1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : ADC1 クロック無効

1 : ADC1 クロック無効

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **USART6EN** : USART6 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART6 クロック無効

1 : USART6 クロック有効

ビット 4 **USART1EN** : USART1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART1 クロック無効

1 : USART1 クロック有効

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM8EN** : TIM8 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM8 クロック無効

1 : TIM8 クロック有効

ビット 0 **TIM1EN** : TIM1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM1 クロック無効

1 : TIM1 クロック有効

6.3.15 低電力モードにおける RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1LPENR)

アドレスオフセット : 0x50

リセット値 : 0x0061 90FF

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 LPEN	DMA1 LPEN	Res.	Res.	Res.	Res.	SRAM1 LPEN
									rw	rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLITF LPEN	Res.	Res.	CRC LPEN	Res.	Res.	Res.	Res.	GPIOH LPEN	GPIOG LPEN	GPIOF LPEN	GPIOE LPEN	GIOD LPEN	GPIOC LPEN	GPIOB LPEN	GPIOA LPEN
rw			rw					rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DMA2LPEN** : SLEEP モード時 DMA2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時 DMA2 クロック無効
1 : SLEEP モード時 DMA2 クロック有効

ビット 21 **DMA1LPEN** : SLEEP モード時 DMA1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時 DMA1 クロック無効
1 : SLEEP モード時 DMA1 クロック有効

ビット 20:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SRAM1LPEN** : SLEEP モード時 SRAM1 インターフェースクロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時 SRAM1 インタフェースクロック無効
1 : SLEEP モード時 SRAM1 インタフェースクロック有効

ビット 15 **FLITFLPEN** : SLEEP モード時フラッシュインタフェースクロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時フラッシュインタフェースクロック無効
1 : SLEEP モード時フラッシュインタフェースクロック有効

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCLPEN** : SLEEP モード時 CRC クロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時 CRC クロック無効
1 : SLEEP モード時 CRC クロック有効

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHLPEN** : SLEEP モード時 IO ポート H クロック有効化
ソフトウェアでセット／クリアされます。
0 : SLEEP モード時 IO ポート H クロック無効
1 : SLEEP モード時 IO ポート H クロック有効

- ビット 6 **GPIOGLPEN** : SLEEP モード時 IO ポート G クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート G クロック無効
1 : SLEEP モード時 IO ポート G クロック有効
- ビット 5 **GPIOFLPEN** : SLEEP モード時 IO ポート F クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート F クロック無効
1 : SLEEP モード時 IO ポート F クロック有効
- ビット 4 **GPIOELPEN** : SLEEP モード時 IO ポート E クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート E クロック無効
1 : SLEEP モード時 IO ポート E クロック有効
- ビット 3 **GPIODLPEN** : SLEEP モード時 IO ポート D クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート D クロック無効
1 : SLEEP モード時 IO ポート D クロック有効
- ビット 2 **GPIOCLPEN** : SLEEP モード時 IO ポート C クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート C クロック無効
1 : SLEEP モード時 IO ポート C クロック有効
- ビット 1 **GPIOBLPEN** : SLEEP モード時 IO ポート B クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート B クロック無効
1 : SLEEP モード時 IO ポート B クロック有効
- ビット 0 **GPIOALPEN** : SLEEP モード時 IO ポート A クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 IO ポート A クロック無効
1 : SLEEP モード時 IO ポート A クロック有効

6.3.16 低電力モードにおける RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2LPENR)

アドレスオフセット : 0x54
リセット値 : 0x0000 00C0
アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSLPEN** : SLEEP モード時 USB OTG FS クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 USB OTG FS クロック無効
1 : SLEEP モード時 USB OTG FS クロック有効

ビット 6 **RNGLPEN** : SLEEP モード時 RNG クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 RNG クロック無効
1 : SLEEP モード時 RNG クロック有効

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

6.3.17 低電力モードにおける RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3LPENR)

アドレスオフセット : 0x58

リセット値 : 0x0000 0003

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPILPEN** : SLEEP モード時 QUADSPI メモリコントローラモジュールクロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 QUADSPI モジュールクロック無効
1 : SLEEP モード時 QUADSPI モジュールクロック有効

ビット 0 **FSMCLPEN** : SLEEP モード時フレキシブルメモリコントローラモジュールクロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 FSMC クロック無効
1 : SLEEP モード時 FSMC クロック有効

6.3.18 低電力モードにおける RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1LPENR)

アドレスオフセット : 0x60

リセット値 : 0x17E6 CDDF

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	PWR LPEN	Res.	CAN2 LPEN	CAN1 LPEN	I2CFMP1 LPEN	I2C3 LPEN	I2C2 LPEN	I2C1 LPEN	Res.	Res.	USART3 LPEN	USART2 LPEN	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 LPEN	SPI2 LPEN	Res.	Res.	WWDG LPEN	RTCAPB LPEN	Res.	TIM14 LPEN	TIM13 LPEN	TIM12 LPEN	TIM7 LPEN	TIM6 LPEN	TIM5 LPEN	TIM4 LPEN	TIM3 LPEN	TIM2 LPEN
rw	rw			rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **PWRLPEN** : SLEEP モード時電源インタフェースクロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時電源インタフェースクロック無効

1 : SLEEP モード時電源インタフェースクロック有効

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **CAN2LPEN** : SLEEP モード時 CAN2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 CAN2 クロック無効

1 : SLEEP モード時 CAN2 クロック有効

ビット 25 **CAN1LPEN** : SLEEP モード時 CAN1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 CAN1 クロック無効

1 : SLEEP モード時 CAN1 クロック有効

ビット 24 **I2CFMP1LPEN** : SLEEP モード時 I2CFMP1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 I2CFMP1 クロック無効

1 : SLEEP モード時 I2CFMP1 クロック有効

ビット 23 **I2C3LPEN** : SLEEP モード時 I2C3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 I2C3 クロック無効

1 : SLEEP モード時 I2C3 クロック有効

ビット 22 **I2C2LPEN** : SLEEP モード時 I2C2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 I2C2 クロック無効

1 : SLEEP モード時 I2C2 クロック有効

ビット 21 **I2C1LPEN** : SLEEP モード時 I2C1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 I2C1 クロック無効

1 : SLEEP モード時 I2C1 クロック有効

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **USART3LPEN** : SLEEP モード時 USART3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 USART3 クロック無効

1 : SLEEP モード時 USART3 クロック有効

ビット 17 **USART2LPEN** : SLEEP モード時 USART2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 USART2 クロック無効

1 : SLEEP モード時 USART2 クロック有効

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **SPI3LPEN** : SLEEP モード時 SPI3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 SPI3 クロック無効

1 : SLEEP モード時 SPI3 クロック有効

ビット 14 **SPI2LPEN** : SLEEP モード時 SPI2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 SPI2 クロック無効

1 : SLEEP モード時 SPI2 クロック有効

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **WWDGLPEN** : SLEEP モード時ウィンドウ型ウォッチドッグクロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時ウィンドウ型ウォッチドッグクロック無効

1 : SLEEP モード時ウィンドウ型ウォッチドッグクロック有効

ビット 10 **RTCAPBEN** : SLEEP モード時 RTC APB クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 RTC APB クロック無効

1 : SLEEP モード時 RTC APB ウォッチドッグクロック有効

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14LPEN** : SLEEP モード時 TIM14 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM14 クロック無効

1 : SLEEP モード時 TIM14 クロック有効

ビット 7 **TIM13LPEN** : SLEEP モード時 TIM13 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM13 クロック無効

1 : SLEEP モード時 TIM13 クロック有効

ビット 6 **TIM12LPEN** : SLEEP モード時 TIM12 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM12 クロック無効

1 : SLEEP モード時 TIM12 クロック有効

ビット 5 **TIM7LPEN** : SLEEP モード時 TIM7 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM7 クロック無効

1 : SLEEP モード時 TIM7 クロック有効

ビット 4 **TIM6LPEN** : SLEEP モード時 TIM6 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM6 クロック無効

1 : SLEEP モード時 TIM6 クロック有効

ビット 3 **TIM5LPEN** : SLEEP モード時 TIM5 クロック有効化
ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM5 クロック無効

1 : SLEEP モード時 TIM5 クロック有効

ビット 2 **TIM4LPEN** : SLEEP モード時 TIM4 クロック有効化
ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM4 クロック無効

1 : SLEEP モード時 TIM4 クロック有効

ビット 1 **TIM3LPEN** : SLEEP モード時 TIM3 クロック有効化
ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM3 クロック無効

1 : SLEEP モード時 TIM3 クロック有効

ビット 0 **TIM2LPEN** : SLEEP モード時 TIM2 クロック有効化
ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM2 クロック無効

1 : SLEEP モード時 TIM2 クロック有効

6.3.19 低電力モードにおける RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2LPENR)

アドレスオフセット : 0x64

リセット値 : 0x0117 F933

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DFSDM1 LPEN	Res.	Res.	Res.	SPI5 LPEN	Res.	TIM11 LPEN	TIM10 LPEN	TIM9 LPEN
							rw				rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTIT EN	SYSC FG LPEN	SPI4LP EN	SPI1 LPEN	SDIO LPEN	Res.	Res.	ADC1 LPEN	Res.	Res.	USART6 LPEN	USART1 LPEN	Res.	Res.	TIM8 LPEN	TIM1 LPEN
rw	rw	rw	rw	rw			rw			rw	rw			rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **DFSDM1LPEN** : SLEEP モード時 DFSDM1 クロック有効化
このビットは、ソフトウェアによってセット/クリアされます。
0 : SLEEP モード時 DFSDM1 クロック無効
1 : SLEEP モード時 DFSDM1 クロック有効

ビット 23:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **SPI5LPEN** : SLEEP モード時 SPI5 クロック有効化
このビットは、ソフトウェアによってセット/クリアされます。
0 : SLEEP モード時 SPI5 クロック無効
1 : SLEEP モード時 SPI5 クロック有効

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11LPEN** : SLEEP モード時 TIM11 クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 TIM11 クロック無効
1 : SLEEP モード時 TIM11 クロック有効

ビット 17 **TIM10LPEN** : SLEEP モード時 TIM10 クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 TIM10 クロック無効
1 : SLEEP モード時 TIM10 クロック有効

ビット 16 **TIM9LPEN** : SLEEP モード時 TIM9 クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 TIM9 クロック無効
1 : SLEEP モード時 TIM9 クロック有効

ビット 15 **EXTITEN** : SLEEP モード時 EXTIT APB および SYSCCTRL PFREE クロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時 EXTIT APB および SYSCCTRL PFREE クロック無効
1 : SLEEP モード時 EXTIT APB および SYSCCTRL PFREE クロック有効

ビット 14 **SYSCFGLPEN** : SLEEP モード時システム設定コントローラクロック有効化
ソフトウェアでセット/クリアされます。
0 : SLEEP モード時システム設定コントローラクロック無効
1 : SLEEP モード時システム設定コントローラクロック有効



ビット 13 **SPI4LPEN** : SLEEP モード時 SPI4 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 SPI4 クロック無効

1 : SLEEP モード時 SPI4 クロック有効

ビット 12 **SPI1LPEN** : SLEEP モード時 SPI1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 SPI1 クロック無効

1 : SLEEP モード時 SPI1 クロック有効

ビット 11 **SDIOLPEN** : SLEEP モード時 SDIO クロックイネーブル

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 SDIO モジュールクロック無効

1 : SLEEP モード時 SDIO モジュールクロック有効

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ADC1LPEN** : SLEEP モード時 ADC1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 ADC1 クロック無効

1 : SLEEP モード時 ADC1 クロック有効

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **USART6LPEN** : SLEEP モード時 USART6 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 USART6 クロック無効

1 : SLEEP モード時 USART6 クロック有効

ビット 4 **USART1LPEN** : SLEEP モード時 USART1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 USART1 クロック無効

1 : SLEEP モード時 USART1 クロック有効

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM8LPEN** : SLEEP モード時 TIM8 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM8 クロック無効

1 : SLEEP モード時 TIM8 クロック有効

ビット 0 **TIM1LPEN** : SLEEP モード時 TIM1 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM1 クロック無効

1 : SLEEP モード時 TIM1 クロック有効

6.3.20 RCC バックアップドメイン制御レジスタ (RCC_BDCR)

アドレスオフセット : 0x70

リセット値 : 0x0000 0000、バックアップドメインリセットによってリセットされます。

アクセス : 0 ≤ ウェイトステート ≤ 3、ワード、ハーフワード、およびバイトアクセス

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

RCC バックアップドメイン制御レジスタ (RCC_BDCR) の LSEON、LSEBYP、RTCSEL、および RTCEN ビットは、バックアップドメインにあります。このため、リセット後、これらのビットは書き込み保護されるので、これらを変更するには、[セクション 5.4.1 : PWR 電源制御レジスタ \(PWR_CR\)](#) の DBP ビットをセットする必要があります。詳細については、[セクション 5.4.2 : PWR 電源制御/ステータスレジスタ \(PWR_PWR\)](#) を参照してください。これらのビットは、バックアップドメインリセット後にのみリセットされます ([セクション 6.1.3 : バックアップドメインリセット](#) を参照)。内部または外部リセットは、これらのビットに影響しません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	Res.	Res.	Res.	LSEMOD	LSEBYP	LSERDY	LSEON
rw						rw	rw					rw	rw	r	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **BDRST** : バックアップドメインソフトウェアリセット

ソフトウェアでセット/クリアされます。

0 : リセットは有効ではありません。

1 : バックアップドメイン全体をリセットします。

ビット 15 **RTCEN** : RTC クロック有効化

ソフトウェアでセット/クリアされます。

0 : RTC クロック無効

1 : RTC クロック有効

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **RTCSEL[1:0]** : RTC クロックソース選択

RTC のクロックソースを選択するために、ソフトウェアによってセットされます。RTC クロックソースが選択されると、バックアップドメインがリセットされない限り、変更はできません。BDRST ビットを使用して、リセットできます。

00 : クロックなし。

01 : LSE オシレータクロックが RTC クロックとして使用されます。

10 : LSI オシレータクロックが RTC クロックとして使用されます。

11 : プログラマブルプリスケラ (RCC クロック設定レジスタ (RCC_CFGR) の RTCPRE[4:0] ビットを使用して選択) で分周された HSE オシレータクロックが RTC クロックとして使用されます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **LSEMOD** : 外部低速オシレータバイパス

低速オシレータのクリスタルモードを選択するために、ソフトウェアによってセット/リセットされます。次の 2 つの消費電力モードがあります。

0 : LSE オシレータ「低電力」モード選択

1 : LSE オシレータ「高駆動」モード選択

ビット 2 LSEBYP : 外部低速オシレータバイパス

デバッグモードのオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。このビットに書き込めるのは、LSE クロックが無効のときだけです。

- 0 : LSE オシレータはバイパスされません。
- 1 : LSE オシレータはバイパスされます。

ビット 1 LSERDY : 外部低速オシレータレディ

外部 32 kHz オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSEON ビットがクリアされた後、LSERDY は、外部低速オシレータクロックの 6 サイクル後にローになります。

- 0 : LSE クロックはレディ状態ではありません。
- 1 : LSE クロックはレディ状態です。

ビット 0 LSEON : 外部低速オシレータ有効化

ソフトウェアでセット/クリアされます。

- 0 : LSE クロックオフ
- 1 : LSE クロックオン

6.3.21 RCC クロック制御およびステータスレジスタ (RCC_CSR)

アドレスオフセット : 0x74

リセット値 : 0x0E00 0000。システムリセットによってリセットされます。ただし、リセットフラグは電源リセット時のみリセットされます。

アクセス : 0 ≤ ウェイトステート ≤ 3、ワード、ハーフワード、およびバイトアクセス

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	BORRS TF	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	rt_w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
														r	rw

ビット 31 LPWRRSTF : 低電力リセットフラグ

低電力管理リセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。

- 0 : 低電力管理リセットは発生していません。
- 1 : 低電力管理リセットが発生しました。

低電力管理リセットの詳細については、[低電力管理リセット](#)を参照してください。

ビット 30 WWDGRSTF : ウィンドウ型ウォッチドッグリセットフラグ

ウィンドウ型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。

- 0 : ウィンドウ型ウォッチドッグリセットは発生していません。
- 1 : ウィンドウ型ウォッチドッグリセットが発生しました。

ビット 29 IWDGRSTF : 独立型ウォッチドッグリセットフラグ

V_{DD} ドメインから独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

- 0 : ウォッチドッグリセットは発生していません。
- 1 : ウォッチドッグリセットが発生しました。

ビット 28 SFTRSTF : ソフトウェアリセットフラグ

ソフトウェアリセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。
0 : ソフトウェアリセットは発生していません。
1 : ソフトウェアリセットが発生しました。

ビット 27 PORRSTF : POR/PDR リセットフラグ

POR/PDR リセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。
0 : POR/PDR リセットは発生していません。
1 : POR/PDR リセットが発生しました。

ビット 26 PINRSTF : PIN リセットフラグ

NRST ピンに外部リセット信号が入力され、リセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。
0 : NRST ピンからのリセットは発生していません。
1 : NRST ピンからリセットが発生しました。

ビット 25 BORRSTF : BOR リセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってクリアされます。
POR/PDR リセットまたは BOR リセットが発生したときに、ハードウェアによってセットされます。
0 : POR/PDR リセットまたは BOR リセットは発生していません。
1 : POR/PDR リセットまたは BOR リセットが発生しました。

ビット 24 RMVF : リセットフラグ解除

リセットフラグをクリアするために、ソフトウェアによってセットされます。
0 : 影響なし。
1 : リセットフラグをクリアします。

ビット 23:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 LSIRDY : 内部低速オシレータレディ

内部 RC 40 kHz オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSION ビットがクリアされた後、LSIRDY は、LSI オシレータクロックの 3 サイクル後にローになります。
0 : LSI RC オシレータはレディ状態ではありません。
1 : LSI RC オシレータはレディ状態です。

ビット 0 LSION : 内部低速オシレータ有効化

ソフトウェアでセット/クリアされます。
0 : LSI RC オシレータオフ
1 : LSI RC オシレータオン

6.3.22 RCC スペクトル拡散クロック生成レジスタ (RCC_SSCGR)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

スペクトル拡散クロック生成は、メイン PLL に対してのみ利用可能です。

RCC_SSCGR レジスタは、メイン PLL が有効になる前またはメイン PLL が無効になった後に書き込む必要があります。

注 : PLL スペクトル拡散クロック生成 (SSCG) 特性の詳細については、デバイスのデータシートの電気的特性のセクションを参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCG EN	SPREAD SEL	Res.	Res.	INCSTEP[14:3]											
r/w	r/w			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCSTEP[2:0]				MODPER[11:0]											
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **SSCGEN** : スペクトル拡散変調有効化
ソフトウェアでセット／クリアされます。
0 : スペクトル拡散変調無効 (書き込みは クリア後 : CR[24]=PLLON ビット)
1 : スペクトル拡散変調有効 (書き込みは セット前 : CR[24] = PLLON ビット)

ビット 30 **SPREADSEL** : 拡散選択
ソフトウェアでセット／クリアされます。
CR[24]=PLLON ビットをセットする前に書き込みます。
0 : センタースプレッド
1 : ダウンスプレッド

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:13 **INCSTEP[14:0]** : インクリメントステップ
ソフトウェアでセット／クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。
変調プロファイルの振幅の設定入力です。

ビット 12:0 **MODPER[11:0]** : 変調周期
ソフトウェアでセット／クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。
変調プロファイルの周期の設定入力です。

6.3.23 RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR)

アドレスオフセット : 0x84

リセット値 : 0x2400 3010

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

このレジスタは、次の式に従って PLLI2S クロック出力を設定するために使用します。

- $f(\text{VCO クロック}) = f(\text{PLLI2S クロック入力}) \times (\text{PLLI2SN} / \text{PLLI2SM})$
- $f(\text{USB OTG FS, SDIO, RNG クロック出力}) = f(\text{VCO クロック}) / \text{PLLQ}$
- $f(\text{DFSDM, I2S クロック出力}) = f(\text{VCO クロック}) / \text{PLLR}$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLLI2SR[2:0]			PLLI2SQ[3:0]				Res.	PLLI2SSRC	Res.	Res.	Res.	Res.	Res.	Res.
	rW	rW	rW	rW	rW	rW	rW		rW						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLI2SN[8:0]								PLLI2SM[5:0]						
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **PLLI2SR[2:0]** : I2S クロックのための PLLI2S 分周比

I2S クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。標準クリスタル使用時の誤差が 0.3 %、オーディオクリスタル使用時の誤差が 0 % となるよう、分周比は I2S ペリフェラル内のプリスケアラ値に従って選択する必要があります。I2S クロック周波数と精度に関する詳細については、I2S の章の [セクション 26.6.4 : クロックジェネレータ](#) を参照してください。

注意 : I2S が正しく動作するには、192 MHz 以下の周波数が必要です。

I2S クロック周波数 = VCO 周波数 / PLLR、ここで $2 \leq \text{PLLR} \leq 7$

000 : PLLR = 0、誤った設定

001 : PLLR = 1、誤った設定

010 : PLLR = 2

...

111 : PLLR = 7

ビット 27:24 **PLLI2SQ[3:0]** : USB OTG FS/SDIO/RNG クロックのための PLLI2S 分周比

USB OTG FS/SDIO/RNG クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。

USB OTG FS/SDIO/RNG クロック周波数 = VCO 周波数 / PLLI2SQ、ここで $2 \leq \text{PLLI2SQ} \leq 15$

0000 : PLLI2SQ = 0、誤った設定

0001 : PLLI2SQ = 1、誤った設定

0010 : PLLI2SQ = 2

0011 : PLLI2SQ = 3

0100 : PLLI2SQ = 4

0101 : PLLI2SQ = 5

...

1111 : PLLI2SQ = 15

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PLLI2SSRC** : PLLI2S 入力クロックソース

PLLI2S クロックソースを選択するために、ソフトウェアによってセット／クリアされます。このビットに書き込めるのは、PLLI2S が無効なときだけです。

0 : PLLCFGR の PLLSRC に応じた HSE または HSI

1 : PLL クロック入力として選択される外部 AFI クロック (CK_I2S_EXT)

ビット 21:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:6 **PLLI2SN[8:0]** : VCOのための PLLI2S 通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット／クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

注意： これらのビットは、VCO 出力周波数が 100~432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。VCO 入力周波数範囲は 1~2 MHz (図 14 および [RCC PLL 設定レジスタ \(RCC_PLLCFGR\)](#) の分周比 M を参照)

VCO 出力周波数 = VCO 入力周波数 × PLLI2SN、ここで $50 \leq \text{PLLI2SN} \leq 432$

000000000 : PLLI2SN = 0、誤った設定

000000001 : PLLI2SN = 1、誤った設定

...

001100010 : PLLI2SN = 50

...

001100011 : PLLI2SN = 99

001100100 : PLLI2SN = 100

001100101 : PLLI2SN = 101

001100110 : PLLI2SN = 102

...

110110000 : PLLI2SN = 432

110110000 : PLLI2SN = 433、誤った設定

...

111111111 : PLLI2SN = 511、誤った設定

注： 通倍数 50~99 は、1 MHz を超える VCO 入力周波数になり得ます。ただし、上記に指定された最小 VCO 出力周波数を満たすことに注意する必要があります。

ビット 5:0 **PLLI2SM[5:0]** : メイン PLL (PLL) およびオーディオ PLL (PLLI2S) 入力クロックのための分周比

PLL および PLLI2S 入力クロックを VCO の前で分周するために、ソフトウェアによってセット／クリアされます。これらのビットに書き込めるのは、PLL および PLLI2S が無効のときだけです。

注意： これらのビットは、VCO 入力周波数が 1 MHz から 2 MHz の間になるように、ソフトウェアで正しく設定する必要があります。PLL ジッタを制限するため 2 MHz の周波数を選択することが推奨されます。

VCO 入力周波数 = PLL 入力クロック周波数 / PLLI2SM、ここで $2 \leq \text{PLLI2SM} \leq 63$

000000 : PLLI2SM = 0、誤った設定

000001 : PLLI2SM = 1、誤った設定...

000010 : PLLI2SM = 2

000011 : PLLI2SM = 3

000100 : PLLI2SM = 4

.....

111110 : PLLI2SM = 62

111111 : PLLI2SM = 63

6.3.24 RCC 専用クロック設定レジスタ (RCC_DCKCFGR)

アドレスオフセット : 0x8C

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKDFSD M1SEL	Res.	Res.	I2S2SRC[1:0]	I2S1RC[1:0]	TIMPRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw			rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKDFSD M1ASEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

ビット 31 **CKDFSDM1SEL** : DFSDM1 カーネルクロック選択

- 0 : カーネルクロックとして APB2 クロックが使用されます。
- 1 : カーネルクロックとしてシステムクロックが使用されます。

ビット 30:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:27 **I2S2SRC[1:0]** : I2S APB2 クロックソース選択 (I2S1/4/5)

- ソフトウェアでセット／クリアされます。
- これらのビットに書き込めるのは、PLL および PLLI2S が無効なときです。
- 00 : I2S APB2 クロック周波数 = $f(\text{PLLI2S_R})$
- 01 : I2S APB2 クロック 周波数 = パッドからの外部 I2S クロック - オルタネート機能入力周波数
- 10 : I2S APB2 クロック周波数 = $f(\text{PLL_R})$
- 11 : I2S APB2 クロック周波数 = PLLSRC に応じた HSI/HSE (PLLCFGR(22))

ビット 26:25 **I2S1SRC[1:0]** : I2S APB1 クロックソース選択 (I2S2/3)

- APB1 I2S クロックの周波数を制御するために、ソフトウェアによってセット／リセットされます。
- これらのビットに書き込めるのは、PLL および PLLI2S が無効なときです。
- 00 : I2S APB1 クロック周波数 = $f(\text{PLLI2S_R})$
- 01 : I2S APB1 クロック 周波数 = パッドからの外部 I2S クロック - オルタネート機能入力周波数
- 10 : I2S APB1 クロック周波数 = $f(\text{PLL_R})$
- 11 : I2S APB1 クロック周波数 = PLLSRC に応じた HSI/HSE (PLLCFGR(22))

ビット 24 **TIMPRE** : タイマクロックプリスケール選択

- APB1 ドメインおよび APB2 ドメインに接続されているすべてのタイマのクロック周波数を制御するために、ソフトウェアによってセット／リセットされます。
- 0 : APB プリスケール (RCC_CFGR レジスタの PPRE1、PPRE2) が分周比 1 に設定されている場合、TIMxCLK = PCLKx となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます :
 $\text{TIMxCLK} = 2 * \text{PCLKx}$
- 1 : APB プリスケール (RCC_CFGR レジスタの PPRE1、PPRE2) が分周比 1、2、4 のいずれかに設定されている場合、TIMxCLK = HCLK となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 4 倍に設定されます :
 $\text{TIMxCLK} = 4 * \text{PCLKx}$

ビット 23 : 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **CKDFSDM1ASEL** : DFSDM1 オーディオクロック選択

- 0 : CK_I2S_APB1 クロックがオーディオクロックとして選択されます。
- 1 : CK_I2S_APB2 クロックがオーディオクロックとして選択されます。

ビット 14:0 予約済みであり、リセット値に保持する必要があります。



6.3.25 RCC クロックゲート有効レジスタ (CKGATENR)

アドレスオフセット : 0x90

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

このレジスタで、指定された IP のクロックゲーティングを有効化または無効化できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTCL_CKEN	RCC_CKEN	FLITF_CKEN	SRAM_CKEN	SPARE_CKEN	CM4DBG_CKEN	AHB2APB2_CKEN	AHB2APB1_CKEN
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **EVTCL_CKEN**

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 6 **RCC_CKEN** : RCC クロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 5 **FLITF_CKEN** : フラッシュインタフェースクロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 4 **SRAM_CKEN** : SRQAM コントローラクロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 3 **SPARE_CKEN** : スペアクロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 2 **CM4DBG_CKEN** : Cortex M4 ETM クロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 1 **AHB2APB2_CKEN** : AHB から APB2 へのブリッジクロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

ビット 0 **AHB2APB1_CKEN** : AHB から APB1 へのブリッジクロック有効化

- 0 : クロックゲーティング有効
- 1 : クロックゲーティング無効、クロックは常に有効

6.3.26 RCC 専用クロック設定レジスタ (RCC_DCKCFGR2)

アドレスオフセット : 0x94

リセット値 : 0x0000 0000

このレジスタで、指定された IP のクロックゲーティングを有効化または無効化できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SDIO SEL	CK48M SEL	Res.			I2CFMP1 SEL[1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw				rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **CKSDIOSEL** : SDIO クロックの選択

0 : CK_48MHz (CK48MSEL ビット定義参照)

1 : クロックシステム

ビット 27 **CK48MSEL** : SDIO/USBFS クロックの選択

0 : f(PLL_Q)

1 : f(PLL12S_Q)

ビット 26:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **I2CFMP1SEL[1:0]** : I2CFMP1 カーネルクロックソースの選択

00 : APB クロックが I2CFMP1 クロックとして選択されます。

01 : システムクロックが I2CFMP1 クロックとして選択されます。

10 : HSI クロックが I2CFMP1 クロックとして選択されます。

11 : APB クロックが I2CFMP1 (「00」と同じ) として選択されます。

ビット 21 : 0 予約済みであり、リセット値に保持する必要があります。

6.3.27 RCC レジスタマップ

表 24 に、レジスタマップとリセット値を示します

表 24. STM32F412xx の RCC レジスタマップとリセット値

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RCC_CR	Res.	Res.	Res.	Res.	PLL12SRDY	PLL12SON	PLL12RDY	PLL12ON	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSE12RDY	HSEON	HSICAL[7:0]							HSITRIM[4:0]				Res.	HS12RDY	HSION			
0x04	RCC_PLLCFGR	Res.	PLL12R[2:0]		PLL12Q[3:0]			Res.		PLL12SRC	Res.	Res.	Res.	Res.	Res.	PLL12P[1:0]	Res.	PLL12N[8:0]					PLL12M[5:0]											
0x08	RCC_CFGR	MCO12[1:0]		MCO12PRE[2:0]			MCO12PRE[2:0]			Res.	MCO12[1:0]		RTC12PRE[4:0]				PPRE12[2:0]		PPRE12[2:0]			Res.			HPRE[3:0]			SWS[1:0]		SW[1:0]				
0x0C	RCC_CIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	Res.	PLL12SRDYC	PLL12RDYC	HSE12DYC	HS12RDYC	LSERDYC	LS12RDYC	Res.	Res.	PLL12SRDYIE	PLL12RDYIE	HSE12DYIE	HS12RDYIE	LSERDYIE	LS12RDYIE	CSSF	Res.	PLL12SRDYF	PLL12RDYF	HSE12DYF	HS12RDYF	LSERDYF	LS12RDYF	
0x10	RCC_AHB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA12RST	DMA1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	Res.	Res.	GPIOH12RST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
0x14	RCC_AHB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x18	RCC_AHB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x1C	予約済み																																	
0x20	RCC_APB1RSTR	Res.	Res.	Res.	PWR12RST	Res.	CAN12RST	CAN1RST	DFSDM1RST	I2C12RST	I2C2RST	I2C1RST	SPI12RST	Res.	Res.	USART12RST	USART2RST	TIM12RST	Res.	SPI12RST	SPI2RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x24	RCC_APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI12RST	Res.	TIM11RST	TIM10RST	TIM9RST	Res.	SYSCFG12RST	Res.	SP12RST	SPI1RST	SDIO1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x28	予約済み																																	
0x2C	予約済み																																	
0x30	RCC_AHB1ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA12EN	DMA1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x34	RCC_AHB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x38	RCC_AHB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x3C	予約済み																																	

表 24. STM32F412xx の RCC レジスタマップとリセット値 (続き)

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	RCC_APB1E NR	Res	Res	Res	PWREN	Res	CAN2EN	CAN1EN	DFSDM1EN	I2C3EN	I2C2EN	I2C1EN	Res	Res	Res	USART2EN	Res	SPI3EN	SPI2EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x44	RCC_APB2E NR	Res	Res	Res	Res	Res	Res	Res	DFSDM1EN	Res	Res	Res	SPI5EN	Res	TIM11EN	TIM10EN	TIM9EN	Res	SYSCFGEN	SPI4EN	SPI1EN	SDIOEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x48	予約済み																																
0x4C	予約済み																																
0x50	RCC_AHB1L PENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA2LPEN	DMA1LPEN	Res	Res	Res	Res	SRAM1LPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x54	RCC_AHB2L PENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x58	RCC_AHB3L PENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x5C	予約済み																																
0x60	RCC_APB1L PENR	Res	Res	Res	PWRLPEN	Res	CAN2LPEN	CAN1LPEN	DFSDM1LPEN	I2C3LPEN	I2C2LPEN	I2C1LPEN	Res	Res	Res	USART3LPEN	USART2LPEN	Res	SPI3LPEN	SPI2LPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x64	RCC_APB2L PENR	Res	Res	Res	Res	Res	Res	Res	DFSDM1LPEN	Res	Res	Res	SPI5LPEN	Res	TIM11LPEN	TIM10LPEN	TIM9LPEN	Res	EXTITEN	SPI3FGLPEN	SPI2FGLPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x68	予約済み																																
0x6C	予約済み																																
0x70	RCC_BDCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x74	RCC_CSR	LPWRSTF	Res	WDGRSTF	Res	WDGRSTF	Res	SFTRSTF	Res	PORRSTF	Res	PADRSTF	Res	BORRSTF	Res	RMVF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x78	予約済み																																
0x7C	予約済み																																
0x80	RCC_SSCGR	SSCGEN	SPREADSEL	Res	Res	INCSTEP[14:0]																MODPER[11:0]											
0x84	RCC_PLLI2S CFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x88	予約済み																																

表 24. STM32F412xx の RCC レジスタマップとリセット値 (続き)

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8C	RCC_DCKCFGR	CKDFSDM1SEL	Res.	Res.	Res.	I2S2SRC[1:0]	I2S1SRC[1:0]		TIMPRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKDFSDM1ASEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x90	CKGATENR																									EVTCL_CKEN	RCC_CKEN	FLITF_CKEN	SRAM_CKEN	SPARE_CKEN	CM4DBG_CKEN	AHB2APB2_CKEN	AHB2APB1_CKEN
0x94	RCC_DCKCFGR2	Res.	Res.	Res.	SDIOSEL	CK48MSEL	Res.	Res.	Res.	I2CFMP1SEL[1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

レジスタ境界アドレスについては、表 1 (49 ページ) を参照してください。



7 汎用 I/O (GPIO)

7.1 GPIO の概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx_IDR、GPIOx_ODR)、32 ビットセット/リセットレジスタ (GPIOx_BSRR)、32 ビットロックレジスタ (GPIOx_LCKR)、2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx_AFRH、GPIOx_AFRL) があります。

7.2 GPIO の主な機能

- 最大 16 個の I/O を制御
- 出力状態：プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態：フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx_ODR へのビット単位の書き込みアクセス用のビットセット/リセットレジスタ (GPIOx_BSRR)
- I/O 設定を凍結可能にするロック機構 (GPIOx_LCKR)
- アナログ機能
- オルタネート機能入力/出力選択レジスタ (入出力ごとに最大 16 のオルタネート機能)
- 2 クロックサイクルで変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能

7.3 GPIO の機能説明

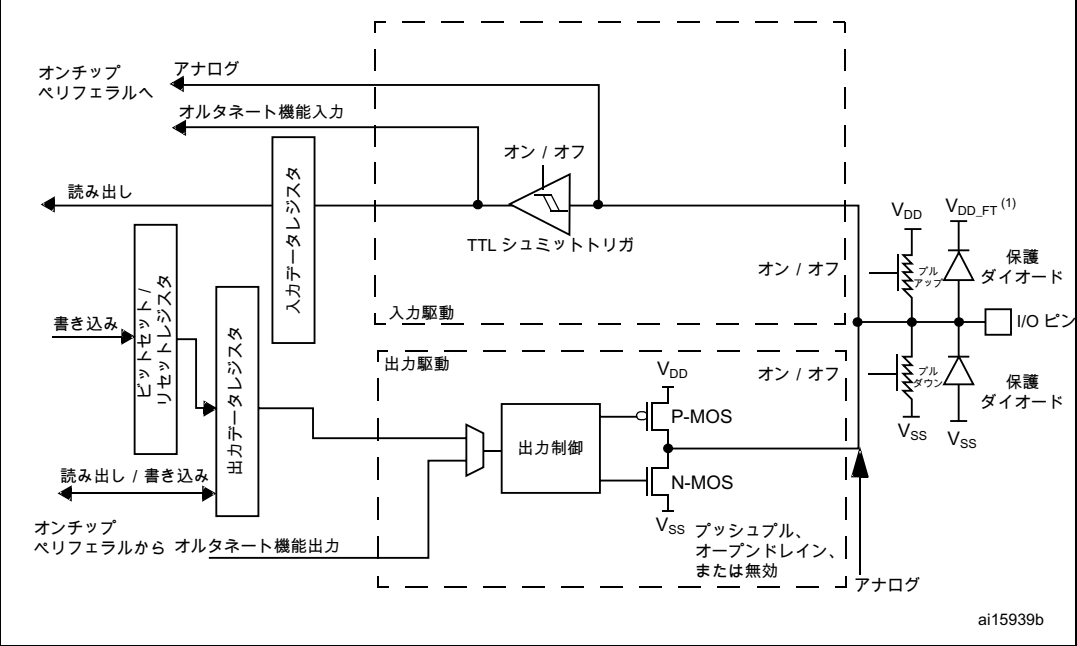
各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx_BSRR レジスタを使用すると、任意の GPIO レジスタにビット単位の読み出し/変更アクセスを行うことができます。これにより、読み出しと変更アクセスの間に IRQ が発生するリスクを回避できます。

図 17 に 5 V トレラント I/O ポートビットの基本構成を示します。表 25 には、実行可能なポートビットの設定を示します。

図 17. 5 V トレラント I/O ポートビットの基本構成



1. V_{DD_FT} は、5 V トレラント I/O に固有の電位であり、 V_{DD} とは異なります。

表 25. ポートビット設定表⁽¹⁾

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]		I/O 設定	
01	0	SPEED [B:A]	0	0	GP 出力	PP
	0		0	1	GP 出力	PP + PU
	0		1	0	GP 出力	PP + PD
	0		1	1	予約済み	
	1		0	0	GP 出力	OD
	1		0	1	GP 出力	OD + PU
	1		1	0	GP 出力	OD + PD
	1		1	1	予約済み (GP 出力 OD)	
10	0	SPEED [B:A]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	予約済み	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	予約済み	

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 25. ポートビット設定表⁽¹⁾ (続き)

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O 設定	
00	x	x	x	0	0	入力	フローティング
	x	x	x	0	1	入力	PU
	x	x	x	1	0	入力	PD
	x	x	x	1	1	予約済み（入力フローティング）	
11	x	x	x	0	0	入力／出力	アナログ
	x	x	x	0	1	予約済み	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 汎用, PP = プッシュプル, PU = プルアップ, PD = プルダウン, OD = オープンドレイン, AF = オルタネート機能

7.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、I/O ポートは入力フローティングモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA15 : JTDI プルアップ
- PA14 : JTCK/SWCLK プルダウン
- PA13 : JTMS/SWDAT プルアップ
- PB4 : NJTRST プルアップ
- PB3 : JTDO フローティング状態

ピンが出力として設定されている場合、出力データレジスタ (GPIOx_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモードで 사용할 수 있습니다 (0 を出力すると、N-MOS のみが有効になります)。

入力データレジスタ (GPIOx_IDR) は、AHB1 クロックサイクルごとに、I/O ピン上のデータをキャプチャします。

すべての GPIO ピンに、内部ウィークプルアップ抵抗とウィークプルダウン抵抗があり、GPIOx_PUPDR レジスタの値によってこれらを有効化/無効化できます。

7.3.2 I/O ピンマルチプレクサとマッピング

本マイクロコントローラの I/O ピンは、マルチプレクサを通してオンボードのペリフェラル/モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間での競合を無くすことができます。

各 I/O ピンは、16 のオルタネート機能入力 (AF0 ~ AF15) を持つマルチプレクサを持ち、これらの入力は、GPIOx_AFRL (ピン 0 ~ 7) と GPIOx_AFRH (ピン 8 から 15) レジスタを通して設定することができます。

- リセット後、すべての I/O はシステムのオルタネート機能 0 (AF0) に接続されます。
- ペリフェラルのオルタネート機能は AF1 ~ AF13 に配置されます
- FPU 搭載 Cortex®-M4 EVENTOUT は AF15 に配置されます。

この仕組みを次の [図 18 : STM32F412xx におけるオルタネート機能の選択](#) に示します。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで利用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従います。

● システム機能

I/O を AF0 に接続し、使用する機能に応じて設定します。

- JTAG/SWD : 各デバイスリセット後、これらのピンはデバッガーストが直ちに使用できる専用ピンとして割り当てられます (GPIO コントローラでは制御されません)
- RTC_REFIN : このピンは入力フローティングモードに設定する必要があります
- MCO1 および MCO2 : これらのピンはオルタネート機能モードに設定する必要があります。

注 : JTAG/SWD ピンの一部またはすべてを無効にすることができるため、該当するピンを GPIO としても使用できます。

詳細については、[セクション 6.2.10 : クロック信号出力](#)を参照してください。

表 26. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ I/O ピン配置				
	PA13 / JTMS/ SWDIO	PA14 / JTCK/ SWCLK	PA15 / JTDI	PB3 / JTDO	PB4/ NJTRST
全 SWJ (JTAG-DP + SW-DP) - リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効					解放

● GPIO

必要とする I/O を、GPIOx_MODER レジスタで出力または入力として設定します。

● ペリフェラルオルタネート機能

ADC については、必要とする I/O を GPIOx_MODER レジスタでアナログとして設定します。

他のペリフェラルでは次のように設定します。

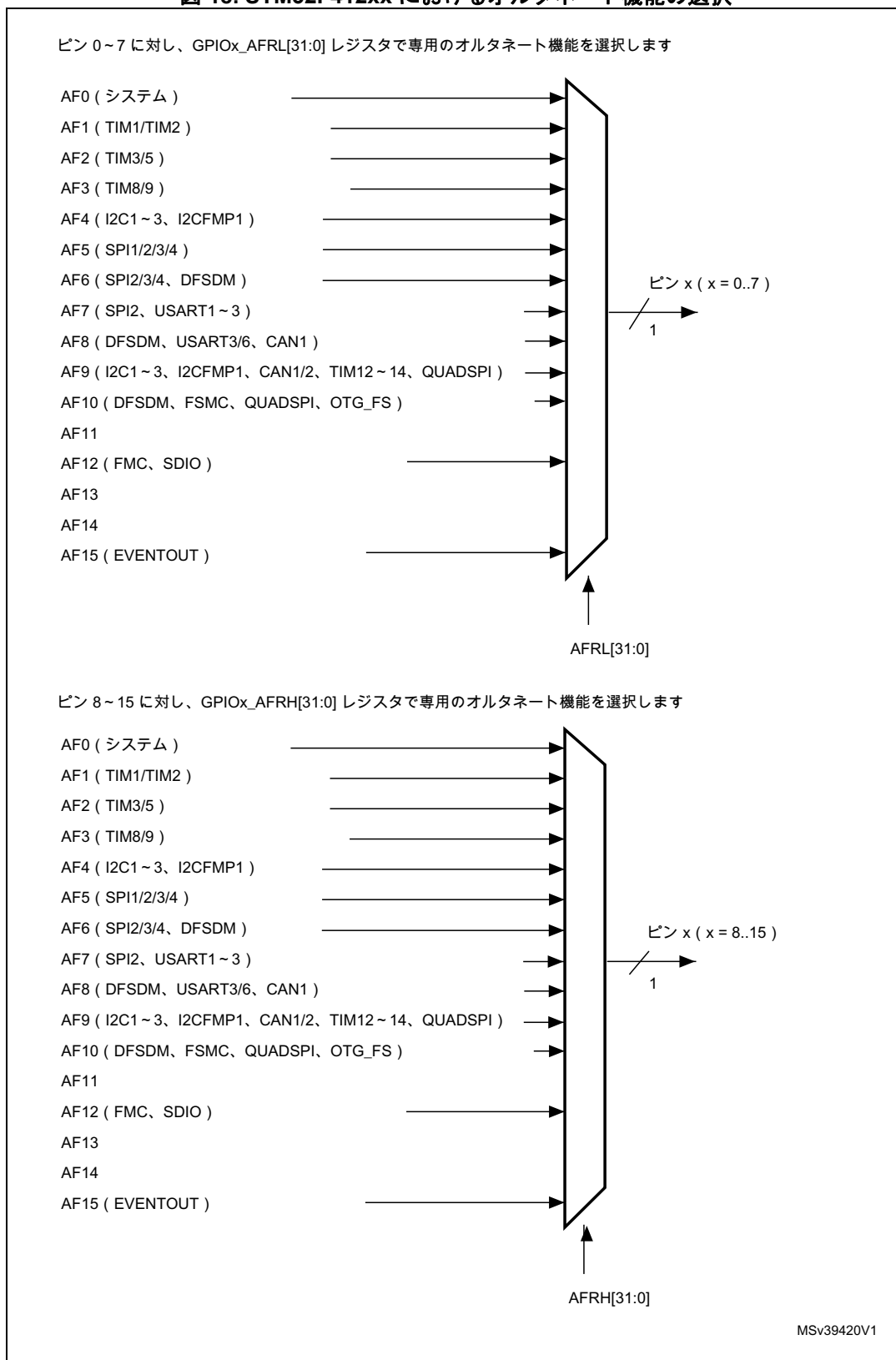
- 必要とする I/O を、GPIOx_MODER レジスタでオルタネート機能として設定します。
- タイプ、プルアップ／プルダウン、出力スピードをそれぞれ GPIOx_OTYPER、GPIOx_PUPDR、GPIOx_OSPEEDR レジスタで選択します。
- I/O を GPIOx_AFR1 または GPIOx_AFRH レジスタで、必要とする AFx に接続します。

● EVENTOUT

AF15 に接続することで、I/O ピンを FPU 搭載 Cortex®-M4 EVENTOUT 信号を出力する用途に設定します。

注 : システムおよびペリフェラルのオルタネート機能 I/O ピンの詳細配置については、データシートの「オルタネート機能配置」表を参照してください。

図 18. STM32F412xx におけるオルタネート機能の選択



7.3.3 I/O ポート制御レジスタ

各 GPIO には 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR) あり、最大 16 個の I/O を設定します。

GPIOx_MODER レジスタは I/O 方向 (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx_OTYPER および GPIOx_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) および速度を選択するために使用されます (I/O の方向がどちらであっても、I/O 速度ピンは、対応する GPIOx_OSPEEDR レジスタの各ビットに直接接続されています)。I/O の方向がどちらであっても、GPIOx_PUPDR レジスタは、プルアップ/プルダウンを選択するために使用されます。

7.3.4 I/O ポートデータレジスタ

各 GPIO には、16 ビットメモリマップドデータレジスタが 2 つあります。入力データレジスタ (GPIOx_IDR) と出力データレジスタ (GPIOx_ODR) です。GPIOx_ODR は出力されるデータを格納し、読み出し/書き込みアクセスが可能です。I/O から入力されるデータは読み出し専用の入力データレジスタ (GPIOx_IDR) に格納されます。

レジスタの説明は、[セクション 7.4.5 : GPIO ポート入力データレジスタ \(GPIOx_IDR\) \(x = A ~ H\)](#) および [セクション 7.4.6 : GPIO ポート出力データレジスタ \(GPIOx_ODR\) \(x = A ~ H\)](#) を参照してください。

7.3.5 I/O データのビット単位の操作

ビットセット/リセットレジスタ (GPIOx_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx_ODR) のビット単位のセット/リセットを可能にします。ビットセット/リセットレジスタは GPIOx_ODR の 2 倍のサイズです。

GPIOx_ODR の各ビットには GPIOx_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。BSRR(i) および BSRR(i+SIZE) です。ビット BSRR(i) に 1 を書き込むと、対応する ODR(i) ビットがセットされます。ビット BSRR(i+SIZE) に 1 を書き込むと、対応する ODR(i) ビットがリセットされます。

GPIOx_BSRR のいかなるビットに 0 を書き込んでも GPIOx_ODR の対応するビットには影響しません。仮に、GPIOx_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx_BSRR レジスタを使用した GPIOx_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx_ODR ビットを固定するものではありません。GPIOx_ODR のビットは常に直接アクセスすることができます。GPIOx_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx_ODR をビットレベルでプログラムする場合は、ソフトウェアで割り込みを無効にする必要はありません。1 回の不可分な AHB1 書き込みアクセスで 1 ビットまたは複数ビットを変更することができます。

7.3.6 GPIO ロック機構

GPIOx_LCKR レジスタへ特定の書き込みシーケンスを行うことにより、GPIO 制御レジスタをロックすることができます。ロックされるレジスタは、GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH です。

GPIOx_LCKR レジスタに書き込むには、特定の書き込み/読み出しシーケンスを行う必要があります。このレジスタのビット 16 に適切な LOCK シーケンスを行う場合、LCKR[15:0] の値を使用して I/O の設定を固定します (この書き込みシーケンス中、LCKR[15:0] の値は同じである必要があります)。あるポートビットに LOCK シーケンスが行われると、次の MCU またはペリフェラルリセット

まで、そのポートビットの値を変更できなくなります。GPIOx_LCKR の各ビットによって、制御レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH) の対応するビットがロックされます。

GPIOx_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンス (セクション 7.4.8 : GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ H) を参照) は、GPIOx_LCKR レジスタへのワード (32 ビット長) アクセスを使用してのみ実行できます。

詳細については、セクション 7.4.8 : GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ H) の LCKR レジスタの説明を参照してください。

7.3.7 I/O オルタネート機能の入力／出力

各 I/O が使用できる 16 個のオルタネート機能入力／出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx_AFRL および GPIOx_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることになります。こうして各 I/O に使用可能な機能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、1 つの I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

どの機能が各 GPIO ピンに多重化されているかについてはデータシートを参照してください。

注： アプリケーションからは、各 I/O に対して使用可能な機能を一度に 1 つだけ選択することができます。

7.3.8 外部割り込み／ウェイクアップライン

すべてのポートに外部割り込み機能があります。外部割り込みラインを使用するには、ポートを入力モードに設定する必要があります。セクション 10.2 : 外部割り込み／イベントコントローラ (EXTI) および セクション 10.2.3 : ウェイクアップイベント管理 を参照してください。

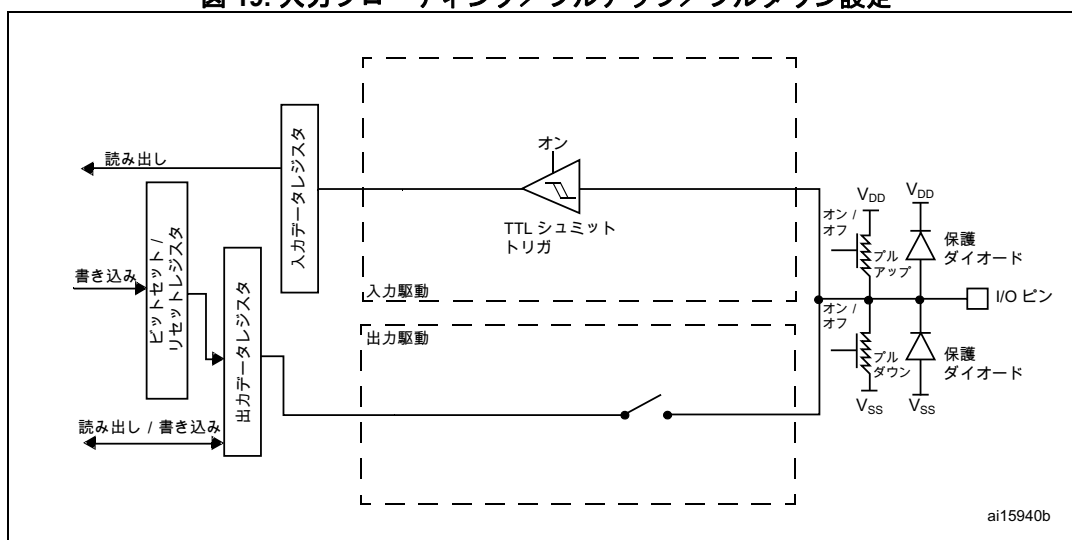
7.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力が有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB1 クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

図 19 は、I/O ポートビットの入力設定を示しています。

図 19. 入力フローティング／プルアップ／プルダウン設定



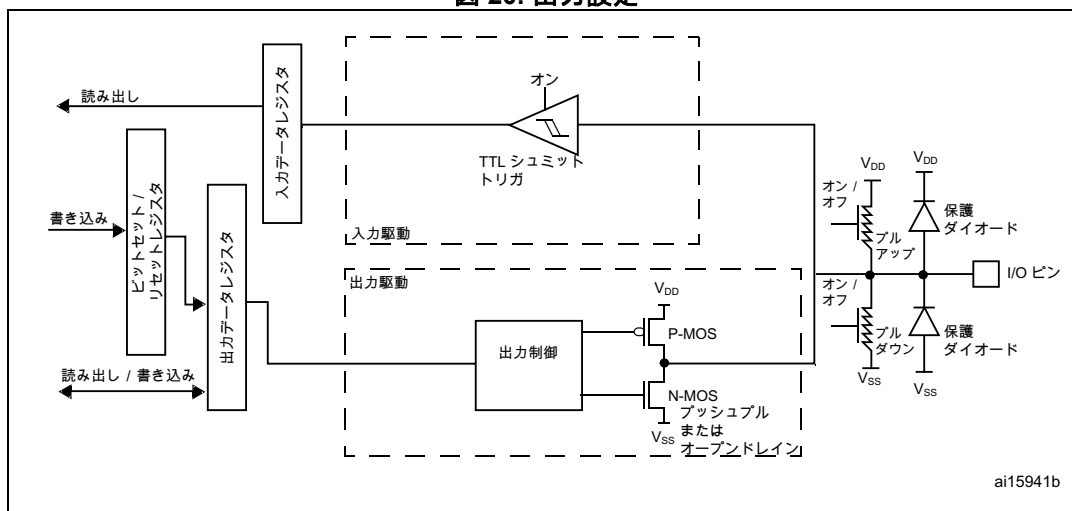
7.3.10 出力設定

I/O ポートが出力としてプログラムされた場合、

- 出力バッファが有効になります。
 - － オープンドレインモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときはポートはハイインピーダンス状態のままです（P-MOS が有効になることはありません）。
 - － プッシュプルモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときは P-MOS が有効になります。
- シュミットトリガ入力があります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB1 クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。
- 出力データレジスタの読み出しアクセスによって、最後に書き込まれたデータが得られます。

図 20 は、I/O ポートビットの出力設定を示しています。

図 20. 出力設定



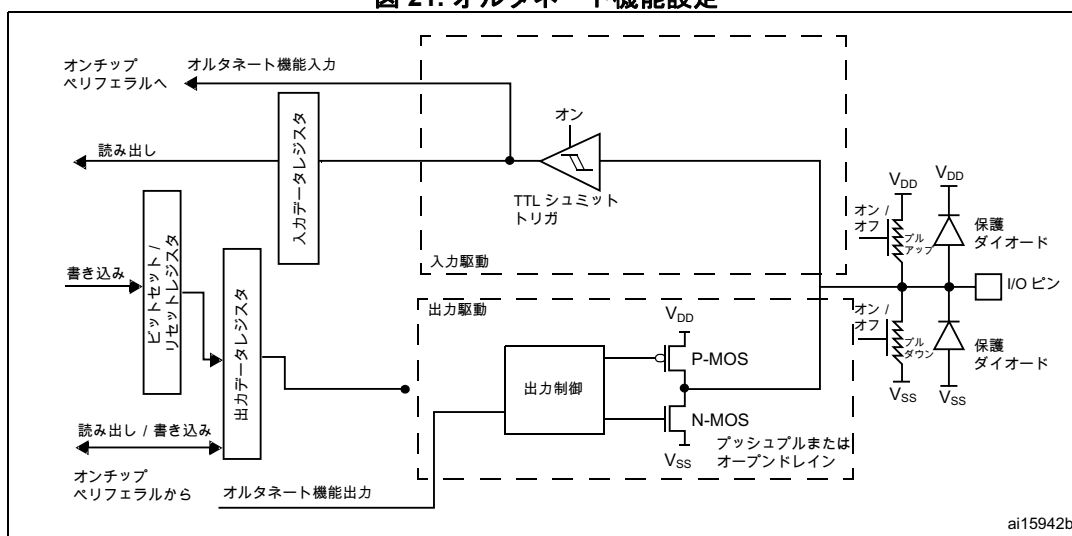
7.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルとして設定することができます。
- 出力バッファが、ペリフェラル (トランスミッタインペブルおよびデータ) からの信号で駆動されます。
- シュミットトリガ入力が有効になります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB1 クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

図 21 は、I/O ポートビットのオルタネート機能設定を示しています。

図 21. オルタネート機能設定



7.3.12 アナログ設定

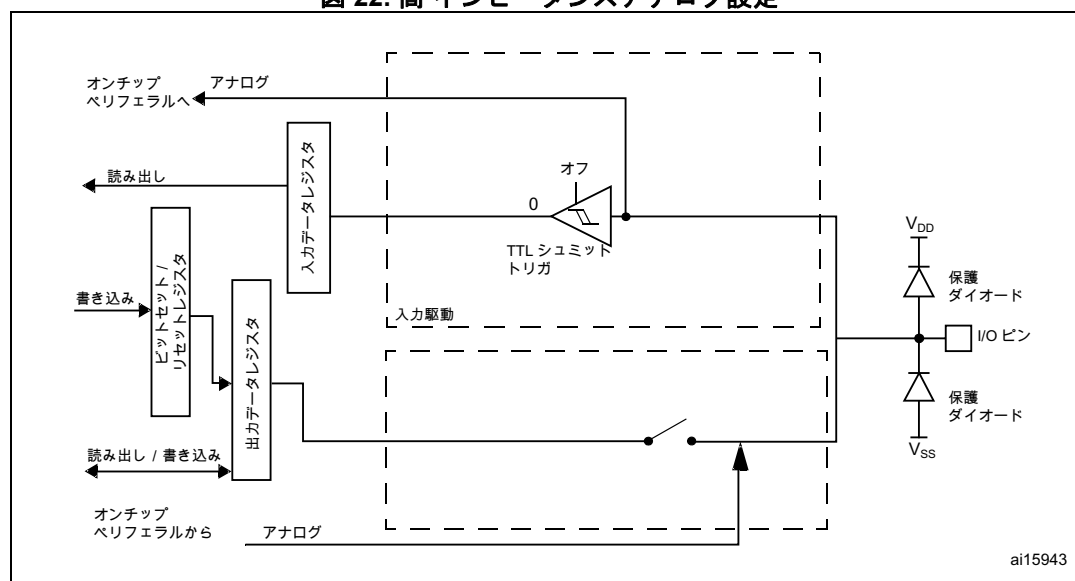
I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗は無効になります。
- 入力データレジスタの読み出しアクセスを行うと、値“0”が得られます。

注： アナログ設定では、I/O ピンは 5 V トレラントにできません。

図 22 は、I/O ポートビットの高インピーダンスアナログ入力設定を示しています。

図 22. 高インピーダンスアナログ設定



7.3.13 OSC32_IN/OSC32_OUT ピンの GPIO PC14/PC15 ポートピンとしての使用

LSE オシレータがオフのとき、LSE オシレータのピン OSC32_IN および OSC32_OUT は、それぞれ汎用 I/O の PC14 および PC15 として使用できます。LSE オシレータがオンのとき、PC14 および PC15 の I/O は、LSE オシレータピン OSC32_IN および OSC32_OUT としてのみ設定できます。この設定は、RCC_BDCR レジスタの LSEON ビットをセットすることによって行います。LSE は、GPIO 機能より優先順位が高くなっています。

注： 1.2 V ドメインの電源がオフになった（デバイスが STANDBY モードに入った）とき、またはバックアップドメインに V_{BAT} から電源供給された（ V_{DD} が供給されない）とき、PC14/PC15 の GPIO が機能しなくなります。この場合、I/O はアナログ入力モードに設定されます。

7.3.14 OSC_IN/OSC_OUT ピンの GPIO PH0/PH1 ポートピンとしての使用

HSE オシレータがオフのとき、HSE オシレータのピン OSC_IN/OSC_OUT は、それぞれ汎用 I/O の PH0/PH1 として使用できます。（リセット後、HSE オシレータはオフになります）。HSE オシレータがオンのとき、PH0/PH1 I/O は、OSC_IN/OSC_OUT HSE オシレータピンとしてのみ設定できます。この設定は、RCC_CR レジスタの HSEON ビットをセットすることによって行います。HSE は、GPIO 機能より優先順位が高くなっています。

7.3.15 RTC 追加機能の選択

STM32F4xx には、1 つの GPIO ピン RTC_AF1 があり、タンパまたはタイムスタンプイベントの検出に使用でき、RTC_ALARM または RTC_CALIB の RTC 出力としても使用できます。

- RTC_AF1 (PC13) は次の目的に使用できます。

RTC_ALARM 出力：この出力は RTC_CR レジスタの OSEL[1:0] ビットによって RTC アラーム A、RTC アラーム B、RTC ウェイクアップのいずれかとして使用できます

- RTC_CALIB 出力：この機能は RTC_CR レジスタの COE[23] をセットすることで有効になります
- RTC_TAMP1：タンパイベント検出
- RTC_TS：タイムスタンプイベント検出

対応するピンの選択は次のように RTC_TAFCR レジスタを通して行います。

- TAMP1INSEL を使用して、どのピンを RTC_TAMP1 タンパ入力として使用するか選択します。
- TSINSEL を使用して、どのピンを RTC_TS タイムスタンプ入力として使用するか選択します。
- ALARMOUTTYPE を使用して、RTC_ALARM をプッシュプルモード出力とするかオープンドレインモード出力とするか選択します。

出力形式は 表 27 に一覧化された優先順位に従います。

表 27. RTC 追加機能⁽¹⁾

ピンの設定と機能	有効	有効	タンパ有効	タイムスタンプ有効	TAMP1INSEL AMPER1 ピン選択	TSINSELTIME STAMP ピン選択	ALARMOUTTYP の設定
アラーム出力 OD	1	無視	無視	無視	無視	無視	0
アラーム出力 PP	1	無視	無視	無視	無視	無視	1
較正出力 PP	0	1	無視	無視	無視	無視	無視
TAMPER1 入力フローティング	0	0	1	0	0	無視	無視
TIMESTAMP および TAMPER1 入力フローティング	0	0	1	1	0	0	無視
TIMESTAMP 入力フローティング	0	0	0	1	無視	0	無視
標準 GPIO	0	0	0	0	無視	無視	無視

1. OD：オープンドレイン、PP：プッシュプル

7.4 GPIO レジスタ

このセクションには、GPIO レジスタの詳細な説明が記載されています。
レジスタビット、レジスタアドレスオフセット、リセット値の概要は、表 28 を参照してください。
GPIO レジスタはバイト（8 ビット）、ハーフワード（16 ビット）、ワード（32 ビット）のいずれかの単位でアクセスできます。

7.4.1 GPIO ポートモードレジスタ (GPIOx_MODER) (x = A~H)

アドレスオフセット : 0x00
リセット値 :
● ポート A 0x0A800 0000
● ポート B 0x0000 0280
● 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 2y : 2y+1 **MODERy[1:0]** : ポート x 設定ビット (y = 0~15)
これらのビットは、I/O 方向モードを設定するために、ソフトウェアによって書き込まれます。
00 : 入力 (リセット状態)
01 : 汎用出力モード
10 : オルタネート機能モード
11 : アナログモード

7.4.2 GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A~H)

アドレスオフセット : 0x04
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。
ビット 15:0 **OTy** : ポート x 設定ビット (y = 0~15)
これらのビットは、I/O ポートの出力タイプを設定するために、ソフトウェアによって書き込まれます。
0 : 出力プッシュプル (リセット状態)
1 : 出力オープンドレイン

7.4.3 GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A~H)

アドレスオフセット : 0x08

リセット値 :

- ポート A 0x0C00 0000
- ポート B 0x0000 00C0
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット $2y : 2y+1$ OSPEEDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

- 00 : ロースピード
- 01 : ミディアムスピード
- 10 : 準高速
- 11 : ハイスピード

注: OSPEEDRy ビット値対 V_{DD} 範囲および外部負荷については製品データシートを参照してください。

7.4.4 GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A~H)

アドレスオフセット : 0x0C

リセット値 :

- ポート A 0x6400 0000
- ポート B 0x0000 0100
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット $2y : 2y+1$ PUPDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

- 00 : プルアップ／プルダウンなし
- 01 : プルアップ
- 10 : プルダウン
- 11 : 予約済み



7.4.5 GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A~H)

アドレスオフセット : 0x10

リセット値 : 0x0000 XXXX (X は未定義)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **IDRy** : ポート入力データ (x = 0..15)

これらのビットは、読み出し専用であり、ワードモードでのみアクセスできます。これらのビットには、対応する I/O ポートの入力値が格納されています。

7.4.6 GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A~H)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ODRy** : ポート出力データ (y = 0..15)

これらのビットは、ソフトウェアによって読み出し／書き込みができます。

注 : ビット単位のセット／リセットのために、GPIOx_BSRR レジスタ (x = A~H) への書き込みによって、ODR ビットを個々にセット／リセットできます。

7.4.7 GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR) (x = A~H)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 **BRy** : ポート x リセットビット y (y= 0~15)

これらのビットは、書き込み専用であり、ワードモード、ハーフワードモード、バイトモードのいずれかでアクセスできます。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをリセットします。

注 : **BSx ビットと BRx ビットの両方がセットされた場合、BSx ビットが優先されます。**

ビット 15:0 **BSy** : ポート x セットビット y (y= 0~15)

これらのビットは、書き込み専用であり、ワードモード、ハーフワードモード、バイトモードのいずれかでアクセスできます。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをセットします。

7.4.8 GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A~H)

このレジスタは、ビット 16 (LCKK) に正しい書き込みシーケンスが行われたときに、ポートビットの設定をロックするために使用されます。ビット [15:0] の値は、GPIO の設定をロックするために使用されます。書き込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU またはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

注 : **GPIOx_LCKR レジスタへの書き込みには特定の書き込みシーケンスが使われます。この書き込みシーケンスではワードアクセス (32 ビット長) のみ可能です。**

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : 32 ビットワードのみ、読み出し/書き込みレジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **LCKK[16]** : ロックキー

このビットは常時読み出しができます。ロックキー書き込みシーケンスを使用しなければ変更できません。

0 : ポート設定ロックキーは無効です。

1 : ポート設定ロックキーは有効です。GPIOx_LCKR レジスタは、MCU リセットまたはペリフェラルリセットが発生するまでロックされます。

ロックキー書き込みシーケンス :

書き込み LCKR[16] = “1” + LCKR[15:0]

書き込み LCKR[16] = “0” + LCKR[15:0]

書き込み LCKR[16] = “1” + LCKR[15:0]

読み出し LCKR

読み出し LCKR[16] = “1” (この読み出し操作は任意だが、ロックが有効であることを確認できる。)

注 : **ロックキー書き込みシーケンスの間は、LCK[15:0] の値を変更することはできません。**

ロックシーケンス中にエラーが発生すると、ロックは中止されます。

ポートの任意のビットの最初のロックシーケンスの後、次の CPU リセットまでは、LCKK ビットのいかなる読み出しアクセスに対しても、“1”が返されます。

ビット 15:0 **LCKy** : ポート x ロックビット y (y = 0~15)

これらのビットは読み出し／書き込みができますが、書き込めるのは LCKK ビットが“0”のときだけです。

0 : ポート設定はロックされません。

1 : ポート設定はロックされます。

7.4.9 **GPIO オルタネート機能下位レジスタ (GPIOx_AFRL) (x = A~H)**

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **AFRLy** : ポート x ビット y (y = 0..7) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRLy 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15



7.4.10 GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A~H)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **AFRHy** : ポート x ビット y (y = 8..15) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRHy 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15

7.4.11 GPIO レジスタマップ

次の表に、GPIO レジスタマップとリセット値を示します。

表 28. GPIO レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
0x00	GPIOA_ MODER	MODER15[1:0]			MODER14[1:0]			MODER13[1:0]			MODER12[1:0]			MODER11[1:0]			MODER10[1:0]			MODER9[1:0]			MODER8[1:0]			MODER7[1:0]			MODER6[1:0]			MODER5[1:0]			MODER4[1:0]			MODER3[1:0]			MODER2[1:0]			MODER1[1:0]			MODER0[1:0]
	リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00	GPIOB_ MODER	MODER15[1:0]			MODER14[1:0]			MODER13[1:0]			MODER12[1:0]			MODER11[1:0]			MODER10[1:0]			MODER9[1:0]			MODER8[1:0]			MODER7[1:0]			MODER6[1:0]			MODER5[1:0]			MODER4[1:0]			MODER3[1:0]			MODER2[1:0]			MODER1[1:0]			MODER0[1:0]
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x00	GPIOx_ MODER (x = C～H)	MODER15[1:0]			MODER14[1:0]			MODER13[1:0]			MODER12[1:0]			MODER11[1:0]			MODER10[1:0]			MODER9[1:0]			MODER8[1:0]			MODER7[1:0]			MODER6[1:0]			MODER5[1:0]			MODER4[1:0]			MODER3[1:0]			MODER2[1:0]			MODER1[1:0]			MODER0[1:0]
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

[illegible]

表 28. GPIO レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	GPIOx_LCKR (x = A~H)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A~H)	AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]				AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (x = A~H)	AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]				AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

セクション：メモリマップでレジスタ境界アドレスについて参照してください。

8 システム設定コントローラ (SYSCFG)

システム設定コントローラは、主にコード領域内でアクセス可能なメモリの再配置、および GPIO に対する外部割り込みライン接続の管理に使用します。

8.1 I/O 補正セル

デフォルトでは I/O 補正セルは使用されません。しかしながら、I/O 出力バッファ速度が 50 MHz または 100 MHz モードに設定されている場合、I/O の状態変化 $t_{r(I/O)out}/t_{r(I/O)out}$ のスルーレート制御のため、補正セルを使用して電源の I/O ノイズを低減することを推奨します。

補正セルが有効になると、補正セルが使用可能であることを示すため READY フラグがセットされ、使用可能となります。I/O 補正セルは、電源電圧が 2.4~3.6 V の範囲にある場合にのみ使用できます。

8.2 SYSCFG レジスタ

8.2.1 SYSCFG メモリ再配置レジスタ (SYSCFG_MEMRMP)

このレジスタは、メモリ再配置時の特定の設定に使用されます。

- アドレス 0x0000 0000 でアクセス可能なメモリのタイプを設定するために、2 つのビットが使用されます。これらのビットを用いて、ソフトウェアで物理的な再配置を選択し、BOOT ピンをバイパスします。
- リセット後、これらのビットは、BOOT ピンで選択された値になります。BOOT0 ピンを 0 に設定してメインフラッシュメモリから起動する場合、このレジスタの値は 0x00 になります。

再割り当てモードでは、CPU がシステムバスではなく ICode バスを通じて外部メモリにアクセスできるため、パフォーマンスを高めることができます。

アドレスオフセット : 0x00

リセット値 : 0x0000 000X (X は BOOT ピンで選択されたメモリモード)

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MEM_MODE** : メモリマッピング選択

ソフトウェアでセット／クリアされます。これらのビットによって、アドレス 0x0000 0000 のメモリ内部のマッピングを制御します。リセット後、これらのビットは、BOOT ピンで選択された値になります。

00 : メインフラッシュメモリは 0x0000 0000 に配置されます。

01 : システムメモリは 0x0000 0000 に配置されます。

10 : 予約済み

11 : 内蔵 SRAM は 0x0000 0000 に配置されます。

注 : アドレス 0x0000 0000 のメモリマッピングの詳細は [図 2: メモリマップ](#) を参照してください。

8.2.2 SYSCFG ペリフェラルモード設定レジスタ (SYSCFG_PMC)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC1D C2
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **ADC1DC2** :

0 : 影響はありません。

1 : このビットの使い方は、AN4073 を参照してください。

注 : これらのビットは、次の条件が満足されたときのみセットできます。

- ADC クロックが 30 MHz 以上

- 複数の ADC 変換が同時に開始せず、サンプリング時間が異なる場合、ADC1DC2 ビットは、ただ 1 つだけ選択する必要があります。

- PWR_CR レジスタで ADCDC1 ビットがセットされている場合、これらのビットをセットすることはできません。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

8.2.3 SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 0~3)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

0111 : PH[x] ピン (EXTI3 および EXTI2 は予約済み)

他の設定 : 予約済み

8.2.4 SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 4~7)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

他の設定 : 予約済み

8.2.5 SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 8~11)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

他の設定 : 予約済み

8.2.6 SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 12~15)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン



8.2.7 SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **PVDL** : PVD ロック

このビットはソフトウェアでセットされます。システムリセットによってのみクリアされます。これは、TIM1/8 のブレーク入力への PVD 接続を有効化/ロックします。また、PWR_CR レジスタの PVDE および PVDS[2:0] ビットもロック（書き込み保護）します。
0 : PVD 割り込みはTIM1/8 のブレーク入力に接続されていません。PVDE および PVDS[2:0] は、読み出しおよび変更が可能です。
1 : PVD 割り込みはTIM1/8 のブレーク入力に接続されています。PVDE および PVDS[2:0] は、読み出し専用です。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CLL** : コア LOCKUP ロック

このビットは、ソフトウェアによってセット/クリアされます。これは、TIM1/8 のブレーク入力によりFPU 搭載 Cortex®-M4コアの LOCKUP（ハードフォールト）出力を有効化/ロックします。
0 : FPU 搭載 Cortex®-M4 LOCKUP 出力は TIM1/8 のブレーク入力に接続されていません。
1 : FPU 搭載 Cortex®-M4 LOCKUP 出力は TIM1/8 のブレーク入力に接続されています。

8.2.8 補正セル制御レジスタ (SYSCFG_CMPCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **READY** : 補正セルレディフラグ
0 : I/O 補正セルはレディ状態ではありません。
1 : I/O 補正セルはレディ状態です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CMP_PD** : 補正セルパワーダウン
0 : I/O 補正セルパワーダウンモード
1 : I/O 補正セルは有効です。

8.2.9 SYSCFG 設定レジスタ (SYSCFG_CFGR)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2CFMP1_SDA	I2CFMP1_SCL
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **I2CFMP1_SDA**
ソフトウェアでセット／クリアされます。このビットがセットされると、GPIO ポートモードレジスタおよび GPIO オルタネート機能選択ビットを介して選択された I2CFMP1_SDA ピンで Fm+ の駆動能力が強制的に有効になります。

ビット 0 **I2CFMP1_SCL**
ソフトウェアでセット／クリアされます。このビットがセットされると、GPIO ポートモードレジスタおよび GPIO オルタネート機能選択ビットを介して選択された I2CFMP1_SCL ピンで Fm+ の駆動能力が強制的に有効になります。

8.2.10 SYSCFG レジスタマップ

次の表に、SYSCFG レジスタマップとリセット値を示します。

表 29. SYSCFG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SYSCFG_MEMRMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM_MODE	
	リセット値																															x	x
0x04	SYSCFG_PMC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC1DC2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																0																
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]				EXTI2[3:0]			EXTI1[3:0]			EXTI0[3:0]					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]				EXTI6[3:0]			EXTI5[3:0]			EXTI4[3:0]					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]				EXTI10[3:0]			EXTI9[3:0]			EXTI8[3:0]					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]				EXTI14[3:0]			EXTI13[3:0]			EXTI12[3:0]					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SYSCFG_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDL	Res.	CLL
	リセット値																													0		0	
0x20	SYSCFG_CMPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	CMP_PD
	リセット値																								0							0	
0x24	SYSCFG_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2CFMP1_SDA	
	リセット値																									0						0	I2CFMP1_SCL

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

9 ダイレクトメモリアクセスコントローラ (DMA)

9.1 DMA の概要

DMA (Direct Memory Access) は、ペリフェラルとメモリ間、およびメモリ間で高速データ転送を行うために使用します。データは、CPU の動作を必要とすることなく、DMA によって高速に移動できます。これにより、CPU リソースを他の操作のためにあけておくことができます。

このDMA コントローラは、高機能なバスマトリックスアーキテクチャに基づいてシステムの帯域幅を最大限に利用するため、強力なデュアル AHB マスタバスアーキテクチャと、独立した FIFO を兼ね備えています。

2 つの DMA コントローラは合計 16 ストリーム (各コントローラごとに 8 つ) を搭載し、それぞれが 1 つ以上のペリフェラルからのメモリアセスリクエストを管理します。各ストリームは、合計で最大 8 チャンネル (リクエスト) を処理することができます。また、それぞれ DMA リクエスト間の優先順位を操作するためのアービタを内蔵しています。

9.2 DMA の主な機能

DMA の主な機能は次のとおりです。

- デュアル AHB マスタバスアーキテクチャ (メモリアクセス専用、およびペリフェラルへのアクセス専用)
- 32 ビットアクセスのみをサポートする AHB スレーブプログラミングインタフェース
- DMA コントローラごとに 8 ストリーム、ストリームごとに最大 8 チャンネル (リクエスト)
- ストリームごとに独立した 4 つの 32 ビットファーストインファーストアウトメモリバッファ (FIFO)。 (FIFO モードまたはダイレクトモードで使用可能)
 - FIFO モード: FIFO サイズの 1/4、1/2 または 3/4 からソフトウェアで選択可能な閾値レベルを持つ
 - ダイレクトモード
各 DMA リクエストによって、直ちにメモリとの転送を開始します。DMA がダイレクトダイレクトモード (FIFO が無効) に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるよう、DMA はデータをメモリから内部 FIFO へ 1 つだけブロードします。
- 各ストリームはハードウェアによって次のように設定することができます。
 - “ペリフェラルからメモリ”、“メモリからペリフェラル”および“メモリ間”転送をサポートするレギュラチャネル
 - メモリ側でダブルバッファリングもサポートするダブルバッファチャネル
- 8 つのストリームのそれぞれは、専用のハードウェア DMA チャネル (リクエスト) に接続されます。
- DMA ストリームリクエスト間の優先順位は、ソフトウェアによってプログラミングできます (最高、高、中、低の 4 レベル)。レベルが等しい場合はハードウェアによって決まります (リクエスト 0 はリクエスト 1 よりも優先、など) です。

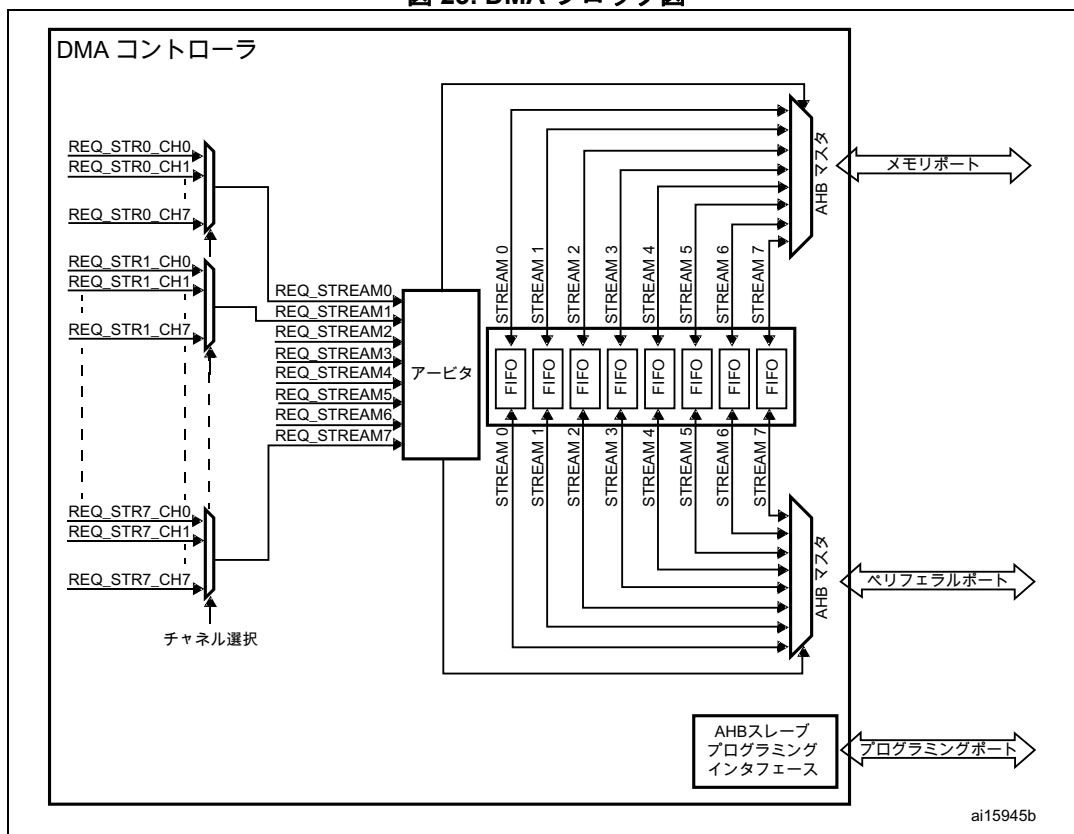
- 各ストリームは、メモリ間転送のためのソフトウェアトリガもサポートしています (DMA2 コントローラでのみ使用可能)。
- 各ストリームリクエストは、発生しうる最大 8 つのチャネルリクエストから選択することができます。この選択は、ソフトウェアによって設定可能であり、これによっていくつかのペリフェラルが DMA リクエストを発生することができるようになります。
- 転送するデータの数、DMA コントローラまたはペリフェラルのいずれかによって管理することができます。
 - DMA フローコントローラ：転送するデータ数は、1 から 65535 の間でソフトウェアによってプログラム可能です。
 - ペリフェラルフローコントローラ：転送されるデータ項目の数は不明であり、ハードウェアによって転送の終了を通知する転送元または転送先ペリフェラルによって制御されます。
- 転送元および転送先の転送幅 (バイト、ハーフワード、ワード) に依存しない：転送元と転送先のデータ幅が異なる場合、DMA によって自動的に必要な転送のパック／アンパックが行われ、バンド幅が最適化されます。この機能は、FIFO モードでのみ使用できます。
- 転送元および転送先に対するインクリメントアドレッシングまたはノンインクリメントアドレッシング
- 4、8、または 16 ビートのインクリメンタルバースト転送をサポート。バーストのサイズはソフトウェアで設定可能であり、通常はペリフェラルの FIFO サイズの半分です。
- サークュラバッファ管理は、各ストリームがサポートします。
- 各ストリームに対する同一の割り込みリクエストにおいて、5 つのイベントフラグ (DMA 1/2 転送、DMA 転送完了、DMA 転送エラー、DMA FIFO エラー、ダイレクトモードエラー) の論理和がとられます。

9.3 DMA の機能説明

9.3.1 概要

図 23 に、DMA のブロック図を示します。

図 23. DMA ブロック図



DMA コントローラはダイレクトメモリ転送を行います。AHB マスタとして、AHB トランザクションを開始するための AHB バスマトリックスの制御を行うことができます。

以下のトランザクションを実行することができます。

- ペリフェラルからメモリ
- メモリからペリフェラル
- メモリ間

この DMA コントローラは 2 つの AHB マスタポートを搭載しています。メモリに接続するための AHB メモリポート、およびペリフェラルに接続するための AHB ペリフェラルポートです。しかし、メモリ間転送を可能とするため、AHB ペリフェラルポートはメモリにもアクセスできる必要があります。

AHB スレーブポートは DMA コントローラをプログラムするために用いられます (32 ビットアクセスのみサポート)。

9.3.2 DMA トランザクション

DMA トランザクションは、一定数のデータ転送の列で構成されています。転送されるデータ項目の数とその幅（8 ビット、16 ビット、または 32 ビット）は、ソフトウェアでプログラム可能です。

各 DMA 転送は次の 3 つの操作で構成されます。

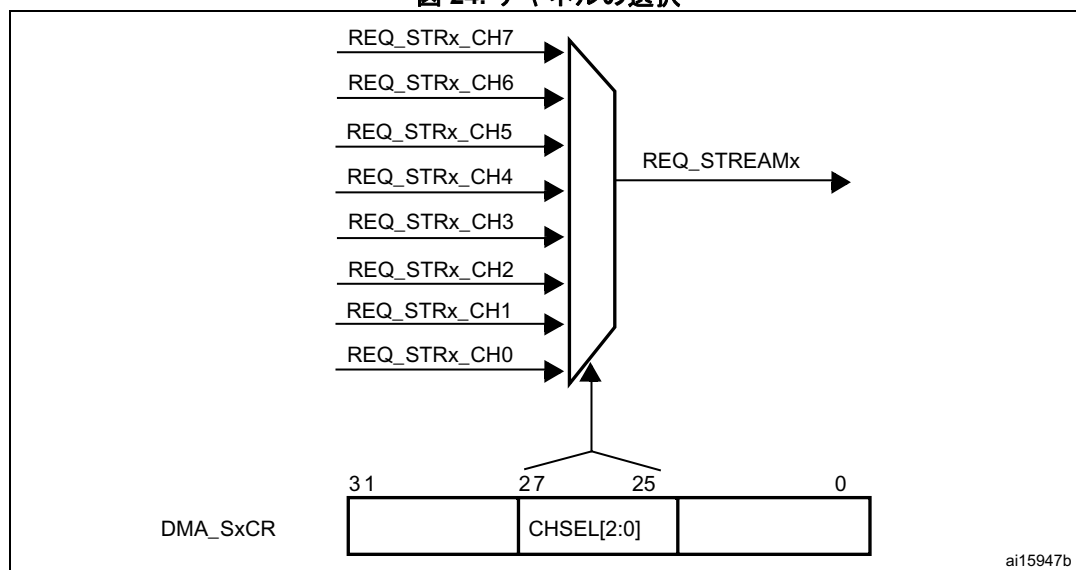
- DMA_SxPAR または DMA_SxM0AR レジスタを通じてアドレス指定されたメモリ上の場所またはペリフェラルデータレジスタからのデータのロード。
- DMA_SxPAR または DMA_SxM0AR レジスタを通じてアドレス指定されたメモリ内の場所またはペリフェラルデータレジスタにロードされたデータの格納。
- このあと予定される DMA トランザクションの数を保持している DMA_SxNDTR レジスタをカウントダウンします。

イベントの終了後、ペリフェラルは DMA コントローラにリクエスト信号を送信します。DMA コントローラは、チャンネルの優先順位に応じて、リクエストを処理します。DMA コントローラがペリフェラルにアクセスするとすぐに、DMA コントローラはペリフェラルに確認応答信号を送信します。ペリフェラルは、DMA コントローラからの確認応答信号を受け取るとすぐにそのリクエストを解除します。ペリフェラルによってリクエストがネゲートされると、DMA コントローラは確認応答信号を解除します。さらにリクエストがある場合、ペリフェラルは次のトランザクションを開始できます。

9.3.3 チャンネル選択

各ストリームは、8 つのチャンネルリクエストの中から選択することができる DMA リクエストに関連付けられています。チャンネル選択は DMA_SxCR レジスタの CHSEL[2:0] ビットによって制御されます。

図 24. チャンネルの選択



ペリフェラルからの 8 つのリクエスト（TIM、ADC、SPI、I2C など）は各チャンネルに個々に接続されており、それらの接続は製品の実装によって異なります。

表 30 および 表 31 に DMA リクエストマッピングの例を示します。

表 30. DMA1 リクエストマッピング

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	SPI3_RX	I2C1_TX	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	-	SPI3_TX
チャンネル 1	I2C1_RX	I2C3_RX	TIM7_UP	I2CFMP1_RX	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
チャンネル 2	TIM4_CH1	I2CFMP1_TX	I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
チャンネル 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
チャンネル 4	-	USART3_RX	-	USART3_TX	-	USART2_RX	USART2_TX	I2CFMP1_TX
チャンネル 5	-	-	TIM3_UP TIM3_CH4	-	TIM3_CH1 TIM3_TRIG	TIM3_CH2	-	TIM3_CH3
チャンネル 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	I2C3_TX	TIM5_UP	USART2_RX
チャンネル 7	I2CFMP1_RX	TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	-	-	I2C2_TX

表 31. DMA2 リクエストマッピング

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	ADC1	-	TIM8_CH1 TIM8_CH2 TIM8_CH3	-	ADC1	-	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
チャンネル 1	-	-	-	-	-	-	-	-
チャンネル 2	-	-	SPI1_TX	SPI5_RX	SPI5_TX	-	-	-
チャンネル 3	SPI1_RX	DFSDM1_FLT1	SPI1_RX	SPI1_TX	DFSDM1_FLT1	SPI1_TX	DFSDM1_FLT0	QUADSPI
チャンネル 4	SPI4_RX	SPI4_TX	USART1_RX	SDIO	SPI4_RX	USART1_RX	SDIO	USART1_TX
チャンネル 5	-	USART6_RX	USART6_RX	SPI4_RX	SPI4_TX	SPI5_TX	USART6_TX	USART6_TX
チャンネル 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
チャンネル 7	DFSDM1_FLT0	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX	SPI5_TX	TIM8_CH4 TIM8_TRIG TIM8_COM

9.3.4 アービタ

アービタは、2 つの AHB マスタポート（メモリポートおよびペリフェラルポート）のそれぞれについて、その優先順位に基づいて 8 つの DMA ストリームリクエストを管理し、ペリフェラル/メモリアクセスシーケンスを起動します。

優先順位は 2 段階で管理されます。

- ソフトウェア：各ストリームの優先順位は、DMA_SxCR レジスタで設定できます。4 つのレベルがあります。
 - － 最優先
 - － 高優先
 - － 中優先
 - － 低優先
- ハードウェア：2 つのリクエストのソフトウェア優先順位レベルが同じ場合、小さな番号のストリームが大きな番号のストリームよりも優先されます。たとえば、ストリーム 2 はストリーム 4 よりも優先されます。



9.3.5 DMA ストリーム

8 つの DMA コントローラストリームのそれぞれに、転送元と転送先の間の単方向転送リンクを備えています。

各ストリームは次のことを行うよう設定することができます。

- レギュラータイプトランザクション：メモリからペリフェラル、ペリフェラルからメモリまたはメモリ間の転送
- ダブルバッファタイプトランザクション：メモリ用に 2 つのメモリポインタを使用したダブルバッファ転送（DMA がバッファの読み書き込みを行っている間、アプリケーションは他のバッファの読み/書きが可能）

転送するデータの量 (65535 以内) はプログラム可能で、ペリフェラル AHB ポートに接続された DMA 転送を要求するペリフェラルのデータ幅に関連しています。転送されるデータ数を格納しているレジスタは、トランザクションの終了ごとにデクリメントされます。

9.3.6 転送元、転送先、および転送モード

転送元および転送先は両方とも、アドレス 0x0000 0000 と 0xFFFF FFFF の間の全 4 GB メモリ領域内のペリフェラルおよびメモリを転送アドレスとして使用できます。

方向は DMA_SxCR レジスタの DIR[1:0] ビットを使用して設定し、メモリからペリフェラル、ペリフェラルからメモリ、およびメモリ間の転送の 3 つの選択肢があります。表 32 に対応する転送元と転送先アドレスを示します。

表 32. 転送元および転送先アドレス

DMA_SxCR レジスタの DIR [1:0] ビット	方向	ソースアドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR
01	メモリからペリフェラル	DMA_SxM0AR	DMA_SxPAR
10	メモリ間転送	DMA_SxPAR	DMA_SxM0AR
11	予約済み	-	-

DMA_SxCR レジスタの PSIZE または MSIZE ビットでプログラムされたデータ幅が、それぞれハーフワードまたはワードである場合、DMA_SxPAR または DMA_SxM0AR/M1AR レジスタに書き込まれたペリフェラルアドレスまたはメモリアドレスは、それぞれワードアドレス境界またはハーフワードアドレス境界で整列する必要があります。

“ペリフェラルからメモリ”モード

図 25 でこのモードを説明しています。

このモードが有効になっている場合（DMA_SxCR レジスタの EN ビットをセットすることにより）、ペリフェラルリクエストが発生するたびに、ストリームは転送元からの転送を開始し、FIFO を満たします。

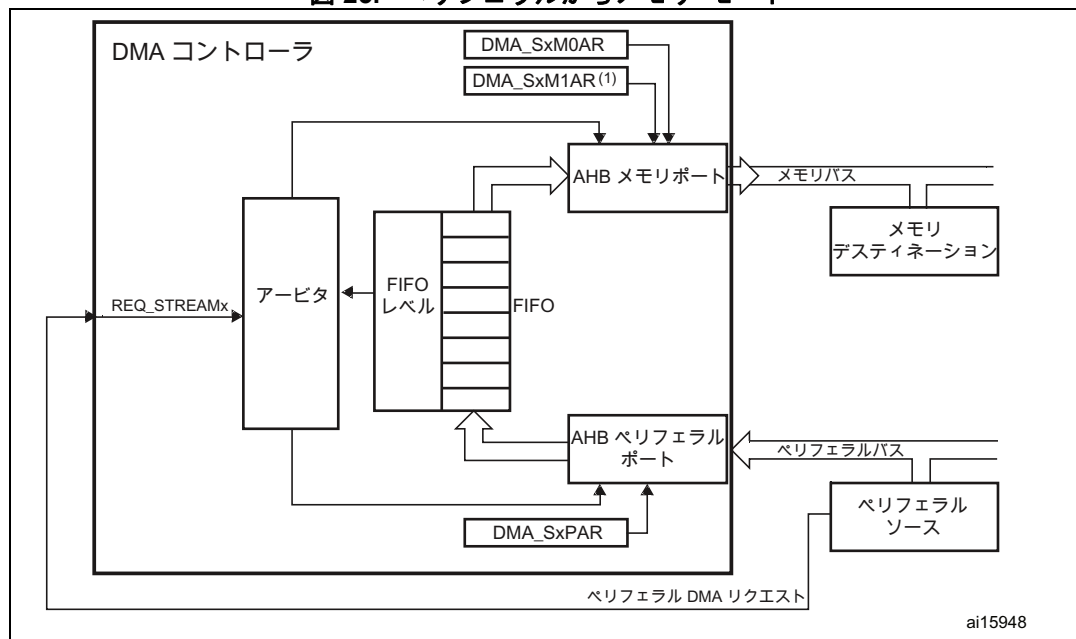
FIFO の閾値レベルに達すると、FIFO の内容が排出され、転送先に格納されます。

DMA_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか（ペリフェラルフローコントローラの場合）、またはソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ペリフェラルから FIFO への各データの転送後、対応するデータはすぐに排出されて転送先に格納されます。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 25. “ペリフェラルからメモリ”モード



1. ダブルバッファモード用。

“メモリからペリフェラル”モード

図 26 でこのモードを説明しています。

このモードが有効になっている場合 (DMA_SxCR レジスタの EN ビットをセットしている場合)、ストリームはすぐに転送元から転送を開始し、FIFO を完全に満たします。

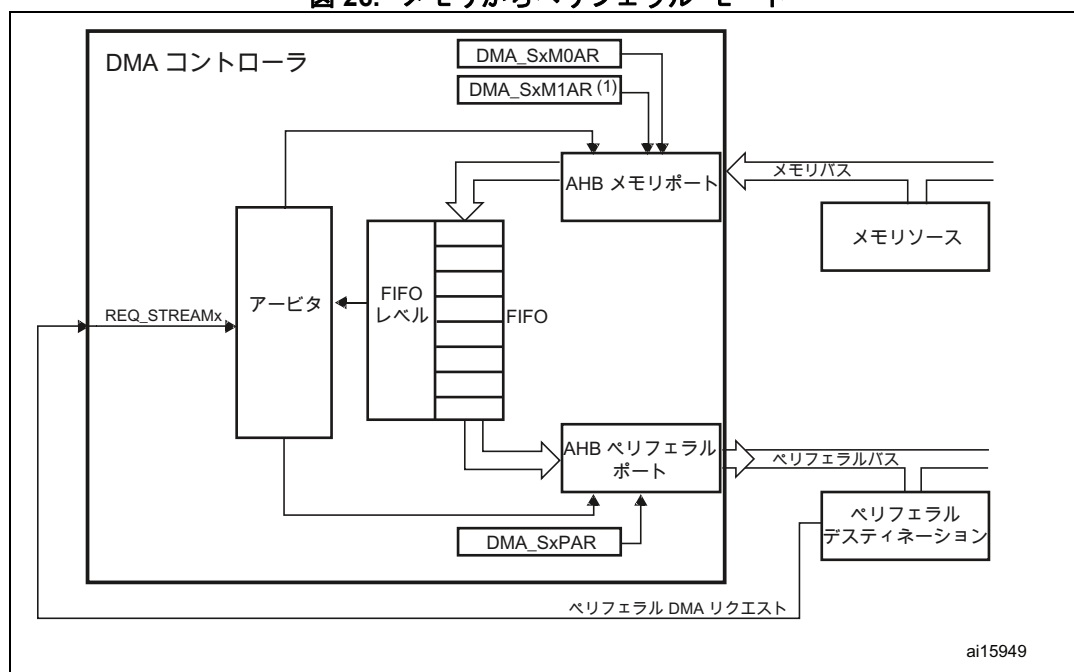
ペリフェラルリクエストが発生するたびに、FIFO のデータが転送されて転送先に格納されます。FIFO のレベルがあらかじめ設定された閾値レベルより低いとき、FIFO はメモリからのデータで完全に再ロードされます。

DMA_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか (ペリフェラルフローコントローラの場合)、またはソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ストリームが有効になると、DMA は内部 FIFO に転送するための最初のデータをプリロードします。ペリフェラルがデータ転送を要求すると、DMA はプリロードされたデータを設定された転送先へすぐに転送します。そしてまた転送すべき次のデータを空の内部 FIFO にリロードします。プリロードデータサイズは、DMA_SxCR レジスタの PSIZE ビットフィールドの値に対応します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 26. “メモリからペリフェラル”モード



1. ダブルバッファモード用。

メモリ間モード

DMA チャンネルは、ペリフェラルからのリクエストによってトリガされなくても機能します。これは [図 27](#) で説明しているメモリ間モードです。

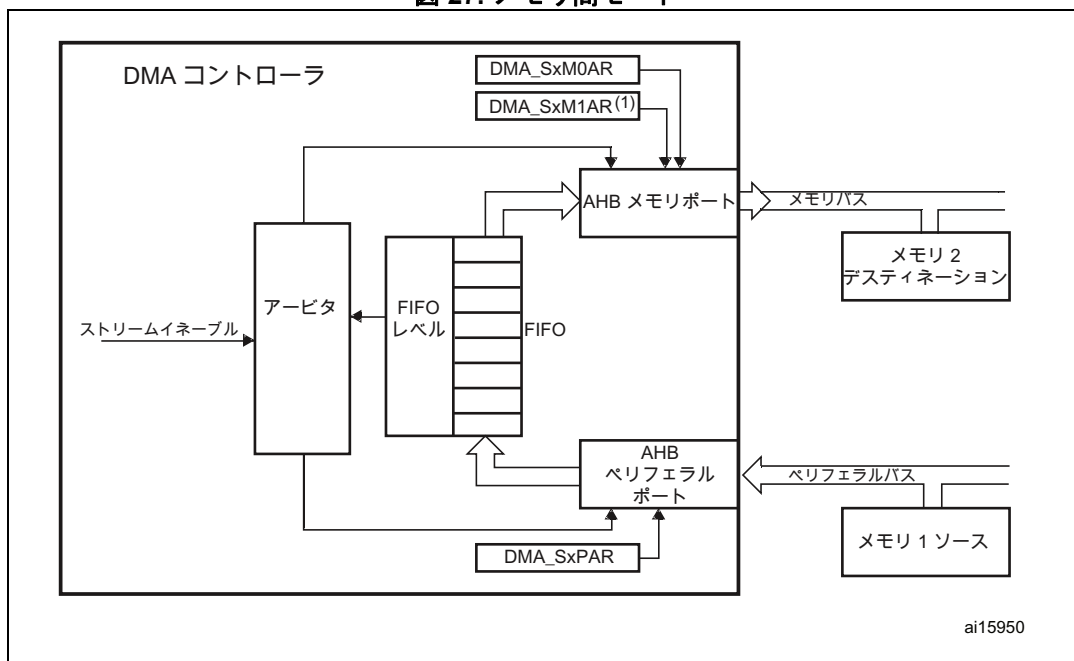
DMA_SxCR レジスタのイネーブルビット (EN) をセットすることによってストリームが有効になっている場合、ストリームはすぐに閾値レベルまで FIFO への書き込みを開始します。FIFO の閾値レベルに達すると、FIFO のデータが転送され、転送先に格納されます。

DMA_SxNDTR レジスタがゼロに達するかソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき転送を停止します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

注： **メモリ間モードを使用する場合、サーキュラモードおよびダイレクトモードは使用できません。**
DMA2 コントローラのみメモリ間転送を行うことができます。

図 27. メモリ間モード



1. ダブルバッファモード用。

9.3.7 ポインタのインクリメント

ペリフェラルやメモリのポインタは、DMA_SxCR レジスタの PINC および MINC ビットに応じて、各転送後に自動的にポストインクリメントまたは一定に保つことができます。

インクリメントモードの無効化は、転送元または転送先ペリフェラルデータを 1 つのレジスタを介してアクセスする場合に便利です。

インクリメントモードが有効な場合、次の転送アドレスは、DMA_SxCR レジスタの PSIZE または MSIZE ビットにプログラムされたデータ幅に応じて、前回のアドレスに 1 (バイトの場合)、2 (ハーフワードの場合)、または 4 (ワードの場合) をインクリメントしたアドレスになります。

パッキング動作を最適化するため、AHB ペリフェラルポート上で転送されるデータのサイズに関わらずペリフェラルアドレスのインクリメントオフセットサイズを固定することができます。DMA_SxCR レジスタの PINCOS ビットは、インクリメントオフセットサイズをペリフェラル AHB ポート上のデータサイズまたは 32 ビットアドレス (この場合アドレスは 4 つインクリメントされます) に調整するために使用されます。PINCOS ビットは AHB ペリフェラルポートにのみ影響します。

PINCOS ビットがセットされている場合、次の転送のアドレスは、PSIZE 値にかかわらず前のアドレスに 4 をインクリメント (自動的に 32 ビットアドレスに調整) したアドレスです。しかしながら、この動作による AHB メモリポートへの影響はありません。

9.3.8 サークュラモード

サーキュラモードを使用すると、サーキュラバッファや連続したデータフロー（ADC スキャンモードなど）を処理できます。この機能は、DMA_SxCR レジスタの CIRC ビットを使用して有効にできます。

サーキュラモードが有効になると、転送されるデータ項目数にはストリーム設定段階でプログラミングされた初期値が自動的に再ロードされ、DMA リクエストの処理が続行されます。

注： サークュラモードでは、メモリ用にバーストモードが設定されている場合、次のルールを守る必要があります。

$DMA_SxNDTR = ((Mburst \text{ ビート}) \times (Msize) / (Psize))$ 、ここで

- $(Mburst \text{ ビート}) = 4, 8 \text{ または } 16$ (DMA_SxCR レジスタの MBURST ビットによって異なる)
- $((Msize) / (Psize)) = 1, 2, 4, 1/2 \text{ または } 1/4$ (Msize と Psize は DMA_SxCR レジスタの MSIZE と PSIZE ビットを表す。これらはバイト依存)
- $DMA_SxNDTR = AHB \text{ ペリフェラルポート上で転送するデータ項目の数}$

例： Mburst ビート=8 (INCR8)、MSIZE=00 (バイト) および PSIZE=01 (ハーフワード)、この場合、DMA_SxNDTR は $(8 \times 1/2 = 4)$ の倍数でなければなりません。

この式を守らないと DMA 動作とデータの整合性は保証されません。

NDTR もペリフェラルバーストサイズの倍数にペリフェラルデータサイズを乗じた値でなければならない、さもないと不正な DMA 動作を引き起こす可能性があります。

9.3.9 ダブルバッファモード

このモードは、すべての DMA1 と DMA2 ストリームに対して使用可能です。

ダブルバッファモードは、DMA_SxCR レジスタの DBM ビットをセットすることで有効になります。

ダブルバッファストリームは通常の（シングルバッファ）ストリームとして動作しますが、2 つのメモリポインタを持っているところが異なります。ダブルバッファモードが有効になっていると、サーキュラモードが自動的に有効になり（DMA_SxCR の CIRC ビットは無視）、トランザクションが終わるごとにメモリポインタをスワップします。

このモードでは、DMA コントローラはトランザクションが終わるごとに、あるメモリターゲットから他へスワップします。これにより、DMA 転送によってある 1 つのメモリ領域がフルまたは使用中である間に、ソフトウェアは別のメモリ領域を処理することができます。ダブルバッファストリームは、[表 33：ダブルバッファモード \(DBM=1\) における転送元および転送先アドレスレジスタ](#)で説明するように両方向に動作することができます（メモリは転送元または転送先のいずれにも設定可能）。

注： ダブルバッファモードでは、ストリームが有効なとき、以下の条件を守ることにより、いつでも AHB メモリポートのベースアドレス (DMA_SxM0AR または DMA_SxM1AR) を更新することができます。

- DMA_SxCR レジスタで CT ビットが“0”のとき DMA_SxM1AR レジスタに書き込むことができます。CT=1 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。
- DMA_SxCR レジスタで CT ビットが“1”のとき DMA_SxM0AR レジスタに書き込むことができます。CT=0 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。

エラー条件を回避するためには、TCIF フラグがアサートされたらできるだけ早くベースアドレスを変更することを推奨します。なぜなら、この時点で、上記 2 つの内いずれかの条件にしたがって DMA_SxCR レジスタの CT の値に応じてターゲットメモリがメモリ 0 から 1（または 1 から 0）に変更されているためです。

他のすべてのモード（ダブルバッファモードを除く）に対し、ストリームが有効になるとすぐに、メモリアドレスレジスタは書き込み保護されます。

表 33. ダブルバッファモード (DBM=1) における転送元および転送先アドレスレジスタ

DMA_SxCR レジスタの DIR [1:0] ビット	方向	ソースアドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR/DMA_SxM1AR
01	メモリからペリフェラル	DMA_SxM0AR/DMA_SxM1AR	DMA_SxPAR
10	設定禁止 ⁽¹⁾		
11	予約済み	-	-

1. ダブルバッファモードが有効になっている場合、サーキュラモードが自動的に有効になります。メモリ間モードはサーキュラモードと互換性がないため、ダブルバッファモードが有効になっているときは、メモリ間モードを設定することはできません。

9.3.10 プログラム可能なデータ幅、パッキング／アンパッキング、エンディアン形式

転送されるデータ項目の数は、ストリームを有効にする前に DMA_SxNDTR (転送データ項目数ビット、NDT) にプログラムする必要があります (フローコントローラがペリフェラルで、DMA_SxCR の PFCTRL ビットがセットされている場合を除く)。

内部 FIFO を使用する場合、転送元と転送先データのデータ幅は DMA_SxCR レジスタの PSIZE と MSIZE ビットを介してプログラム可能です (8、16、または 32 ビットが可能)。

PSIZE と MSIZE が等しくない場合：

- DMA_SxNDTR レジスタに設定されている転送データ項目数のデータ幅は、ペリフェラルバスの幅 (DMA_SxCR レジスタの PSIZE ビットで設定されている) と同じです。たとえば、ペリフェラルからメモリ、メモリからペリフェラル、またはメモリ間転送のケースで PSIZE[1:0] ビットがハーフワード用に設定されている場合、転送されるバイト数は 2×NDT です。
- DMA コントローラは、転送元と転送先の両方でリトルエンディアンアドレッシングのみ取り扱うことができます。これについては、[表 34：パッキング／アンパッキングおよびエンディアンの動作 \(ビット PINC=MINC=1\)](#) で説明されています。

このパッキング／アンパッキング手順には、データが完全にパック／アンパックされる前に操作が中断された場合にデータが破損する危険性があります。したがって、データの一貫性を確保するため、ストリームはバースト転送を生成するように設定することができます。この場合、バーストに属する各転送グループは分割できません ([セクション 9.3.11：シングル転送とバースト転送](#)を参照)。

ダイレクトモード (DMA_SxFCR レジスタで DMDIS=0 のとき) では、データのパッキング／アンパッキングはできません。この場合には、転送元と転送先の転送データ幅が異なってはなりません。両方が等しく、DMA_SxCR の PSIZE ビットで定義します (MSIZE ビットは無視されます)。

表 34. パッキング／アンパッキングおよびエンディアンの動作 (ビット PINC=MINC=1)

AHB メモリポート幅	AHB ペリフェラルポート幅	転送するデータ項目の数 (NDT)	メモリ転送番号	メモリポートアドレス／バイトレーン	ペリフェラル転送番号	ペリフェラルポートアドレス／バイトレーン	
						PINCOS = 1	PINCOS = 0
8	8	4	1	0x0 / B0[7:0]	1	0x0 / B0[7:0]	0x0 / B0[7:0]
			2	0x1 / B1[7:0]	2	0x4 / B1[7:0]	0x1 / B1[7:0]
			3	0x2 / B2[7:0]	3	0x8 / B2[7:0]	0x2 / B2[7:0]
			4	0x3 / B3[7:0]	4	0xC / B3[7:0]	0x3 / B3[7:0]
8	16	2	1	0x0 / B0[7:0]	1	0x0 / B1[B0[15:0]	0x0 / B1[B0[15:0]
			2	0x1 / B1[7:0]	2	0x4 / B3[B2[15:0]	0x2 / B3[B2[15:0]
			3	0x2 / B2[7:0]			
			4	0x3 / B3[7:0]			

表 34. パッキング／アンパッキングおよびエンディアン動作 (ビット PINC=MINC=1)

AHB メモリ ポート幅	AHB ペリ フェラル ポート 幅	転送する データ 項目の数 (NDT)	メモリ 転送番号	メモリポート アドレス/ バイトレーン	ペリフェラル 転送番号	ペリフェラルポートアドレス/ バイトレーン	
						PINCOS = 1	PINCOS = 0
8	32	1	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1	0x0 / B3[B2:B1]B0[31:0]	0x0 / B3[B2:B1]B0[31:0]
16	8	4	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
16	16	2	1 2	0x0 / B1[B0:15:0] 0x2 / B1[B0:15:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
16	32	1	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1	0x0 / B3[B2:B1]B0[31:0]	0x0 / B3[B2:B1]B0[31:0]
32	8	4	1	0x0 / B3[B2:B1]B0[31:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
32	16	2	1	0x0 / B3[B2:B1]B0[31:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
32	32	1	1	0x0 / B3[B2:B1]B0 [31:0]	1	0x0 / B3[B2:B1]B0 [31:0]	0x0 / B3[B2:B1]B0 [31:0]

注： ペリフェラルポートは、転送元または転送先として設定可能です (メモリ間転送の場合はメモリ転送元としての設定も可能です)。

PSIZE、MSIZE、および NDT[15:0] は、最後の転送が不完全にならないよう必ず設定する必要があります。これはペリフェラルポートのデータ幅 (PSIZE ビット) がメモリポートのデータ幅 (MSIZE ビット) より小さい場合に発生する可能性があります。この制約の一覧を表 35 に示します。

表 35. PSIZE と MSIZE に対する NDT に関する制限事項

PSIZE[1:0] of DMA_SxCR	MSIZE[1:0] of DMA_SxCR	NDT[15:0] of DMA_SxNDTR
00 (8 ビット)	01 (16 ビット)	2 の倍数でなければならない
00 (8 ビット)	10 (32 ビット)	4 の倍数でなければならない
01 (16 ビット)	10 (32 ビット)	2 の倍数でなければならない

9.3.11 シングル転送とバースト転送

DMA コントローラは、シングル転送を生成することもできれば、4、8、または 16 ビートのインクリメンタルバースト転送を生成することもできます。

バーストのサイズは、DMA_SxCR レジスタの MBURST[1:0] と PBURST[1:0] ビットを使用して 2 つの AHB ポートに対して別々にソフトウェアで設定されます。

バーストサイズは、バースト内のビート数を示します。転送するバイト数ではありません。

データの一貫性を確保するため、バーストを形成する転送の各グループは分割できません。AHB 転送はロックされ、AHB バスマトリックスのアービタは、バースト転送シーケンス中 DMA マスタからバスの使用を奪取しません。

シングルまたはバーストの設定によって各 DMA リクエストによって AHB ペリフェラルポート上で開始される転送の数は異なります。

- AHB ペリフェラルポートがシングル転送用に設定されている場合、各 DMA リクエストによって、DMA_SxCR レジスタの PSIZE[1:0] ビットに応じた、バイト、ハーフワード、またはワードのデータ転送が生成されます。
- AHB ペリフェラルポートがバースト転送用に設定されている場合、各 DMA リクエストによって、DMA_SxCR レジスタの PBURST[1:0] ビットと PSIZE[1:0] ビットに応じた、4、8、または 16 ビートのバイト、ハーフワード、またはワードの転送が生成されます。

MBURST と MSIZE ビットに関しても、AHB メモリポートに対して上記と同じことを考慮する必要があります。

ダイレクトモードでは、ストリームはシングル転送のみ生成することができ、MBURST[1:0] と PBURST[1:0] ビットはハードウェアで決まっています。

アドレスポインタ (DMA_SxPAR または DMA_SxM0AR レジスタ) は、バーストブロック内のすべての転送が転送サイズに等しいアドレス境界で整列するように選択する必要があります。

バースト設定は、AHB プロトコルを守るよう選択する必要があります。ここで、1 個のスレーブに割り当てることができる最小アドレス空間が 1 KB であるため、バーストが 1 KB アドレス境界を超えることはできません。つまり、バーストブロック転送が 1 KB アドレス境界を超えてはならず、超えた場合には AHB エラーが生成される可能性があります。このエラーは DMA レジスタで報告されません。

9.3.12 FIFO

FIFO の構造

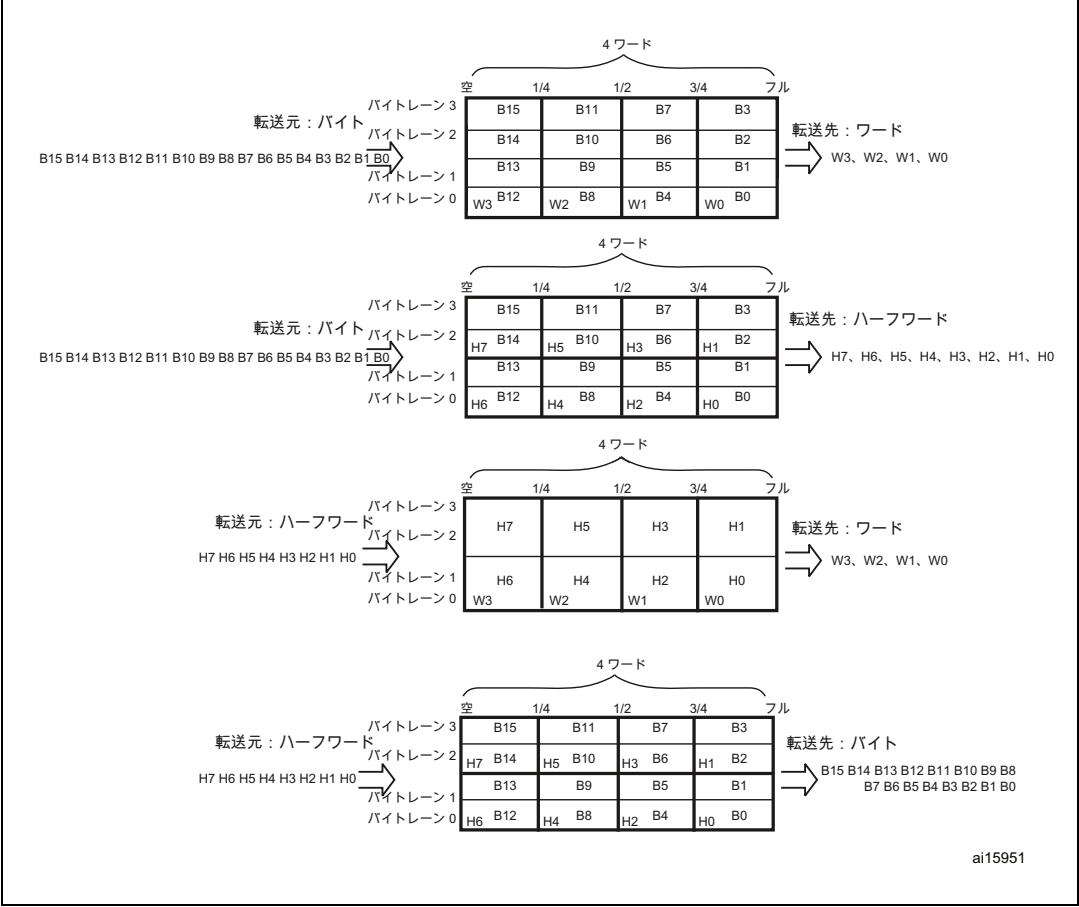
FIFO は転送元からのデータを転送先へ送信する前に一時的に格納するために使用します。

各ストリームには独立した 4 ワードの FIFO があり、閾値レベルはソフトウェアで 1/4、1/2、3/4、またはフルに設定できます。

FIFO 閾値レベルを使用できるようにするには、DMA_SxFCR レジスタの DMDIS ビットをセットしてダイレクトモードを無効にする必要があります。

FIFO の構造は、転送元および転送先のデータ幅によって異なり、[図 28 : FIFO の構造](#)で説明しています。

図 28. FIFO の構造



FIFO 閾値とバーストの設定

FIFO の閾値 (DMA_SxFCR レジスタの FTH[1:0] ビット) とメモリバーストのサイズ (DMA_SxCR レジスタの MBURST[1:0]) を選択する場合は注意が必要です。FIFO 閾値が指す内容はメモリバースト転送の整数倍に一致する必要があります。一致しない場合、ストリームが有効なとき FIFO エラー (DMA_HISR または DMA_LISR レジスタの FEIFx フラグ) が発生し、ストリームが自動的に無効になります。可能な設定と禁止されている設定の説明は表 36 を参照してください。禁止されている設定は、表内でグレイでハイライトされています。

表 36. FIFO 閾値設定

MSIZE	FIFO レベル	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
バイト	1/4	4 ビートの 1 バースト	設定禁止	設定禁止
	1/2	4 ビートの 2 バースト	8 ビートの 1 バースト	
	3/4	4 ビートの 3 バースト	設定禁止	
	フル	4 ビートの 4 バースト	8 ビートの 2 バースト	

表 36. FIFO 閾値設定 (続き)

MSIZE	FIFO レベル	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16	
ハーフワード	1/4	設定禁止	設定禁止	設定禁止	
	1/2	4 ビートの 1 パースト			
	3/4	設定禁止			
	フル	4 ビートの 2 パースト	8 ビートの 1 パースト		
ワード	1/4	設定禁止	設定禁止		
	1/2				
	3/4				
	フル	4 ビートの 1 パースト			

いかなる場合でも、パーストサイズにデータサイズを乗じた値が FIFO サイズを超えることはできません (可能なデータサイズは1 (バイト)、2 (ハーフワード) または 4 (ワード))。

次のいずれかの条件が発生した場合、DMA 転送の終了時に、パースト転送が不完全となる可能性があります。

- AHB ペリフェラルポート設定において、データ項目の合計数 (DMA_SxNDTR レジスタに設定) が、パーストサイズにデータサイズを乗じた値の倍数ではない
- AHB メモリポート設定において、メモリに転送する FIFO 内の残りのデータ項目数が、パーストサイズにデータサイズを乗じた値の倍数ではない

このような場合、DMA ストリーム設定時にパーストトランザクションが要求されていても、転送すべき残りのデータは、DMA によってシングルモードで管理されます。

注： ペリフェラル AHB ポートでパースト転送が要求されており、FIFO が使用されている場合 (DMA_SxCR レジスタの DMDIS=1)、DMA ストリーム方向に応じて、恒久的なアンダーランまたはオーバーラン状態を回避するため、次のルールを守ることが必要です。

$(PBURST \times PSIZE) = FIFO_SIZE$ (4 ワード) の場合、PSIZE = 1、2、または 4 であり、かつ、PBURST = 4、8、または 16 の時、FIFO_Threshold = 3/4 は禁止。

このルールによって、ペリフェラルからの要求を一度に処理するのに十分な FIFO スペースを確実に確保できます。

FIFO フラッシュ

FIFO は DMA_SxCR レジスタの EN ビットをリセットすることによりストリームが無効化された場合、およびストリームが“ペリフェラルからメモリ”転送またはメモリ間転送を管理するよう設定された場合、フラッシュすることができます。ストリームを無効化したときに FIFO にデータが残っている場合、DMA コントローラは引き続き残りのデータを転送先へ転送します (ストリームが事実上無効化されているとしても)。このフラッシュが完了すると、DMA_LISR または DMA_HISR レジスタの転送完了ステータスビット (TCIFx) がセットされます。

この場合、残りのデータカウンタ DMA_SxNDTR は値を保持し、現在いくつかのデータ項目が転送先メモリに格納できるかを示します。

FIFO フラッシュ動作中に、FIFO 内のメモリに転送する残りのデータ項目数 (バイト単位) がメモリデータ幅より少ない場合 (たとえば、MSIZE がワードに設定されているとき FIFO 内に 2 バイト)、データは DMA_SxCR レジスタの MSIZE ビットに設定されたデータ幅で送信されます。つまり、メモリには不要な値が書き込まれることになります。ソフトウェアで DMA_SxNDTR レジスタを読み出すことにより、正しいデータが格納されたメモリ領域 (開始アドレスと終了アドレス) を判断することができます。



FIFO 内の残りのデータ項目数がバーストサイズよりも小さい場合 (AHB メモリポート上のバーストを管理するストリームを設定するため DMA_SxCR レジスタの MBURST ビットがセットされている場合)、FIFO フラッシュを完了するためシングルトランザクションが生成されます。

ダイレクトモード

デフォルトでは、FIFO はダイレクトモードで動作し (DMA_SxFCR の DMDIS ビットがリセットされています)、FIFO 閾値レベルは使用しません。このモードは、システムが各 DMA リクエスト後にメモリとの即時シングル転送を必要とする場合に便利です。

DMA がダイレクトモード (FIFOが無効) に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるように、DMA はデータをメモリから内部 FIFO へ 1 つプリロードします。

FIFO の飽和を避けるためには、対応するストリームを高い優先度で設定することを推奨します。

このモードは、次のような転送に制限されています。

- 転送元および転送先の転送幅が等しく、両方とも DMA_SxCR の PSIZE[1:0] ビットによって定義されている (MSIZE[1:0] ビットは無視される)
- バースト転送は不可 (DMA_SxCR の PBURST[1:0] と MBURST[1:0] ビットは無視される)

メモリ間転送を実行する際はダイレクトモードを使用することはできません。

9.3.13 DMA 転送の完了

DMA_LISR または DMA_HISR ステータスレジスタの TCIFx ビットを設定することにより、さまざまなイベントによって転送を終了させることができます。

- DMA フローコントローラモード：
 - “メモリからペリフェラル”モードで DMA_SxNDTR カウンタがゼロに達した
 - (DMA_SxCR レジスタの EN ビットをクリアすることにより) ストリームが転送終了する前に無効化され、(転送がペリフェラルからメモリまたはメモリ間の場合) 残りのすべてのデータが FIFO からメモリにフラッシュされた
- ペリフェラルフローコントローラモード：
 - 最後の外部バーストまたはシングルリクエストがペリフェラルから生成され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき) 残りのデータが FIFO からメモリへ転送された
 - ストリームがソフトウェアによって無効化され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき)、残りのデータが FIFO からメモリへ転送された

注： **転送完了は、“ペリフェラルからメモリ”モードの場合のみ、メモリに転送される FIFO 内の残りのデータによって異なります。この条件は、“メモリからペリフェラル”モードには適用されません。**

ストリームがノンサーキュラモードで設定されている場合、転送終了後 (転送すべきデータの数がゼロに達したとき)、DMA は停止し (DMA_SxCR レジスタの EN ビットがハードウェアでクリアされます)、ソフトウェアによってストリームを再プログラムして再有効化 (DMA_SxCR レジスタの EN ビットを設定) しない限り、DMA リクエストは処理されません。

9.3.14 DMA 転送の中断

DMA 転送は、いつでも中断して再開することもできれば、DMA 転送が終了する前に完全に無効化することもできます。

これには、2 つのケースがあります。

- ストリームは、転送を無効にし、停止した場所以降再開しません。ストリームを無効化するために DMA_SxCR レジスタの EN ビットをクリアする以外、特に行うことはありません。ストリームを無効にするのに時間を要する可能性があります（まず、進行中の転送を完了します）。転送の終了を示すために転送完了割り込みフラグ (DMA_LISR または DMA_HISR レジスタの TCIF) がセットされます。その時点で、DMA_SxCR の EN ビットの値が“0”となって、ストリームが中断されます。ソフトウェアによって、ストリームが中断される前に転送されたデータ項目の数を判断できるように、DMA_SxNDTR レジスタに、ストリームが停止した時点で残っているデータ項目の数を保持しています。
- ストリームは、DMA_SxNDTR レジスタ内の転送されるべき残りのデータ項目数が 0 に到達する前に転送を中断します。その目的は、ストリームを再度有効にすることで後から転送を再開することにあります。転送が停止した時点から再開するにあたり、ソフトウェアが既に収集したデータ項目数を知るため DMA_SxCR レジスタの EN ビットに書き込んで（その後それが“0”であることを確認します）ストリームを無効にした後で DMA_SxNDTR レジスタを読み出す必要があります。その後、
 - － アドレスポインタを調整するために、ペリフェラルやメモリのアドレスを更新する必要があります。
 - － SxNDTR レジスタは、転送すべきデータ項目の残数（ストリームを無効にしたとき読み出される値）で更新する必要があります。
 - － ストリームは、その後再度有効化して転送が停止したところから再開することができます。

注： *ストリームの中断による、転送の終了を示すため、転送完了割り込みフラグ (DMA_LISR または DMA_HISR の TCIF) がセットされることに注意してください。*

9.3.15 フローコントローラ

転送するデータの数を制御するものはフローコントローラとして知られています。このフローコントローラは、DMA_SxCR レジスタの PFCTRL ビットを使用して、各ストリームごとに個別に設定されます。

フローコントローラは次のように使用することができます。

- DMA コントローラ：この場合、DMA ストリームを有効にする前に、転送するデータ項目数を DMA_SxNDTR レジスタにソフトウェアでプログラムします。
- ペリフェラル転送元またはペリフェラル転送先：この場合、転送されるデータ項目数は不明です。最後のデータが転送されるとき、ペリフェラルがハードウェアによって DMA コントローラに知らせます。この機能は、転送の終了を通知できるペリフェラルでのみサポートされています。すなわち、

特定のストリームでペリフェラルフローコントローラを使用した場合、DMA_SxNDTR に書き込まれる値は DMA 転送に影響しません。実際に、いかなる値が書き込まれても、以下のスキームを守るためストリームが有効になるとすぐに、ハードウェアによって強制的に 0xFFFF になります。

- 想定されるストリームの中断：最終データを示すハードウェア信号（シングルまたはバースト）がペリフェラルによって送信される前に、DMA_SxCR レジスタの EN ビットをソフトウェアによって 0 にリセットし、ストリームを停止します。このようなケースでは、“ペリフェラルからメモリ”DMA 転送の場合、ストリームが停止し、FIFO フラッシュが作動します。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA の完了を示します。DMA 転送中に転送されたデータ項目数を知るには、DMA_SxNDTR レジスタを読み出し次の式を適用します。
 - － 転送されたデータ数 = 0xFFFF – DMA_SxNDTR

- 最終データを示すハードウェア信号の受信による通常ストリームの中断: ペリフェラルが最後の転送 (シングルまたはバースト) を要求しこの転送が完了したときストリームは自動的に中断されます。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA 転送の完了を示します。転送されたデータ項目数を知るには、DMA_SxNDTR レジスタを読み出し上記と同じ式を適用します。
- DMA_SxNDTR レジスタが 0 に到達: 対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、強制的な DMA 転送の完了を示します。最終データを示すハードウェア信号 (シングルまたはバースト) がまだアサートされていない場合でも、ストリームは自動的に停止します。既に転送されたデータは失われません。つまり、ペリフェラルフローコントローラモードの場合でも、1 つの DMA のトランザクションにより最大 65535 のデータ項目が管理できることとなります。

注: メモリ間モードに設定されているとき、DMA は常にフローコントローラとなり PFCTRL ビットはハードウェアによって強制的に 0 になります。

 ペリフェラルフロー制御モードでは、サーキュラモードは禁止されています。

9.3.16 実現可能な DMA 設定の概要

表 37 に、実現可能なさまざまな DMA 設定をまとめています。禁止されている設定は、表内でグレイでハイライトされています。

表 37. 実現可能な DMA 設定

DMA 転送モード	転送元	転送先	フロー コントローラ	サーキュラ モード	転送タイプ	ダイレクト モード	ダブルバッファ モード
ペリフェラルから メモリ	AHB ペリフェラル ポート	AHB メモリポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリから ペリフェラル	AHB メモリポート	AHB ペリフェラル ポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリ間転送	AHB ペリフェラル ポート	AHB メモリポート	DMA のみ	設定禁止	シングル	設定禁止	設定禁止
					バースト		



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

9.3.17 ストリーム設定手順

DMA ストリーム x の設定は、次の手順で行います (x はストリーム番号です)。

1. ストリームが有効になっている場合、DMA_SxCR レジスタの EN ビットをリセットすることによって無効にし、その後、進行中のストリーム動作がないことを確認するため、このビットを読み出します。このビットに 0 を書き込んでもすぐに効果はありません。それは現在行われているすべての転送が完了した後、実際に 0 が書き込まれるからです。EN ビットが 0 として読み出される場合、ストリームが設定可能な状態であることを意味します。したがって、ストリームの設定を開始する前に EN ビットがクリアされるのを待つ必要があります。以前のデータブロックの DMA 転送からステータスレジスタ (DMA_LISR と DMA_HISR) に設定されたすべてのストリーム専用ビットは、ストリームを再度有効にする前にクリアする必要があります。
2. DMA_SxPAR レジスタにペリフェラルポートレジスタアドレスを設定します。ペリフェラルイベントの後、データはこのアドレスとペリフェラルポートの間で転送されます。
3. DMA_SxMA0R レジスタ (およびダブルバッファモードの場合は DMA_SxMA1R レジスタ) にメモリアドレスを設定します。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
4. 転送されるデータ項目の合計数を DMA_SxNDTR レジスタで設定します。各ペリフェラルイベントまたはバーストの各ビートの後、この値はデクリメントされます。
5. DMA_SxCR レジスタの CHSEL[2:0] を使用して DMA チャンネル (リクエスト) を選択します。
6. ペリフェラルをフローコントローラとして使用することを意図し、かつこの機能をサポートしている場合、DMA_SxCR レジスタの PFCTRL ビットをセットします。
7. DMA_SxCR レジスタの PL[1:0] ビットを使用して、ストリームの優先順位を設定します。
8. FIFO の使用方法を設定します (有効または無効にする、送信と受信における閾値)。
9. DMA_SxCR レジスタで以下の項目を設定します。データ転送方向、ペリフェラルとメモリのインクリメントモード/固定モード、シングルトランザクションまたはバーストランザクション、ペリフェラルとメモリのデータ幅、サーキュラモード、ダブルバッファモード、ハーフ転送やフル転送後の割り込み、およびエラー。
10. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

ストリームが有効になると、そのストリームに接続されているペリフェラルからの DMA リクエストを処理できます。

AHB 転送先ポート上でデータの半分が転送されると、1/2 転送フラグ (HTIF) がセットされ、1/2 転送割り込みイネーブルビット (HTIE) がセットされている場合には、割り込みが生成されます。転送が終了すると、転送完了フラグ (TCIF) がセットされ、転送完了割り込みイネーブルビット (TCIE) がセットされている場合には、割り込みが生成されます。

警告： DMA ストリームリクエストに接続されているペリフェラルを切断するには、まず、ペリフェラルが接続されている DMA ストリームを切断し、次いで EN ビット=0 になるのを待つ必要があります。その後はじめてペリフェラルを安全に無効にすることができます。

9.3.18 エラー管理

DMA コントローラによって、次のエラーを検出することができます。

- **転送エラー**：転送エラー割り込みフラグ (TEIFx) は次の場合にセットされます。
 - DMA 読み出しまたは書き込みアクセス時にバスエラーが発生した場合
 - ダブルバッファモードでは、メモリアドレスレジスタの書き込みアクセスはソフトウェアによって要求されますが、ストリームが有効になり、メモリアドレスレジスタへの書き込みによって影響を受けるのは現在のターゲットメモリです ([セクション 9.3.9: ダブルバッファモード](#)を参照)。
- **FIFO エラー**：FIFO エラー割り込みフラグ (FEIFx) は次の場合にセットされます。
 - FIFO アンダーラン状態が検出された場合
 - FIFO オーバーラン状態が検出された場合 (リクエストと転送は DMA が内部で管理するため、このエラーはメモリ間モードでは検出されない)
 - FIFO 閾値レベルがメモリバーストのサイズに適合していないときにストリームが有効化された場合 ([表 36: FIFO 閾値設定](#)を参照)
- **ダイレクトモードエラー**：ダイレクトモードで動作していて DMA_SxCR レジスタの MINC ビットがクリアされたときのみ、“ペリフェラルからメモリ”モードにおいてダイレクトモードエラー割り込みフラグ (DMEIFx) がセットされることがあります。このフラグは、メモリバスの使用権が与えられていなかったことにより、前回のデータがまだ完全にメモリに転送されていない状態で DMA リクエストが発生したときにセットされます。このケースでは、このフラグは 2 つのデータ項目が同じ転送先アドレスに連続して転送されたことを示し、転送先がこの状況を管理できない場合に、このことが問題となる可能性があります。

ダイレクトモードでは、FIFO エラーフラグが以下の条件でセットされることもあります。

- “ペリフェラルからメモリ”モードでは、複数のペリフェラルリクエストのためにメモリバスの使用が許可されない場合、FIFO が飽和 (オーバーラン) することがあります。
- “メモリからペリフェラル”モードでは、ペリフェラルリクエストが発生する前に、メモリバスの使用が許可されていない場合、アンダーラン状態が発生する可能性があります。

バーストサイズと FIFO 閾値レベルが適合しないことによって TEIFx または FEIFx フラグがセットされた場合、欠陥のあるストリームは対応するストリーム設定レジスタ (DMA_SxCR) の EN ビットがハードウェアによってクリアされることにより、自動的に無効化されます。

オーバーランまたはアンダーラン状態によって DMEIFx または FEIFx フラグがセットされる場合、欠陥のあるストリームは自動的に無効化されず、DMA_SxCR レジスタの EN ビットをリセットすることによってストリームを無効化するかどうかはソフトウェアに任されています。なぜならば、この種のエラーが発生してもデータの損失がないためです。

DMA_LISR または DMA_HISR レジスタにあるストリームのエラー割り込みフラグ (TEIF、FEIF、DMEIF) がセットされると、DMA_SxCR または DMA_SxFCR レジスタの対応する割り込みイネーブルビット (TEIE、FEIE、DMIE) がセットされていれば、割り込みが発生します。

注： **FIFO オーバーランまたはアンダーラン状態が発生した場合、オーバーランまたはアンダーラン状態がクリアされるまではペリフェラルリクエストがストリームによって承認されないため、データが失われることはありません。この承認に時間がかかりすぎた場合、ペリフェラル自体がその内部バッファのオーバーランまたはアンダーラン状態を検出し、データが失われる可能性があります。**

9.4 DMA 割り込み

各 DMA ストリームに対し、次のイベントにて割り込みを生成することができます。

- 1/2 転送終了
- 転送完了
- 転送エラー
- FIFO エラー（オーバーラン、アンダーラン、または FIFO レベルのエラー）
- ダイレクトモードエラー

柔軟性を持たせるため、表 38 に示すよう個別の割り込みイネーブル制御ビットが使用できます。

表 38. DMA 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
1/2 転送	HTIF	HTIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE
FIFO オーバーラン／アンダーラン	FEIF	FEIE
ダイレクトモードエラー	DMEIF	DMEIE

注：イネーブル制御ビットを '1' に設定する前に、対応するイベントフラグをクリアする必要があります。さもないと直ちに割り込みが生成されてしまいます。

9.5 DMA レジスタ

DMA レジスタには、ワード（32 ビット）単位でアクセスすることができます。

9.5.1 DMA ロー割り込みステータスレジスタ (DMA_LISR)

アドレスオフセット：0x00

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Res.	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **TCIFx**：ストリーム x 転送完了割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0：ストリーム x に転送完了イベントは発生していません。

1：ストリーム x に転送完了イベントが発生しました。

- ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割り込みフラグ (x=3..0)
- このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に 1/2 転送イベントは発生していません。
- 1 : ストリーム x に 1/2 転送イベントが発生しました。
- ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割り込みフラグ (x=3..0)
- このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に転送エラーは発生していません。
- 1 : ストリーム x に転送エラーが発生しました。
- ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割り込みフラグ (x=3..0)
- このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x にダイレクトモードエラーは発生していません。
- 1 : ストリーム x にダイレクトモードエラーが発生しました。
- ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。
- ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割り込みフラグ (x=3..0)
- このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に FIFO エラーイベントは発生していません。
- 1 : ストリーム x に FIFO エラーイベントが発生しました。

9.5.2 DMA ハイ割り込みステータスレジスタ (DMA_HISR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Res.	FEIF4
				r	r	r	r		r	r	r	r	r		r

- ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。
- ビット 27、21、11、5 **TCIFx** : ストリーム x 転送完了割り込みフラグ (x=7..4)
- このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に転送完了イベントは発生していません。
- 1 : ストリーム x に転送完了イベントが発生しました。
- ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割り込みフラグ (x=7..4)
- このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に 1/2 転送イベントは発生していません。
- 1 : ストリーム x に 1/2 転送イベントが発生しました。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

- ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割り込みフラグ (x=7..4)
- このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に転送エラーは発生していません。
- 1 : ストリーム x に転送エラーが発生しました。
- ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割り込みフラグ (x=7..4)
- このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x にダイレクトモードエラーは発生していません。
- 1 : ストリーム x にダイレクトモードエラーが発生しました。
- ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。
- ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割り込みフラグ (x=7..4)
- このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。
- 0 : ストリーム x に FIFO エラーイベントは発生していません。
- 1 : ストリーム x に FIFO エラーイベントが発生しました。

9.5.3 DMA ロー割り込みフラグクリアレジスタ (DMA_LIFCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Res.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Res.	CFEIF0
				w	w	w	w		w	w	w	w	w		w

- ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。
- ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割り込みフラグ (x=3..0)
- このビットに 1 を書き込むと DMA_LISR レジスタの対応する TCIFx フラグをクリアします。
- ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割り込みフラグ (x=3..0)
- このビットに 1 を書き込むと DMA_LISR レジスタの対応する HTIFx フラグをクリアします。
- ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割り込みフラグ (x=3..0)
- このビットに 1 を書き込むと DMA_LISR レジスタの対応する TEIFx フラグをクリアします。
- ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割り込みフラグ (x=3..0)
- このビットに 1 を書き込むと DMA_LISR レジスタの対応する DMEIFx フラグをクリアします。
- ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。
- ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割り込みフラグ (x=3..0)
- このビットに 1 を書き込むと DMA_LISR レジスタの対応する CFEIFx フラグをクリアします。

9.5.4 DMA ハイ割り込みフラグクリアレジスタ (DMA_HIFCR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Res.	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res.	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Res.	CFEIF4
				w	w	w	w		w	w	w	w	w		w

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する TCIFx フラグをクリアします。

ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する HTIFx フラグをクリアします。

ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する TEIFx フラグをクリアします。

ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する DMEIFx フラグをクリアします。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する CFEIFx フラグをクリアします。

9.5.5 DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0~7)

このレジスタは当該ストリームの設定に使用されます。

アドレスオフセット : $0x10 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Res.	CT	DBM	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:25 **CHSEL[2:0]** : チャンネル選択

これらのビットは、ソフトウェアによってセット／クリアされます。

000 : チャンネル 0 が選択されます。

001 : チャンネル 1 が選択されます。

010 : チャンネル 2 が選択されます。

011 : チャンネル 3 が選択されます。

100 : チャンネル 4 が選択されます。

101 : チャンネル 5 が選択されます。

110 : チャンネル 6 が選択されます。

111 : チャンネル 7 が選択されます。

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ビット 24:23 **MBURST** : メモリバースト転送設定

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : シングル転送

01 : INCR4 (4 ビートのインクリメンタルバースト)

10 : INCR8 (8 ビートのインクリメンタルバースト)

11 : INCR16 (16 ビートのインクリメンタルバースト)

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ダイレクトモードでは、これらのビットは EN ビット=“1”になり次第ハードウェアによって強制的に 0x0 になります。

ビット 22:21 **PBURST[1:0]** : ペリフェラルバースト転送設定

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : シングル転送

01 : INCR4 (4 ビートのインクリメンタルバースト)

10 : INCR8 (8 ビートのインクリメンタルバースト)

11 : INCR16 (16 ビートのインクリメンタルバースト)

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ダイレクトモードでは、これらのビットはハードウェアによって強制的に 0x0 になります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 CT : 現在のターゲット (ダブルバッファモード時のみ)

このビットは、ハードウェアによってセット／クリアされます。また、ソフトウェアによって書き込むこともできます。

0 : 現在のターゲットメモリはメモリ 0 (DMA_SxM0AR ポインタによってアドレス指定) です。

1 : 現在のターゲットメモリはメモリ 1 (DMA_SxM1AR ポインタによってアドレス指定) です。

このビットは、EN が '0' のときのみ書き込むことができ、最初の転送の対象メモリ領域を示します。ストリームが有効になると、このビットはステータスフラグとして動作し、どのメモリ領域が現在のターゲットかを示します。

ビット 18 DBM : ダブルバッファモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : 転送終了時にバッファを切り替えません

1 : DMA 転送終了時にメモリターゲットを切り替えます

このビットは保護されており、EN が "0" のときのみ書き込みできます。

ビット 17:16 PL[1:0] : 優先順位レベル

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : 低

01 : 中

10 : 高

11 : 最優先

これらのビットは保護されており EN が "0" のときのみ書き込みできます。

ビット 15 PINCOS : ペリフェラルインクリメントオフセットサイズ

このビットは、ソフトウェアによってセット／クリアされます。

0 : ペリフェラルアドレス計算のためのオフセットサイズは PSIZE と関連付けられています。

1 : ペリフェラルアドレス計算のためのオフセットサイズは 4 に固定されています (32 ビットアライメント)。

ビット PINC="0" の場合、このビットは意味を持ちません。

このビットは保護されており EN="0" のときのみ書き込みできます。

ダイレクトモードが選択されているか、PBURST が "00" ではない場合、ストリームが有効になったとき (EN ビット="1") このビットはハードウェアによって強制的にローになります。

ビット 14:13 MSIZE[1:0] : メモリデータサイズ

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : バイト (8 ビット)

01 : ハーフワード (16 ビット)

10 : ワード (32 ビット)

11 : 予約済み

これらのビットは保護されており EN が "0" のときのみ書き込みできます。

ダイレクトモードでは、EN ビット="1" になり次第、MSIZE はハードウェアによって強制的に PSIZE と同じ値になります。

ビット 12:11 PSIZE[1:0] : ペリフェラルデータサイズ

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : バイト (8 ビット)

01 : ハーフワード (16 ビット)

10 : ワード (32 ビット)

11 : 予約済み

これらのビットは保護されており EN が "0" の場合のみ書き込みできます。

ビット 10 MINC : メモリインクリメントモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : メモリアドレスポインタは固定されています。

1 : メモリアドレスポインタは、各データ転送後にインクリメントされます (MSIZE に応じてインクリメント)

このビットは保護されており、EN が "0" のときのみ書き込みできます。

ビット 9 PINC : ペリフェラルインクリメントモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : ペリフェラルアドレスポインタは固定されています

1 : ペリフェラルアドレスポインタは、各データ転送後にインクリメントされます (PSIZE に応じてインクリメント)

このビットは保護されており、EN が“0”のときのみ書き込みできます。

ビット 8 CIRC : サーキュラモード

このビットは、ソフトウェアによってセット／クリアされ、かつハードウェアによってクリアできません。

0 : サーキュラモードは無効です。

1 : サーキュラモードは有効です。

ペリフェラルがフローコントローラ (PFCTRL ビット =1) となっていて、ストリームが有効なとき (ビット EN=1)、このビットは自動的にハードウェアによって 0 になります。

DBM ビットがセットされている場合、ストリームが有効になり次第 (EN ビット =“1”) ハードウェアにより強制的に 1 になります。

ビット 7:6 DIR[1:0] : データ転送方向

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : ペリフェラルからメモリ

01 : メモリからペリフェラル

10 : メモリ間転送

11 : 予約済み

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

ビット 5 PFCTRL : ペリフェラルフローコントローラ

このビットは、ソフトウェアによってセット／クリアされます。

0 : DMA がフローコントローラ

1 : ペリフェラルがフローコントローラ

このビットは保護されており、EN が“0”のときのみ書き込みできます。

メモリ間モードが選択されている場合 (DIR[1:0] ビット =10)、このビットは自動的にハードウェアにより強制的に 0 になります。

ビット 4 TCIE : 転送完了割り込み有効化

このビットは、ソフトウェアによってセット／クリアされます。

0 : TC 割り込みは無効です。

1 : TC 割り込みは有効です。

ビット 3 HTIE : 1/2 転送割り込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : HT 割り込みは無効です。

1 : HT 割り込みは有効です。

ビット 2 TEIE : 転送エラー割り込み有効化

このビットは、ソフトウェアによってセット／クリアされます。

0 : TE 割り込みは無効です。

1 : TE 割り込みは有効です。

ビット 1 DMEIE : ダイレクトモードエラー割り込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DME 割り込みは無効です。

1 : DME 割り込みは有効です。

ビット 0 **EN** : ストリームイネーブル/ロー読み出し時のストリームレディフラグ

このビットは、ソフトウェアによってセット/クリアされます。

0 : ストリームは無効です。

1 : ストリームは有効です。

このビットは、次のときハードウェアによってクリアできます。

- DMA 転送終了時 (ストリームが設定準備完了)
- AHB マスタバス上で転送エラーが発生した場合
- メモリ AHB ポート上の FIFO 閾値がバーストの大きさと適合しない場合

このビットが 0 として読み出された場合、ソフトウェアによって設定レジスタと FIFO ビットレジスタをプログラムすることができます。EN ビットが 1 として読み出された場合、これらのレジスタの書き込みは禁止されています。

注 : 新しい転送を開始するために EN ビットを“1”に設定する前に、DMA_LISR または DMA_HISR レジスタのストリームに対応するイベントフラグをクリアする必要があります。

9.5.6 DMA ストリーム x データ数レジスタ (DMA_HIFCR) (x=0~7)

アドレスオフセット : $0x14 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **NDT[15:0]** : 転送するデータ項目の数

転送されるデータ項目数 (0~65535)。このレジスタに書き込めるのは、ストリームが無効のときだけです。ストリームが有効になると、このレジスタは読み出し専用になり、送信されるべき残りのデータ項目を示します。このレジスタは、各 DMA 転送後にデクリメントされます。

転送が完了すると、このレジスタは、ゼロのままにすることもできれば (ストリームが通常モードのとき)、次の場合には事前にプログラムされた値を自動的にリロードすることもできます。

- ストリームがサーキュラモードに設定されている場合
- EN ビットを“1”に設定することにより、ストリームが再び有効になった場合

このレジスタの値が 0 の場合、ストリームが有効であっても、トランザクションは処理できません。

9.5.7 DMA ストリーム x ペリフェラルアドレスレジスタ (DMA_SxPAR) (x=0~7)

アドレスオフセット : $0x18 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

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

ビット 31:0 **PAR[31:0]** : ペリフェラルアドレス

データの読み出し／書き込みが行われるペリフェラルデータレジスタのベースアドレス。
これらのビットは書き込み保護されており、DMA_SxCR レジスタの EN ビット = "0" のときのみ書き込みできます。

9.5.8 DMA ストリーム x メモリ 0 アドレスレジスタ (DMA_SxM0AR) (x=0~7)

アドレスオフセット : $0x1C + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **M0A[31:0]** : メモリ 0 アドレス

- データの読み出し／書き込みが行われるメモリ領域 0 のベースアドレス。
これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。
- ストリームが無効 (DMA_SxCR レジスタの EN = "0") の場合、または
 - ストリームが有効 (DMA_SxCR レジスタの EN = "1") で DMA_SxCR レジスタのビット CT="1" (ダブルバッファモード) の場合。

9.5.9 DMA ストリーム x メモリ 1 アドレスレジスタ (DMA_SxM1AR) (x=0~7)

アドレスオフセット : $0x20 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1A[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 31:0 **M1A[31:0]** : メモリ 1 アドレス (ダブルバッファモードの場合に使用)

データの読み出し/書き込みが行われるメモリ領域 1 のベースアドレス。

このレジスタは、ダブルバッファモードでのみ使用されます。

これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。

- ストリームが無効 (DMA_SxCR レジスタの EN =“0”) の場合、または
- ストリームが有効 (DMA_SxCR レジスタの EN =“1”) で DMA_SxCR レジスタのビット CT=“0”の場合。

9.5.10 DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0~7)

アドレスオフセット : $0x24 + 0x24 \times \text{ストリーム番号}$

リセット値 : 0x0000 0021

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **FEIE** : FIFO エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : FE 割り込みは無効です。

1 : FE 割り込みは有効です。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5:3 **FS[2:0]** : FIFO ステータス

これらのビットは読み出し専用です。

000 : $0 < \text{FIFO レベル} < 1/4$

001 : $1/4 \leq \text{FIFO レベル} < 1/2$

010 : $1/2 \leq \text{FIFO レベル} < 3/4$

011 : $3/4 \leq \text{FIFO レベル} < \text{フル}$

100 : FIFO はエンプティです。

101 : FIFO はフルです。

その他 : 意味を持ちません。

これらのビットは、ダイレクトモードには無関係です (DMDIS ビットがゼロ)。

ビット 2 DMDIS : ダイレクトモードディセーブル

このビットは、ソフトウェアによってセット／クリアされます。ハードウェアによってセットすることができます。

0 : ダイレクトモードは有効です。

1 : ダイレクトモードは無効です。

このビットは保護されており、EN が“0”のときのみ書き込みできます。

このビットは、メモリ間モードが選択されており (DMA_SxCR の DIR ビットが “10”)、DMA_SxCR レジスタの EN ビットが “1” の場合、ハードウェアによってセットされます。その理由は、ダイレクトモードではメモリ間転送設定が使用できないからです。

ビット 1:0 FTH[1:0] : FIFO 閾値選択

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : FIFO フルの 1/4

01 : FIFO フルの 1/2

10 : FIFO フルの 3/4

11 : FIFO フル

これらのビットは、DMIS 値がゼロのときダイレクトモードで使用されません。

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

9.5.11 DMA レジスタマップ

表 39 に DMA レジスタの要約を示します。

表 39. DMA レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0000	DMA_LISR	Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2	Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	予約済み	予約済み	
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	0	
0x0004	DMA_HISR	Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6	Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	予約済み	予約済み	
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	0	
0x0008	DMA_LIFCR	Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	TEIF3	CDMEIF3	予約済み	CCEIF3	CTCIF2	CHTIF2	CCEIF2	CDMEIF2	Res.	CCEIF2	Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CCEIF1	CDMEIF1	Res.	CCEIF1	CTCIF0	CHTIF0	CCEIF0	CDMEIF0	予約済み	予約済み	
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	0	
0x000C	DMA_HIFCR	Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CCEIF7	CDMEIF7	予約済み	CCEIF7	CTCIF6	CHTIF6	CCEIF6	CDMEIF6	Res.	CCEIF6	Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CCEIF5	CDMEIF5	Res.	CCEIF5	CTCIF4	CHTIF4	CCEIF4	CDMEIF4	予約済み	予約済み	
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	0	
0x0010	DMA_S0CR	Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Res.		CT	DBM	PL[1:0]	PNCOS		MSIZE[1:0]		PSIZE[1:0]		MNC	PNC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	リセット値					0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	DMA_S0NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0018	DMA_S0PAR	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x001C	DMA_S0M0AR	M0A[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0020	DMA_S0M1AR	M1A[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0024	DMA_S0FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS		FTH	[1:0]		
	リセット値																								0		1	0	0	0	0	0	1	
0x0028	DMA_S1CR	Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Res.		CT	DBM	PL[1:0]	PNCOS		MSIZE[1:0]		PSIZE[1:0]		MNC	PNC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	リセット値					0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	DMA_S1NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0030	DMA_S1PAR	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	



表 39. DMA レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0034	DMA_S1M0AR	M0A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0038	DMA_S1M1AR	M1A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x003C	DMA_S1FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]					
	リセット値																								0	1		0	0	0	0	0	1			
0x0040	DMA_S2CR	Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]			PBURST[1:0]			Res.	CT	DBM	PL[1:0]		PINCOS		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	リセット値					0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA_S2NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NDT[15:]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	DMA_S2PAR	PA[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x004C	DMA_S2M0AR	M0A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0050	DMA_S2M1AR	M1A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0054	DMA_S2FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]					
	リセット値																								0	1	0	0	0	0	0	0	1			
0x0058	DMA_S3CR	Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]			PBURST[1:0]			Res.	CT	DBM	PL[1:0]		PINCOS		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	リセット値					0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	DMA_S3NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NDT[15:]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0060	DMA_S3PAR	PA[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0064	DMA_S3M0AR	M0A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0068	DMA_S3M1AR	M1A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x006C	DMA_S3FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]					
	リセット値																								0	1	0	0	0	0	0	1				

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0070	DMA_S4CR	Res.	Res.											CT	DBM	PL[1:0]		PINCOS					MINC	PINC	CIRC	DIR [1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN			
	リセット値					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0074	DMA_S4NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NDT[15:]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0078	DMA_S4PAR	PA[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x007C	DMA_S4M0AR	M0A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0080	DMA_S4M1AR	M1A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0084	DMA_S4FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE		FS[2:0]			DMDIS	FTH [1:0]				
	リセット値																									0		1	0	0	0	0	1			
0x0088	DMA_S5CR	Res.	Res.	Res.	Res.									CT	DBM	PL[1:0]		PINCOS					MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN			
	リセット値					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x008C	DMA_S5NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NDT[15:]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0090	DMA_S5PAR	PA[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0094	DMA_S5M0AR	M0A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0098	DMA_S5M1AR	M1A[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x009C	DMA_S5FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]			DMDIS	FTH [1:0]				
	リセット値																									0		1	0	0	0	0	1			
0x00A0	DMA_S6CR	Res.	Res.	Res.	Res.									CT	DBM	PL[1:0]		PINCOS					MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN</			

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などにつきましては如何なる責任にも負いません。

表 39. DMA レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00B8	DMA_S7CR	Res.	Res.	Res.	Res.	CHSEL[2:0]			MBURST[1:0]			PBURST[1:0]		Res.	CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCtrl	TCIE	HTIE	TEIE	DMEIE	EN	
	リセット値					0	0	0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00BC	DMA_S7NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:..]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C0	DMA_S7PAR	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C4	DMA_S7M0AR	M0A[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C8	DMA_S7M1AR	M1A[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00CC	DMA_S7FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]			DMDIS		FTH [1:0]		
	リセット値																								0		1	0	0	0	0	0	1	

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

10 割り込みとイベント

10.1 ネスト化されたベクタ割り込みコントローラ (NVIC)

10.1.1 NVIC の機能

ネスト化されたベクタ割り込みコントローラ NVIC には、次のような機能があります。

- 52 本のマスク可能な割り込みチャネル (FPU 搭載 Cortex®-M4 の 16 本の割り込みラインは数に含めない)
- 16 のプログラム可能な優先レベル (4 ビットの割り込み優先順位を使用)
- 遅延時間の少ない例外および割り込み処理
- 電源管理制御
- システム制御レジスタの実装

NVIC とプロセッサコアのインタフェースは密に結合され、割り込み処理の遅延時間を小さくし、遅れて到着した割り込みを効率的に処理できます。

コア例外を含むすべての割り込みは、NVIC によって管理されます。例外と NVIC プログラミングの詳細については、プログラミングマニュアル PM0214 を参照してください。

10.1.2 SysTick 較正值レジスタ

SysTick 較正值は 10500 に固定され、SysTick クロックを 10.5 MHz (HCLK を 84 MHz に設定した状態で HCLK/8) に設定した状態で 1 ms という基準タイムベースを与えます。

10.1.3 割り込みベクタと例外ベクタ

表 40 は、STM32F412xx デバイスのベクタテーブルです。

10.2 外部割り込み／イベントコントローラ (EXTI)

イベント／割り込みリクエストを生成するための外部割り込み／イベントコントローラは、最大 23 のエッジ検出回路で構成されます。各入力ラインは、個別に設定可能であり、タイプ (割り込みまたはイベント) と対応するトリガイイベント (立ち上がり、立ち下がり、または両方) を選択できます。また、ラインごとにマスクをかけることもできます。ペンディングレジスタは、割り込みリクエストのステータスラインを維持します。

表 40. STM32F412xxのベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-3	固定	リセット	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスカブル割り込み、クロックセキュリティシステム	0x0000 0008
-	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
-	0	設定可能	MemManage	メモリ管理	0x0000 0010
-	1	設定可能	BusFault	プリフェッチ異常、メモリアクセス異常	0x0000 0014
-	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
-	-	-	-	予約済み	0x0000 001C~ 0x0000 002B
-	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
-	4	設定可能	デバッグモニタ	デバッグモニタ	0x0000 0030
-	-	-	-	予約済み	0x0000 0034
-	5	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038
-	6	設定可能	SysTick	システムティックタイマ	0x0000 003C
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割り込み	0x0000 0040
1	8	設定可能	PVD	PVD 検出割り込み (EXTI ライン経由)	0x0000 0044
2	9	設定可能	TAMP_STAMP	タンバおよびタイムスタンプ割り込み (EXTI ライン経由)	0x0000 0048
3	10	設定可能	RTC_WKUP	RTC ウェイクアップ割り込み (EXTI ライン経由)	0x0000 004C
4	11	設定可能	FLASH	フラッシュグローバル割り込み	0x0000 0050
5	12	設定可能	RCC	RCC グローバル割り込み	0x0000 0054
6	13	設定可能	EXTI0	EXTI ライン 0 割り込み	0x0000 0058
7	14	設定可能	EXTI1	EXTI ライン 1 割り込み	0x0000 005C
8	15	設定可能	EXTI2	EXTI ライン 2 割り込み	0x0000 0060
9	16	設定可能	EXTI3	EXTI ライン 3 割り込み	0x0000 0064
10	17	設定可能	EXTI4	EXTI ライン 4 割り込み	0x0000 0068
11	18	設定可能	DMA1_Stream0	DMA1 ストリーム 0 グローバル割り込み	0x0000 006C
12	19	設定可能	DMA1_Stream1	DMA1 ストリーム 1 グローバル割り込み	0x0000 0070
13	20	設定可能	DMA1_Stream2	DMA1 ストリーム 2 グローバル割り込み	0x0000 0074
14	21	設定可能	DMA1_Stream3	DMA1 ストリーム 3 グローバル割り込み	0x0000 0078
15	22	設定可能	DMA1_Stream4	DMA1 ストリーム 4 グローバル割り込み	0x0000 007C
16	23	設定可能	DMA1_Stream5	DMA1 ストリーム 5 グローバル割り込み	0x0000 0080
17	24	設定可能	DMA1_Stream6	DMA1 ストリーム 6 グローバル割り込み	0x0000 0084
18	25	設定可能	ADC	ADC1 グローバル割り込み	0x0000 0088
19	26	設定可能	CAN1_TX	CAN1 TX 割り込み	0x0000 008C



表 40. STM32F412xxのベクタテーブル（続き）

順位	優先順位	優先種別	項目（略称）	説明	アドレス
20	27	設定可能	CAN1_RX0	CAN1 RX0 割り込み	0x0000 0090
21	28	設定可能	CAN1_RX1	CAN1 RX1 割り込み	0x0000 0094
22	29	設定可能	CAN1_SCE	CAN1 SCE 割り込み	0x0000 0098
23	30	設定可能	EXTI9_5	EXTI ライン [9:5] 割り込み	0x0000 009C
24	31	設定可能	TIM1_BRK_TIM9	TIM1 ブレーク割り込み および TIM9 グローバル割り込み	0x0000 00A0
25	32	設定可能	TIM1_UP_TIM10	TIM1 更新割り込み および TIM10 グローバル割り込み	0x0000 00A4
26	33	設定可能	TIM_TRG_COM_TIM11	TIM1 トリガおよびコムテーション割り込み および TIM11 グローバル割り込み	0x0000 00A8
27	34	設定可能	TIM1_CC	TIM1 キャプチャ / 比較割り込み	0x0000 00AC
28	35	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 00B0
29	36	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 00B4
30	37	設定可能	TIM4	TIM4 グローバル割り込み	0x0000 00B8
31	38	設定可能	I2C1_EVT	I2C1 グローバルイベント割り込み	0x0000 00BC
32	39	設定可能	I2C1_ERR	I2C1 グローバルエラー割り込み	0x0000 00C0
33	40	設定可能	I2C2_EVT	I2C2 グローバルイベント割り込み	0x0000 00C4
34	41	設定可能	I2C2_ERR	I2C2 グローバルエラー割り込み	0x0000 00C8
35	42	設定可能	SPI1	SPI1 グローバル割り込み	0x0000 00CC
36	43	設定可能	SPI2	SPI2 グローバル割り込み	0x0000 00D0
37	44	設定可能	USART1	USART1 グローバル割り込み	0x0000 00D4
38	45	設定可能	USART2	USART2 グローバル割り込み	0x0000 00D8
39	46	設定可能	USART3	USART3 グローバル割り込み	0x0000 00DC
40	47	設定可能	EXTI15_10	EXTI ライン [15:10] 割り込み	0x0000 00E0
41	48	設定可能	EXTI17 / RTC アラーム	EXTI ライン 17 割り込み / RTC アラーム（A および B）割り込み （EXTIライン経由）	0x0000 00E4
42	49	設定可能	EXTI18/OTG_FS WKUP	EXTI ライン 18 割り込み / USB OTG FS ウェイクアップ割り込み（EXTI ライン経由）	0x0000 00E8
43	50	設定可能	TIM8_BRK_TIM12	TIM8 ブレーク割り込み TIM12 グローバル割り込み	0x0000 00EC
44	51	設定可能	TIM8_UP_TIM13	TIM8 更新割り込み TIM13 グローバル割り込み	0x0000 00F0
45	52	設定可能	TIM8_TRG_COM_TIM14	TIM8 トリガおよびコムテーション割り込み TIM14 グローバル割り込み	0x0000 00F4
46	53	設定可能	TIM8_CC	TIM8 キャプチャ / 比較割り込み	0x0000 00F8
47	54	設定可能	DMA1_Stream7	DMA1 グローバル割り込みチャンネル 7	0x0000 00FC
48	55	設定可能	FSMC	FSMC グローバル割り込み	0x0000 0100

表 40. STM32F412xxのベクタテーブル（続き）

順位	優先順位	優先種別	項目（略称）	説明	アドレス
49	56	設定可能	SDIO	SDIO グローバル割り込み	0x0000 0104
50	57	設定可能	TIM5	TIM5 グローバル割り込み	0x0000 0108
51	58	設定可能	SPI3	SPI3 グローバル割り込み	0x0000 010C
54	61	設定可能	TIM6	TIM6 グローバル割り込み	0x0000 0118
55	62	設定可能	TIM7	TIM7 グローバル割り込み	0x0000 011C
56	63	設定可能	DMA2_Stream0	DMA2 ストリーム 0 グローバル割り込み	0x0000 0120
57	64	設定可能	DMA2_Stream1	DMA2 ストリーム 1 グローバル割り込み	0x0000 0124
58	65	設定可能	DMA2_Stream2	DMA2 ストリーム 2 グローバル割り込み	0x0000 0128
59	66	設定可能	DMA2_Stream3	DMA2 ストリーム 3 グローバル割り込み	0x0000 012C
60	67	設定可能	DMA2_Stream4	DMA2 ストリーム 4 グローバル割り込み	0x0000 0130
61	68	設定可能	DFSDM1_FLT0	SD フィルタ 0 グローバル割り込み	0x0000 0134
62	69	設定可能	DFSDM1_FLT1	SD フィルタ 1 グローバル割り込み	0x0000 0138
63	70	設定可能	CAN2_TX	CAN2 TX 割り込み	0x0000 013C
64	71	設定可能	CAN2_RX0	BXCAN2 RX0 割り込み	0x0000 0140
65	72	設定可能	CAN2_RX1	BXCAN2 RX1 割り込み	0x0000 0144
66	73	設定可能	CAN2_SCE	CAN2 SCE 割り込み	0x0000 0148
67	74	設定可能	OTG_FS	USB OTG FS グローバル割り込み	0x0000 014C
68	75	設定可能	DMA2_Stream5	DMA2 ストリーム 5 グローバル割り込み	0x0000 0150
69	76	設定可能	DMA2_Stream6	DMA2 ストリーム 6 グローバル割り込み	0x0000 0154
70	77	設定可能	DMA2_Stream7	DMA2 ストリーム 7 グローバル割り込み	0x0000 0158
71	78	設定可能	USART6	USART6 グローバル割り込み	0x0000 015C
72	79	設定可能	I2C3_EV	I ² C3 イベント割り込み	0x0000 0160
73	80	設定可能	I2C3_ER	I ² C3 エラー割り込み	0x0000 0164
80	87	設定可能	RNG	RNG グローバル割り込み	0x0000 0180
81	88	設定可能	FPU	FPU グローバル割り込み	0x0000 0184
84	91	設定可能	SPI4	SPI4 グローバル割り込み	0x0000 0190
85	92	設定可能	SPI5	SPI5 グローバル割り込み	0x0000 0194
92	99	設定可能	Quad-SPI	Quad-SPI グローバル割り込み	0x0000 01B0
95	102	設定可能	I2CFMP1 イベント	I2CFMP1 イベント割り込み	0x0000 01BC
96	103	設定可能	I2CFMP1 エラー	I2CFMP1 エラー割り込み	0x0000 01C0

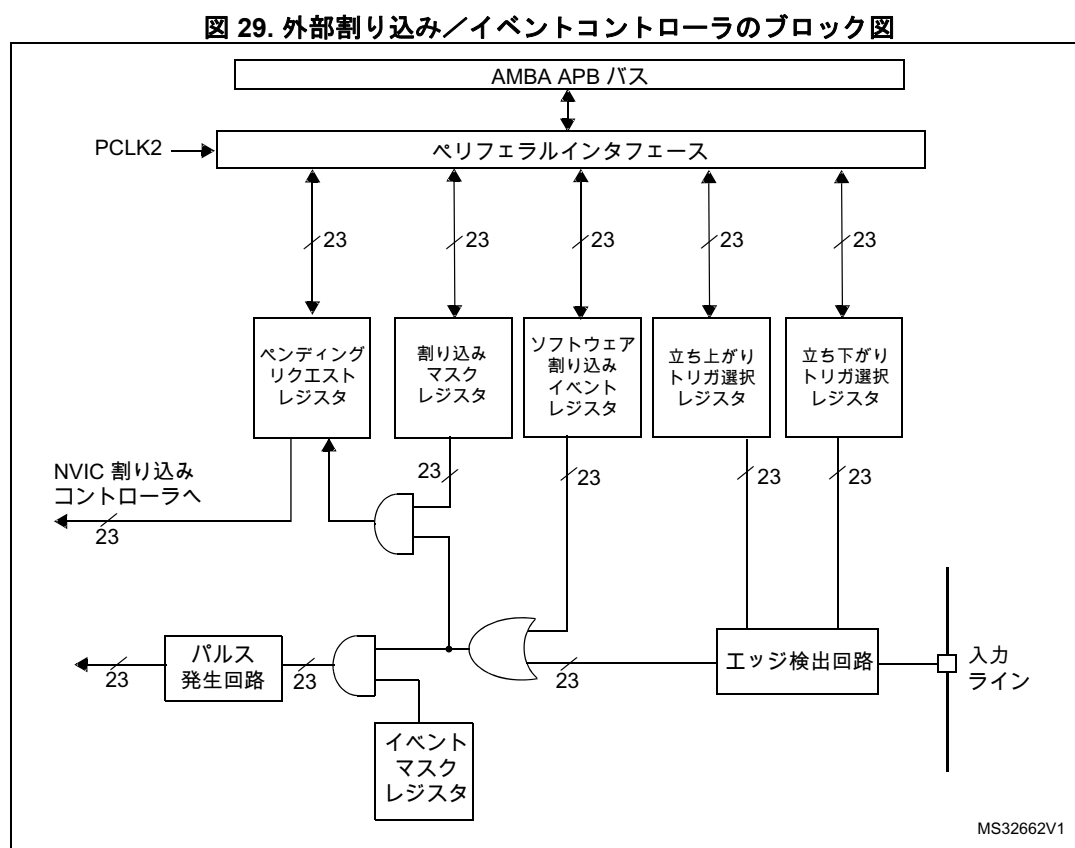
10.2.1 EXTI の主な機能

EXTI コントローラの主な機能は次のとおりです。

- 割り込み／イベントラインごとに独立したトリガとマスク
- 割り込みラインごとに専用のステータスビット
- 最大 23 本のソフトウェアイベント／割り込みリクエストの生成
- APB2 クロック周期よりも短いパルス幅の外部信号の検出。このパラメータの詳細については、STM32F4xx データシートの電気的特性のセクションを参照してください。

10.2.2 EXTI ブロック図

図 29 にブロック図を示します。



10.2.3 ウェイクアップイベント管理

STM32F4xx は、外部イベントや内部イベントを処理することによって、コアをウェイクアップできます (WFE)。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、FPU 搭載 Cortex®-M4 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャンネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードで設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャンネルペンディングビットをクリアする必要はありません。

外部ラインをウェイクアップイベントとして使用する方法については、[セクション 10.2.4: 機能詳細](#)を参照してください。

10.2.4 機能詳細

割り込みを生成するには、割り込みラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、割り込みマスクレジスタの対応するビットに "1" を書き込むことによって割り込みリクエストを有効にします。選択されたエッジが外部割り込みラインで発生すると、割り込みリクエストが生成されます。割り込みラインに対応するペンディングビットもセットされます。このリクエストは、ペンディングレジスタに "1" を書き込むことによってリセットされます。

イベントを生成するには、イベントラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、イベントマスクレジスタの対応するビットに "1" を書き込むことによってイベントリクエストを有効にします。選択されたエッジがイベントラインで発生すると、イベントパルスが生成されます。イベントラインに対応するペンディングビットはセットされません。

割り込み/イベントリクエストは、ソフトウェア割り込み/イベントレジスタに "1" を書き込むことによって、ソフトウェアによって生成することもできます。

ハードウェア割り込みの選択

23 本のラインを割り込みソースとして設定するには、次の手順を使用します。

- 23 本の割り込みラインのマスクビットを設定します (EXTI_IMR)。
- 割り込みラインのトリガ選択ビットを設定します (EXTI_RTISR および EXTI_FTISR)。
- 23 本のラインのいずれかからの割り込みを正しく処理できるように、外部割り込みコントローラ (EXTI) にマップされた NVIC IRQ チャンネルを制御するイネーブルビットとマスクビットを設定します。

ハードウェアイベントの選択

23 本のラインをイベントソースとして設定するには、次の手順を使用します。

- 23 本のイベントラインのマスクビットを設定します (EXTI_EMR)。
- イベントラインのトリガ選択ビットを設定します (EXTI_RTISR および EXTI_FTISR)。

ソフトウェア割り込み/イベントの選択

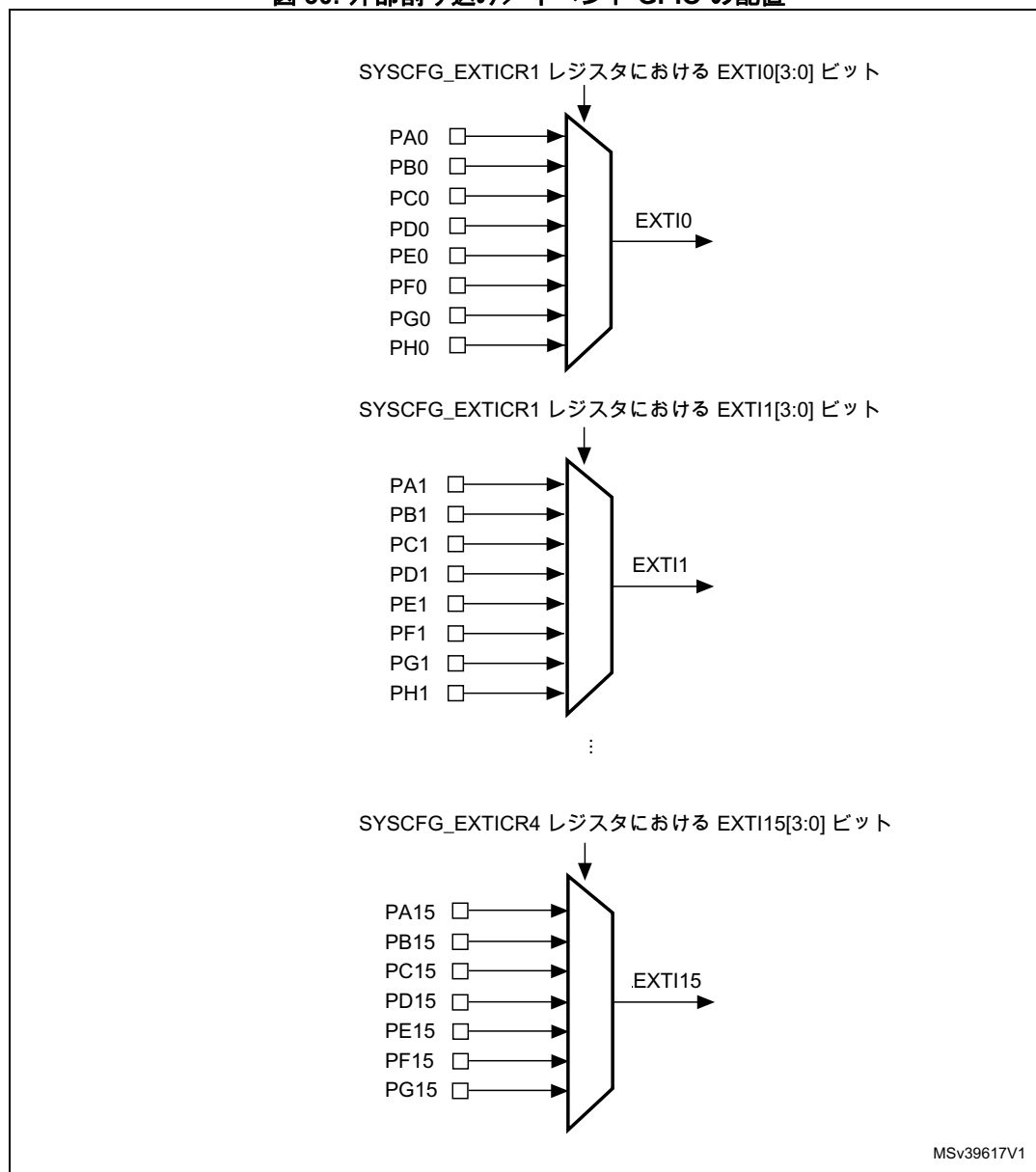
23 本のラインは、ソフトウェア割り込み/イベントラインとして設定できます。ソフトウェア割り込みの生成には、次の手順を使用します。

- 23 本の割り込み/イベントラインのマスクビットを設定します (EXTI_IMR、EXTI_EMR)。
- ソフトウェア割り込みレジスタの必要なビットをセットします (EXTI_SWIER)。

10.2.5 外部割り込み／イベントラインの配置

最大 STM32F412xx は、次のように、16 本の外部割り込み／イベントラインに接続されます。

図 30. 外部割り込み／イベント GPIO の配置



他の 5 本の EXTI ラインは次のように接続されます。

- EXTI ライン 16 は PVD 出力に接続されます。
- EXTI ライン 17 は RTC アラームイベントに接続されます。
- EXTI ライン 18 は USB OTG FS ウェイクアップイベントに接続されます。
- EXTI ライン 21 は RTC のタンパイイベントおよびタイムスタンプイベントに接続されます。
- EXTI ライン 22 は RTC ウェイクアップイベントに接続されます。

10.3 EXTI レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 : レジスタに関する略記](#)を参照してください。

10.3.1 割り込みマスクレジスタ (EXTI_IMR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR22	MR21	Res.	Res.	MR18	MR17	MR16
									rW	rW			rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **MR[22:21]** : ライン x の割り込みマスク

- 0 : ライン x からの割り込みリクエストはマスクされます。
- 1 : ライン x からの割り込みリクエストはマスクされません。

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **MR[18:0]** : ライン x の割り込みマスク

- 0 : ライン x からの割り込みリクエストはマスクされます。
- 1 : ライン x からの割り込みリクエストはマスクされません。

10.3.2 イベントマスクレジスタ (EXTI_EMR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR22	MR21	Res.	Res.	MR18	MR17	MR16
									rW	rW			rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **MR[22:21]** : ライン x のイベントマスク

- 0 : ライン x からのイベントリクエストはマスクされます。
- 1 : ライン x からのイベントリクエストはマスクされません。

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **MR[18:0]** : ライン x のイベントマスク

- 0 : ライン x からのイベントリクエストはマスクされます。
- 1 : ライン x からのイベントリクエストはマスクされません。



10.3.3 立ち上がりトリガ選択レジスタ (EXTI_RTSR)

アドレスオフセット : 0x08
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	Res.	Res.	TR18	TR17	TR16
									r/w	r/w			r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **TR[22:21]** : ライン x の立ち上がりトリガイベント設定ビット

- 0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
- 1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **TR[18:0]** : ライン x の立ち上がりトリガイベント設定ビット

- 0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
- 1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

注 : 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。
**EXTI_RTSR レジスタへの書き込み中に外部割り込みラインで立ち上がりエッジが発生した場合、ペ
ンディングビットはセットされません。**
**同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することが
できます。この設定では、両方のエッジでトリガ条件が生成されます。**

10.3.4 立ち下がりトリガ選択レジスタ (EXTI_FTSR)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	Res.	Res.	TR18	TR17	TR16
									r/w	r/w			r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **TR[22:21]** : ライン x の立ち下がりトリガイイベント設定ビット

- 0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です
1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **TR[18:0]** : ライン x の立ち下がりトリガイイベント設定ビット

- 0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です
1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

注 : 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。
**EXTI_FTSR レジスタへの書き込み中に外部割り込みラインで立ち下がりエッジが発生した場合、ペ
ンディングビットはセットされません。**
**同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することが
できます。この設定では、両方のエッジでトリガ条件が生成されます。**

10.3.5 ソフトウェア割り込みイベントレジスタ (EXTI_SWIER)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

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

- ビット 31:23 予約済みであり、リセット値に保持する必要があります。
- ビット 22:21 **SWIER[22:21]** : ライン x のソフトウェア割り込み
- 割り込みが EXTI_IMR レジスタのライン x で有効になっている場合、SWIERx ビットが“0”のときに“1”を書き込むと EXTI_PR レジスタの対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。
- EXTI_PR レジスタの対応するビットをクリアする（対応ビットに“1”を書き込む）と、このビットはクリアされます。
- ビット 20:19 予約済みであり、リセット値に保持する必要があります。
- ビット 18:0 **SWIER[18:0]** : ライン x のソフトウェア割り込み
- 割り込みが EXTI_IMR レジスタのライン x で有効になっている場合、SWIERx ビットが“0”のときに“1”を書き込むと EXTI_PR レジスタの対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。
- EXTI_PR レジスタの対応するビットをクリアする（対応ビットに“1”を書き込む）と、このビットはクリアされます。

10.3.6 ペンディングレジスタ (EXTI_PR)

アドレスオフセット : 0x14
リセット値 : 定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR22	PR21	Res.	Res.	PR18	PR17	PR16
									rc_w1	rc_w1			rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:21 **PR[22:21]** : ペンディングビット

- 0 : トリガリクエストは発生していません。
- 1 : 選択されたトリガリクエストが発生しました。
このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。
このビットは、“1”をプログラムすることによってクリアされます。

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **PR[18:0]** : ペンディングビット

- 0 : トリガリクエストは発生していません。
- 1 : 選択されたトリガリクエストが発生しました。
このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。
このビットは、“1”をプログラムすることによってクリアされます。

10.3.7 EXTI レジスタマップ

表 41 に、EXTI レジスタマップとリセット値を示します。

表 41. 外部割り込み／イベントコントローラのレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR [22:21]		Res.	Res.	MR[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR [22:21]		Res.	Res.	MR[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR [22:21]		Res.	Res.	TR[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_FTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR [22:21]		Res.	Res.	TR[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER [22:21]		Res.	Res.	SWIER[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR [22:21]		Res.	Res.	PR[18:0]																		
	リセット値										0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

セクション: レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

11 フレキシブルスタティックメモリコントローラ (FSMC)

フレキシブルスタティックメモリコントローラ (FSMC) には、次の 1 つのメモリコントローラが含まれています。

- NOR/PSRAM メモリコントローラ

11.1 FSMC の主な特長

FSMC 機能ブロックは、同期および非同期のスタティックメモリとのインタフェースを構成します。主要目的は、

- AHB トランザクションを該当する外部デバイスプロトコルへ翻訳すること
- 外部メモリデバイスのアクセス時間条件を満たすようにすること

すべての外部メモリが、アドレス信号、データ信号、制御信号をコントローラと共用します。各外部デバイスを固有のチップセレクトを使ってアクセスします。FSMC は外部デバイス毎度に 1 回だけアクセスします。

FSMC コントローラの主要機能は次の通りです。

- 次のスタティックメモリにマップされたデバイスとのインタフェース
 - スタティックランダムアクセスメモリ (SRAM)
 - NOR 型フラッシュメモリ/OneNAND フラッシュメモリ
 - PSRAM (4 メモリバンク)
- NOR フラッシュメモリ、PSRAM などの同期デバイスに対するアクセスを高速化するバーストモードのサポート
- 非同期アクセスと同期アクセス用のプログラマブルな連続クロック出力
- 8 ビット、16 ビット幅のデータバス
- 各メモリバンクに対する独立したチップセレクト制御
- メモリバンクごとに独立した設定
- PSRAM、SRAM デバイスで使用する書き込みイネーブルとバイトレーン選択出力
- 外部非同期ウェイト制御
- 16x32 ビットの深さの書き込み FIFO

書き込み FIFO はすべてのメモリコントローラに共通であり、以下の構成となっています。

- 書き込みデータ FIFO は、メモリへ書き込む AHB データ (最大 32 ビット) と、AHB 転送 (バーストモードまたは非シーケンシャルモード) 用の 1 ビットを格納
- 書き込みアドレス FIFO は、AHB アドレス (最大 28 ビット) と AHB データサイズ (最大 2 ビット) を格納バーストモードで動作する場合、ページ境界 (PSRAM) を跨ぐとき以外は、開始アドレスのみを格納この場合、AHB バーストは 2 つの FIFO エントリに分割されます。

書き込み FIFO は、FSMC_BCR1 レジスタの WFDIS ビットをセットすることによって無効にできません。

開始時に、ユーザアプリケーションから FSMC ピンを設定する必要があります。アプリケーションで使用しない FSMC I/O ピンは、他の用途に使用することができます。

外部デバイスタイプおよびその特性を定義する FSMC レジスタは、通常、起動時に設定され、次のリセットまたは電源投入まで変更されません。ただし、設定値は何時でも変更することができます。



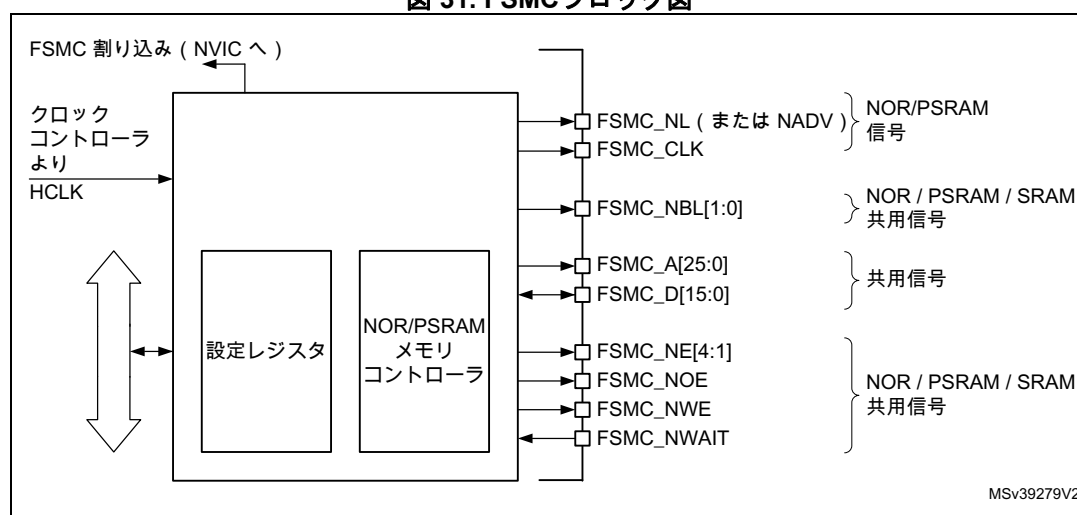
11.2 ブロック図

FSMC は 次のメインブロックで構成されます。

- AHB インタフェース (FSMC 設定レジスタを含む)
- NOR フラッシュ /PSRAM/SRAM コントローラ

以下にブロック図を示します。

図 31. FSMC ブロック図



11.3 AHB インタフェース

AHB スレーブインタフェースは、内部 CPU および他のバスマスタペリフェラルから外部スタティックメモリへのアクセスを可能にします。

AHB トランザクションは、外部デバイスプロトコルに変換されます。特に、選択された外部メモリが 16 または 8 ビット幅の場合、AHB 上の 32 ビット幅のトランザクションは、16 または 8 ビットの連続した複数のアクセスに分割されます。FSMC チップセレクト (FSMC_NEX) は、拡張モードが有効になっているときのアクセスモード D の場合を除き、複数の連続アクセス間にトグルしません。

FSMC は、以下の条件下で AHB エラーを生成します。

- 有効化していない FSMC バンク（バンク1~4）の読み出しまたは書き込みを行ったとき。
- FSMC_BCRx レジスタの FACCEN ビットがリセットされている状態で NOR フラッシュバンクの読み出しまたは読み出しを行ったとき。

この AHB エラーの影響は、読み出し／書き込みアクセスを試みた AHB マスタに依存します。

- アクセスが FPU 搭載 Cortex®-M4 CPU から行われた場合は、ハードフォルト割り込みが発生します。
- アクセスが DMA コントローラから行われた場合は、DMA 転送エラーが発生し、対応する DMA チャンネルが自動的に無効化されます。

AHB クロック (HCLK) は、FSMC のリファレンスクロックです。

11.3.1 サポートされるメモリおよびトランザクション

一般的なトランザクション規則

リクエストされる AHB トランザクションのデータサイズは、アクセスされる外部デバイスのデータ幅が固定であっても、8、16、または 32 ビット幅です。このため、転送に矛盾が生じることがあります。

したがって、次のようないくつかの単純なトランザクション規則を守る必要があります。

- AHB トランザクションサイズとメモリデータサイズが等しい場合：
この場合は問題ありません。
- AHB トランザクションサイズがメモリサイズより大きい場合：
この場合、FSMC は、AHB トランザクションを外部データ幅に合わせて、連続した小さなメモリアクセスに分割します。FSMC チップセレクト (FSMC_NEX) は、複数の連続アクセス間にトグルしません。
- AHB トランザクションサイズがメモリサイズより小さい場合：
外部デバイスのタイプによっては、転送に一貫性がなくなる場合があります：
 - － バイト選択機能を持つデバイス (SRAM、ROM、PSRAM) に対するアクセス
この場合、FSMC は、読み出し／書き込みトランザクションを許可して、バイトレーン NBL[1:0] を通じて正しいデータにアクセスします。
書き込み対象バイトは NBL[1:0] によりアドレス指定されます。
すべてのメモリバイトが読み出され (読み出しトランザクション中 NBL[1:0] はローに駆動されます)、不要なバイトは無視されます。
 - － バイト選択機能を持たないデバイス (NOR) に対するアクセス
この状況は、16 ビット幅のフラッシュメモリへのバイトアクセスが要求されたときに発生します。バイトモードでデバイスをアクセスできないため (フラッシュメモリに対しては 16 ビットワードの読み出し／書き込みのみが可能)、書き込みトランザクションと読み出しトランザクションが可能です (コントローラは 16 ビットメモリワード全体を読み出して、必要なバイトのみを使用)。

NOR 型フラッシュ /PSRAM のラップサポート

同期メモリに対するラップバーストモードはサポートされていません。メモリは、不定長のリニアバーストモードに構成する必要があります。

設定レジスタ

FSMC は、レジスタのセットとして設定することができます。NOR フラッシュ/PSRAM コントローラレジスタの詳細については、[セクション 11.5.6](#) を参照してください。

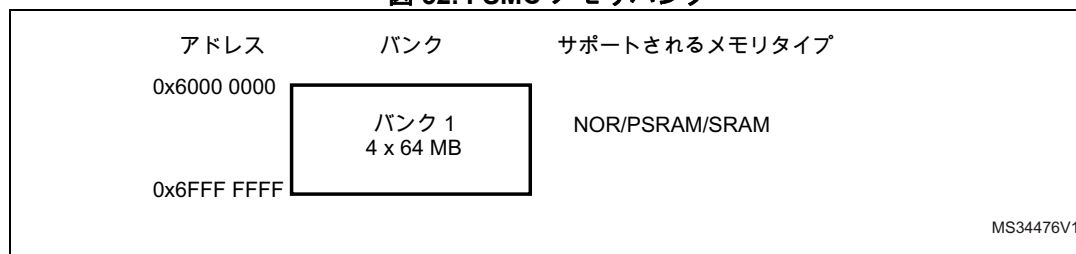
11.4 外部デバイスアドレスマッピング

FSMC から見ると、外部メモリは、それぞれ 256 MB の固定サイズのバンクに分けられています ([図 32](#) を参照)。

- バンク 1 は、4 つまでの NOR フラッシュメモリまたは PSRAM メモリデバイスのアドレス指定に使用されます。このバンクは、4 つの専用チップセレクトを持つ 4 つの NOR/PSRAM サブバンクに分割されます。
 - バンク 1 - NOR/PSRAM 1
 - バンク 1 - NOR/PSRAM 2
 - バンク 1 - NOR/PSRAM 3
 - バンク 1 - NOR/PSRAM 4
- バンク 3 は、NAND フラッシュメモリデバイスのアドレス指定に使用されます。この空間の MPUメモリ属性をソフトウェアによってデバイスに再設定する必要があります。

各バンクで使用されるメモリのタイプは、ユーザアプリケーションから設定レジスタを使って設定することができます。

図 32. FSMC メモリバンク



11.4.1 NOR/PSRAM アドレスマッピング

HADDR[27:26] ビットは、[表 42](#) に示される 4 つのメモリバンクの 1 つを選択するために使用されます。

表 42. NOR/PSRAM バンク選択

HADDR[27:26] ⁽¹⁾	選択されるバンク
00	バンク 1 - NOR/PSRAM 1
01	バンク 1 - NOR/PSRAM 2
10	バンク 1 - NOR/PSRAM 3
11	バンク 1 - NOR/PSRAM 4

1. HADDR は、外部メモリに変換される内部 AHB アドレスラインです。

HADDR[25:0] ビットは、外部メモリアドレスを含みます。HADDR はバイトアドレスですが、メモリはワードレベルでアクセスされるので、メモリに対して実際に発行されるアドレスは、次の表に示されるように、メモリのデータ幅に応じて変わります。

表 43. NOR/PSRAM 外部メモリアドレス

メモリ幅 ⁽¹⁾	メモリに発行されるデータアドレス	最大メモリ容量 (ビット)
8 ビット	HADDR[25:0]	64 Mバイト x 8 = 512 M ビット
16 ビット	HADDR[25:1] >> 1	64 Mバイト/2 x 16 = 512 M ビット

1. 外部メモリが 16 ビット幅の場合、FSMC は、内部で HADDR[25:1] を使用して、外部メモリ FSMC_A[24:0] に対するアドレスを生成します。
外部メモリの幅にかかわらず、FSMC_A[0] は外部メモリアドレス A[0] に接続されます。

11.5 NOR 型フラッシュ／PSRAM コントローラ

FSMC は、以下のタイプのメモリを駆動するのに適した信号タイミングを発生します。

- 非同期 SRAM および ROM
 - 8 ビット
 - 16 ビット
- PSRAM (Cellular RAM)
 - 非同期モード
 - 同期アクセスに対するバーストモード
 - マルチプレクスまたは非マルチプレクス
- NOR フラッシュメモリ
 - 非同期モード
 - 同期アクセスに対するバーストモード
 - マルチプレクスまたは非マルチプレクス

FSMC は、バンクごとに 1 つのチップセレクト信号 NE[4:1] を出力します。その他の信号（アドレス、データ、および制御）はすべて、共有されます。

FSMC は、次のようなプログラム可能なタイミングにより、広範囲なデバイスをサポートしています。

- プログラム可能なウェイトステート (15 まで)
- プログラム可能なバスターンアラウンドサイクル (15 まで)
- プログラム可能な出カインーブルおよび書き込みインーブル遅延 (15 まで)
- 独立した読み出しおよび書き込みタイミングとプロトコルにより、広範囲なメモリおよびタイミングをサポート
- プログラム可能な連続クロック (FSMC_CLK) 出力

FSMC クロック (FSMC_CLK) は HCLK クロックの約数です。このクロックは、同期アクセス中にのみ、または FSMC_BCR1 レジスタの CCKEN ビット設定に応じて非同期アクセスおよび同期アクセス中に、選択した外部デバイスへ供給することができます。

- CCLKEN ビットがリセットされている場合、FSMC は同期アクセス（読み出し／書き込みトランザクション）中にのみクロック (CLK) を生成します。
- CCLKEN ビットがセットされている場合、FSMC は非同期アクセス中および同期アクセス中に連続クロックを生成します。FSMC_CLK 連続クロックを生成するためには、バンク 1 を同期モードに設定する必要があります（[セクション 11.5.6 : NOR/PSRAM コントローラレジスタ](#)を参照）。すべての同期メモリに対して同じクロックが使用されるため、連続出力クロックを生成

して同期アクセスを実行する場合、AHB データサイズはメモリデータ幅 (MWID) に一致する必要があります。そうしないと、AHB データトランザクションに応じて、FSMC_CLK 周波数が変化します (FSMC_CLK 分周比の式については、[セクション 11.5.5: 同期トランザクション](#)を参照)。

各バンクのサイズは 64 MB に固定されます。各バンクは、専用レジスタによって設定されます ([セクション 11.5.6: NOR/PSRAM コントローラレジスタ](#)を参照)。

プログラム可能なメモリパラメータは、アクセスタイミング ([表 44](#) を参照) を含み、ウェイト管理をサポートします (バーストモードでの PSRAM および NOR フラッシュへのアクセス)。

表 44. プログラム可能な NOR/PSRAM のアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
アドレスセットアップ	アドレスセットアップフェーズ時間	非同期	AHB クロックサイクル (HCLK)	0	15
アドレスホールド	アドレスホールドフェーズ時間	非同期、マルチプレクス I/O	AHB クロックサイクル (HCLK)	1	15
データセットアップ	データセットアップフェーズ時間	非同期	AHB クロックサイクル (HCLK)	1	256
バスターン	バスターンアラウンドフェーズ時間	非同期および同期読み出し/書き込み	AHB クロックサイクル (HCLK)	0	15
クロック分周比	1 つのメモリクロックサイクル (CLK) に必要な AHB クロックサイクル (HCLK) 数	同期	AHB クロックサイクル (HCLK)	2	16
データ遅延	バーストの最初のデータの前にメモリに発行されるクロックサイクル数	同期	メモリクロックサイクル (CLK)	2	17

11.5.1 外部メモリインタフェース信号

[表 45](#)、[表 46](#) および [表 47](#) に、NOR フラッシュメモリ、SRAM、および PSRAM へのインタフェースで一般に使用される信号を示します。

注: 接頭辞「N」は、アクティブラーの信号を表します。

NOR フラッシュメモリ、非マルチプレクス I/O

表 45. 非マルチプレクス I/O NOR フラッシュメモリ

FSMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書き込みインーブル



表 45. 非マルチプレクス I/O NOR フラッシュメモリ (続き)

FSMC 信号名	I/O	機能
NL(= NADV)	O	ラッチイネーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FSMC への NOR 型フラッシュウェイト入力信号

最大容量は、512 M ビット (26 アドレスライン) です。

NOR フラッシュメモリ、16 ビットマルチプレクス I/O

表 46. 16 ビットマルチプレクス I/O NOR フラッシュメモリ

FSMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書き込みイネーブル
NL(= NADV)	O	ラッチイネーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FSMC への NOR 型フラッシュウェイト入力信号

最大容量は 512 M ビットです。

PSRAM/SRAM、非マルチプレクス I/O

表 47. 非マルチプレクス I/O PSRAM/SRAM

FSMC 信号名	I/O	機能
CLK	O	クロック (PSRAM 同期アクセス専用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインーブル
NWE	O	書き込みイネーブル
NL(= NADV)	O	PSRAM 入力の場合のみアドレス有効 (メモリ信号名 : NADV)
NWAIT	I	FSMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出力。バイト 0 およびバイト 1 制御 (上位および下位バイトイネーブル)

最大容量は 512 M ビットです。



PSRAM、16 ビットマルチプレクス I/O

表 48. 16 ビットマルチプレクス I/O PSRAM

FSMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインエーブル
NWE	O	書き込みイネーブル
NL(= NADV)	O	アドレスバリット PSRAM 入力 (メモリ信号名: NADV)
NWAIT	I	FSMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出力。バイト 0 およびバイト 1 制御 (上位および下位バイトイネーブル)

最大容量は、512 M ビット (26 アドレスライン) です。

11.5.2 サポートされるメモリおよびトランザクション

以下の表 49 に、NOR フラッシュメモリ、PSRAM、SRAM に対してメモリデータバスが 16 ビット幅の場合にサポートされているデバイス、アクセスモード、トランザクションの例を示します。FSMC では許容されない (すなわち非サポート) トランザクションは、この例で灰色表示しています。

表 49. NOR フラッシュ/PSRAM : サポートされているメモリおよびトランザクションの例

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	許可/ 禁止	コメント
NOR 型 フラッシュ (マルチプレクス I/O と非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FSMC アクセスに分割
	非同期	W	32	16	Y	2 つの FSMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされていません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-

表 49. NOR フラッシュ/PSRAM : サポートされているメモリおよびトランザクションの例 (続き)

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	許可/ 禁止	コメント
PSRAM (マルチプレクス I/O および 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FSMC アクセスに分割
	非同期	W	32	16	Y	2 つの FSMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされて いません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-
	同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	同期	W	16/32	16	Y	-
SRAM と ROM	非同期	R	8/16	16	Y	-
	非同期	W	8/16	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	32	16	Y	2 つの FSMC アクセスに分割
	非同期	W	32	16	Y	2 つの FSMC アクセスに 分割、バイトレーン NBL[1:0] の使用

11.5.3 一般的なタイミング規則

信号の同期

- すべてのコントローラ出力信号は、内部クロック (HCLK) の立ち上がりエッジで変化します。
- 同期モード (読み出しまたは書き込み) では、すべての出力信号が HCLK の立ち上がりエッジで変化します。CLKDIV の値によらず、すべての信号は次のように変化します。
 - NOEL/NWEL/ NEL/NADVL/ NADVH /NBLL/ アドレスの有効出力は、FSMC_CLK クロックの立ち下がりエッジで変化します。
 - NOEH/ NWEH / NEH/ NOEH/NBLH/ アドレスの無効出力は、FSMC_CLK クロックの立ち上がりエッジで変化します。

11.5.4 NOR フラッシュ /PSRAM コントローラ非同期トランザクション

非同期スタティックメモリ (NOR フラッシュ、PSRAM、SRAM)

- 信号は、内部クロック HCLK によって同期されます。このクロックはメモリには発行されません。
- FSMC は、NOE 信号をネゲートする前に、常にデータをサンプリングします。これにより、メモリのデータホールドタイミングに関する制約が満たされます (チップインーブルハイからデータ遷移までの最小時間は通常 0 ns です)。
- 拡張モードを有効にした場合 (FSMC_BCRx レジスタの EXTMOD ビットをセット)、最大 4 つの拡張モード (A、B、C、D) が使用できます。書き込み動作および読み出し動作で、A、B、C、D モードをミックスすることができます。例えば、読み出し動作をモード A で実行し、書き込み動作をモード B で実行することができます。
- 拡張モードを無効にした場合 (FSMC_BCRx レジスタで EXTMOD ビットをリセット)、FSMC は次のようにモード 1 またはモード 2 で動作することができます。
 - SRAM/PSRAM メモリタイプを選択した場合 (FSMC_BCRx レジスタで MTYP = 0x0 または 0x01)、モード 1 がデフォルトモードです。
 - NOR メモリタイプを選択した場合 (FSMC_BCRx レジスタで MTYP = 0x10)、モード 2 がデフォルトモードです。

モード 1 - SRAM/PSRAM (CRAM)

次の図に、サポートされているモードに対する読み出しトランザクションと書き込みトランザクションを示します。それに続いて、FSMC_BCRx レジスタ、および FSMC_BTRx/FSMC_BWTRx レジスタの必要とされる設定を示します。

図 33. モード 1 読み出しアクセス波形

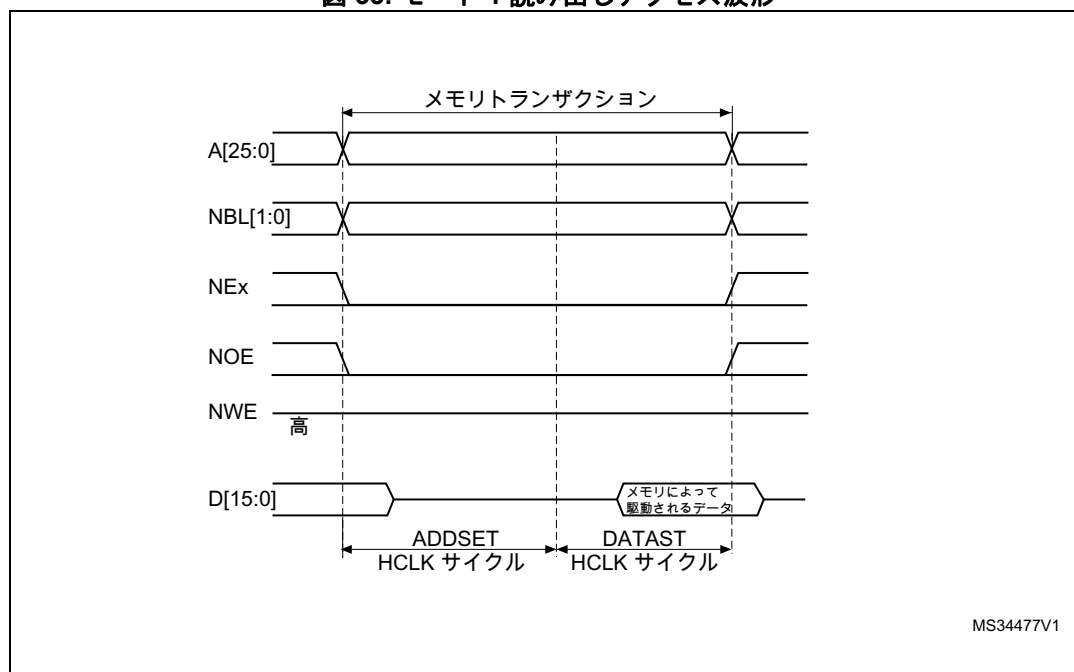
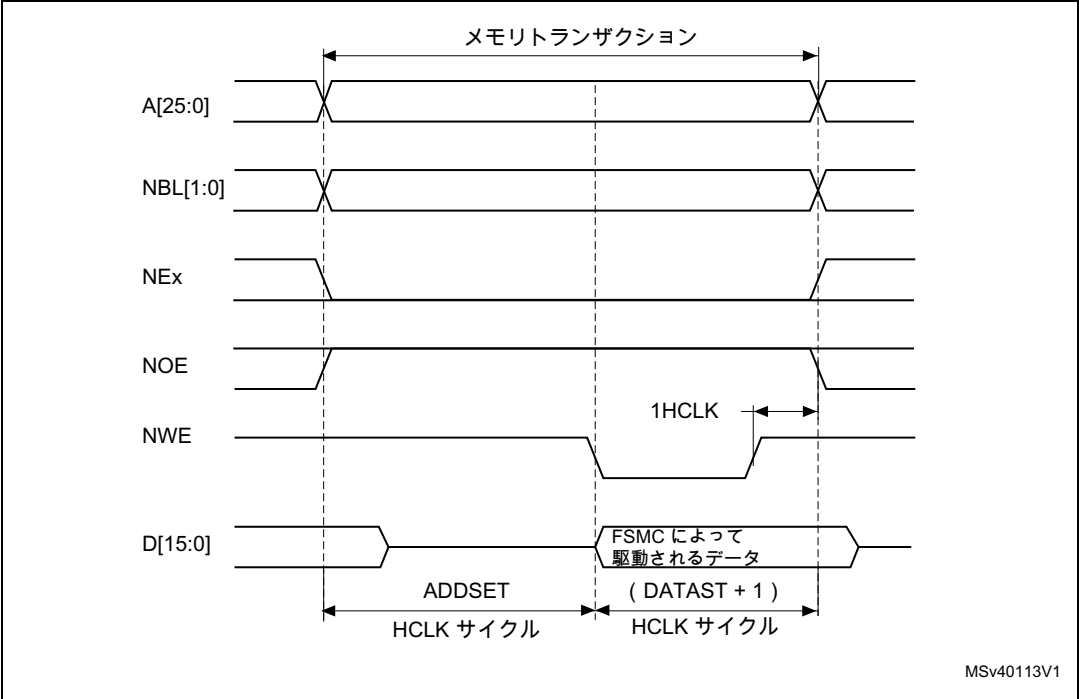


図 34. モード 1 書き込みアクセス波形



書き込みトランザクションの最後の 1 HCLK サイクルは、NWE の立ち上がりエッジ後のアドレスおよびデータホールド時間の保証に役立ちます。この HCLK サイクルがあるため、DATAST の値はゼロより大きくなければなりません (DATAST>0)。

表 50. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	予約済み	0x0
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。

表 50. FSMC_BCRx ビットフィールド (続き)

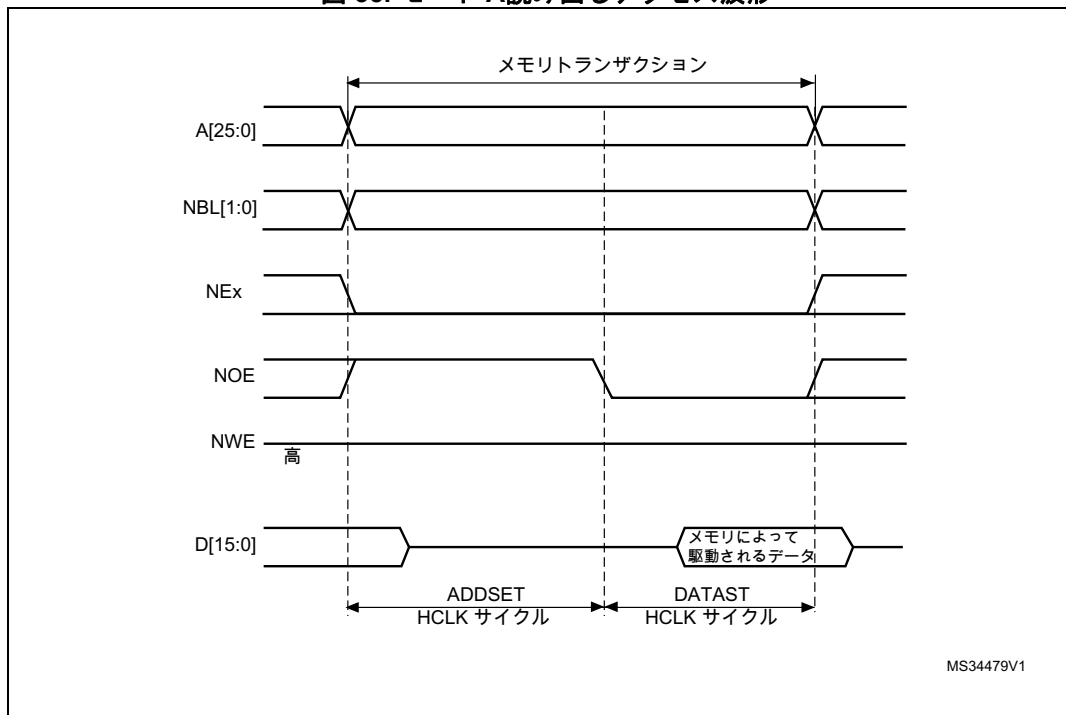
ビット番号	ビット名	設定値
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。
1	MUXE	0x0
0	MBKEN	0x1

表 51. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	無視
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	2 番目のアクセスフェーズの時間 (書き込みアクセスの場合は DATAST + 1 HCLK サイクル、読み出しアクセスの場合は DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

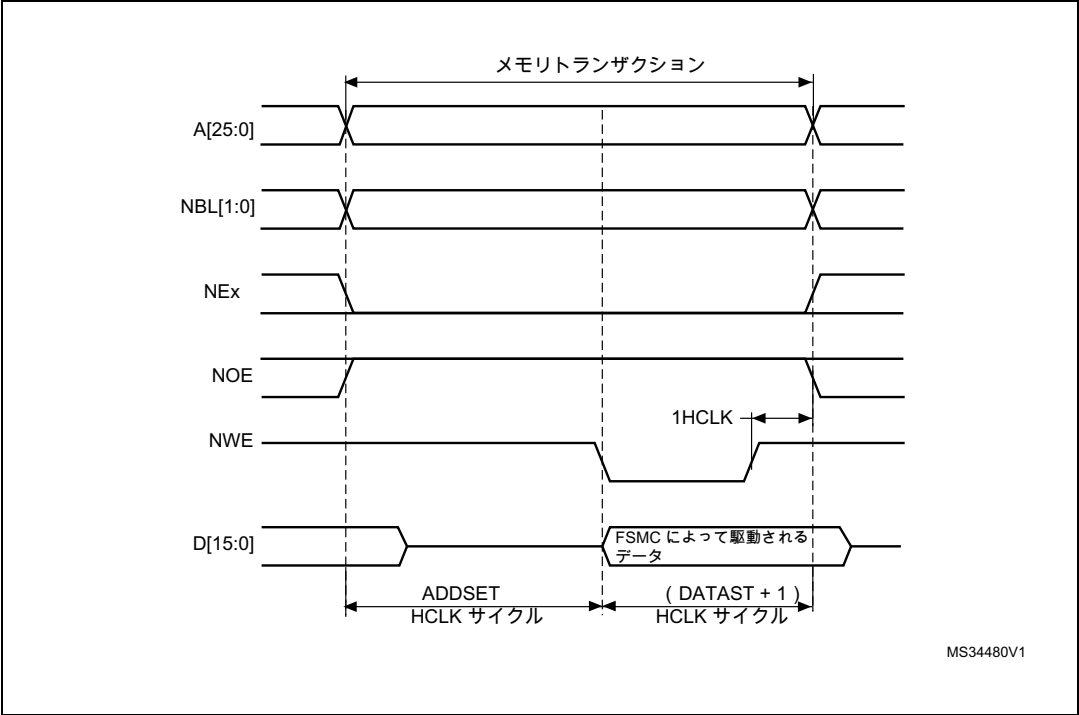
モード A - SRAM/PSRAM (CRAM) OE トグル

図 35. モード A 読み出しアクセス波形



1. NBL[1:0] は読み出しアクセス中ローレベルに駆動されます。

図 36. モード A 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグルリングと、独立した読み出しおよび書き込みのタイミングです。

表 52. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
11	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。

表 52. FSMC_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。
1	MUXEN	0x0
0	MBKEN	0x1

表 53. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

表 54. FSMC_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

モード 2/B - NOR 型フラッシュ

図 37. モード 2 およびモード B 読み出しアクセス波形

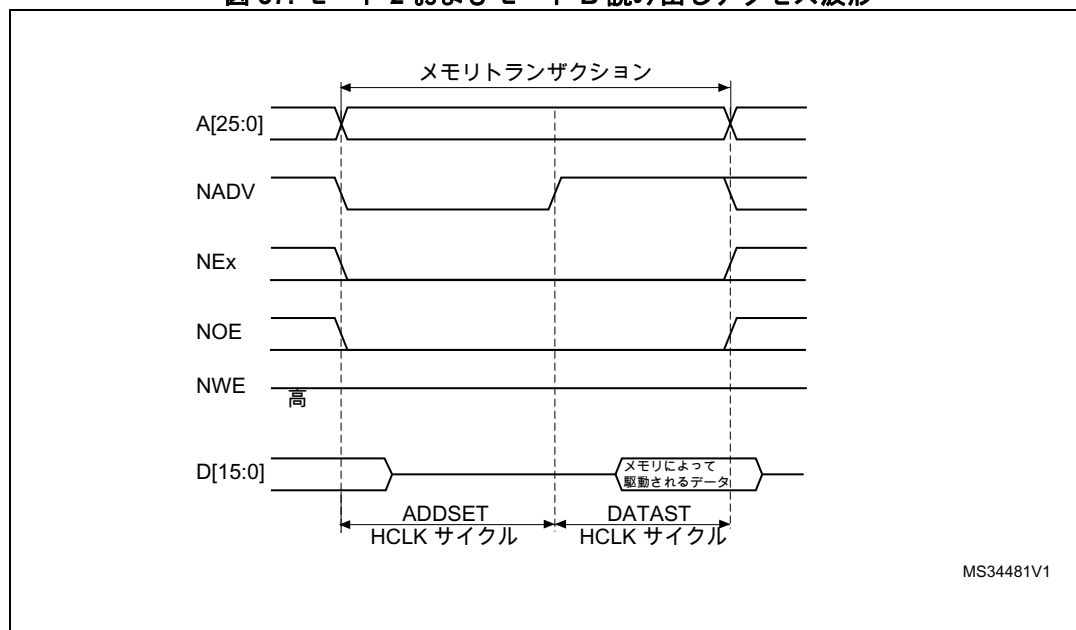
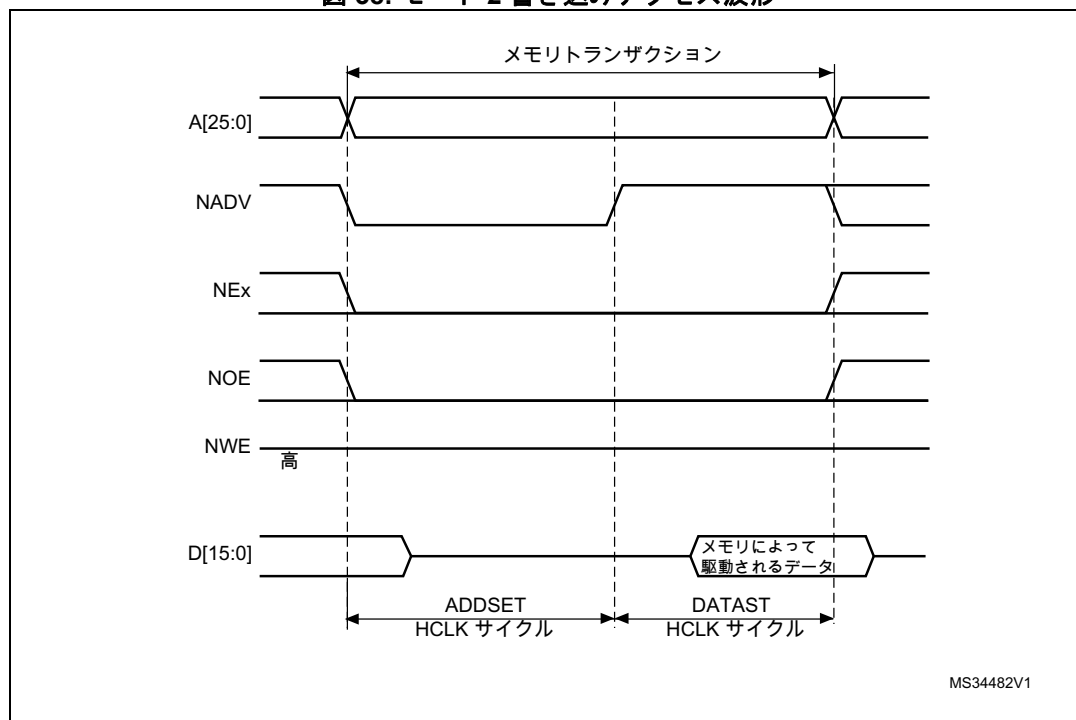
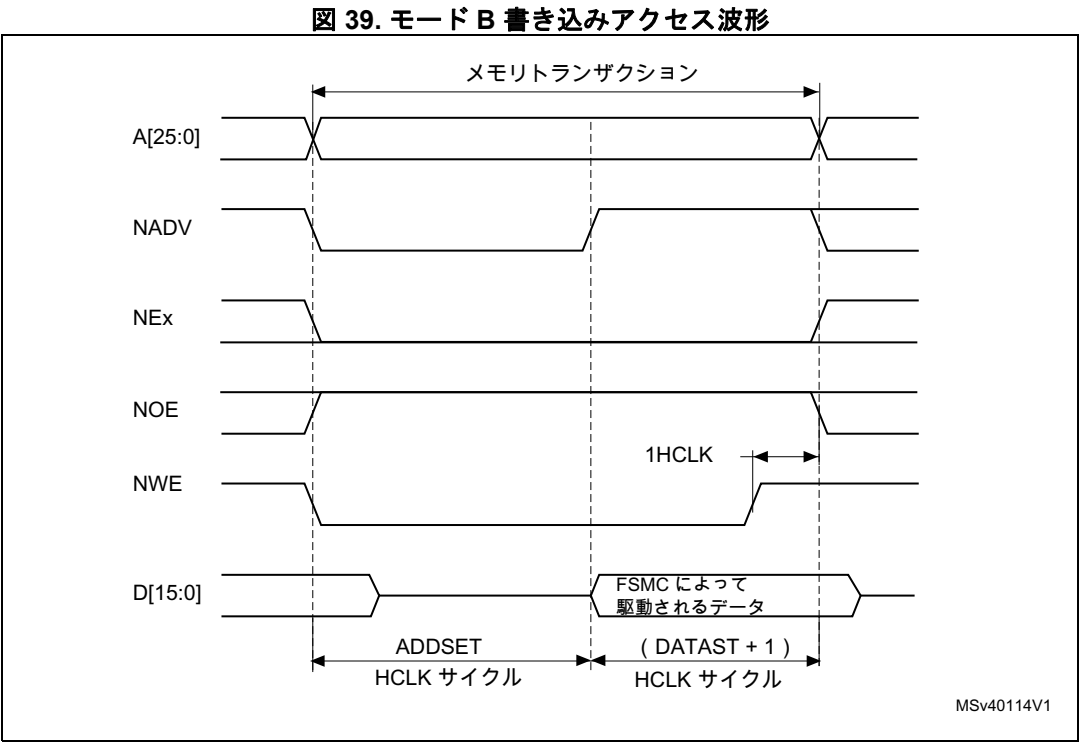


図 38. モード 2 書き込みアクセス波形





モード 1 と比較した場合の相違点は、NWE のトグルリングと拡張モードがセットされたとき（モード B）の独立した読み出しおよび書き込みのタイミングです。

表 55. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0（非同期モードで無効）
18:16	CPSIZE	0x0（非同期モードで無効）
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	モード B では 0x1、モード 2 では 0x0
13	WAITEN	0x0（非同期モードで無効）
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。

表 55. FSMC_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
3-2	MTYP	0x2 (NOR 型フラッシュメモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 56. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 57. FSMC_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

注： *FSMC_BWTRx レジスタは、拡張モードがセットされている場合 (モード B) のみ有効です。それ以外の場合には、その内容は意味を持ちません。*

モード C - NOR フラッシュ - OE トグリング

図 40. モード C 読み出しアクセス波形

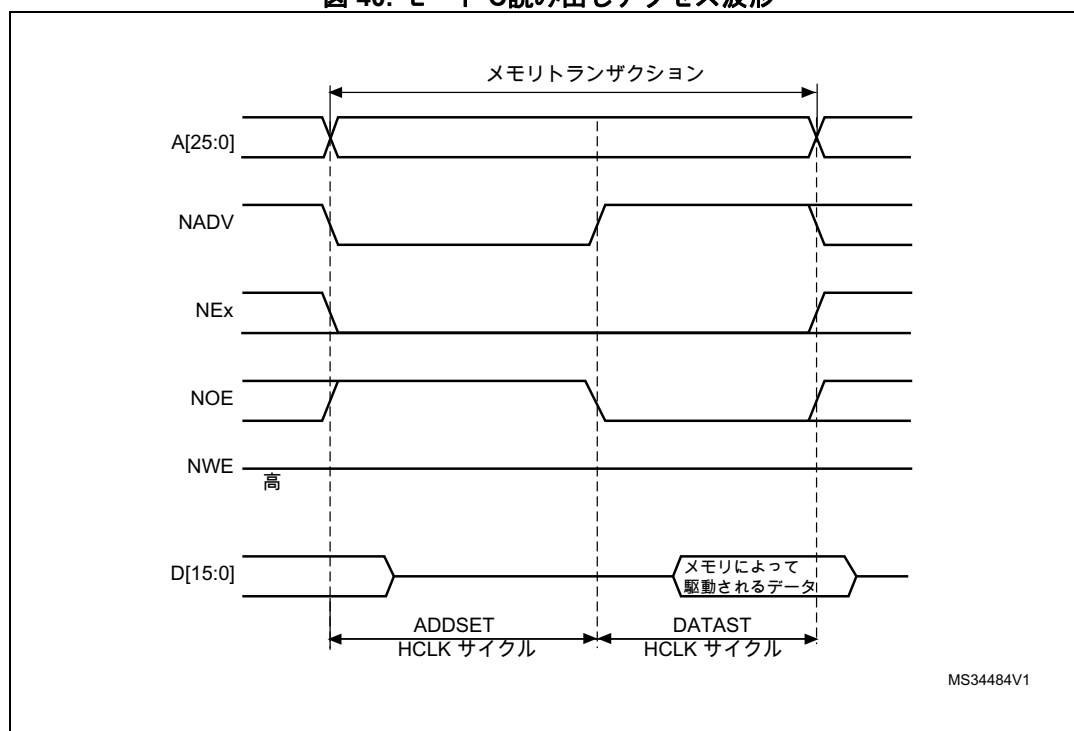
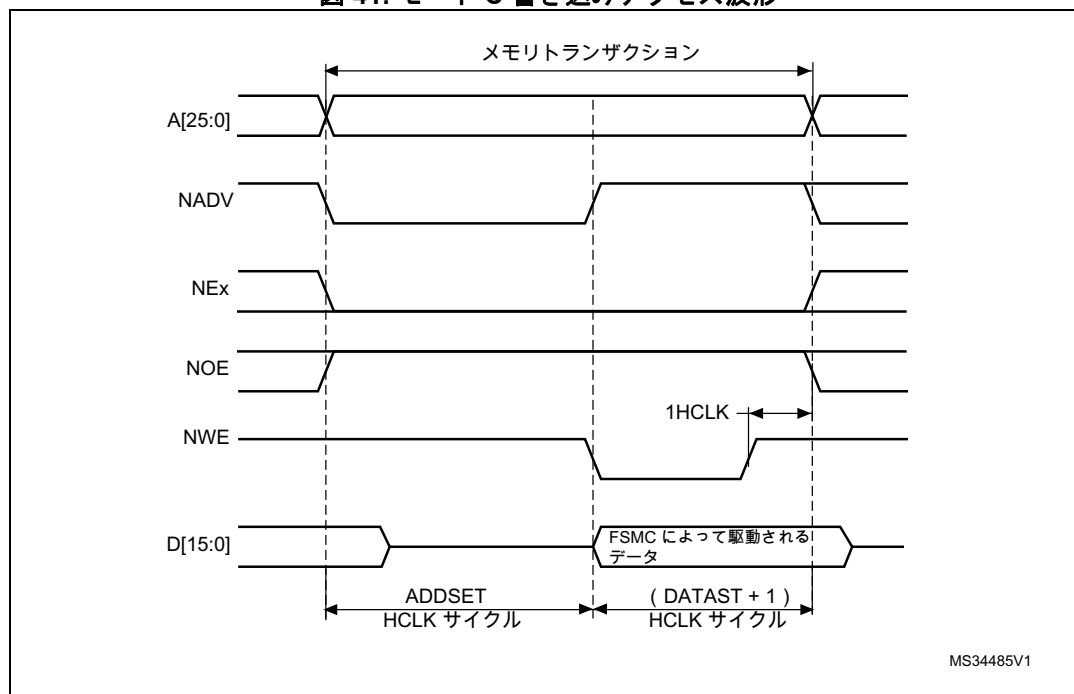


図 41. モード C 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグリングと、独立した読み出しおよび書き込みのタイミングです。

表 58. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCAWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x02 (NOR 型フラッシュメモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 59. FSMC_BTRx ビットフィールド

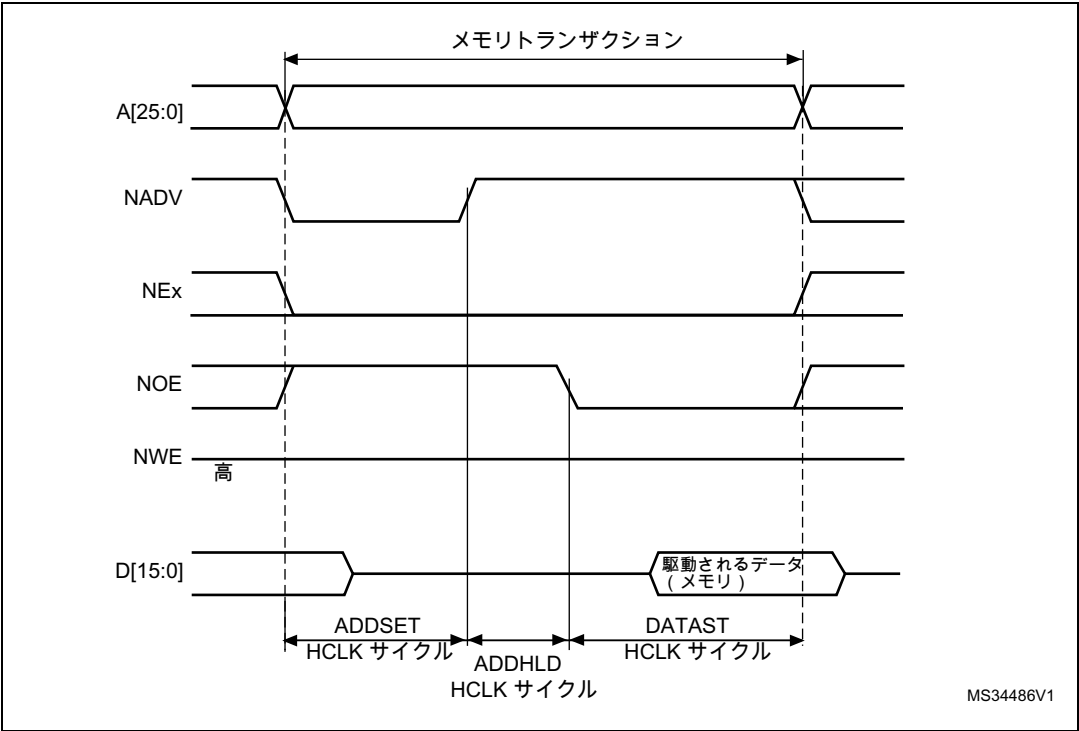
ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

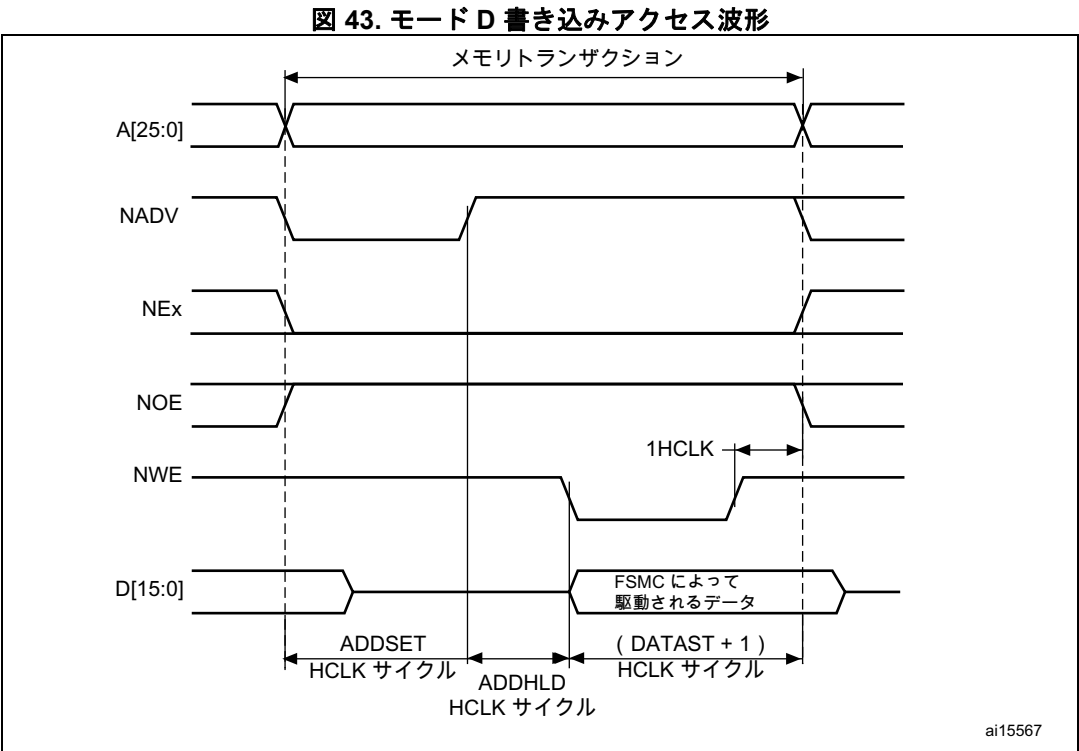
表 60. FSMC_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x2
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15:8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

モード D - 拡張アドレスによる非同期アクセス

図 42. モード D 読み出しアクセス波形





モード 1 と比較した場合の相違点は、NADV が変化した後、NOE のトグルと独立した読み出し書き込みタイミングです。

表 61. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCAWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。

表 61. FSMC_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
3-2	MTYP	必要に応じて設定します。
1	MUXEN	0x0
0	MBKEN	0x1

表 62. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	読み出しにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

表 63. FSMC_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST1 HCLK サイクル)
7-4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

マルチプレクスモード - NOR 型フラッシュメモリに対するマルチプレクス非同期アクセス

図 44. マルチプレクス読み出しアクセス波形

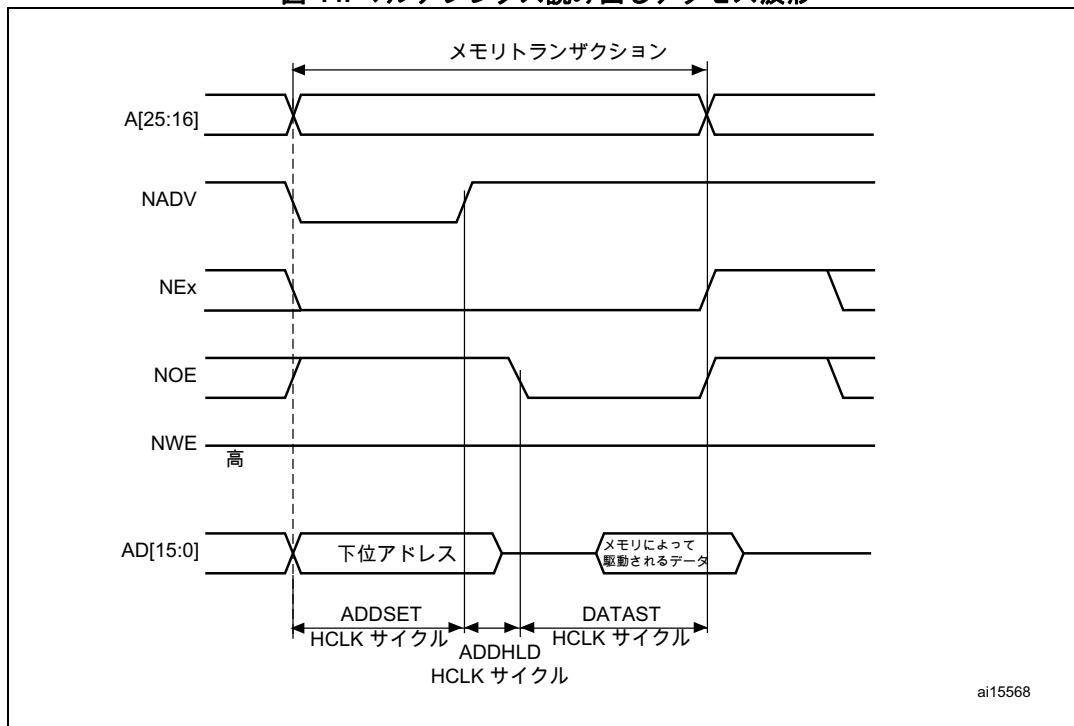
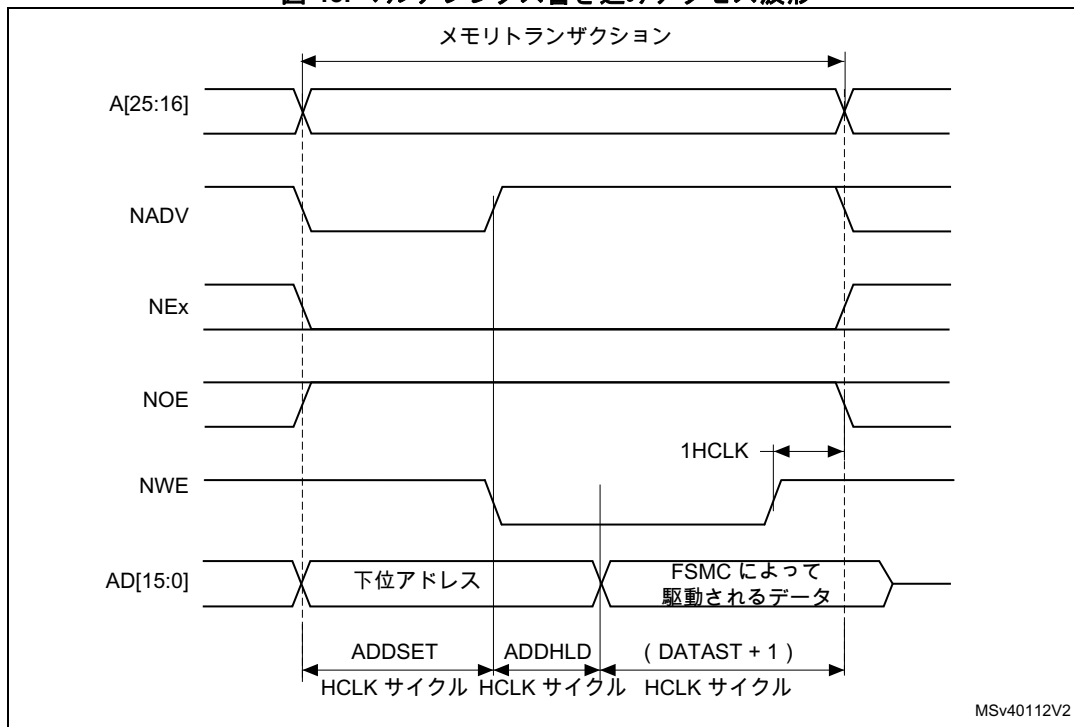


図 45. マルチプレクス書き込みアクセス波形



モード D との相違点は、データバス上の下位アドレスバイトのドライブです。

表 64. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCAWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x2 (NOR 型フラッシュメモリ)
1	MUXEN	0x1
0	MBKEN	0x1

表 65. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	2 番目のアクセスフェーズの時間 (読み出しアクセスの場合は DATAST HCLK サイクル、書き込みアクセスの場合は DATAST+1 HCLK サイクル)。
7-4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)。
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

非同期アクセスにおけるウェイト管理

非同期メモリが、データの受け取りまたは提供の準備がまだできていないことを知らせるために、WAIT 信号をアサートした場合、FSMC_BCRx レジスタの ASYNCWAIT ビットをセットする必要があります。

WAIT 信号がアクティブな場合 (WAITPOL ビットに依存して、ハイまたはロー)、DATAST ビットによってプログラムされる 2 番目のアクセスフェーズ (データセットアップフェーズ) は、WAIT ビットが非アクティブになるまで延長されます。データセットアップフェーズとは異なり、ADDSET および ADDHLD ビットによってプログラムされる最初のアクセスフェーズ (アドレスセットアップおよびアドレスホールドフェーズ) は、WAIT に対応しないので、延長されません。

データセットアップフェーズは、メモリトランザクションが終了する 4 HCLK サイクル前に WAIT が検出されるようにプログラムする必要があります。次のようなケースを考慮する必要があります。

1. メモリが、次のようにトグルする NOE/NWE に合わせた WAIT 信号をアサートする場合。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + \text{max_wait_assertion_time}$$

2. メモリが、NEx (またはトグルしない NOE/NWE) に合わせた WAIT 信号をアサートする場合。
次の条件が成立するならば、

$$\text{max_wait_assertion_time} > \text{address_phase} + \text{hold_phase}$$

以下の通りです。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + (\text{max_wait_assertion_time} - \text{address_phase} - \text{hold_phase})$$

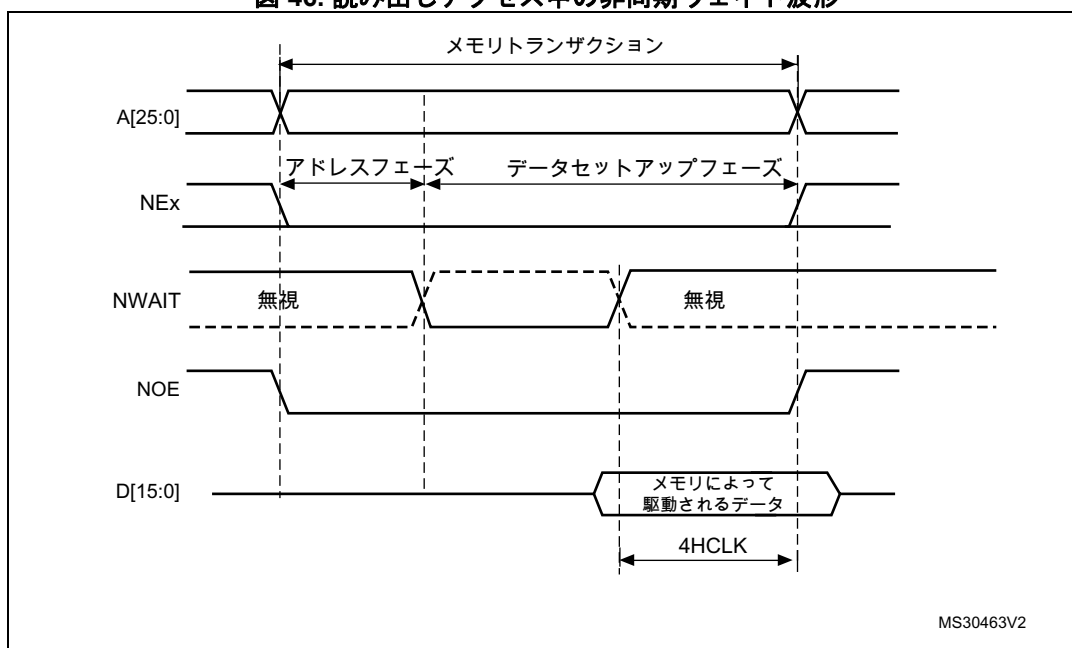
それ以外の場合は、

$$\text{DATAST} \geq 4 \times \text{HCLK}$$

ここで、max_wait_assertion_time は、NEx/NOE/NWE がローになったときにメモリが WAIT 信号をアサートするために必要な最大時間です。

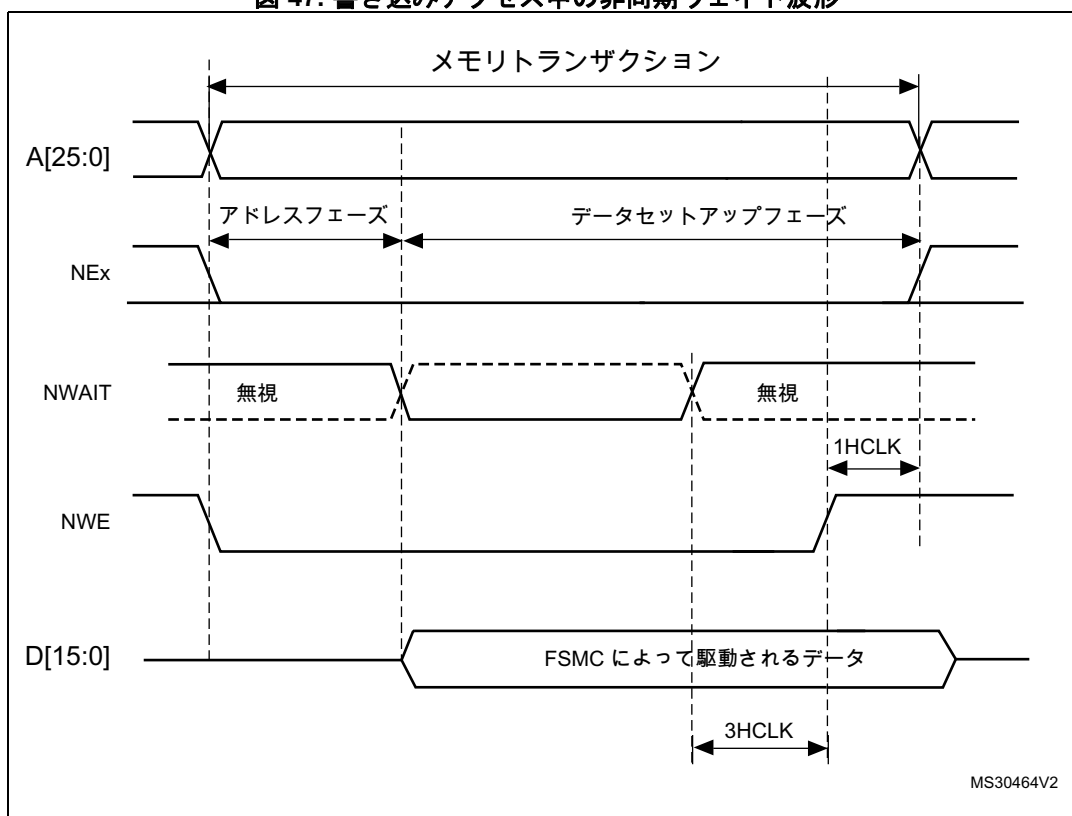
図 46 と 図 47 に、非同期メモリによって WAIT が解除された後、メモリアクセスフェーズに追加される HCLK クロックのサイクル数を示します (上記のケースに無関係)。

図 46. 読み出しアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FSMC_BCRx レジスタの WAITPOL ビット設定値に依存します。

図 47. 書き込みアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FSMC_BCRx レジスタの WAITPOL ビット設定値に依存します。

11.5.5 同期トランザクション

メモリクロック FSMC_CLK は、HCLK の約数です。この値は CLKDIV 値と MWID/ AHB データサイズに依存し、次式で与えられます。

$$\text{FSMC_CLK divider ratio} = \max(\text{CLKDIV} + 1, \text{MWID}(\text{AHB data size}))$$

MWID サイズが16 ビットまたは 8 ビットの場合、FSMC_CLK の分周比は常に CLKDIV のプログラムされた値で決まります。

例：

- CLKDIV = 1、MWID = 16ビット、AHB データサイズ = 8 ビットの場合、FSMC_CLK = HCLK / 2

NOR 型フラッシュメモリは、NADV のアサーションから CLK がハイになるまでの最小時間を指定します。この制約を満たすために、FSMC は、同期アクセスの最初の内部クロックサイクルでは (NADV アサーションの前)、クロックをメモリに供給しません。これにより、メモリクロックの立ち上がりエッジは、NADV ローパルスの中で発生します。

データ遅延と NOR メモリ遅延

データ遅延は、データをサンプリングする前のウェイトサイクル数です。DATLAT の値は、NOR 型フラッシュ設定レジスタで指定された遅延の値と一致しなければなりません。FSMC は、NADV がローのときのクロックサイクルをデータ遅延カウントに含めません。

注意： 一部の NOR フラッシュメモリは、データ遅延カウントに NADV ローサイクルを含めるので、NOR フラッシュの遅延と FSMC DATLAT パラメータの厳密な関係は、次のいずれかです。

- NOR 型フラッシュ遅延 = (DATLAT + 2) CLK クロックサイクル、または
- NOR 型フラッシュ遅延 = (DATLAT + 3) CLK クロックサイクル

最近のメモリの中には、遅延フェーズにおいて NWAIT をアサートするものがあります。そのような場合には、DATLAT を最小値に設定することができます。結果として、FSMC はデータをサンプリングして、データが有効であるかどうか評価するのに十分な時間、ウェイトします。このように、FSMC は、メモリが遅延から抜けたことを検出し、実際のデータが処理されます。

その他のメモリは、遅延中には NWAIT をアサートしません。この場合、FSMC とメモリの両方について、遅延が正しく設定されなければなりません。そうしないと、正しいデータの代わりに無効なデータが取られたり、有効なデータがメモリサイクルの初期フェーズで失われることになります。

シングルバースト転送

選択されたバンクが同期アクセスのバーストモードに設定されている場合、例えば、16 ビットメモリで AHB シングルバーストトランザクションがリクエストされると、FSMC は長さ 1 (AHB 転送が 16 ビットの場合) または 2 (AHB 転送が 32 ビットの場合) のバーストトランザクションを実行して、最後のデータがストローブされると、チップセレクトをデアサートします。

このような転送はサイクル数の点で、非同期読み出し動作に比較して最も効率的ではありません。しかし、時間がかかるメモリアクセスモードの再プログラムには、まず、ランダム非同期アクセスが必要です。

Cellular RAM 1.5 ページ境界跨ぎ

Cellular RAM 1.5 では、ページ境界を跨ぐバーストアクセスは許可されていません。メモリページサイズに従って FSMC_BCR1 レジスタの CPSIZE ビットを設定することで、FSMC コントローラは、メモリページサイズに達した時点で自動的にバーストアクセスを分割できます。



ウェイト管理

同期 NOR フラッシュメモリでは、NWAIT がプログラムされた遅延期間（(DATLAT+2) CLK サイクルに対応）後に評価されます。

NWAIT がアクティブの場合（WAITPOL = 0 ではローレベル、WAITPOL = 1 ではハイレベル）、ウェイトステートはNWAIT が非アクティブ（WAITPOL = 0 ではハイレベル、WAITPOL = 1 ではローレベル）になるまでが挿入されます。

NWAIT が非アクティブのときには、データはただちに（ビット WAITCFG=1）または次のクロックエッジで（ビット WAITCFG=0）、有効とみなされます。

NWAIT 信号によるウェイトステートの挿入時には、コントローラはメモリへのクロックパルスの送信を続け、チップセレクトと出力イネーブル信号を維持します。データは有効と見なしません。

バーストモードでの NOR フラッシュの NWAIT 信号には、2 つのタイミング設定があります。

- フラッシュメモリは、ウェイトステートの 1 データサイクル前に NWAIT 信号をアサートします（リセット後のデフォルト）。
- フラッシュメモリは、ウェイトステートの間に NWAIT 信号をアサートします。

FSMC は FSMC_BCRx レジスタ（x = 0..3）の WAITCFG ビットを使って、各チップセレクトに対する両 NOR フラッシュウェイトステート設定をサポートします。

図 48. 待ち設定波形

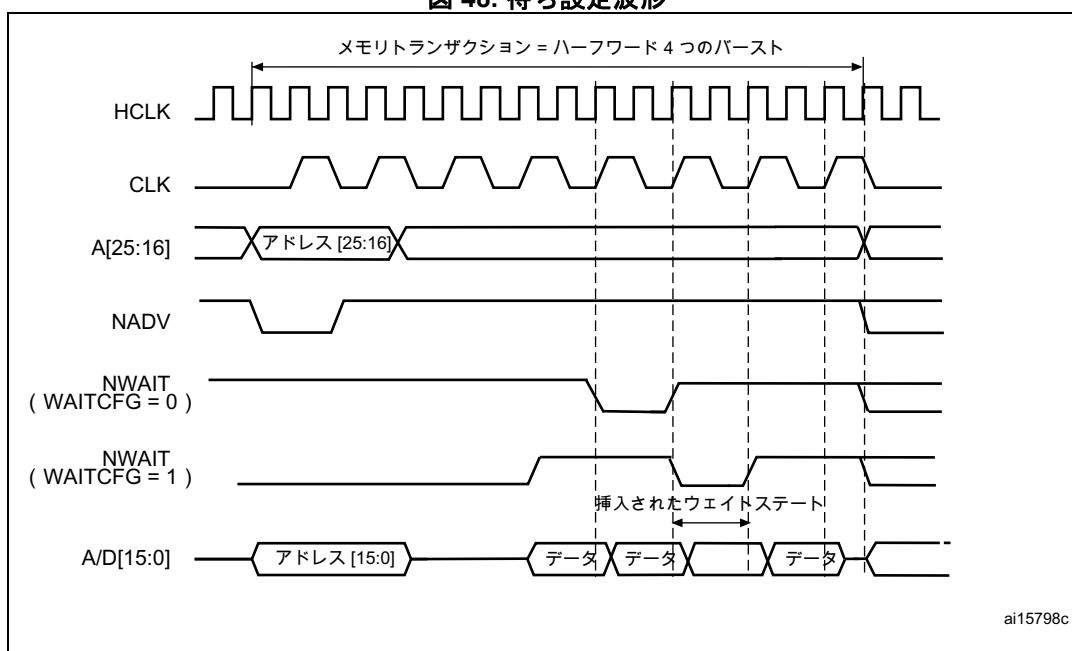
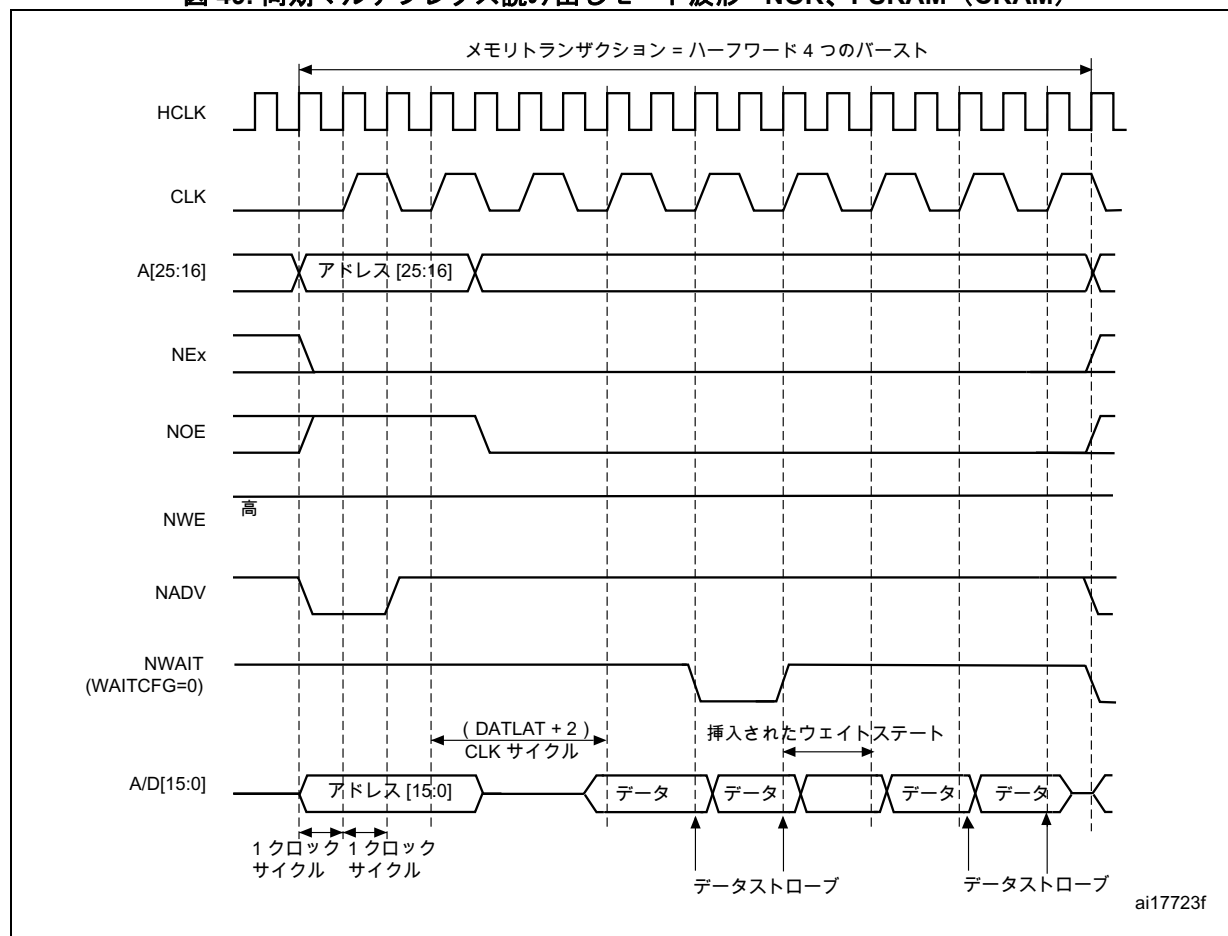


図 49. 同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)



1. バイトレーン出力 (NBL は示されていません。NOR アクセス時にはハイに保たれ、PSRAM (CRAM) アクセス時にはローに保たれます。)

表 66. FSMC BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	同期読み出し時は影響しません。
18:16	CPSIZE	0x0（非同期モードで無効）
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は1にセット、そうでなければ0のままとします。
12	WREN	同期読み出し時は影響しません。
11	WAITCFG	メモリに応じて設定します。
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。

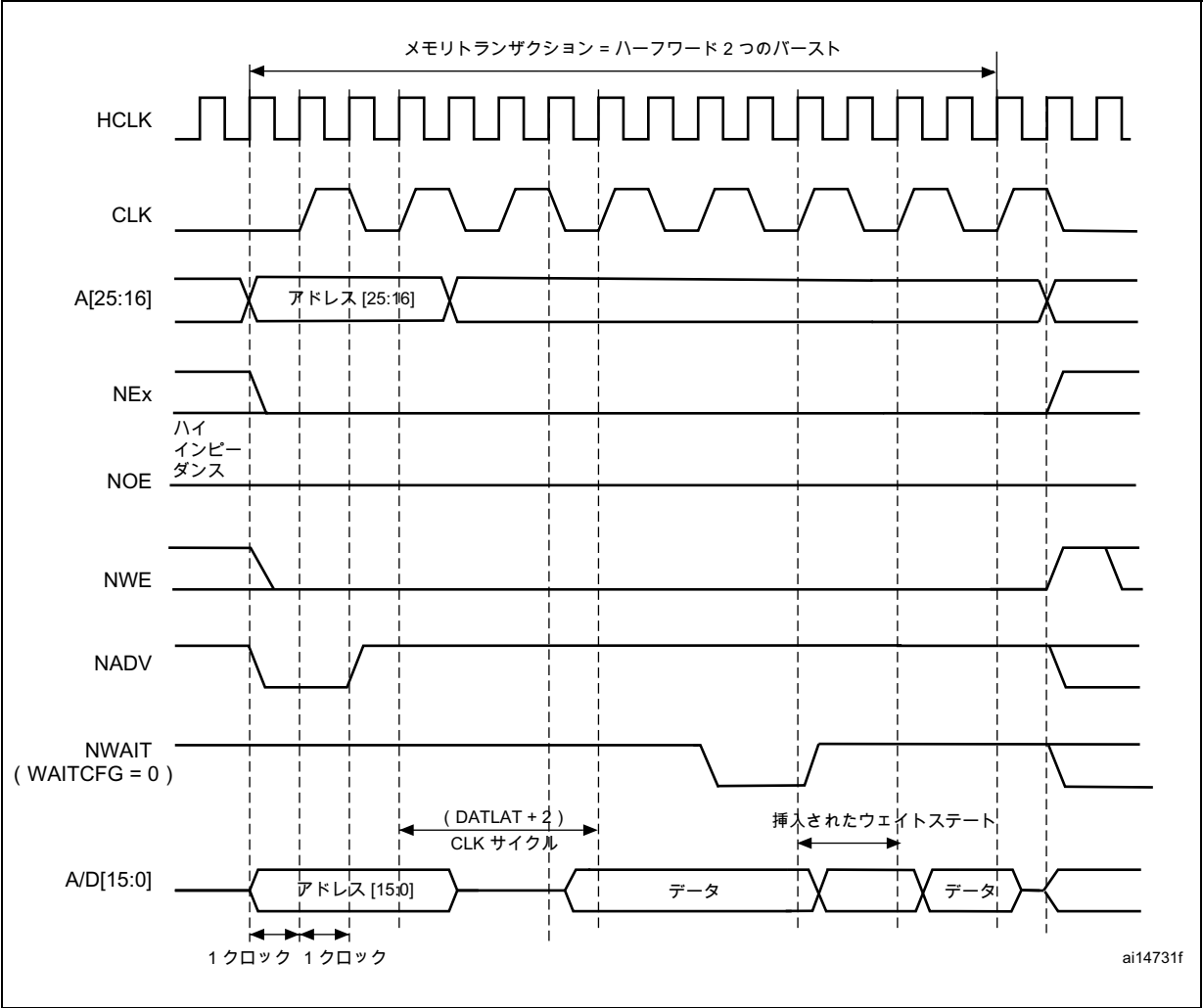
表 66. FSMC_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
8	BURSTEN	0x1
7	予約済み	0x1
6	FACCEN	メモリサポートに従って設定します (NOR 型フラッシュメモリ)
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1 または 0x2
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 67. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	データ遅延
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 の場合、CLK=HCLK となります。 0x1 の場合、CLK=2×HCLK となります。 ..
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視

図 50. 同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)



- メモリは、NWAIT 信号を 1 サイクル前に発行しなければならないので、WAITCFG を 0 にプログラムする必要があります。
- バイトレーン (NBL) 出力は、図に示されておらず、NEx がアクティブな間はローに保たれます。

表 68. FSMC_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x1
18:16	CPSIZE	必要に応じて (CRAM 1.5 は 0x1)
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	0x1

表 68. FSMC_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
11	WAITCFG	0x0
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	同期読み出し時は影響しません。
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 69. FSMC_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 の場合、CLK=HCLK となります。 0x1 の場合、CLK=2×HCLK となります。
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視

11.5.6 NOR/PSRAM コントローラレジスタ

SRAM/NOR 型フラッシュチップセレクト制御レジスタ 1..4 (FSMC_BCR1..4)

アドレスオフセット : $8 * (x - 1)$, $x = 1..4$

リセット値 : バンク 1 は 0x0000 30DB、バンク 2~4 は 0x0000 30D2

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WFDIS	CCLK EN	CBURST RW	CPSIZE[2:0]		
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC WAIT	EXT MOD	WAIT EN	WREN	WAIT CFG	Res.	WAIT POL	BURST EN	Res.	FACC EN	MWID		MTYP[1:0]		MUX EN	MBK EN
rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31 : 22 予約済みであり、リセット値のままにしておかなければなりません。

ビット 21 **WFDIS** : 書き込み FIFO 無効化

このビットは、FSMC コントローラで使用される書き込み FIFO を無効にします。

0 : 書き込み FIFO 有効 (リセット後のデフォルト)

1 : 書き込み FIFO 無効

注 : *FSMC_BCR2..4 レジスタの WFDIS ビットは無視されます。このビットは FSMC_BCR1 レジスタからのみ有効化できます。*

ビット 20 **CCLKEN** : 連続 クロック有効化

このビットは、外部メモリデバイスに対する FSMC_CLK クロック出力を有効化します。

0 : FSMC_CLK は同期メモリアクセス (読み出し/書き込みトランザクション) 時にのみ発生されます。FSMC_CLK クロック比は、FSMC_BCRx レジスタに設定した CLKDIV 値により指定されます (リセット後のデフォルト)。

1 : FSMC_CLK は、非同期アクセスと同期アクセス時に連続的に発生されます。CCLKEN がセットされると、FSMC_CLK クロックがアクティブになります。

注 : *FSMC_BCR2..4 レジスタの CCLKEN ビットは無視されます。このビットは FSMC_BCR1 レジスタからのみ有効化できます。FSMC_CLK 連続クロックが発生するときは、バンク 1 を同期モードに設定する必要があります。*

注 : *CCLKEN ビットがセットされている場合、FSMC_CLK クロック比は FSMC_BTR1 レジスタの CLKDIV 値で指定されます。FSMC_BWTR1 の CLKDIV は、無視されます。*

注 : *同期モードを使用し、かつ CCLKEN ビットがセットされる場合、バンク 1 以外のバンクに接続される同期メモリはすべて、同じクロックで駆動されます (他のバンクに対する FSMC_BTR2..4 レジスタと FSMC_BWTR2..4 レジスタの CLKDIV 値は無効です)。*

ビット 19 **CBURSTRW** : 書き込みバースト有効化

バーストモードでの PSRAM (CRAM) 動作では、このビットは書き込み動作時に同期アクセスを可能にします。同期読み出しアクセスのイネーブルビットは、FSMC_BCRx レジスタの BURSTEN ビットです。

0 : 書き込み動作は、常に非同期モードで実行されます。

1 : 書き込み動作は、同期モードで実行されます。

ビット 18:16 CPSIZE[2:0] : CRAM ページサイズ

これらのビットは、ページ間でアドレス境界を跨ぐバーストアクセスができないCellular RAM 1.5 に使用されます。これらのビットを設定すると、FSMC コントローラは、メモリページサイズに達した時点で自動的にバーストアクセスを分割します（ページサイズについてはメモリのデータシートを参照）。

000 : ページ境界を跨ぐ場合、バーストは分割されません（リセット後のデフォルト）

001 : 128 バイト

010 : 256 バイト

011 : 512 バイト

100 : 1024 バイト

その他 : 予約済み

ビット 15 ASYNCWAIT : 非同期転送中のウェイト信号

このビットは、非同期プロトコル時でも、FSMC が NWAIT 信号を有効化／無効化できるようにします。

0 : 非同期プロトコル動作中は、NWAIT 信号は考慮されません（リセット後のデフォルト）。

1 : 非同期プロトコル動作中に NWAIT 信号が考慮されます。

ビット 14 EXTMOD : 拡張モード有効化

このビットを使うと、FSMC が FSMC_BWTR レジスタで非マルチプレクス非同期アクセスの書き込みタイミングをプログラムできるため、読み出しと書き込みで異なるタイミングが可能になります。

0 : FSMC_BWTR レジスタ内部の値は考慮されません（リセット後のデフォルト）。

1 : FSMC_BWTR レジスタ内部の値が考慮されます。

注 : 拡張モードを無効化した場合、FSMC は次のようにモード1 またはモード2 で動作することができます。

- － モード1は、SRAM/PSRAM メモリタイプを選択した場合（MTYP = 0x0 または 0x01）のデフォルトモードです。
- － モード2は、NOR メモリタイプを選択した場合（MTYP = 0x10）のデフォルトモードです。

ビット 13 WAITEN : ウェイトイネーブルビット

このビットは、同期モードでフラッシュメモリをアクセスするとき NWAIT 信号を使用したウェイトステートの挿入を有効化／無効化します。

0 : NWAIT 信号は無効です（レベルは考慮されず、プログラムされたフラッシュ遅延時間後にウェイトステートは挿入されません）。

1 : NWAIT 信号は有効です（プログラムされた遅延時間後にレベルが考慮され、アサートされた場合に、ウェイトステートが挿入されます）（リセット後のデフォルト）。

ビット 12 WREN : 書き込みイネーブルビット

このビットは、FSMC によるバンクへの書き込み動作の有効／無効を示します。

0 : FSMC によるバンクへの書き込み動作は無効にされています。AHB エラーが報告されます。

1 : FSMC によるバンクへの書き込み動作は有効にされています（リセット後のデフォルト）。

ビット 11 WAITCFG : ウェイトタイミング設定

NWAIT 信号は、メモリからのデータが有効か、または同期モードでフラッシュメモリをアクセスするときウェイトステートを挿入すべきかを表示します。この設定ビットは、ウェイトステートの1クロック前、またはウェイトステート中に、メモリによって NWAIT がアサートされるかどうかを定義します。

0 : NWAIT 信号は、ウェイトステートの1データサイクル前にアクティブです（リセット後のデフォルト）。

1 : NWAIT 信号は、ウェイトステートの間アクティブです（PSRAM には不使用）。

ビット 10 予約済みであり、リセット値のままにしておかなければなりません。**ビット 9 WAITPOL : ウェイト信号極性ビット**

同期モードまたは非同期モードで使用されるメモリからのウェイト信号の極性を定義します。

0 : NWAIT はアクティブラウです（リセット後のデフォルト）。

1 : NWAIT はアクティブハイです。

ビット 8 BURSTEN : パーストイネーブルビット

このビットは、読み出し動作での同期アクセスを有効化/無効化します。これは、パーストモードでの同期メモリ動作に対してのみ有効です。

0 : パーストアクセスモードは無効です (リセット後のデフォルト)。読み出しアクセスは非同期モードで実行されます。

1 : パーストモードを有効化。読み出しアクセスは同期モードで実行されます。

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6 FACCEN : フラッシュアクセス有効化

NOR 型フラッシュメモリアクセス動作を有効にします。

0 : 対応する NOR 型フラッシュメモリアクセスは無効です。

1 : 対応する NOR フラッシュメモリアクセスは有効です (リセット後のデフォルト)。

ビット 5:4 MWID[1:0] : メモリデータバス幅。

外部メモリデバイスの幅を定義します。すべてのタイプのメモリに対して有効です。

00 : 8 ビット

01 : 16 ビット (リセット後のデフォルト)

10 : 予約済み

11 : 予約済み

ビット 3:2 MTYP[1:0] : メモリタイプ

対応するメモリバンクに接続される外部メモリのタイプを定義します。

00 : SRAM (バンク 2...4 のリセット後のデフォルト)

01 : PSRAM (CRAM)

10 : NOR 型フラッシュ / OneNAND フラッシュ (バンク 1 のリセット後のデフォルト)

11 : 予約済み

ビット 1 MUXEN : アドレス / データマルチプレクシングイネーブルビット

このビットがセットされているときには、データバス上でアドレスとデータがマルチプレクスされます。NOR および PSRAM メモリでのみ有効です。

0 : アドレス/データはマルチプレクスされません。

1 : アドレス/データはデータバス上でマルチプレクスされます (リセット後のデフォルト)。

ビット 0 MBKEN : メモリバンクイネーブルビット。

メモリバンクを有効にします。リセット後は、バンク 1 は有効であり、その他はすべて無効です。無効なバンクにアクセスすると、AHB バス上で ERROR が発生します。

0 : 対応するメモリバンクは無効です

1 : 対応するメモリバンクは有効です。

SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1..4 (FSMC_BTR1..4)

アドレスオフセット : 0x04 + 8 * (x - 1)、x = 1..4

リセット値 : 0x0FFF FFFF

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。FSMC_BCRx レジスタの EXTMOD ビットがセットされている場合、このレジスタは、書き込みと読み出しアクセスに分割されます。すなわち、読み出しアクセスを設定するためのレジスタ（このレジスタ）と、書き込みアクセスを設定するためのレジスタ（FSMC_BWTRx レジスタ）の 2 つのレジスタが使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ACCMOD		DATLAT				CLKDIV				BUSTURN			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST								ADDHLD				ADDSET			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値のままにしておかなければなりません。

ビット 29:28 **ACCMOD[1:0]** : アクセスモード

タイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FSMC_BCRx レジスタの EXTMOD ビットが 1 のときのみ考慮されます。

- 00 : アクセスモード A
- 01 : アクセスモード B
- 10 : アクセスモード C
- 11 : アクセスモード D

ビット 27:24 **DATLAT[3:0]** : (下のビット説明参照) 同期メモリのデータ遅延

読み出し／書き込みバーストモードを有効にした同期 アクセスの場合 (BURSTEN / CBURSTRW ビットをセット)、最初のデータの読み書きの前にメモリに発行するメモリクロックサイクル数 (+2) を定義します。

このタイミングパラメータは、HCLK 周期ではなく、FSMC_CLK 周期で表されます。

非同期アクセスでは、この値は無視されます。

0000 : 最初のバーストアクセスでは、2 CLK クロックサイクルのデータ遅延

1111 : 最初のバーストアクセスでは、17 CLK クロックサイクルのデータ遅延 (リセット後のデフォルト)

ビット 23:20 **CLKDIV[3:0]** : クロック分周比 (FSMC_CLK 信号)

FSMC_CLK クロック出力信号の周期を定義します。HCLK サイクル数で表されます。

0000 : FSMC_CLK 周期 = 1 × HCLK 周期

0001 : FSMC_CLK 周期 = 2 × HCLK 周期

0010 : FSMC_CLK 周期 = 3 × HCLK 周期

1111 : FSMC_CLK 周期 = 16 × HCLK 周期 (リセット後のデフォルト値)

非同期 NOR 型フラッシュ、SRAM、または PSRAM アクセスでは、この値は無視されます。

注 : **FSMC_CLK 分周比の式については、セクション 11.5.5: 同期トランザクションを参照してください。**



ビット 19:16 BUSTURN[3:0] : バスターンアラウンドフェーズ時間

これらのビットは、書き込みから読み出しまで（および読み出しから書き込みまで）のトランザクションの終わりに遅延を追加するためにソフトウェアで書き込まれます。この遅延を使うと、連続トランザクション間の最小時間（NEx ハイから NEx ローまでの tEH_{EL}）と、読み出しアクセス後にメモリがデータバスを開放するために要する最大時間（tEH_{QZ}）を次のように一致させることができます。プログラムされたバスターンアラウンド遅延が非同期読み出し（マルチプレクスまたはモード D）または書き込みトランザクションと、スタティックバンクへのその他の同期／非同期読み出しまたは書き込みとの間に挿入されます。このバンクは読み出しの場合、同じまたは異なるものに、書き込みの場合、マルチプレクスまたはモード D 以外で異なるものに設定できます。

一部のケースでは、プログラムされた BUSTURN の値にかかわらず、バスターンアラウンド遅延は固定です。

次の場合があてはまります。

- バスターンアラウンド遅延は、マルチプレクスまたはモード D 以外の同じスタティックメモリバンクへの 2 つの連続した非同期書き込み転送の間には挿入されません。
- 次の間隔で 1 FSMC クロックサイクルのバスターンアラウンド遅延が発生します。
 - マルチプレクスまたはモード D 以外の同じスタティックメモリバンクへの 2 つの連続した非同期読み出し転送
 - マルチプレクスまたはモード D 以外のすべてのスタティックバンクまたはダイナミックバンクへの非同期／同期書き込みに対する非同期読み出し
 - 非同期（モード 1、2、A、B、C）読み出しと別のスタティックバンクからの読み出し
- 次の間隔で 2 FSMC クロックサイクルのバスターンアラウンド遅延が発生します。
 - 同じバンクに対する 2 つの連続した同期書き込み（バーストまたはシングル）
 - 同期書き込み（バーストまたはシングル）アクセスと、スタティックメモリバンクへの非同期書き込み／読み出し転送（読み出しの場合、バンクを同じものまたは異なるものに設定できます）
 - 2 つの連続した同期読み出し（バーストまたはシングル）、それに続く別のスタティックメモリバンクへのすべての同期／非同期読み出しまたは書き込み
- 次の間隔で 3 FSMC クロックサイクルのバスターンアラウンド遅延が発生します。
 - 異なるスタティックバンクに対する 2 つの連続した同期書き込み（バーストまたはシングル）
 - 同期書き込み（バーストまたはシングル）アクセスと、同じまたは異なるバンクからの同期読み出し

0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル

...

1111 : BUSTURN フェーズ時間 = 15 × HCLK クロックサイクル追加（リセット後のデフォルト値）

ビット 15:8 DATAST[7:0] : データフェーズ時間

これらのビットは、非同期アクセスで使用するデータフェーズの時間を定義するためにソフトウェアにより書き込まれます（図 33 ~ 図 45 を参照）。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 × HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 × HCLK クロックサイクル

...

1111 1111 : DATAST フェーズ時間 = 255 × HCLK クロックサイクル（リセット後のデフォルト値）

各メモリタイプおよびアクセスモードでのデータフェーズ時間については、それぞれの図（図 33 ~ 図 45）を参照してください。

例：モード 1、書き込みアクセス、DATAST=1：データフェーズ時間 = DATAST+1 = 2 HCLK クロックサイクル

注： 同期アクセスでは、この値は無視されます。

ビット 7:4 ADDHLD[3:0] : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアから書き込まれ (図 33 ~ 図 45を参照)、モード D またはマルチプレクスアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 * HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 * HCLK クロックサイクル

...

1111 : ADDHLD フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスホールドフェーズ時間については、それぞれの図 ((図 33 ~ 図 45) を参照してください。

注 : 同期アクセスでは、この値は使用されず、アドレスホールドフェーズは、常に 1 メモリクロック周期の長さです。

ビット 3:0 ADDSET[3:0] : アドレスセットアップフェーズ時間

これらのビットは、アドレスセットアップフェーズの時間を定義するために、ソフトウェアによって書き込まれ (図 33 ~ 図 45を参照)、SRAM、ROM、非同期 NOR フラッシュ、および PSRAM で使用されます。

0000 : ADDSET フェーズ時間 = 0 * HCLK クロックサイクル

...

1111 : ADDSET フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスセットアップフェーズ時間については、それぞれの図 ((図 33 ~ 図 45) を参照してください。

注 : 同期アクセスでは、この値は無視されます。

マルチプレクスモードとモード D では、ADDSET の最小値は 1 です。

注 : PSRAM (CRAM) は内部リフレッシュがあるため遅延 (レイテンシ) が可変になっています。したがって、これらのメモリは必要に応じてレイテンシ期間を延長するために、レイテンシフェーズ全体に NWAIT 信号を発行します。

PSRAM (CRAM) では、フィールド DATLAT を 0 に設定しなければなりません。こうすると、FSMC は、すぐにレイテンシフェーズを終了して、メモリからの NWAIT 信号のサンプリングを開始し、メモリがレディ状態になると、読み出しまたは書き込みを開始します。

この方法は、古い世代のフラッシュメモリと異なり、NAWAIT 信号を発行する最新世代の同期フラッシュメモリでも使用できます (使用中の特定のフラッシュメモリのデータシートを参照してください)。

SRAM/NOR 型フラッシュ書き込みタイミングレジスタ 1..4 (FSMC_BWTR1..4)

アドレスオフセット : $0x104 + 8 * (x - 1)$ 、 $x = 1..4$

リセット値 : 0x0FFF FFFF

このレジスタには各メモリバンクの制御情報が格納されます。SRAM、PSRAM、NOR フラッシュメモリに使用されます。FSMC_BCRx レジスタの EXTMOD ビットがセットされているときには、このレジスタは書き込みアクセスについてアクティブになります。

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

ビット 31:30 予約済みであり、リセット値のままにしておかなければなりません。



ビット 29:28 **ACCMOD[1:0]** : アクセスモード。

次のタイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FSMC_BCRx レジスタの EXTMOD ビットが 1 にセットされている場合のみ考慮されます。

00 : アクセスモード A

01 : アクセスモード B

10 : アクセスモード C

11 : アクセスモード D

ビット 27:20 予約済みであり、リセット値のままにしておかなければなりません。

ビット 19:16 **BUSTURN[3:0]** : バスターンアラウンドフェーズ時間

プログラムされたバスターンアラウンド遅延が非同期書き込み転送と、スタティックバンクへのその他の同期／非同期読み出しまたは書き込みとの間に挿入されます。このバンクは読み出しの場合、同じまたは異なるものに、書き込みの場合、マルチプレクスまたはモード D 以外で異なるものに設定できます。

一部のケースでは、プログラムされた BUSTURN の値にかかわらず、バスターンアラウンド遅延は固定です。次の場合が当てはまります。

- バスターンアラウンド遅延は、マルチプレクスまたはモード D 以外の同じスタティックメモリバンクへの 2 つの連続した非同期書き込み転送の間には挿入されません。
- 次の間隔で 2 FSMC クロックサイクルのバスターンアラウンド遅延が発生します。
 - 同じバンクに対する 2 つの連続した同期書き込み（バーストまたはシングル）
 - 同期書き込み（バーストまたはシングル）転送と、スタティックメモリバンクへの非同期書き込み／読み出し転送
- 次の間隔で 3 FSMC クロックサイクルのバスターンアラウンド遅延が発生します。
 - 異なるスタティックバンクに対する 2 つの連続した同期書き込み（バーストまたはシングル）
 - 同期書き込み（バーストまたはシングル）転送と、同じまたは異なるバンクからの同期読み出し

0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル

...

1111 : BUSTURN フェーズ時間 = 15 HCLK クロックサイクル追加（リセット後のデフォルト値）

ビット 15:8 **DATAST[7:0]** : データフェーズ時間

これらのビットは、非同期 SRAM、PSRAM、NOR フラッシュメモリアクセスで使用するデータフェーズの時間を定義するために、ソフトウェアによって書き込まれます（図 33 ~ 図 45 を参照）。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 * HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 * HCLK クロックサイクル

...

1111 1111 : DATAST フェーズ時間 = 255 * HCLK クロックサイクル（リセット後のデフォルト値）

ビット 7:4 **ADDHLD[3:0]** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアによって書き込まれ（図 42 ~ 図 45 を参照）、非同期マルチプレクスアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 * HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 * HCLK クロックサイクル

...

1111 : ADDHLD フェーズ時間 = 15 * HCLK クロックサイクル（リセット後のデフォルト値）

注： 同期 NOR 型フラッシュアクセスでは、この値は使用されず、アドレスホールドフェーズは常に 1 フラッシュクロック周期の長さです。

ビット 3:0 **ADDSET[3:0]** : アドレスセットアップフェーズ時間。

これらのビットは、アドレスセットアップフェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ (図 33 ~ 図 45 を参照)、非同期アクセスで使用されます。

0000 : ADDSET フェーズ時間 = 0 * HCLK クロックサイクル

...

1111 : ADDSET フェーズ時間 = 15 * HCLK クロックサイクル (リセット後のデフォルト値)

注 : 同期アクセスでは、この値は使用されず、アドレスセットアップフェーズは常に 1 フラッシュクロック周期の長さです。マルチプレクスモードでは、ADDSET の最小値は 1 です。

11.6 FSMC レジスタマップ

次の表に FSMC レジスタの一覧を示します。

表 70. FSMC レジスタマップ

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FSMC_BCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WFDIS	CCLKEN	CBURSTW	CPSIZE [2:0]			ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	リセット値											0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1
0x08	FSMC_BCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	リセット値													0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0
0x10	FSMC_BCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	リセット値													0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0
0x18	FSMC_BCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTW	CPSIZE [2:0]			ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	リセット値													0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0
0x04	FSMC_BTR1	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]			CLKDIV[3:0]			BUSTURN[3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]								
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	FSMC_BTR2	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]			CLKDIV[3:0]			BUSTURN[3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]								
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	FSMC_BTR3	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]			CLKDIV[3:0]			BUSTURN[3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]								
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	FSMC_BTR4	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]			CLKDIV[3:0]			BUSTURN[3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]								
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x104	FSMC_BWTR1	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]			DATAST[7:0]							ADDHLD[3:0]			ADDSET[3:0]						
	リセット値			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 70. FSMC レジスタマップ (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10C	FSMC_BWTR2	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]							ADDHLD[3:0]				ADDSET[3:0]				
	リセット値			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x114	FSMC_BWTR3	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]							ADDHLD[3:0]				ADDSET[3:0]				
	リセット値			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x11C	FSMC_BWTR4	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]							ADDHLD[3:0]				ADDSET[3:0]				
	リセット値			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。



12 Quad SPI インタフェース (QUADSPI)

12.1 概要

QUADSPI は、シングル、デュアル、またはQuad SPI フラッシュメモリを対象とする特殊な通信インタフェースです。このインタフェースは、次の 3 つのモードのいずれかで動作できます。

- インダイレクトモード：すべての動作は QUADSPI レジスタを使用して実行されます。
- ステータスポーリングモード：外部フラッシュメモリステータスレジスタが定期的に読み出され、フラグのセット時に割り込みを生成することができます。
- メモリマップドモード：外部フラッシュメモリはマイクロコントローラのアドレス空間に配置され、システムによって内部メモリであるかのようにみなされます。

デュアルフラッシュモードを使用すると、スループットと容量の両方を 2 倍に増やすことができます。このモードでは、2 つのQuad SPI フラッシュメモリに同時にアクセスします。

12.2 QUADSPI の主な機能

- 3 つの機能モード：インダイレクト、ステータスポーリング、メモリマップド
- 2 つのフラッシュメモリに並列にアクセスすることで 8 ビットの同時送受信を可能にするデュアルフラッシュモード
- SDR および DDR サポート
- インダイレクトモードとメモリマップドモードの両方で完全にプログラム可能な OP コード
- インダイレクトモードとメモリマップドモードの両方で完全にプログラム可能なフレームフォーマット
- 受信および送信用の内蔵 FIFO
- 8、16、32 ビットのデータアクセスを許可
- インダイレクトモードで動作するための DMA チャンネル
- FIFO 閾値、タイムアウト、動作完了、アクセスエラーによる割り込み生成

12.3 QUADSPI の機能説明

12.3.1 QUADSPI ブロック図

図 51. QUADSPI ブロック図 (デュアルフラッシュモードが無効のとき)

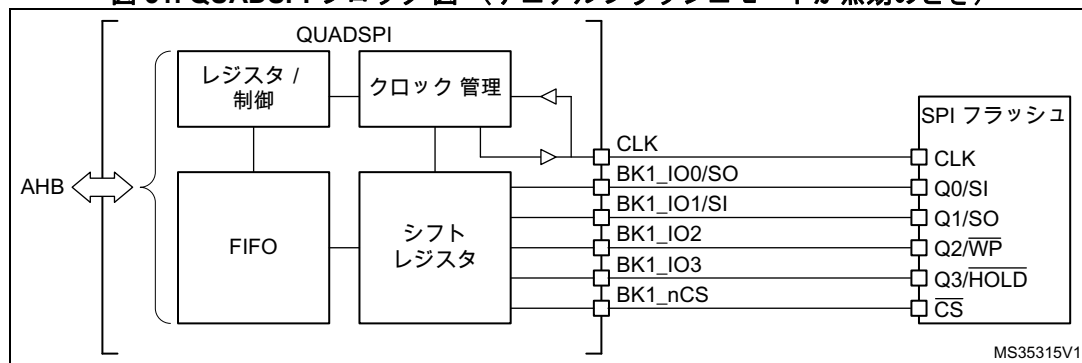
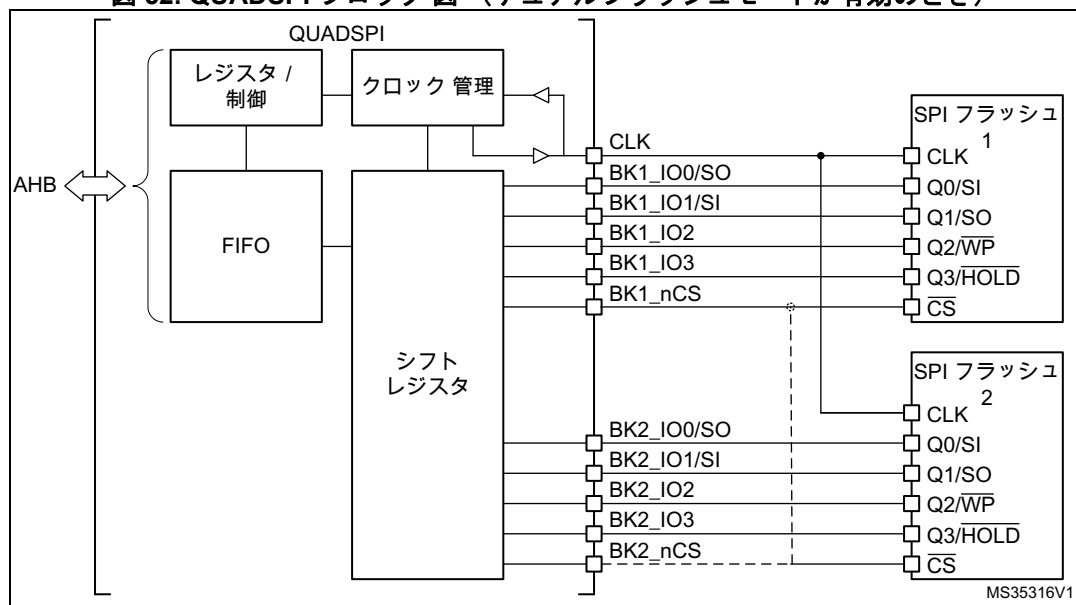


図 52. QUADSPI ブロック 図 (デュアルフラッシュモードが有効のとき)



QUADSPI は、シングル フラッシュメモリとのインタフェースでは 6 つの信号、デュアルフラッシュモードの 2 つのフラッシュメモリ (フラッシュ 1 および フラッシュ 2) とのインタフェースでは 10 ~11 の信号を使用します。

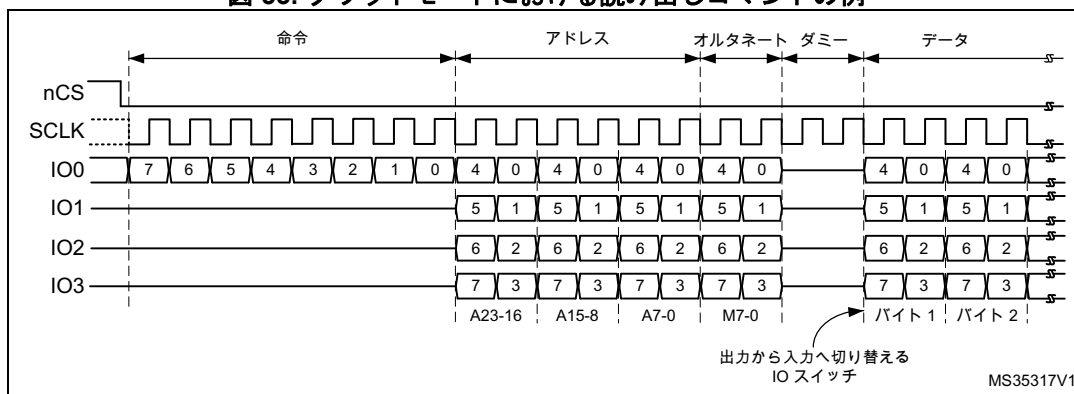
- CLK - クロック出力 (フラッシュ 1 およびフラッシュ 2)
- BK1_IO0/SO - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル出力 (フラッシュ 1 のみ)
- BK1_IO1/SI - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル入力 (フラッシュ 1 のみ)
- BK1_IO2 - クワッドモードでの双方向入出力 (フラッシュ 1 のみ)
- BK1_IO3 - クワッドモードでの双方向入出力 (フラッシュ 1 のみ)
- BK2_IO0/SO - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル出力 (フラッシュ 2 のみ)
- BK2_IO1/SI - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル入力 (フラッシュ 2 のみ)
- BK2_IO2 - クワッドモードでの双方向入出力 (フラッシュ 2 のみ)
- BK2_IO3 - クワッドモードでの双方向入出力 (フラッシュ 2 のみ)
- BK1_nCS - チップセレクト出力 (アクティブロー) (フラッシュ 1 のみ)。デュアルフラッシュモードで QUADSPI が常時使用される場合は、フラッシュ 2 にも使用可能
- BK2_nCS - チップセレクト出力 (アクティブロー) (フラッシュ 2 のみ)。デュアルフラッシュモードで QUADSPI が常時使用される場合は、フラッシュ 1 にも使用可能

12.3.2 QUADSPI コマンドシーケンス

QUADSPI はコマンドを使用してフラッシュメモリと通信します。各コマンドには、命令、アドレス、オルタネートバイト、ダミー、データの 5 つのフェーズが含まれます。これらのフェーズのいずれかをスキップするように設定することができますが、命令、アドレス、オルタネートバイト、データの少なくとも 1 つのフェーズが必要です。

nCS は各コマンドの開始前に立ち下がり、各コマンドの終了前に再び立ち上がります。

図 53. クワッドモードにおける読み出しコマンドの例



命令フェーズ

このフェーズでは、QUADSPI_CCR[7:0] レジスタの INSTRUCTION フィールドに設定された 8 ビットの命令がフラッシュメモリに送信され、実行すべき動作のタイプが指定されます。

ほとんどのフラッシュメモリは IO0/SO 信号 (シングル SPI モード) から同時に 1 ビットの命令しか受信できませんが、命令フェーズでは、オプションで、同時に 2 ビット (デュアル SPI モードの IO0/IO1 経由) または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) を送信することができます。この設定には、QUADSPI_CCR[9:8] レジスタの IMODE[1:0] フィールドを使用することができます。

IMODE = 00 の場合、命令フェーズはスキップされますので、コマンドシーケンスはアドレスフェーズが存在する場合はそこから開始されます。

アドレスフェーズ

アドレスフェーズでは、動作のアドレスを示す 1~4 バイトがフラッシュメモリに送信されます。送信されるアドレスバイト数は、QUADSPI_CCR[13:12] レジスタの ADSIZE[1:0] フィールドに設定されます。インダイレクトモードと自動ポーリングモードでは、送信されるアドレスバイトは QUADSPI_AR レジスタの ADDRESS[31:0] フィールドに指定されます。一方、メモリマップドモードの場合は、アドレスは AHB (Cortex® から、または DMA から) 経由で直接与えられます。

アドレスフェーズで同時に送信できるのは、1 ビット (シングル SPI モードの SO 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI_CCR[11:10] レジスタの ADMODE[1:0] フィールドを使用することができます。

ADMODE = 00 の場合、アドレスフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

オルタネートバイトフェーズ

オルタネートバイトフェーズでは、一般に動作モードを制御する 1~4 バイトがフラッシュメモリに送信されます。送信されるオルタネートバイト数は、QUADSPI_CCR[17:16] レジスタの ABSIZE[1:0] フィールドに設定されます。送信されるバイトは QUADSPI_ABR レジスタで指定されます。

オルタネートバイトフェーズで同時に送信できるのは、1 ビット (シングル SPI モードの SO 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI_CCR[15:14] レジスタの ABMODE[1:0] フィールドを使用することができます。

ABMODE = 00 の場合、オルタネートバイトフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

場合によっては、オルタネートバイトフェーズ中にフルバイトではなく 4bit を送信する必要があるかもしれません。たとえば、デュアルモード使用時で、オルタネートバイトに 2 サイクルだけが使われる場合などです。この場合、ファームウェアによって Quad モード (ABMODE = 11) を使用し、ALTERNATE のビット 7 および 3 を“1” (IO3 ラインをハイで保持) に、ビット 6 および 2 を“0” (IO2 ラインをローで保持) にセットしてバイトを送信できます。送信される 4bit の上位 2 ビットは ALTERNATE のビット 4:3 にセットされ、下位 2 ビットはビット 1 および 0 にセットされます。例えば、4bit の値 2 (0010) が IO0/IO1 で送信される場合、ALTERNATE は 0x8A (1000_1010) にセットされている必要があります。

ダミーサイクルフェーズ

ダミーサイクルフェーズでは、データの送受信は一切なく、1~31 サイクルが実行されます。これは、より高いクロック周波数が使われている場合に、データフェーズの準備をする時間をフラッシュメモリに与えるためです。このフェーズ中に実行されるサイクル数は、QUADSPI_CCR[22:18] レジスタの DCYC[4:0] フィールドで指定されます。SDR モードと DDR モードのどちらでも、処理時間にはフル CLK サイクルの数が指定されます。

DCYC = 00 の場合、ダミーサイクルフェーズはスキップされますので、コマンドシーケンスはデータフェーズが存在する場合は直接そこに進みます。

ダミーサイクルフェーズの動作モードは DMODE によって決まります。

データ信号を出力モードから入力モードへ変更するのに十分な「ターンアラウンド」時間を確保するために、デュアルまたはクワッドモードを使用してフラッシュメモリからデータを受信する場合、少なくとも 1 ダミーサイクルが必要です。

データフェーズ

データフェーズ中は、フラッシュメモリに対して何バイトでも送受信することができます。

インダイレクトモードと自動ポーリングモードでは、送受信するバイト数は QUADSPI_DLR レジスタで指定されます。

インダイレクト書き込みモードでは、フラッシュメモリに送信されるデータを QUADSPI_DR レジスタに書き込む必要があります。一方、インダイレクト読み出しモードでは、フラッシュメモリから受信するデータは QUADSPI_DR レジスタから読み出して取得します。

メモリマップドモードでは、読み出されるデータは、直接 AHB を経由して Cortex または DMA へ送り戻されます。

データフェーズで同時に送受信できるのは、1 ビット (シングル SPI モードの SO/SI 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI_CCR[15:14] レジスタの ABMODE[1:0] フィールドを使用することができます。

DMODE = 00 の場合、データフェーズはスキップされますので、コマンドシーケンスは nCS を立ち上げることですぐに終了します。この設定はインダイレクト書き込みモードのみで使用してください。

12.3.3 QUADSPI シングルインタフェースプロトコルモード

シングル SPI モード

レガシー SPI モードでは、1 ビットの順次送受信のみが可能です。このモードでは、データは SO 信号 (I/O を IO0 と共有) を経由してフラッシュメモリへ送信されます。フラッシュメモリからのデータは、SI 信号 (I/O を IO1 と共有) を経由して受信されます。

QUADSPI_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“01”にセットすることで、このシングルビットモードを使用できるよう、フェーズごとに個別に設定することができます。

各フェーズをシングルモードに設定する場合

- IO0 (SO) は出力モードです。
- IO1 (SI) は入力モードです (ハイインピーダンス)。
- IO2 は出力モードで、強制的に“0”に設定されます (書き込み保護機能を無効にするため)。
- IO3 は出力モードで、強制的に“1”に設定されます (ホールド機能を無効にするため)。

これは、DMODE = 01 であれば、ダミーフェーズにも該当します。

デュアル SPI モード

デュアル SPI モードでは、IO0/IO1 信号を経由して 2 ビットが同時に送受信されます。

QUADSPI_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“10”にセットすることで、このデュアル SPI モードを使用できるよう、フェーズごとに個別に設定することができます。

各フェーズをデュアルモードに設定する場合

- IO0/IO1 は、データフェーズの読み出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。
- IO2 は出力モードで、強制的に“0”に設定されます。
- IO3 は出力モードで、強制的に“1”に設定されます。

ダミーフェーズでは、DMODE = 01 の場合、IO0/IO1 は常にハイインピーダンスです。

Quad SPI モード

Quad SPI モードでは、IO0/IO1/IO2/IO3 信号を経由して 4 ビットが同時に送受信されます。

QUADSPI_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“11”にセットすることで、このQuad SPI モードを使用できるよう、フェーズごとに個別に設定することができます。

クワッドモードに設定されている各フェーズでは、IO0/IO1/IO2/IO3 はすべてデータフェーズの読み出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。

ダミーフェーズでは、DMODE = 11 の場合、IO0/IO1/IO2/IO3 はすべて常にハイインピーダンスです。

IO2 と IO3 はQuad SPI モードでのみ使用されます。Quad SPI モードを使用できるように設定されたフェーズが 1 つもない場合、IO2 および IO3 に対応するピンは QUADSPI がアクティブなときでも、他の機能に使用することができます。

SDR モード

DDRM ビット (QUADSPI_CCR[31]) は、デフォルトで 0 に設定されており、QUADSPI はシングルデータレート (SDR) モードで動作します。

SDR モードでは、QUADSPI が IO0/SO、IO1、IO2、IO3 信号を駆動している場合、これらの信号は CLK の立ち下がりエッジでのみ遷移します。

SDR モードでデータを受信する場合、QUADSPI はフラッシュメモリも CLK の立ち下がりエッジを使用してデータを送信するとみなします。デフォルトでは (SSHIFT = 0 の場合)、信号は次の CLK の (立ち上がり) エッジを使用してサンプリングされます。

DDR モード

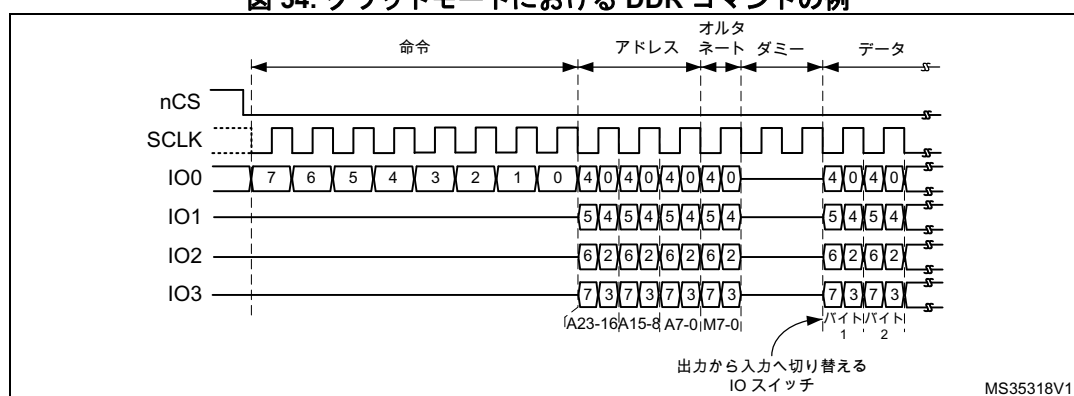
DDRM ビット (QUADSPI_CCR[31]) が 1 にセットされている場合、QUADSPI はダブルデータレート (DDR) モードで動作します。

DDR モードでは、QUADSPI がアドレス/オルタネートバイト/データフェーズで IO0/SO、IO1、IO2、IO3 信号を駆動している場合、CLK の立ち下がりエッジおよび立ち上がりエッジのそれぞれでビットを送信します。

命令フェーズは DDRM の影響は受けません。命令は常に CLK の立ち下がりエッジを使用して送信されます。

DDR モードでデータを受信する場合、QUADSPI はフラッシュメモリも CLK の立ち上がりエッジと立ち下がりエッジの両方を使用してデータを送信するとみなします。DDRM = 1 の場合、ファームウェアは SSHIFT ビット (QUADSPI_CR[4]) をクリアする必要があります。したがって、信号は 1/2 CLK サイクル後にサンプリングされます (下図の逆エッジで)。

図 54. クワッドモードにおける DDR コマンドの例



デュアルフラッシュモード

DFM ビット (QUADSPI_CR[6]) が 1 の場合、QUADSPI はデュアルフラッシュモードです。このモードでは、2 つの外部Quad SPI フラッシュメモリ (フラッシュ 1 と フラッシュ 2) が使用されており、サイクルごとに 8 ビットの送受信 (DDR モードでは 16 ビット) を行うことで、スループットおよび容量を効果的に倍にします。

どちらのフラッシュメモリも同じ CLK を使用し、オプションで同じ nCS 信号を使用しますが、それぞれが個別の IO0、IO1、IO2、IO3 信号を有しています。

デュアルフラッシュモードは、SDR モードまたは DDR モードと組み合わせて使用できるだけでなく、シングルビットモード、デュアルビットモード、クワッドビットモードとも組み合わせて使用できます。

FSIZE[4:0] (QUADSPI_DCR[20:16]) で指定されているフラッシュメモリのサイズは、フラッシュメモリの総容量 (コンポーネント 1 個のサイズの 2 倍) を反映していなければなりません。

アドレス X が偶数の場合、QUADSPI はアドレス X に対してフラッシュ 1 のアドレス X/2 にあるバイトを与え、アドレス X+1 に対してフラッシュ 2 のアドレス X/2 にあるバイトを与えます。言い換えれば、偶数アドレスにあるバイトはすべてフラッシュ 1 に、奇数アドレスにあるバイトはすべてフラッシュ 2 に格納されます。

デュアルフラッシュモードでフラッシュメモリステータスレジスタを読み出す場合、シングルフラッシュモードで読み出すバイト数の 2 倍のバイトを読み出す必要があります。つまり、各フラッシュメモリがステータスレジスタのフェッチ命令後に有効な 8 ビットを提供する場合、QUADSPI はデータ長 2 バイト (16 ビット) で設定する必要があります。それにより QUADSPI は各フラッシュメモリから 1 バイトを受信します。各フラッシュメモリが 16 ビットのステータスを提供する場合、QUADSPI は、4 バイトを読み出して、デュアルフラッシュモードの両方のフラッシュメモリのステータスビットをすべて取得できるよう設定する必要があります。リザルト (データレジスタ内) の最下位バイトはフラッシュ 1 ステータスレジスタの最下位バイト、次のバイトはフラッシュ 2 ステータスレジスタの最下位バイトとなります。さらに、データレジスタの 3 番目のバイトはフラッシュ 1 の 2 番目のバイト、4 番目のバイトはフラッシュ 2 の 2 番目のバイトとなります (フラッシュメモリに 16 ビットステータスレジスタがある場合)。

偶数バイトは常にデュアルフラッシュモードでアクセスする必要があります。このため、DRM = 1 のとき、データ長フィールドのビット 0 (QUADSPI_DLR[0]) は 1 のままとなります。

デュアルフラッシュモードでは、フラッシュ 1 インタフェース信号の動作は基本的に通常モードの動作と同じです。フラッシュ 2 インタフェース信号の波形は、命令、アドレス、オルタネートバイト、およびタミーサイクルフェーズの間は、フラッシュ 1 とまったく同じです。つまり、どちらのフラッシュメモリも常に同じ命令や同じアドレスを受信するということです。データフェーズ中は、2 つのバス BK1_IOx と BK2_IOx は並列してデータ転送を行います。フラッシュ 1 に対する送信 (または受信) データはフラッシュ 2 のものとは異なります。

12.3.4 QUADSPI インダイレクトモード

インダイレクトモードでは、QUADSPI レジスタに書き込むことでコマンドが開始され、データレジスタへの書き込み/読み出しによってデータが転送されます。その方法はその他の通信ペリフェラルの場合と同様です。

FMODE = 00 (QUADSPI_CCR[27:26]) の場合、QUADSPI はインダイレクト書き込みモードです。このモードでは、データフェーズ中にフラッシュメモリへのバイトの送信が行われます。データは、データレジスタ (QUADSPI_DR) への書き込みによって提供されます。

FMODE = 01 の場合、QUADSPI はインダイレクト読み出しモードです。このモードでは、データフェーズ中にフラッシュメモリからのバイトの受信が行われます。データは、QUADSPI_DR の読み出しによって復旧されます。

送受信するバイト数はデータ長レジスタ (QUADSPI_DLR) で指定されます。QUADSPI_DLR = 0xFFFF_FFFF (すべて“1”) の場合、データ長は不定長と見なされ、QUADSPI はフラッシュメモリの終了 (FSIZE によって定義される) までデータを転送し続けるのみです。転送されるバイトが無い場合は、DMODE (QUADSPI_CCR[25:24]) を“00”にセットする必要があります。

QUADSPI_DLR = 0xFFFF_FFFF かつ FSIZE = 0x1F (4 GB のフラッシュメモリを示す最大値) の場合、このような特殊なケースでは、転送は無限に続けられ、アボートリクエスト後または QUADSPI の無効化後にやっと停止します。最後のメモリアドレスが読み出された後も (アドレス 0xFFFF_FFFF で)、アドレス = 0x0000_0000 で読み出しが続けられます。

プログラムされた送信または受信バイト数に達すると、TCF がセットされ、TCIE = 1 であれば割り込みが生成されます。データ数が定義されていない場合、外部 SPI メモリの限界に達した時点で、QUADSPI_CR に定義されているフラッシュメモリのサイズに応じて TCF がセットされます。



コマンド開始のトリガ

コマンドは、基本的に、ファームウェアがそのコマンドに必要な最期の情報を提供するとすぐに開始されます。インダイレクトモードでコマンドを開始する方法は、QUADSPI の設定に応じて、3 通りあります。コマンドは以下の動作の直後に開始されます。

1. アドレスを必要としない場合 (ADMODE = 00) およびファームウェアによるデータの提供を必要としない場合 (FMODE = 01 または DMODE = 00) は、INSTRUCTION[7:0] (QUADSPI_CCR) への書き込み動作直後
2. アドレスを必要とする場合 (ADMODE != 00) およびファームウェアによるデータの提供を必要としない場合 (FMODE = 01 または DMODE = 00) は、ADDRESS[31:0] (QUADSPI_AR) への書き込み動作直後
3. アドレスを必要とする場合 (ADMODE != 00) およびファームウェアによるデータの提供を必要とする場合 (FMODE = 00 および DMODE != 00) は、DATA[31:0] (QUADSPI_DR) への書き込み動作直後

オルタネートバイトレジスタ (QUADSPI_ABR) への書き込みが通信開始をトリガすることはありません。オルタネートバイトが必要である場合、事前にプログラムしておく必要があります。

コマンドが開始されたらすぐに、BUSY ビット (QUADSPI_SR の ビット5) が自動的にセットされます。

FIFO およびデータの管理

インダイレクトモードでは、データは QUADSPI に内蔵されている 32 バイト FIFO を通ります。FLEVEL[5:0] (QUADSPI_SR[13:8]) は、現在 FIFO に保持されているバイト数を示します。

インダイレクト書き込みモード (FMODE = 00) では、ファームウェアは QUADSPI_DR に書き込む時点で FIFO にデータを追加します。書き込みにより FIFO に追加されるバイト数は、ワード書き込みで 4 バイト、ハーフワード書き込みで 2 バイト、バイト書き込みで 1 バイトです。ファームウェアが FIFO に追加するバイト数が多すぎる (DL[31:0] によって示される数より多い) 場合、余分なバイトは、書き込み動作の終了時 (TCF がセットされるとき) に FIFO から一掃されます。

QUADSPI_DR へのバイト/ハーフワードアクセスは、32 ビットレジスタの最下位バイト/ハーフワードにのみ実行されます。

FTHRES[3:0] を使用して FIFO 閾値が定義されます。閾値に達すると、FTF (FIFO 閾値フラグ) がセットされます。インダイレクト読み出しモードでは、FIFO から読み出される有効バイト数が閾値を超えると FTF がセットされます。また、FTHRES の設定に関わらず、フラッシュメモリから最後のバイトが読み出された後に FIFO 内にデータが残っている場合も FTF がセットされます。インダイレクト書き込みモードでは、FIFO 内の空のバイト数が閾値を超えると FTF がセットされます。

FTIE = 1 の場合、FTF がセットされた時点で割り込みが発生します。DMAEN = 1 の場合、FTF がセットされた時点で DMA 転送が開始されます。閾値条件が真でなくなる (CPU または DMA によって十分なデータが転送された後) とすぐに、HW によって FTF がクリアされます。

インダイレクト読み出しモードでは、FIFO がフルになると、QUADSPI は一時的にフラッシュメモリからのバイトの読み出しを停止して、オーバーランを回避します。フラッシュメモリの読み出しは、FIFO 内の 4 バイトが空になるまで再開されませんのでご注意ください (FLEVEL ≤ 11 の場合)。したがって、FTHRES ≥ 13 の場合、QUADSPI は確実にフラッシュメモリからのデータ取得を再開できるものとして、アプリケーションによる十分なバイトの読み出しが行われる必要があります。そうしないと、11 < FLEVEL < FTHRES である限り、FTF フラグは 0 のままとなります。

12.3.5 QUADSPI ステータスフラグポーリングモード

自動ポーリングモードでは、QUADSPI は定期的にコマンドを開始させて、定義された数のステータスバイト（最大 4）を読み出します。受信バイトをマスクして、ステータスビットの一部を分離することができます。また、選択されたビットに値が定義されている場合は割り込みを生成することができます。

フラッシュメモリへのアクセス方法はインダイレクト読み出しモードの場合と同様です。アドレスを必要としない場合（ADMODE = 00）は、QUADSPI_CCR が書き込まれるとすぐにアクセスが開始されます。反対にアドレスを必要とする場合は、QUADSPI_AR が書き込まれた時点で最初のアクセスが始まります。この時点で BUSY がハイになり、定期的なアクセスが実行される間もハイのままです。

MASK[31:0] (QUADSPI_PSMAR) の内容を使用して、自動ポーリングモードにおけるフラッシュメモリからのデータがマスクされます。MASK[n] = 0 の場合、リザルトのビット n はマスクされ、考慮されません。MASK[n] = 1 かつ bit[n] の値が MATCH[n] (QUADSPI_PSMAR) の値と同じである場合、ビット n との一致が見られます。

ポーリング一致モードビット (PMM、QUADSPI_CR[23]) が 0 の場合、「AND」一致モードがアクティブになります。つまり、ステータス一致フラグ (SMF) は、マスクされていないすべてのビットで一致が検出された場合にのみセットされます。

PMM = 1 の場合、「OR」一致モードがアクティブになります。つまり、SMF は、マスクされていない任意のビットで一致が検出された場合にセットされます。

SMIE = 1 の場合に SMF がセットされると、割り込みが発生します。

自動ポーリングモード停止 (APMS) ビットがセットされている場合、一致が検出されるとすぐに動作は停止し、BUSY フラグが 0 になります。あるいは、BUSY は“1”のままで、アボートが検出される、または QUADSPI が無効化される (EN = 0) まで定期的なアクセスが継続します。

データレジスタ (QUADSPI_DR) には、最後に受信したステータスバイト (FIFO の無効化) が含まれます。データレジスタの内容は、一致ロジックで使用されるマスクの影響を受けません。FTF ステータスビットは新たなステータスの読み出しが完了するとすぐにセットされ、データが読み出された直後にクリアされます。

12.3.6 QUADSPI メモリマップドモード

メモリマップドモードに設定された場合、外部 SPI デバイスは内部メモリとみなされます。

QUADSPI ペリフェラルを正しく設定し、有効にする前に、QUADSPI フラッシュバンク領域にアクセスすることは禁じられています。

フラッシュメモリの容量がもっと大きい場合でも、アドレス指定できるのは 256 MB までです。

FSIZE で規定されている範囲外のアドレスにアクセスすると、256 MB の範囲内であっても、AHB エラーが発生します。このエラーの影響は、アクセスを試みた AHB マスタに依存します。

- Cortex® CPU の場合は、ハードフォールト割り込みが発生します。
- DMA の場合は、DMA 転送エラーが発生し、対応する DMA チャネルが自動的に無効になります。

バイト、ハーフワード、およびワードアクセスはすべてサポートされます。

XIP (Execute In Place) 動作のサポートが実装されています。この場合、QUADSPI は次のマイクロコントローラのアクセスを予測し、あらかじめ次のアドレスにバイトをロードします。その後のアクセスが実際に連続したアドレスに対して行われる場合、値がすでにプリフェッチされているので、アクセスは高速で完了します。

デフォルトでは、QUADSPI がプリフェッチ動作を停止することなく、フラッシュメモリへのアクセスが長時間なくとも、前の読み出し動作はアクティブな状態を保ち、nCs もローに保持されます。nCS

がローのときにフラッシュメモリはより多く電力消費する傾向があるため、アプリケーションがタイムアウトカウンタ (TCEN = 1、QUADSPI_CR[3]) の起動を要求する場合があります、FIFO がプリフェッチデータでいっぱいになっているときに、アクセスのない状態で TIMEOUT[15:0] (QUADSPI_LPTR) のサイクル周期が経過すると nCS が開放されます。

最初のメモリマップドアクセスが発生するとすぐに、BUSY はハイになります。プリフェッチ動作の影響で、BUSY の立ち下がりは、タイムアウトが起こる、アボートが検出される、またはペリフェラルが無効化されるまで起こりません。

12.3.7 QUADSPI フラッシュメモリの設定

デバイス設定レジスタ (QUADSPI_DCR) を使用して、外部 SPI フラッシュメモリの特性を指定することができます。

FSIZE[4:0] フィールドは、次の式を使用して外部メモリのサイズを定義します。

$$\text{フラッシュメモリ内のバイト数} = 2^{[FSIZE+1]}$$

FSIZE+1 はフラッシュメモリのアドレスを指定するために必要とされる有効なアドレスビット数です。フラッシュメモリの容量はインダイレクトモードでは最大 4 GB (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 MB に制限されます。

DFM = 1 の場合、FSIZE は 2 つのフラッシュメモリの総容量を示します。

QUADSPI が 2 つのコマンドを実行する場合、一方のコマンドはもう一方のコマンドの直後に実行され、デフォルトでは 1 CLK サイクル中の 2 つのコマンドの間にハイレベルのチップセレクト信号 (nCS) を立ち上げます。フラッシュメモリがコマンド間により長い時間を必要とする場合、チップセレクトハイ時間 (CSHT) フィールドを使用して、nCS がハイに保たなければならない CLK サイクルの回数 (最大 8) を指定することができます。

クロックモード (CKMODE) ビットは、コマンド間の CLK 信号ロジックレベルを示します (nCS = 1 の場合)。

12.3.8 QUADSPI 遅延データサンプリング

デフォルトでは、QUADSPI がフラッシュメモリに駆動されているデータのサンプリングを行うタイミングは、フラッシュメモリが信号を駆動してから 1/2 CLK サイクル後です。

外部信号の遅延が発生した場合、それは後でデータをサンプリングする際に有用となるかもしれません。SSHIFT ビット (QUADSPI_CR[4]) を使用することで、データのサンプリングを 1/2 CLK サイクルだけシフトすることができます。

クロックのシフトは DDR モードではサポートされていないので、DDRM ビットがセットされた時点で SSHIFT ビットをクリアする必要があります。

12.3.9 QUADSPI の設定

QUADSPI の設定は次の 2 つのフェーズで行われます。

- QUADSPI IP の設定
- QUADSPI フラッシュメモリの設定

QUADSPI は、設定および有効化が行われると、インダイレクトモード、ステータスポーリングモード、メモリマップドモードの 3 つの動作モードのいずれかで使用することができます。

QUADSPI IP の設定

QUADSPI IP は QUADSPI_CR を使用して設定します。ユーザは、受信データに対してクロックプリスケアラ分周比とサンプリングシフトを設定します。

DDR モードは DDRM ビットによって設定することができます。このモードを有効にすると、アドレスおよびオルタネートバイトが両方のクロックエッジに送信され、両方のクロックエッジでデータの送受信が行われます。DDRM ビットの設定にかかわらず、命令は常に SDR モードで送信されます。

DMA リクエストは DMAEN ビットを設定することで有効化されます。割り込みを使用する場合、それぞれのイネーブルビットもこのフェーズ中にセットされます。

DMA リクエスト生成または割り込み生成の FIFO レベルは FTHRES ビットで設定されます。

タイムアウトの発生が必要な場合、TCEN ビットをセットして、QUADSPI_LPTR レジスタにタイムアウト値を設定することができます。

デュアルフラッシュモードは DFM を “1” にセットすることで有効にできます。

QUADSPI フラッシュメモリの設定

対象の外部フラッシュメモリに関連するパラメータは QUADSPI_DCR レジスタによって設定されます。ユーザは FSIZE ビットでフラッシュメモリのサイズを、CSHT ビットでチップセレクトハイ時間の最小値を、MODE ビットで機能モード（モード 0 またはモード 3）を設定します。

12.3.10 QUADSPI の使用

動作モードは FMODE[1:0] (QUADSPI_CCR[27:26]) を使用して選択されます。

インダイレクトモードの手順

FMODE が 00 に設定されている場合、インダイレクト書き込みモードが選択され、データをフラッシュメモリに送信することができます。FMODE = 01 に設定されている場合、インダイレクト読み出しモードが選択され、データをフラッシュメモリから読み出すことができます。

QUADSPI がインダイレクトモードで使用される場合、フレームは次のように構成されます。

1. QUADSPI_DLR に、読み出される／書き込まれるデータのバイト数を指定します。
2. QUADSPI_CCR に、フレームフォーマット、モード、および命令コードを指定します。
3. QUADSPI_ABR に、アドレスフェーズの直後に送信されるオプションのオルタネートバイトを指定します。
4. QUADSPI_CR に、動作モードを指定します。FMODE = 00 (インダイレクト書き込みモード) かつ DMAEN = 1 の場合、QUADSPI_AR は QUADSPI_CR より前に指定されなければなりません。そうしないと、QUADSPI_AR が更新される前に QUADSPI_DR が DMA によって書き込まれる可能性があるからです (DMA コントローラがすでに有効な場合)。
5. QUADSPI_AR に、ターゲットアドレスを指定します。
6. QUADSPI_DR により FIFO に対してデータの読み出し／書き込みを行います。

制御レジスタ (QUADSPI_CR) に書き込む場合は次の設定値を指定してください。

- イネーブルビット (EN) を “1” にセット
- RAM へのデータ転送用の DMA 有効ビット (DMAEN)
- タイムアウトカウンタイネーブルビット (TCEN)
- サンプリングシフト設定 (SSHIFT)
- FTF フラグをセットするタイミングを示す FIFO 閾値レベル (FTHRES)
- 割り込みイネーブル
- 自動ポーリングモードのパラメータ：一致モードおよび STOP モード (FMODE = 11 のとき有効)
- クロックプリスケアラ

通信設定レジスタ (QUADSPI_CCR) に書き込む場合は次のパラメータを指定してください。

- INSTRUCTION ビットによる命令バイト
- IMODE ビット (1、2、または4 ライン) により命令を送信する方法
- ADMODE ビット (なし、または1、2、4 ライン) によりアドレスを送信する方法
- ADSIZE ビットによるアドレスサイズ (8、16、24、または32 ビット)
- ADMODE ビット (なし、または1、2、4 ライン) によりオルタネートバイトを送信する方法
- ABSIZE ビットによるオルタネートバイト数 (1、2、3、または4)
- DBMODE ビットによるダミーバイトの有無
- DCYC ビットによるダミーバイト数
- DMODE ビットによるデータの送受信方法 (なし、または1、2、4 ライン)

アドレスレジスタ (QUADSPI_AR) とデータレジスタ (QUADSPI_DR) のどちらも特定のコマンドに関して更新する必要がない場合、QUADSPI_CCR が書き込まれるとすぐにコマンドシーケンスが開始されます。これは、ADMODE と DMODE がどちらも 00 である場合、またはインダイレクト読み出しモード (FMODE = 01) で ADMODE = 00 の場合に該当します。

アドレスを必要とする場合 (ADMODE が 00 以外) で、データレジスタに書き込む必要がない場合 (FMODE = 01 または DMODE = 00)、QUADSPI_AR が書き込まれてアドレスが更新されるとすぐにコマンドシーケンスが開始されます。

データ転送時 (FMODE = 00 かつ DMODE! = 00)、QUADSPI_DR による FIFO への書き込みによって通信の開始がトリガされます。

ステータスフラグポーリングモード

ステータスフラグポーリングモードは、FMODE フィールド (QUADSPI_CCR[27:26]) を 10 にセットすることで有効になります。このモードでは、プログラムされたフレームが送信され、定期的にデータが取得されます。

各フレームに読み込まれるデータの最大量は 4 バイトです。QUADSPI_DLR で もっと大量のデータが要求されても、そのリクエストは無視され、4 バイトだけが読み出されます。

QUADSPI_PISR レジスタには周期性が指定されます。

ステータスデータが取得されると、そのデータを次の目的のために内部で処理することができます。

- ステータス一致フラグをセットし、有効になっていれば割り込みを生成する。
- ステータスバイトの定期的な取得を自動的に停止する。

受信した値は QUADSPI_PSMKR に格納されている値でマスクし、QUADSPI_PSMAR に格納されている値で論理和または論理積をとることができます。

一致する場合、ステータス一致フラグがセットされ、有効になっていれば割り込みが生成されます。また、AMPS ビットがセットされている場合、QUADSPI は自動的に停止する可能性があります。

いずれにしろ、一番最後に取得した値は QUADSPI_DR で使用可能です。

メモリマップドモード

メモリマップドモードでは、外部フラッシュメモリは内部メモリとみなされますが、アクセス中にいくらかの遅延があります。このモードでは外部フラッシュメモリに対して読み出し動作のみが許可されます。

メモリマップドモードに移行するには、QUADSPI_CCR レジスタの FMODE を 11 にセットします。

プログラムされた命令およびフレームは、AHB マスタがメモリマップド空間にアクセスしているときに送信されます。

FIFO は、リニア読み出しを予測するプリフェッチバッファとして使用されます。このモードでの QUADSPI_DR へのアクセスはすべてゼロで返されます。

メモリマップドモードでは、データ長レジスタ (QUADSPI_DLR) は意味を持ちません。

12.3.11 1 回限りの命令の送信

一部のフラッシュメモリ (Winbound など) が提供するモードは、最初のコマンドシーケンスでのみ命令を送信し、その後のコマンドは直接アドレスから開始されるというものです。SIOO ビット (QUADSPI_CCR[28]) を使用して、そのような機能を活用することができます。

SIOO はすべての機能モードで有効です (インダイレクト、自動ポーリング、メモリマップド)。SIOO ビットがセットされている場合、QUADSPI_CCR への書き込み後の最初のコマンドに対してのみ命令が送信されます。その後のコマンドシーケンスは、命令フェーズでは、QUADSPI_CCR への書き込みがあるまでスキップされます。

IMODE = 00 (命令なし) の場合、SIOO は効果がありません。

12.3.12 QUADSPI エラー管理

エラーは次の場合に生成できます。

- インダイレクトモードまたはステータスフラグポーリングモードでは、QUADSPI_AR に間違っただレスがプログラムされている場合 (QUADSPI_DCR の FSIZE[4:0] で定義されているフラッシュメモリのサイズによる)、TEF がセットされ、有効になっていれば割り込みが生成されます。
- また、インダイレクトモードでは、アドレスおよびデータ長がフラッシュメモリのサイズを超えている場合は、アクセスがトリガされるとすぐに TEF がセットされます。
- メモリマップドモードでは、AHB マスタによる範囲外のアクセスが行われた場合または QUADSPI が無効な場合は、誤った AHB リクエストへの応答として AHB エラーが生成されます。
- メモリマップドモードが無効なときに AHB マスタがメモリマップド空間にアクセスすると、誤った AHB リクエストへの応答として AHB エラーが生成されます。

12.3.13 QUADSPI の BUSY ビットおよびアボート機能

QUADSPI がフラッシュメモリでの動作を開始すると、BUSY ビットが QUADSPI_SR に自動的にセットされます。

インダイレクトモードでは、QUADSPI が要求されたコマンドシーケンスを完了し、FIFO が空になると、BUSY ビットがリセットされます。

自動ポーリングモードでは、一致 (APMS = 1 のとき) またはアボートの検出により、最後の定期的なアクセスが完了して初めて BUSY ビットはローになります。

メモリマップドモードでの最初のアクセス後、タイムアウトイベントまたはアボートの検出により、BUSY ビットはローになります。

QUADSPI_CR レジスタの ABORT ビットをセットすることによって、動作を中止することができます。アボートが完了すると、BUSY ビットおよび ABORT ビットは自動的にリセットされ、FIFO が一掃されます。

注： フラッシュメモリには、ステータスレジスタへの書き込み動作が中止された場合に誤動作するものもあります。

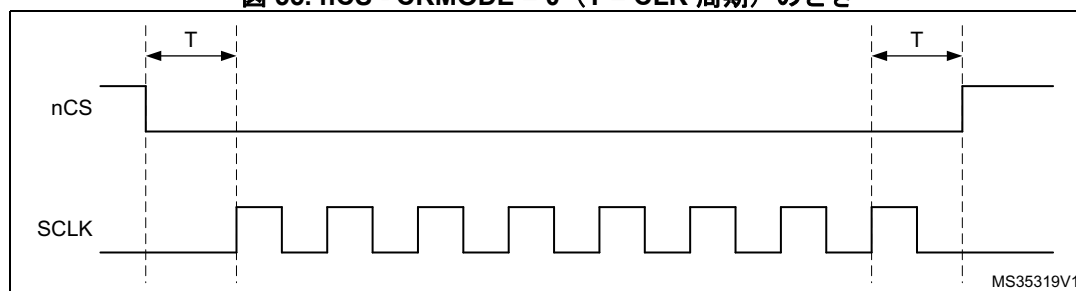


12.3.14 nCS の動作

デフォルトでは nCS はハイで、外部フラッシュメモリは選択解除されています。nCS の立ち下がり動作が始まる前に、立ち上がりは動作終了の直後に起こります。

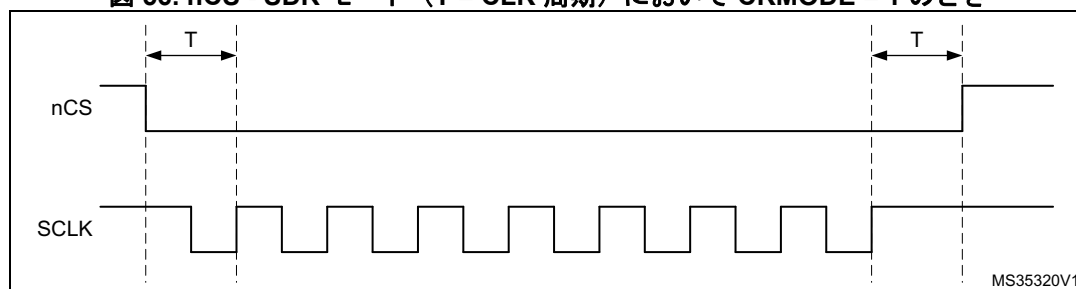
CKMODE = 0 (「モード0」で、この状態では進行中の動作がない場合は CLK がローに保持されている) のとき、nCS の立ち下がり動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前に、nCS の立ち上がりは動作の最後の CLK エッジ立ち上がりから 1 CLK サイクル後に起こります (図 55 を参照)。

図 55. nCS - CKMODE = 0 (T = CLK 周期) のとき



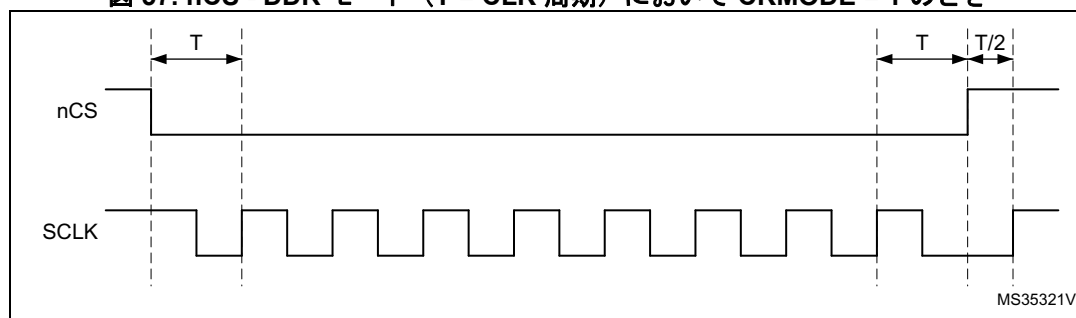
CKMODE = 1 (「モード3」で、この状態では進行中の動作がない場合に CLK がハイになる)、かつ DDRM = 0 (SDR モード) のとき、nCS は動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前でさらに立ち下がり、動作の最後の CLK エッジ立ち上がりから 1 CLK サイクル後に立ち上がります (図 56 を参照)。

図 56. nCS - SDR モード (T = CLK 周期) において CKMODE = 1 のとき



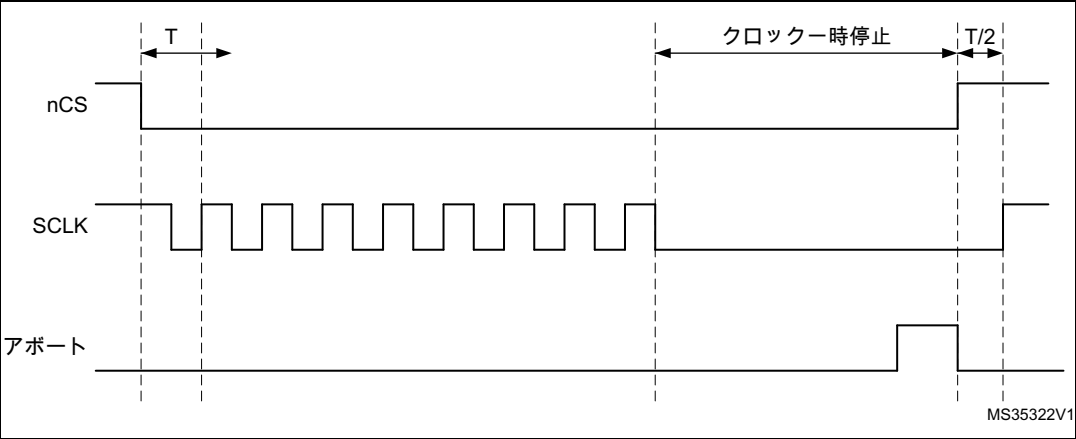
CKMODE = 1 (モード3) かつ DDRM = 1 (DDR モード) の場合、nCS の立ち下がり動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前に、nCS の立ち上がりは動作の最後の アクティブ CLK エッジ立ち上がりから 1 CLK サイクル後に起こります (図 57 を参照)。DDR 動作は立ち下がりエッジで終わらなければならないため、nCS が立ち上がる時 CLK はローとなり、その後 CLK は 1/2 CLK サイクルだけ後ろ向きに立ちあがります。

図 57. nCS - DDR モード (T = CLK 周期) において CKMODE = 1 のとき



読み出し動作で FIFO がフルのまま、または書き込み動作で FIFO が空のままであれば、ファームウェアが FIFO を処理するまでその動作は停止し、CLK はローのままとなります。動作が停止した時点でアボートが発生した場合、アボートが要求された直後に nCS の立ち上がりが起こり、CLK は 1/2 CLK サイクル後に立ち上がります (図 58 を参照)。

図 58. nCS - アボートが発生し (T = CLK 周期)、CKMODE = 1 のとき



デュアルフラッシュモード以外するとき (DFM = 0) はフラッシュ 1 のみがアクセスされるので、BK2_nCS はハイのままです。デュアルフラッシュモードでは、BK2_nCS は BK1_nCS とまったく同じように動作します。したがって、フラッシュ 2 が存在し、アプリケーションがずっとデュアルフラッシュモードのままであれば、フラッシュ 2 は BK1_nCS を使用することができ、BK2_nCS を出力するピンを他の機能に使用することができます。

12.4 QUADSPI 割り込み

割り込みは次のイベントによって生成できます。

- タイムアウト
- ステータス一致
- FIFO 閾値
- 転送完了
- 転送エラー

高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 71. QUADSPI 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
タイムアウト	TOF	TOIE
ステータス一致	SMF	SMIE
FIFO 閾値	FTF	FTIE
転送完了	TCF	TCIE
転送エラー	TEF	TEIE

12.5 QUADSPI レジスタ

12.5.1 QUADSPI 制御レジスタ (QUADSPI_CR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTHRES					FSEL	DFM	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
			rw	rw	rw	rw	rw	rw	rw		rw	rw	w1s	rw	w1s

ビット 31:24 **PRESCALER[7:0]** : クロックプリスケアラ

このフィールドは、AHB クロック (値 + 1) に基づいて CLK を生成するスケアラ係数を定義します。

0 : $F_{CLK} = F_{AHB}$ 、QUADSPI CLK (迂回されたプリスケアラ) として直接的に使用される AHB クロック

1 : $F_{CLK} = F_{AHB}/2$

2 : $F_{CLK} = F_{AHB}/3$

...

255 : $F_{CLK} = F_{AHB}/256$

クロック分周比が奇数の場合、CLK のデューティサイクルは 50% ではありません。クロック信号は、ローに保持された時間より 1 サイクル分長く、ハイに保持されます。

このフィールドは、BUSY が 0 のときにのみ修正できます。

ビット 23 **PMM** : ポーリング一致モード

このビットは、自動ポーリングモード時に「一致」を決定するために使うべき方法を示します。
0 : AND 一致モード SMF は、フラッシュメモリから受信したマスクされていないすべてのビットが一致レジスタの対応するビットと一致している場合に、セットされます。

1 : OR 一致モード SMF は、フラッシュメモリから受信したマスクされていないビットのいずれかが一致レジスタの対応するビットと一致している場合に、セットされます。

このビットは、BUSY が 0 のときにのみ修正できます。

ビット 22 **APMS** : 自動ポーリングモード停止

このビットは、一致後に自動ポーリングを停止するかどうかを決定します。

0 : 自動ポーリングモードは、アボートまたは QUADSPI の無効化によってのみ停止されます。

1 : 自動ポーリングモードは、一致が検出されるとすぐに停止されます。

このビットは、BUSY が 0 のときにのみ修正できます。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **TOIE** : タイムアウト割り込み有効化

このビットは、タイムアウト割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 19 **SMIE** : ステータス一致割り込み有効化

このビットは、ステータス一致割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 18 FTIE : FIFO 閾値割り込み有効化

このビットは、FIFO 閾値割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 17 TCIE : 転送完了割り込み有効化

このビットは、転送完了割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 16 TEIE : 転送エラー割り込み有効化

このビットは、転送エラー割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 FTHRES[4:0] FIFO 閾値レベル

インダイレクトモードでは、FIFO 閾値フラグ (FTF、QUADSPI_SR[2]) がセットされる FIFO 内のバイトの閾値数を定義します。

インダイレクト書き込みモード (FMODE = 00)の場合

0 : FTF は、FIFO 内に書き込みのできる空きバイトが 1 バイト以上ある場合にセットされます。

1 : FTF は、FIFO 内に書き込みのできる空きバイトが 2 バイト以上ある場合にセットされます。

...

31 : FTF は、FIFO 内に書き込みのできる空きバイトが 32 バイト以上ある場合にセットされます。

インダイレクト読み出しモード (FMODE = 01)の場合

0 : FTF は、FIFO から読み出せる有効なバイトが 1 バイト以上ある場合にセットされます。

1 : FTF は、FIFO から読み出せる有効なバイトが 2 バイト以上ある場合にセットされます。

...

31 : FTF は、FIFO から読み出せる有効なバイトが 32 バイト以上ある場合にセットされます。

DMAEN = 1 の場合、対応するチャンネル用の DMA コントローラは FTHRES 値を変更する前に無効にする必要があります。..

ビット 7 FSEL : フラッシュメモリの選択

このビットは、シングルフラッシュモード (DFM = 0) でアドレス指定するフラッシュメモリを選択します。

0 : フラッシュ 1 が選択されます。

1 : フラッシュ 2 が選択されます。

このビットは、BUSY が 0 のときにのみ修正できます。

このビットは、DFM = 1 のときは無視されます。

ビット 6 DFM : デュアルフラッシュモード

このビットはデュアルフラッシュモードをアクティブにします。このモードでは、2 つの外部フラッシュメモリを同時に使用して、スループットおよび容量を倍にします。

0 : デュアルフラッシュモードは無効です。

1 : デュアルフラッシュモードは有効です。

このビットは、BUSY が 0 のときにのみ修正できます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 SSHIFT : サンプリングシフト

デフォルトでは、QUADSPI はフラッシュメモリによるデータ駆動の 1/2 CLK サイクル後にデータをサンプリングします。このビットを使用することで、外部信号遅延を考慮して、後でデータをサンプリングすることができます。

0 : シフトなし

1 : 1/2 サイクルのシフト

ファームウェアは、DDR モード (DDRM = 1) では SSHIFT = 0 であることを予測する必要があります。

このフィールドは、BUSY が 0 のときにのみ修正できます。

ビット 3 TCEN : タイムアウトカウンタ有効化

このビットは、メモリマップドモードが選択されている (FMODE = 11) 場合にのみ有効です。ある一定の時間アクセスがない場合 (この時間は TIMEOUT[15:0] (QUADSPI_LPTR) によって定義されます)、このビットをアクティブにすることでチップセレクト (nCS) がリリリースされます (これにより消費電力が削減されます)。

タイムアウトカウンタを有効にします。

デフォルトでは、QUADSPI がブリフエッチ動作を停止することではなく、フラッシュメモリへのアクセスが長時間なくとも、前の読み出し動作はアクティブな状態を保ち、nCS もローに保持されます。nCS がローのときにフラッシュメモリはより多く電力を消費する傾向があるため、アプリケーションがタイムアウトカウンタ (TCEN=1, QUADSPI_CR[3]) の起動を要求する場合があります、FIFO がブリフエッチデータでいっぱいになっているときに、アクセスのない状態で TIMEOUT[15:0] (QUADSPI_LPTR) のサイクル周期が経過すると nCS が開放されます。

0 : タイムアウトカウンタは無効です。したがって、チップセレクト (nCS) はメモリマップドモードにおける 1 回のアクセス後は無限にアクティブなままです。

1 : タイムアウトカウンタは有効です。したがって、チップセレクトは、インアクティブなフラッシュメモリの TIMEOUT[15:0] サイクル後にメモリマップドモードでリリースされます。

このビットは、BUSY が 0 のときにのみ修正できます。

ビット 2 DMAEN : DMA イネーブル

インダイレクトモードでは、DMA を使用して、QUADSPI_DR レジスタ経由でデータを入出力することができます。DMA 転送は、FTF (FIFO 閾値フラグ) がセットされた時点で開始されます。

0 : インダイレクトモードでは DMA は無効です。

1 : インダイレクトモードでは DMA は有効です。

ビット 1 ABORT : アバートリクエスト

このビットは進行中のコマンドシーケンスを停止します。すべてのアバートが完了すると自動的にリセットされます。

このビットは現在の転送を中止させます。

ポーリングモードまたはメモリマップドモードでは、このビットは APM ビットまたは DM ビットもリセットします。

0 : アバートは要求されていません。

1 : アバートが要求されています。

ビット 0 EN : 有効化

QUADSPI を有効にします。

0 : QUADSPI は無効です。

1 : QUADSPI は有効です。

12.5.2 QUADSPI デバイス設定レジスタ (QUADSPI_DCR)

アドレスオフセット : 0x0004

リセット値 : 0x0000 0000

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

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **FSIZE[4:0]** : フラッシュメモリサイズ

このフィールドは、次の式を使用して外部メモリのサイズを定義します。

フラッシュメモリ内のバイト数 = $2^{FSIZE+1}$

FSIZE+1 はフラッシュメモリのアドレスを指定するために必要とされる有効なアドレスビット数です。フラッシュメモリの容量はインダイレクトモードでは最大 4 GB (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 MB に制限されます。

DFM = 1 の場合、FSIZE は 2 つのフラッシュメモリの総容量を示します。

このフィールドは、BUSY が 0 のときにのみ修正できます。

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **CSHT[2:0]** : チップセレクトハイ時間

CSHT+1 は、フラッシュメモリに対して発行されるコマンド間でチップセレクト (nCS) がハイに保たなければならない CLK サイクルの最小数を定義します。

0 : nCS は、フラッシュメモリコマンド間の少なくとも 1 サイクルの間はハイに保持されます。

1 : nCS は、フラッシュメモリコマンド間の少なくとも 2 サイクルの間はハイに保持されます。

...

7 : nCS は、フラッシュメモリコマンド間の少なくとも 8 サイクルの間はハイに保持されます。

このフィールドは、BUSY が 0 のときにのみ修正できます。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CKMODE** : モード 0 / モード 3

このビットは、コマンド間での CLK のレベルを示します (nCS = 1 のとき)。

0 : CLK は nCS がハイである間はローに保持される必要があります (チップセレクトがリリースされる)。これは、モード 0 と呼ばれます。

1 : CLK は nCS がハイである間はハイに保持される必要があります (チップセレクトがリリースされる)。これは、モード 3 と呼ばれます。

このフィールドは、BUSY が 0 のときにのみ修正できます。

12.5.3 QUADSPI ステータスレジスタ (QUADSPI_SR)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
		r	r	r	r	r	r			r	r	r	r	r	r

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **FLEVEL[5:0]** : FIFO レベル

このフィールドは FIFO 内に保持されている有効バイト数を示します。FIFO が空の場合、FLEVEL = 0、フルになっている場合、32です。メモリマップドモードおよび自動ステータスポーリングモードでは、FLEVEL はゼロです。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **BUSY** : ビジー

このビットは動作が進行中のときにセットされます。このビットは、フラッシュメモリの動作が終わり、FIFO が空になると自動的にクリアされます。

ビット 4 **TOF** : タイムアウトフラグ

このビットは、タイムアウトが発生するとセットされます。CTOF に 1 を書き込むことによってクリアされます。

ビット 3 **SMF** : ステータス一致フラグ

このビットは、マスクされていない受信データが一致レジスタ (QUADSPI_PSMAR) の対応するビットと一致している場合に、自動ポーリングモードでセットされます。CSMF に 1 を書き込むことによってクリアされます。

ビット 2 **FTF** : FIFO 閾値フラグ

インダイレクトモードでは、このビットは FIFO 閾値に達した時点で、またはフラッシュメモリからの読み出し完了後に FIFO 内にデータが残っている場合にセットされます。閾値条件が真でなくなるとすぐに自動的にクリアされます。

自動ポーリングモードでは、このビットはステータスレジスタが読み出されるたびにセットされ、データレジスタが読み出された時点でクリアされます。

ビット 1 **TCF** : 転送完了フラグ

このビットは、プログラムされたデータ数が転送された時点でインダイレクトモードでセットされるか、または転送が中止された時点でいずれかのモードでセットされます。また、CTCF に 1 を書き込むことによってクリアされます。

ビット 0 **TEF** : 転送エラーフラグ

このビットは、インダイレクトモードで無効なアドレスにアクセスされているときに、インダイレクトモードでセットされます。CTEF に 1 を書き込むことによってクリアされます。

12.5.4 QUADSPI フラグクリアレジスタ (QUADSPI_FCR)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
											w1o	w1o		w1o	w1o

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CTOF** : タイムアウトフラグのクリア

このビットに 1 を書き込むと、QUADSPI_SR レジスタの TOF フラグがクリアされます。

ビット 3 **CSMF** : ステータス一致フラグのクリア

このビットに 1 を書き込むと、QUADSPI_SR レジスタの SMF フラグがクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CTCF** : 転送完了フラグのクリア

このビットに 1 を書き込むと、QUADSPI_SR レジスタの TCF フラグがクリアされます。

ビット 0 **CTEF** : 転送エラーフラグのクリア

このビットに 1 を書き込むと、QUADSPI_SR レジスタの TEF フラグがクリアされます。

12.5.5 QUADSPI データ長レジスタ (QUADSPI_DLR)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

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

ビット 31:0 **DL[31:0]** : データ長

インダイレクトモードおよびステータスポーリングモードで取得されるデータ数 (値 + 1) ステータスポーリングモードの場合、3 より大きい値 (4 バイトを示す) を使用する必要があります。インダイレクトモードのすべて1は、不定長を意味します。この場合、FSIZE によって定義されているメモリの終了まで QUADSPI が継続します。

0x0000_0000 : 1 バイトが転送されます。

0x0000_0001 : 2 バイトが転送されます。

0x0000_0002 : 3 バイトが転送されます。

0x0000_0003 : 4 バイトが転送されます。

...

0xFFFF_FFFD : 4,294,967,294 (4G-2) バイトが転送されます。

0xFFFF_FFFE : 4,294,967,295 (4G-1) バイトが転送されます。

0xFFFF_FFFF : 不定長 -- すべてのバイトがフラッシュメモリの終了 (FSIZE によって定義) まで転送されます。FSIZE = 0x1F の場合、読み出しが無限に続けられます。

デュアルフラッシュモード (DFM=1) では、DL[0] は、たとえこのビットに 0 が書き込まれても 1 のままとなります。したがって、アクセスのたびに必ず偶数バイトが転送されます。

このフィールドはメモリマップドモード (FMODE = 10) では効果がありません。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.6 QUADSPI 通信設定レジスタ (QUADSPI_CCR)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DDRM	DHHC	Res.	SIOO	FMODE[1:0]		DMODE		Res.	DCYC[4:0]				ABSIZE		
rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INSTRUCTION[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **DDRM** : データレートモードを倍にします。

このビットは、アドレス、オルタナートバイト、およびデータフェーズの DDR モードをセットします。

0 : DDR モードは無効です。

1 : DDR モードは有効です。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 30 **DHHC** : DDR ホールド

DDR モードで、データの出力を 1/4 QUADSPI 出カクロックサイクルだけ遅延させます。

0 : アナログ遅延を使用してデータ出力を遅延させます。

1 : データの出力を 1/4 QUADSPI 出カクロックサイクルだけ遅延させます。

この機能は、DDR モードでのみアクティブです。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **SIOO** : モードにつき 1 回のみ命令を送信します。

[セクション 12.3.11 : 1 回限りの命令の送信 \(291 ページ\)](#) を参照してください。このビットは、IMODE = 00 のときには効果がありません。

0 : トランザクションごとに命令を送信します。

1 : 最初のコマンドに対してのみ命令を送信します。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 27:26 **FMODE[1:0]** : 機能モード

このフィールドは、動作の QUADSPI 機能モードを定義します。

00 : インダイレクト書き込みモード

01 : インダイレクト読み出しモード

10 : 自動ポーリングモード

11 : メモリマップドモード

DMAEN = 1 の場合、対応するチャネル用の DMA コントローラは FMODE 値を変更する前に無効にする必要があります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 25:24 **DMODE[1:0]** : データモード

このフィールドは、動作のデータフェーズのモードを次のように定義します。

00 : データはありません。

01 : データは 1 本のライン上にあります。

10 : データは 2 本のライン上にあります。

11 : データは 4 本のライン上にあります。

また、このフィールドは動作のダミーフェーズモードも決定します。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **DCYC[4:0]** : ダミーサイクル数

このフィールドはダミーフェーズの処理時間を定義します。SDR モードと DDR モードの両方で、CLK サイクルの数を指定します (0~31)。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 17:16 **ABSIZE[1:0]** : オルタネートバイトサイズ

このビットはオルタネートバイトのサイズを定義します。

00 : 8 ビットオルタネートバイト

01 : 16 ビットオルタネートバイト

10 : 24 ビットオルタネートバイト

11 : 32 ビットオルタネートバイト

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 15:14 **ABMODE[1:0]** : オルタネートバイトモード

このフィールドは、動作のオルタネートバイトフェーズのモードを次のように定義します。

00 : オルタネートバイトはありません。

01 : オルタネートバイトは 1 本のライン上にあります。

10 : オルタネートバイトは 2 本のライン上にあります。

11 : オルタネートバイトは 4 本のライン上にあります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 13:12 **ADSIZE[1:0]** : アドレスサイズ

このビットはアドレスサイズを定義します。

00 : 8 ビットアドレス

01 : 16 ビットアドレス

10 : 24 ビットアドレス

11 : 32 ビットアドレス

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

- ビット 11:10 **ADMODE[1:0]** : アドレスモード
- このフィールドは、動作のアドレスフェーズのモードを次のように定義します。
- 00 : アドレスはありません。
 - 01 : アドレスは 1 本のライン上にあります。
 - 10 : アドレスは 2 本のライン上にあります。
 - 11 : アドレスは 4 本のライン上にあります。
- このフィールドは、BUSY が 0 のときにのみ書き込みができます。
- ビット 9:8 **IMODE[1:0]** : 命令モード
- このフィールドは、動作の命令フェーズのモードを次のように定義します。
- 00 : 命令はありません。
 - 01 : 命令は 1 本のライン上にあります。
 - 10 : 命令は 2 本のライン上にあります。
 - 11 : 命令は 4 本のライン上にあります。
- このフィールドは、BUSY が 0 のときにのみ書き込みができます。
- ビット 7 : 0 **INSTRUCTION[7: 0]** : 命令
- 外部 SPI デバイスに送信される命令
- このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.7 QUADSPI アドレスレジスタ (QUADSPI_AR)

- アドレスオフセット : 0x0018
- リセット値 : 0x0000 0000

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

- ビット 31:0 **ADDRESS[31 0]** : アドレス
- 外部フラッシュメモリに送信されるアドレス
- このフィールドへの書き込みは、BUSY = 0 または FMODE = 11（メモリマップドモード）の場合は無視されます。
- デュアルフラッシュモードでは、アドレスは常に偶数でなければならないので、ADDRESS[0] は自動的に 0 に固定されます。

12.5.8 QUADSPI オルタネートバイトレジスタ (QUADSPI_ABR)

アドレスオフセット : 0x001C

リセット値 : 0x0000 0000

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

ビット 31:0 **ALTERNATE[31: 0]** : オルタネートバイト

アドレスの直後に外部 SPI デバイスに送信されるオプションのデータ
このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.9 QUADSPI データレジスタ (QUADSPI_DR)

アドレスオフセット : 0x0020

リセット値 : 0x0000 0000

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

ビット 31:0 **DATA[31: 0]** : データ

外部 SPI デバイスに対して送受信されるデータ
インダイレクト書き込みモードでは、このレジスタに書き込まれたデータはいったん FIFO に格納されてから、データフェーズ時にフラッシュメモリに送信されます。FIFO がフルの場合、書き込まれるデータ量を受け入れるだけの十分な空間が FIFO にできるまで、書き込み動作は停止されません。
インダイレクト読み出しモードでは、このレジスタを読み出すことで (FIFO 経由)、フラッシュメモリから受信したデータが提供されます。FIFO のバイト数が読み出し動作で要求したバイト数ほど多くない場合で、BUSY = 1 のとき、データ量が十分になるか、転送が完了するかのいずれかが早く起こるほうまで、読み出し動作は停止されます。
自動ポーリングモードでは、このレジスタはフラッシュメモリから読み出された最後のデータを格納します (マスキングなし)。
このレジスタへのワード、ハーフワード、バイトアクセスはサポートされます。インダイレクト書き込みモードでの書き込みにより FIFO に追加されるバイト数は、バイト書き込みで 1 バイト、ハーフワード書き込みで 2 バイト、ワード書き込みで 4 バイトです。同様に、インダイレクト読み出しモードでの読み出しにより FIFO から削除されるバイト数は、バイト読み出しで 1 バイト、ハーフワード読み出しで 2 バイト、ワード読み出しで 4 バイトです。インダイレクトモードでのアクセスは、このレジスタのボトムに整列されなければなりません。つまり、バイト読み出しでは DATA[7:0] が、ハーフワード読み出しでは DATA[15:0] が読み出されます。



12.5.10 QUADSPI ポーリングステータスマスクレジスタ (QUADSPI_PSMKR)

アドレスオフセット : 0x0024

リセット値 : 0x0000 0000

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

ビット 31:0 **MASK[31:0]** : ステータスマスク

ポーリングモードで受信したステータスバイトに適用されるマスク

ビット n の場合

0 : 自動ポーリングモードで受信したデータのビット n はマスクされ、その値は一致ロジックでは考慮されません。

1 : 自動ポーリングモードで受信したデータのビット n はマスクされず、その値は一致ロジックで考慮されます。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.11 QUADSPI ポーリングステータス一致レジスタ (QUADSPI_PSMAR)

アドレスオフセット : 0x0028

リセット値 : 0x0000 0000

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

ビット 31:0 **MATCH[31:0]** : ステータス一致

一致を得るためにマスクされたステータスレジスタと比較される値

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.12 QUADSPI ポーリングインターバルレジスタ (QUADSPI_PIR)

アドレスオフセット : 0x002C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **INTERVAL[15: 0]** : ポーリングインターバル

自動ポーリングフェーズ時の読み出し間の CLK サイクル数

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.13 QUADSPI 低電力タイムアウトレジスタ (QUADSPI_LPTR)

アドレスオフセット : 0x0030

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **TIMEOUT[15: 0]** : タイムアウト期間

メモリマップドモードでのアクセス終了のたびに、QUADSPI はその後のバイトをプリフェッチし、それらを FIFO に保持します。このフィールドは、FIFO がフルになってから nCS を立ち上げるまでに QUADSPI が待つ CLK サイクルの数を示し、フラッシュメモリを消費電力がより少ない状態にします。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

12.5.14 QUADSPI レジスタマップ

表 72. QUADSPI レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	QUADSPI_CR	PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE	Res.	Res.	Res.	FTHRES [4:0]				FSEL	DFM	Res.	5	SSHIFT	TCEN	DMAEN	ABORT	EN
	リセット値	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0		Res.	Res.	Res.	0	0	0	0	0	0	0	0	0	0	0	0
0x0004	QUADSPI_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSIZE[4:0]				Res.	Res.	Res.	Res.	Res.	CSHT			Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKMODE	
	リセット値												0	0	0	0	0		Res.	Res.	Res.			0	0	0						0	
0x0008	QUADSPI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLEVEL[6:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF		
	リセット値																	0	0	0	0	0	0	0			0	0	0	0	0	0	
0x000C	QUADSPI_FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	CTCF	CTEF	
	リセット値																											0	0			0	0
0x0010	QUADSPI_DLR	DL[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0014	QUADSPI_CCR	DDRM	DHHC	Res.	SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]				ABSIZE[1:0]		ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]								
	リセット値	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	QUADSPI_AR	ADDRESS[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	QUADSPI_ABR	ALTERNATE[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	QUADSPI_DR	DATA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	QUADSPI_PSMKR	MASK[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	QUADSPI_PSMAR	MATCH[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	QUADSPI_PIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTERVAL[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	QUADSPI_LPTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEOUT[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2](#) を参照してください。

13 アナログデジタルコンバータ (ADC)

13.1 ADC の概要

この 12 ビット ADC は、逐次比較型アナログデジタルコンバータです。最大 19 の多重化チャネルを持ち、16 の外部ソース、2 つの内部ソース、および V_{BAT} チャネルの測定を行うことができます。これらのチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めで 16 ビットのデータレジスタに格納されます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

13.2 ADC の主な機能

- 12 ビット、10 ビット、8 ビット、または 6 ビットの設定可能な分解能
- 変換終了時、インジェクト変換終了時、および、アナログウォッチドッグイベントまたはオーバーランイベント時に割り込みを生成
- シングルおよび連続変換モード
- チャネル 0 からチャネル“n”を自動変換するためのスキャンモード
- 組み込みのデータコヒーレンシによるデータ配置
- チャネル単位でプログラム可能なサンプリング時間
- レギュラ変換とインジェクト変換の両方について極性設定可能な外部トリガオプション
- 不連続モード
- デュアル／トリプルインタリーブモード変換間の設定可能な遅延
- ADC 電源仕様：高速時 2.4 V から 3.6 V、低速時 1.8 V から低下可能
- ADC 入力電圧範囲： $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- レギュラチャネル変換中の DMA リクエストの生成

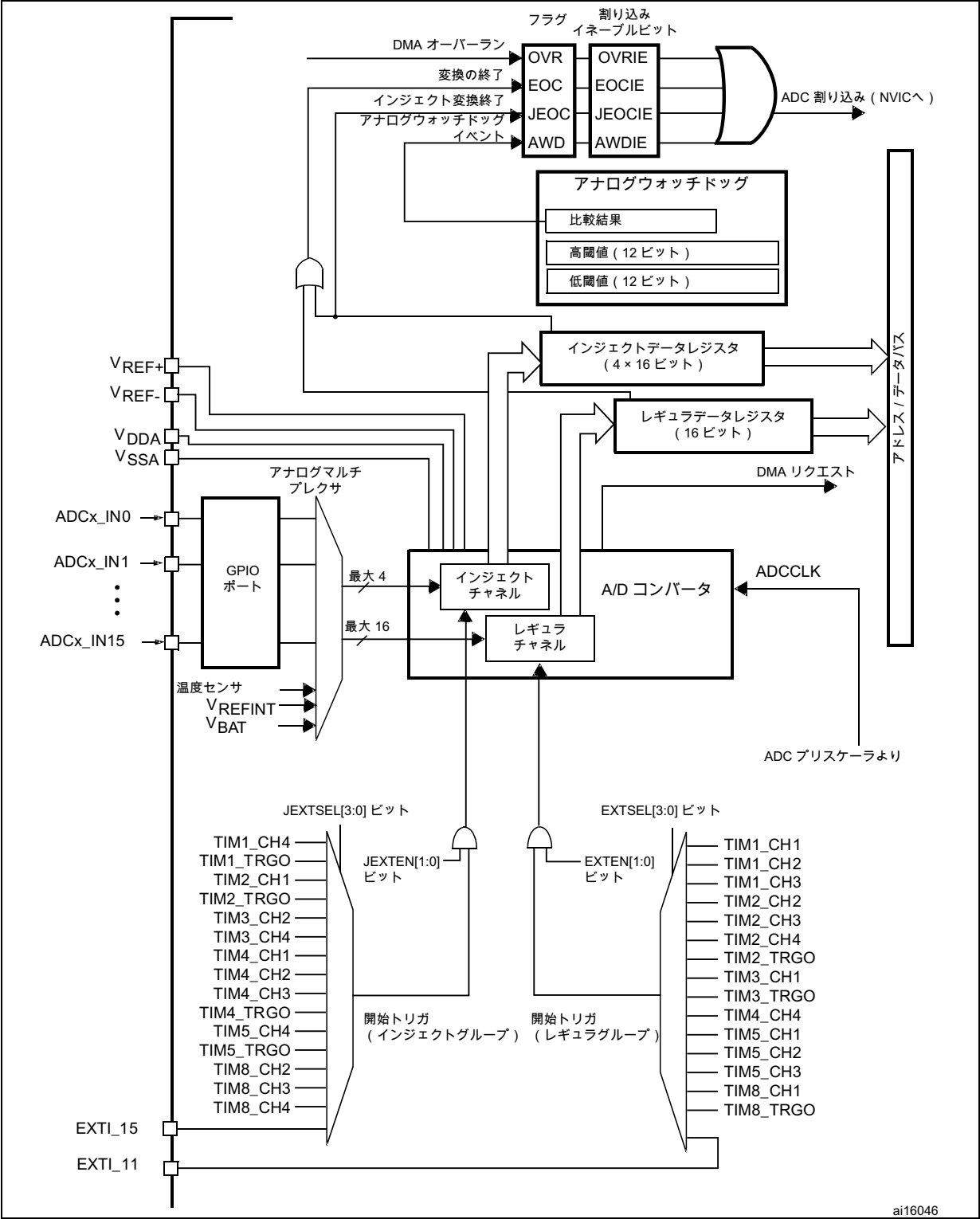
 59 に、ADC のブロック図を示します。

注： V_{REF-} ピンが用意されているパッケージでは、このピンを V_{SSA} に固定する必要があります。

13.3 ADC の機能詳細

 59 に 1 つの ADC ブロック図を、表 73 に ADC ピンの説明を示します。

図 59. 1つの ADC ブロック図



ai16046

表 73. ADC ピン

名前	信号タイプ	説明
V _{REF+}	入力、アナログ基準電圧正	ADC のハイレベル／正基準電圧 $1.8\text{ V} \leq V_{\text{REF+}} \leq V_{\text{DDA}}$
V _{DDA}	入力、アナログ電源供給	アナログ電源供給、V _{DD} に等しく、 $2.4\text{ V} \leq V_{\text{DDA}} \leq V_{\text{DD}}$ (3.6 V) フルスピード時 $1.8\text{ V} \leq V_{\text{DDA}} \leq V_{\text{DD}}$ (3.6 V) 低速時
V _{REF-}	入力、アナログ基準電圧負	ADC のローレベル／負基準電圧 $V_{\text{REF-}} = V_{\text{SSA}}$
V _{SSA}	入力、アナログ供給グラウンド	V _{SS} に等しいアナログ電源供給のグラウンド
ADCx_IN[15:0]	アナログ入力信号	16 チャンネルのアナログ入力

13.3.1 ADC のオン／オフ制御

ADC は、ADC_CR2 レジスタの ADON ビットをセットすることで、電源オンの状態になります。ADON ビットが最初にセットされると、ADC はパワーダウンモードからウェイクアップします。

SWSTART ビットまたは JSWSTART ビットのいずれかがセットされると変換が開始します。

ADON ビットをクリアすることによって、変換を停止し、ADC をパワーダウンモードにできます。このモードでは、ADC は電力をほとんど消費しません（数 μA 程度）。

13.3.2 ADC クロック

この ADC には 2 つのクロックスキームがあります。

- アナログ回路のクロック：ADCCLK
このクロックは、プログラム可能なプリスケアラによって分周された APB2 クロックによって生成され、ADC は $f_{\text{CLK2}}/2$ 、 $/4$ 、 $/6$ または $/8$ で動作することができます。ADCCLK の最大値に関してはデータシートを参照してください。
- デジタルインタフェース用のクロック（レジスタの読み書きアクセスに使用）
このクロックは APB2 クロックと同一です。デジタルインタフェースクロックは、RCC APB2 ペリフェラルクロックイネーブルレジスタ（RCC_APB2ENR）を使用して ADC ごとに有効／無効にすることができます。

13.3.3 チャンネル選択

16 の多重化チャンネルがあります。変換は、レギュラとインジェクトの 2 つのグループに構成することができます。各グループは、任意のチャンネルに対して任意の順序で行うことができる一連の変換で構成されます。たとえば、次のような順で変換順を設定することができます：ADC_IN3、ADC_IN8、ADC_IN2、ADC_IN2、ADC_IN0、ADC_IN2、ADC_IN2、ADC_IN15

- **レギュラグループ**は、最大 16 の変換で構成されます。レギュラチャンネルと変換シーケンス内での順序は、ADC_SQRx レジスタで選択する必要があります。レギュラグループの合計変換数は、ADC_SQR1 レジスタの L[3:0] ビットに書き込む必要があります。
- **インジェクトグループ**は、最大 4 つの変換で構成されます。インジェクトチャンネルと変換シーケンス内での順序は、ADC_JSQR レジスタで選択する必要があります。インジェクトグループの合計変換数は、ADC_JSQR レジスタの L[1:0] ビットに書き込む必要があります。

変換中に ADC_SQRx または ADC_JSQR レジスタが変更された場合、現在の変換はリセットされ、新しい開始パルスが ADC に送られて、新しく選ばれたグループを変換します。

温度センサ、V_{REFINT}、および V_{BAT} 内部チャネル

- 温度センサは VBAT と共用で ADC1_IN18 チャネルに内部的に接続されています。一度に選択できるのは、温度センサまたは VBAT のうち、どちらか 1 つだけです。温度センサと VBAT 変換を同時に設定すると、VBAT 変換のみ行われます。

内部基準電圧 VREFINT は ADC1_IN17 に接続されています。

V_{BAT} チャネルは ADC1_IN18 チャネルに接続されています。これも、インジェクトまたはレギュラチャネルとして変換を行うことができます。

13.3.4 シングル変換モード

シングル変換モードでは、ADC は変換を 1 回行います。CONT ビットが 0 の時、次のいずれかによってこのモードが開始します。

- ADC_CR2 レジスタの SWSTART ビットをセットすること (レギュラチャネルのみ)
- JSWSTART ビットをセットすること (インジェクトチャネル)
- 外部トリガ (レギュラまたはインジェクトチャネル)

選択されたチャネルの変換が完了すると、

- レギュラチャネルが変換された場合：
 - 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
 - EOC (end of conversion) フラグがセットされます。
 - EOCIE ビットがセットされている場合、割り込みが生成されます。
- インジェクトチャネルが変換された場合：
 - 変換されたデータは 16 ビットの ADC_JDR1 レジスタに格納されます。
 - JEOP (end of conversion injected) フラグがセットされます。
 - JEOCIE ビットがセットされている場合、割り込みが生成されます。

その後、ADC が停止します。

13.3.5 連続変換モード

連続変換モードでは、ADC は 1 回の変換が終了すると次の変換に移ります。CONT ビットが 1 の時、外部トリガまたは ADC_CR2 レジスタの SWSTRT ビットをセットすることによって、このモードが開始されます (レギュラチャネルのみ)。

各変換後、

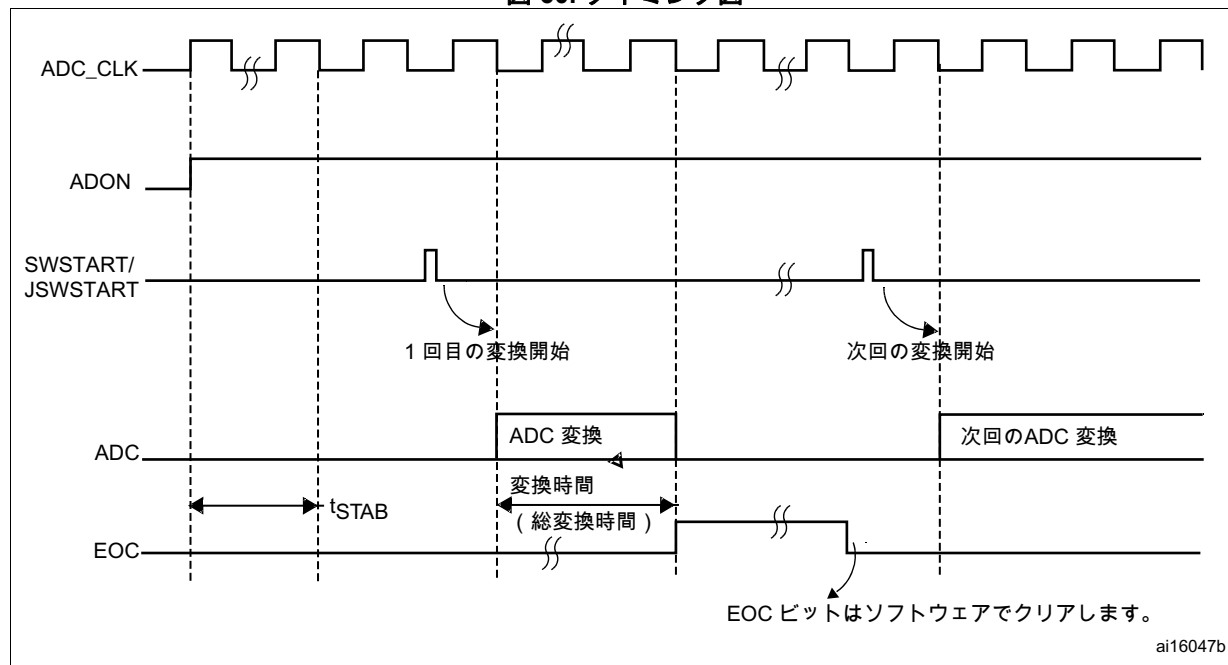
- レギュラチャネルのグループが変換された場合：
 - 最後に変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
 - EOC (end of conversion) フラグがセットされます。
 - EOCIE ビットがセットされている場合、割り込みが生成されます。

注： インジェクトチャネルを連続して変換することはできません。唯一の例外は、連続モードに設定されたレギュラチャネルの後にインジェクトチャネルを自動的に変換するように設定されている (JAUTO ビットを使用) 場合です。自動インジェクションセクションを参照してください。

13.3.6 タイミング図

図 60 に示すように、ADC は正確な変換を開始する前に、安定時間 t_{STAB} を必要とします。ADC 変換の開始後 15 クロックサイクルで EOC フラグがセットされ、変換されたデータが 16 ビットの ADC データレジスタに格納されます。

図 60. タイミング図



ai16047b

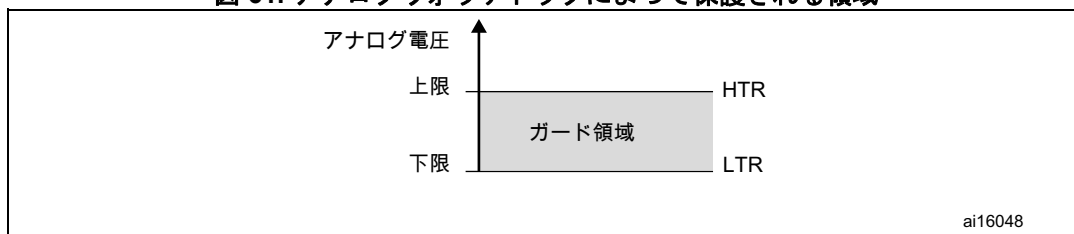
13.3.7 アナログウォッチドッグ

アナログウォッチドッグ (AWD) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。これらの閾値は、ADC_HTR および ADC_LTR 16 ビットレジスタの下位 12 ビットに設定します。ADC_CR1 レジスタの AWDIE ビットを使用して、割り込みを有効にできます。

閾値は、ADC_CR2 レジスタの ALIGN ビットによって選択された配置とは無関係です。アナログ電圧は配置される前に低閾値および高閾値と比較されます。

表 74 に、1 つまたは複数のチャンネル上でアナログウォッチドッグを有効にするためにどのようにして ADC_CR1 レジスタを設定するかを示します。

図 61. アナログウォッチドッグによって保護される領域



ai16048

表 74. アナログウォッチドッグチャンネル選択

アナログウォッチドッグによって保護されるチャンネル	ADC_CR1 レジスタ制御ビット (x = 無視)		
	AWDSGL ビット	AWDEN ビット	JAWDEN ビット
なし	x	0	0
すべてのインジェクトチャンネル	0	0	1
すべてのレギュラチャンネル	0	1	0
すべてのレギュラおよびインジェクトチャンネル	0	1	1
1 つの ⁽¹⁾ インジェクトチャンネル	1	0	1
1 つの ⁽¹⁾ レギュラチャンネル	1	1	0
1 つの ⁽¹⁾ レギュラまたはインジェクトチャンネル	1	1	1

1. AWDCH[4:0] ビットによって選択

13.3.8 スキャンモード

このモードは、アナログチャンネルのグループをスキャンするために使用されます。

スキャンモードを選択するには、ADC_CR1 レジスタの SCAN ビットをセットします。このビットがセットされると、ADC は ADC_SQRx レジスタ（レギュラチャンネルの場合）または ADC_JSQR レジスタ（インジェクトチャンネルの場合）で選択されたすべてのチャンネルをスキャンします。グループの各チャンネルに対して 1 回の変換が行われます。各変換が完了すると、グループ内の次のチャンネルが自動的に変換されます。CONT ビットがセットされている場合、グループの最後の選択チャンネルでレギュラチャンネル変換は停止せず、選択チャンネルの最初から続行します。

DMA ビットがセットされている場合、変換されたデータ（ADC_DR レジスタに格納されている）は、各レギュラチャンネル変換後、ダイレクトメモリアクセス（DMA）コントローラを使用して、レギュラチャンネルグループから SRAM に転送されます。

次の場合に ADC_SR レジスタの EOC ビットがセットされます。

- EOCs ビットが 0 にクリアされている場合、各レギュラグループシーケンス終了後
- EOCs ビットが 1 にセットされている場合、各レギュラチャンネル変換終了後

インジェクトチャンネルから変換されたデータは常に ADC_JDRx レジスタに格納されます。

13.3.9 インジェクトチャンネルの管理

トリガインジェクション

トリガインジェクションを使用するには、ADC_CR1 レジスタの JAUTO ビットがクリアされていなければなりません。

1. 外部トリガによって、または、ADC_CR2 レジスタの SWSTART ビットをセットすることによって、レギュラチャンネルのグループの変換を開始します。
2. レギュラチャンネルグループの変換中に外部インジェクトトリガが発生した場合または JSWSTART ビットがセットされた場合、現在の変換はリセットされ、インジェクトチャンネルシーケンスはスキャンワンスモードに切り替わります。
3. その後、レギュラチャンネルグループのレギュラ変換が、最後に中断されたレギュラ変換から再開されます。
インジェクト変換中にレギュライベントが発生した場合、インジェクト変換は中断されず、イン

ジェクトシーケンスの終了時にレギュラシーケンスが実行されます。図 62 に対応するタイミング図を示します。

注： トリガインジェクションを使用する場合は、トリガイベントの間隔がインジェクションシーケンスより長くなるようにしなければなりません。たとえば、シーケンスの長さが 30 ADC クロックサイクルの場合 (3 クロック周期のサンプリング時間を持つ 2 つの変換)、トリガの最小間隔は 31 ADC クロックサイクルでなければなりません。

自動インジェクション

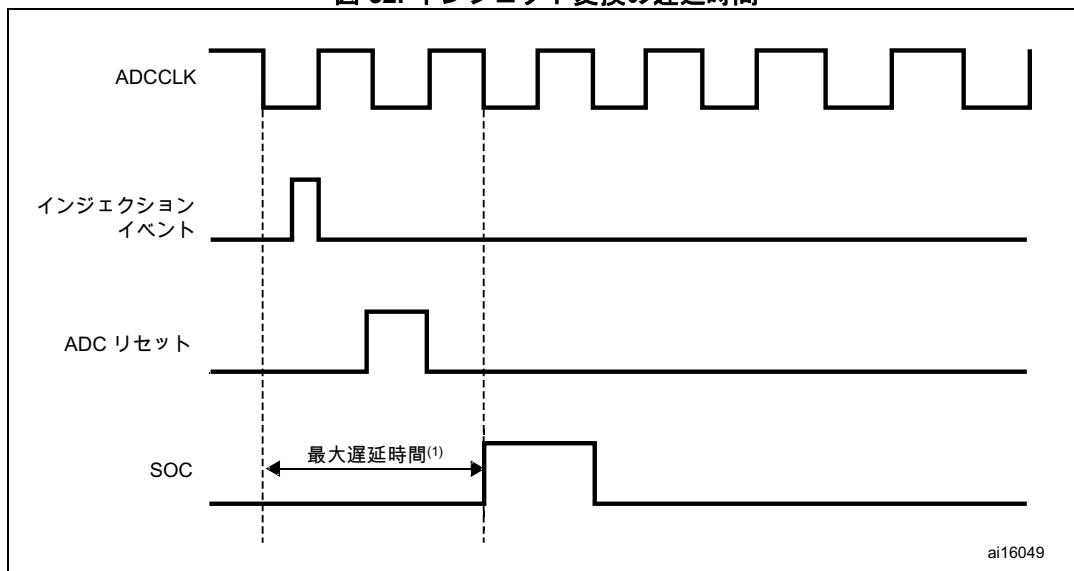
JAUTO ビットがセットされている場合、レギュラグループチャネルの変換後、インジェクトグループチャネルが自動的に変換されます。これを使用して、ADC_SQRx レジスタおよび ADC_JSQR レジスタにプログラミングされた最大 20 回の変換シーケンスを実行することができます。

このモードでは、インジェクトチャネルの外部トリガは無効でなければなりません。

JAUTO ビットに加えて CONT ビットもセットされている場合、レギュラチャネルに続いてインジェクトチャネルが連続して変換されます。

注： 自動インジェクトモードと不連続モードを同時に使用することはできません。

図 62. インジェクト変換の遅延時間



1. 最大遅延時間の値は、STM32F412xx のデータシートの電気的特性に記載されています。

13.3.10 不連続モード

レギュラグループ

このモードは、ADC_CR1 レジスタの DISCEN ビットをセットすることによって有効になります。これを使用して、ADC_SQRx レジスタで選択された変換シーケンスの一部である短い変換シーケンス ($n \leq 8$) を変換できます。n の値は、ADC_CR1 レジスタの DISCNUM[2:0] ビットに書き込むことによって指定します。

外部トリガが発生すると、ADC_SQRx レジスタで選択された次の n 個の変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_SQR1 レジスタの L[3:0] ビットによって定義されます。

例：

- $n = 3$ 、変換されるチャネル = 0、1、2、3、6、7、9、10
- 最初のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。
- 2 番目のトリガ：3、6、7 が変換されます。各変換で EOC イベントが生成されます。
- 3 番目のトリガ：9、10 が変換されます。各変換で EOC イベントが生成されます。
- 4 番目のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。

注： **レギュラグループが不連続モードで変換されるときには、ロールオーバーは発生しません。**
すべてのサブグループが変換されると、次のトリガでは、最初のサブグループの変換が開始されます。
上の例では、4 番目のトリガで最初のサブグループのチャネル 0、1、および 2 が再変換されます。

インジェクトグループ

このモードは、ADC_CR1 レジスタの JDISCEN ビットをセットすることで有効になります。これを使用して、ADC_JSQR レジスタで選択されたシーケンスを、外部トリガイイベント後に 1 チャネルずつ変換することができます。

外部トリガが発生すると、ADC_JSQR レジスタで選択された次のチャネル変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_JSQR レジスタの JL[1:0] ビットによって定義されます。

例：

- n = 1、変換されるチャネル = 1、2、3
- 最初のトリガ：チャネル 1 が変換されます。
- 2 番目のトリガ：チャネル 2 が変換されます。
- 3 番目のトリガ：チャネル 3 が変換され、JEOC イベントが生成されます。
- 4 番目のトリガ：チャネル 1 が変換されます。

注： **すべてのインジェクトチャネルが変換されると、次のトリガでは、最初のインジェクトチャネルの変換が開始されます。** **上の例では、4 番目のトリガで最初のインジェクトチャネル 1 が再変換されます。**
自動インジェクトモードと不連続モードを同時に使用することはできません。
レギュラグループとインジェクトグループの両方で不連続モードを同時に設定することはできません。 **不連続モードは、1 つのグループ変換に対してのみ有効にする必要があります。**

13.4 データの配置

ADC_CR2 レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。データは、[図 63](#) および [図 64](#) に示すように、右詰めまたは左詰めに配置できます。

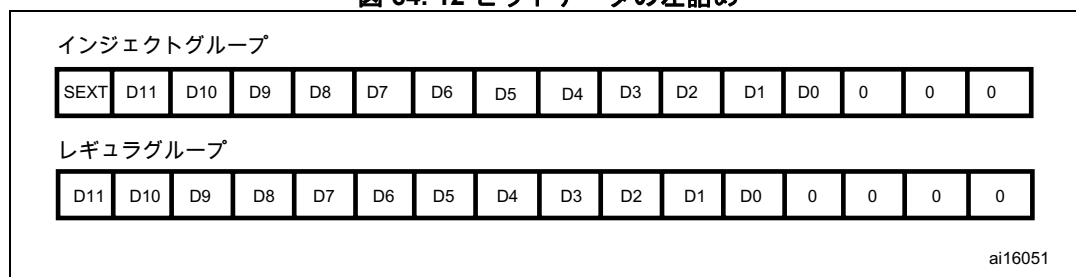
インジェクトグループチャネルの変換後のデータ値は、ADC_JOFRx レジスタに書き込まれたユーザ定義オフセットだけ減算されるので結果は負の値になることもあります。SEXT ビットは、拡張符号値を表します。

レギュラグループのチャネルの場合、オフセットは引かれられないので、12 ビットのみが有効です。

図 63. 12 ビットデータの右詰め

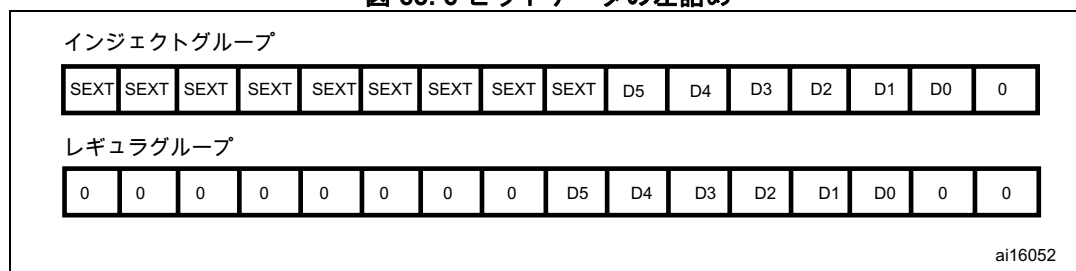


図 64. 12 ビットデータの左詰め



特殊なケース：左詰めの場合、バイト単位に配置される場合で分解能が 6 ビットに設定されている場合 ([図 65](#)) を除きデータはハーフワード単位に配置されます。

図 65. 6 ビットデータの左詰め



13.5 チャンネル単位でプログラム可能なサンプリング時間

ADC は、一定の ADCCLK サイクル数だけ入力電圧をサンプリングしますが、この時間は、ADC_SMPR1 および ADC_SMPR2 レジスタの SMP[2:0] ビットを使用して変更できます。チャンネルごとに異なるサンプリング時間でサンプリングすることができます。

合計変換時間は、次のように計算されます。

$$T_{conv} = \text{サンプリング時間} + 12 \text{ サイクル}$$

例：

ADCCLK = 30 MHz、サンプリング時間が 3 サイクルのとき：

$$T_{conv} = 3 + 12 = 15 \text{ サイクル} = 0.5 \mu\text{s} \text{ (APB2 が 60 MHz の場合)}$$

13.6 外部トリガによる変換およびトリガ極性

変換は、外部イベント（タイマキャプチャ、EXTI ラインなど）によってトリガできます。EXTEN[1:0] 制御ビット（レギュラ変換の場合）または JEXTEN[1:0] ビット（インジェクト変換の場合）が “0b00” 以外の場合、選択した極性で外部イベントによって変換をトリガすることができます。表 75 に EXTEN[1:0] と JEXTEN[1:0] の値とトリガ極性の対応を示します。

表 75. トリガ極性の設定

転送元	EXTEN[1:0] / JEXTEN[1:0]
トリガ検出は無効です。	00
立ち上がりエッジで検出します。	01
立ち下がりエッジで検出します。	10
立ち上がりエッジと立ち下がりエッジの両方で検出します。	11

注： 外部トリガの極性は動作中に変更することができます。

EXTSEL[3:0] および JEXTSEL[3:0] 制御ビットを使用してレギュラおよびインジェクトグループの変換をトリガできるイベントを 16 のイベントから選択します。

表 76 に、レギュラ変換に使用できる外部トリガを示します。

表 76. レギュラチャネルの外部トリガ

転送元	タイプ	EXTSEL[3:0]
TIM1_CH1 イベント	オンチップタイマからの内部信号	0000
TIM1_CH2 イベント		0001
TIM1_CH3 イベント		0010
TIM2_CH2 イベント		0011
TIM2_CH3 イベント		0100
TIM2_CH4 イベント		0101
TIM2_TRGO イベント		0110
TIM3_CH1 イベント		0111
TIM3_TRGO イベント		1000
TIM4_CH4 イベント		1001
TIM5_CH1 イベント		1010
TIM5_CH2 イベント		1011
TIM5_CH3 イベント		1100
TIM8_CH1 イベント		1101
TIM8_TRGO イベント		1110
EXTI ライン 11	外部ピン	1111

表 77 に、インジェクト変換に使用できる外部トリガを示します。

表 77. インジェクトチャネルの外部トリガ

転送元	接続タイプ	JEXTSEL[3:0]
TIM1_CH4 イベント	オンチップタイマからの内部信号	0000
TIM1_TRGO イベント		0001
TIM2_CH1 イベント		0010
TIM2_TRGO イベント		0011
TIM3_CH2 イベント		0100
TIM3_CH4 イベント		0101
TIM4_CH1 イベント		0110
TIM4_CH2 イベント		0111
TIM4_CH3 イベント		1000
TIM4_TRGO イベント		1001
TIM5_CH4 イベント		1010
TIM5_TRGO イベント		1011
TIM8_CH2 イベント		1100
TIM8_CH3 イベント		1101
TIM8_CH4 イベント		1110
EXTI ライン 15	外部ピン	1111

ソフトウェアソーストリガイベントは、ADC_CR2 レジスタの SWSTART (レギュラ変換の場合) ビットまたは JSWSTART (インジェクト変換の場合) ビットをセットすることによって生成できます。

レギュラグループ変換は、インジェクトトリガによって中断することが可能です。

注： トリガの選択は動作中に変更することができます。ただし、選択が変更されるとき 1 APB クロックサイクルの期間トリガの検出が無効になります。これは、変更時の誤検出を避けるためです。

13.7 高速変換モード

ADC の分解能を下げることによって、高速変換を行うことができます。RES ビットを使用してデータレジスタで利用できるビット数を選択します。各精度に対する最小変換時間は、次のとおりです。

- 12 ビット : $3 + 12 = 15$ ADCCLK サイクル
- 10 ビット : $3 + 10 = 13$ ADCCLK サイクル
- 8 ビット : $3 + 8 = 11$ ADCCLK サイクル
- 6 ビット : $3 + 6 = 9$ ADCCLK サイクル

13.8 データ管理

13.8.1 DMA の使用

変換されたレギュラチャネルの値は特定のデータレジスタに格納されるので、複数のレギュラチャネルの変換には DMA の使用が便利です。これによって、ADC_DR レジスタにすでに格納されているデータの損失を防ぐことができます。

DMA モードが有効 (ADC_CR2 レジスタの DMA ビットが 1 に設定されている) な場合、レギュラチャネルが変換されるごとに DMA リクエストが生成されます。これにより、変換されたデータを ADC_DR レジスタからソフトウェアで選択した場所へ転送することができます。

これにもかかわらず、データが失われる場合 (オーバーラン)、ADC_SR レジスタの OVR ビットがセットされ割り込みが生成されます (OVRIE イネーブルビットがセットされている場合)。続いて DMA 転送が無効になり、その後の DMA リクエストは受け付けられません。この場合、DMA リクエストが行われると、実行中のレギュラ変換は中断され、その後のレギュラトリガは無視されます。そのときは、必要な変換済みチャネルデータを正しいメモリ位置へ転送するために OVR フラグおよび使用している DMA ストリームの DMAEN ビットをクリアして DMA と ADC を再初期化する必要があります。その後初めて、変換を再開することができ、データ転送を再び有効にすることができます。インジェクトチャネル変換はオーバーランエラーの影響を受けません。

DMA モードで OVR = 1 の場合、最後の有効なデータが転送されたあとに DMA リクエストがブロックされます。つまり、RAM へ転送されたデータはすべて有効とみなすことができます。

最後の DMA 転送 (DMA コントローラの DMA_SxNTR レジスタに設定されている転送数) が終了すると、

- ADC_CR2 レジスタの DDS ビットが 0 にクリアされている場合、DMA コントローラへの新規の DMA リクエストは発生しません (これによりオーバーランエラーの発生を防ぎます)。ただし、DMA ビットはハードウェアによってクリアされることはありません。新たな転送を開始するにはそこに 0 を書き込み、そのあと 1 を書き込む必要があります。
- DDS ビットが 1 にセットされている場合、リクエストの生成は継続することができます。これにより DMA をダブルバッファサーキュラモードに設定することができます。

DMA 使用時に ADC を OVR 状態から回復するには次の手順に従います。

1. DMA を再初期化します (転送先アドレスと NDTR カウンタを調整します)
2. ADC_SR レジスタの ADC OVR ビットをクリアします。
3. ADC をトリガして変換を開始します。

13.8.2 DMA を使用しない変換シーケンスの管理

変換が十分遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、変換シーケンスの終わりだけではなく各変換が終わるたびに EOC ステータスビットをセットするため、ADC_CR2 レジスタの EOCS ビットをセットする必要があります。EOCS = 1 のとき、オーバーラン検出は自動的に有効になります。したがって、変換が終了するごとに EOC がセットされ ADC_DR レジスタを読み出すことができます。オーバーラン管理は DMA を使用するときと同じです。

EOCS がセットされているとき ADC を OVR 状態から回復するには次の手順に従います。


1. ADC_SR レジスタの ADC OVR ビットをクリアします。
2. ADC をトリガして変換を開始します。

13.8.3 DMA およびオーバーラン検出を使用しない変換

毎回データの読み出しをせずに ADC に 1 つまたは複数のチャンネルを変換させると便利な場合があります (たとえば、アナログウォッチドッグがある場合)。そのためには、DMA を無効 (DMA = 0) にし、シーケンスの終了時にのみ EOC ビットをセットする必要があります (EOCS = 0)。この設定では、オーバーラン検出が無効になります。

13.9 温度センサ

温度センサを使用して、デバイスの温度 (T_A) を測定できます。

 66 に、温度センサのブロック図を示します。

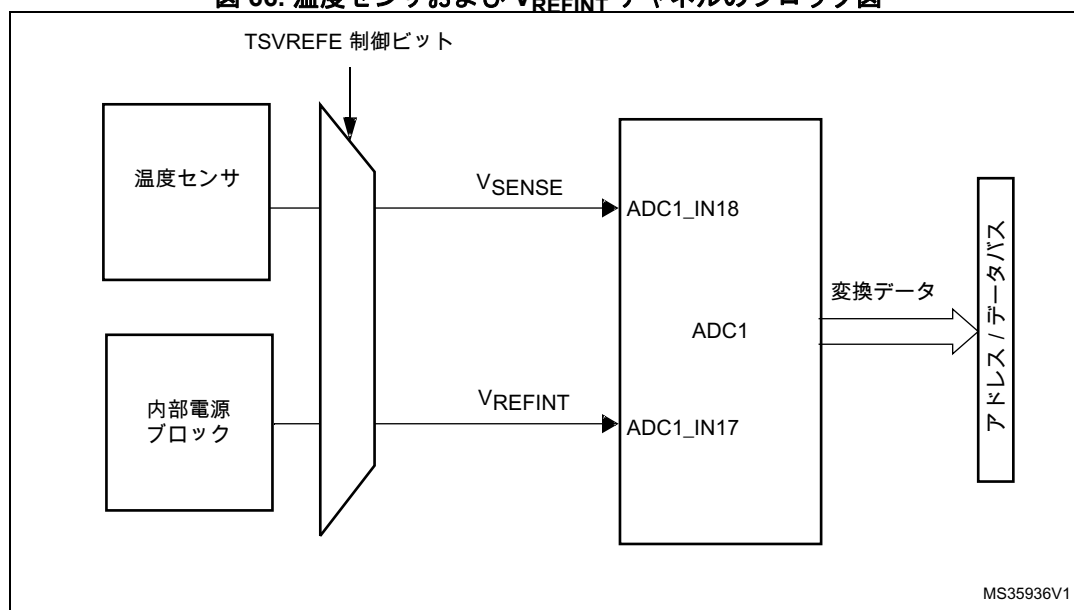
使用しないときには、センサをパワーダウンモードにすることができます。

注： 両方の内部チャンネル、すなわち、ADC1_IN18 (温度センサ) および ADC1_IN17 (VREFINT) の変換を有効にするには、TSVREFE ビットをセットする必要があります。

主な特長

- サポートしている温度範囲：-40 ~ 125 °C
- 精度：±1.5 °C

図 66. 温度センサおよび V_{REFINT} チャンネルのブロック図



1. V_{SENSE} が ADC1_IN18 に入力されます。

温度の読み出し

温度センサを使用するには：

- ADC1_IN18 入力チャンネルを選択します。
- データシートに指定されている最小サンプリング時間より長いサンプリング時間を選択します。
- ADC_CCR レジスタの TSVREFE ビットをセットして、温度センサをパワーダウンモードからウェイクアップします。
- SWSTART ビットをセットすることによって（または外部トリガによって）ADC 変換を開始します。
- ADC データレジスタの V_{SENSE} データを読み出します。
- 次の式を使用して、温度を計算します。

$$\text{温度 (摂氏)} = \{(V_{SENSE} - V_{25}) / \text{Avg_Slope}\} + 25$$

ここで、

- V_{25} = 25° C における V_{SENSE} 値
- Avg_Slope = 温度対 V_{SENSE} 曲線の平均傾斜 (mV/°C または $\mu\text{V}/^\circ\text{C}$)

V_{25} および Avg_Slope の実際の値については、データシートの電気的特性のセクションを参照してください。

注： センサがパワーダウンモードからウェイクアップして、正しいレベルで V_{SENSE} を出力できるようになるまでには時間がかかります（スタートアップ時間）。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADON ビットと TSVREFE ビットを同時にセットしてください。

温度センサの出力電圧は、温度に比例して変化します。この一次関数のオフセットは、プロセスのばらつきにより、チップごとに異なります（チップ間で最大 45 °C）。

内部温度センサは、絶対温度の代わりに温度変化を検出するアプリケーションに適しています。正確な計測温度が必要な場合は、外部温度センサを使用してください。

13.10 バッテリ充電監視

ADC_CCR レジスタの VBATE ビットはバッテリー電圧に切り替えるために使用します。V_{BAT} 電圧は V_{DDA} より高くなることがあるので、ADC の正しい動作を確保するために、V_{BAT} ピンはブリッジ分圧回路に内部接続されています。

VBATE をセットするとブリッジが自動的に有効になり以下を接続します。

- VBAT/4 を ADC1_IN18 入力チャネル

注： VBAT および温度センサは同じ ADC 内部チャネル (ADC1_IN18) に接続されます。一度に 1 つの変換 (温度センサまたは VBAT のどちらか一方) を選択する必要があります。両方の変換を同時に有効すると、VBAT 変換のみ行われます。

13.11 ADC 割り込み

アナログウォッチドッグステータスビットがセットされ、かつオーバーランステータスビットがセットされていれば、レギュラおよびインジェクトグループの変換終了時、割り込みを生成することができます。高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

ADC_SR レジスタには他に 2 つのフラグがありますが、これらは割り込みには無関係です。

- JSTRT (インジェクトグループチャネルの変換開始)
- STRT (レギュラグループチャネルの変換開始)

表 78. ADC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
レギュラグループの変換終了	EOC	EOCIE
インジェクトグループの変換終了	JEOC	JEOCIE
アナログウォッチドッグステータスビットのセット	AWD	AWDIE
オーバーラン	OVR	OVRIE

13.12 ADC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワードレベル (32 ビット) で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

13.12.1 ADC ステータスレジスタ (ADC_SR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **OVR** : オーバーラン

このビットはデータが失われたとき (シングルモードまたはデュアル/トリプルモード時) ハードウェアによってセットされます。ソフトウェアによってクリアされます。オーバーラン検出は DMA = 1 または EOCS = 1 のときのみ有効です。

0 : オーバーランは発生していません。
1 : オーバーランが発生しました。

ビット 4 **STRT** : レギュラチャネル開始フラグ

このビットは、レギュラチャネルの変換が開始したときにハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : レギュラチャネルの変換は開始されていません。
1 : レギュラチャネルの変換が開始されました。

ビット 3 **JSTRT** : インジェクトチャネル開始フラグ

このビットは、インジェクトグループの変換が開始すると、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : インジェクトグループの変換は開始されていません。
1 : インジェクトグループの変換が開始されました。

ビット 2 JEOPC : インジェクトチャネルの変換終了フラグ

このビットは、グループ内のすべてのインジェクトチャネル変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : 変換が完了していません。
- 1 : 変換が完了しました。

ビット 1 EOC : レギュラチャネルの変換終了フラグ

このビットは、レギュラチャネルグループの変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによって、または ADC_DR レジスタを読み出すことによってクリアされます。

- 0 : 変換が終了していない (EOCS=0)、または変換のシーケンスが終了していません (EOCS=1)。
- 1 : 変換が終了した (EOCS=0)、または変換のシーケンスが終了しました (EOCS=1)。

ビット 0 AWD : アナログウォッチドッグフラグ

このビットは、変換された電圧が、ADC_LTR および ADC_HTR レジスタでプログラミングされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : アナログウォッチドッグイベントは発生していません。
- 1 : アナログウォッチドッグイベントが発生しました。

13.12.2 ADC 制御レジスタ 1 (ADC_CR1)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	OVRIE	RES		AWDEN	JAWDEN	Res.	Res.	Res.	Res.	Res.	Res.
					rw	rw	rw	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **OVRIE** : オーバーラン割り込みイネーブル

このビットは、オーバーラン割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : オーバーラン割り込みは無効です。

1 : オーバーラン割り込みは有効です。OVR ビットがセットされると、割り込みが生成されます。

ビット 25:24 **RES[1:0]** : 分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

00 : 12 ビット (最小 15 ADCCLK サイクル)

01 : 10 ビット (最小 13 ADCCLK サイクル)

10 : 8 ビット (最小 11 ADCCLK サイクル)

11 : 6 ビット (最小 9 ADCCLK サイクル)

ビット 23 **AWDEN** : アナログウォッチドッグ有効 (レギュラチャネル)

このビットは、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルに対するアナログウォッチドッグは無効です。

1 : レギュラチャネルに対するアナログウォッチドッグは有効です。

ビット 22 **JAWDEN** : アナログウォッチドッグ有効 (インジェクトチャネル)

このビットは、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルに対するアナログウォッチドッグは無効です。

1 : インジェクトチャネルに対するアナログウォッチドッグは有効です。

ビット 21:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **DISCNUM[2:0]** : 不連続モードチャネルカウント

これらのビットは、外部トリガの受信後に、不連続モードで変換されるレギュラチャネルの数を定義するために、ソフトウェアによって書き込まれます。

000 : 1 チャネル

001 : 2 チャネル

...

111 : 8 チャネル

ビット 12 **JDISCEN** : インジェクトチャネルの不連続モード

このビットは、グループのインジェクトチャネルに対する不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルの不連続モードは無効です。

1 : インジェクトチャネルの不連続モードは有効です。

ビット 11 DISCEN : レギュラチャネルの不連続モード

このビットは、レギュラチャネルに対する不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルの不連続モードは無効です。

1 : レギュラチャネルの不連続モードは有効です。

ビット 10 JAUTO : 自動インジェクトグループ変換

このビットは、レギュラグループ変換の後、自動インジェクトグループ変換を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 自動インジェクトグループ変換は無効です。

1 : 自動インジェクトグループ変換は有効です。

ビット 9 AWDSGL : スキャンモードでの単一チャネルのウォッチドッグイネーブル

このビットは、AWDCH[4:0] ビットによって指定されたチャネルに対するアナログウォッチドッグを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : すべてのチャネルでアナログウォッチドッグは有効です。

1 : 単一チャネルでアナログウォッチドッグは有効です。

ビット 8 SCAN : スキャンモード

このビットは、スキャンモードを有効／無効にするために、ソフトウェアによってセット／クリアされます。スキャンモードでは、ADC_SQRx および ADC_JSQRx レジスタを通じて選択された入力に変換されます。

0 : スキャンモードは無効です。

1 : スキャンモードは有効です。

注 : EOCIE ビットがセットされている場合、EOC 割り込みが生成されます。

– EOCs ビットが 0 にクリアされている場合、各レギュラグループシーケンス終了後

– EOCs ビットが 1 にセットされている場合、各レギュラチャネル変換終了後

注 : JEOCIE ビットがセットされている場合、最後のチャネルの変換が終了したときのみ、JEOC 割り込みが生成されます。

ビット 7 JEOCIE : インジェクトチャネルの割り込みイネーブル

このビットは、インジェクトチャネルの変換終了割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : JEOC 割り込みは無効です。

1 : JEOC 割り込みは有効です。JEOC ビットがセットされると、割り込みが生成されます。

ビット 6 AWDIE : アナログウォッチドッグ割り込みイネーブル

このビットは、アナログウォッチドッグ割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグ割り込みは無効です。

1 : アナログウォッチドッグ割り込みは有効です。

ビット 5 EOCIE : EOC 割り込みイネーブル

このビットは、変換終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOC 割り込みは無効です。

1 : EOC 割り込みは有効です。EOC ビットがセットされると、割り込みが生成されます。

ビット 4:0 **AWDCH[4:0]** : アナログウォッチドッグチャンネル選択ビット

これらのビットは、ソフトウェアによってセット／クリアされます。アナログウォッチドッグによって保護される入力チャンネルを選択します。

注 : **00000** : ADC アナログ入力チャンネル0
 00001 : ADC アナログ入力チャンネル1
 ...
 01111 : ADC アナログ入力チャンネル15
 10000 : ADC アナログ入力チャンネル16
 10001 : ADC アナログ入力チャンネル17
 10010 : ADC アナログ入力チャンネル18
 その他の値は予約済みです。

13.12.3 **ADC 制御レジスタ 2 (ADC_CR2)**

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SWSTART	EXTEN		EXTSEL[3:0]				Res.	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ALIGN	EOCS	DDS	DMA	Res.	Res.	Res.	Res.	Res.	Res.	CONT	ADON
				rW	rW	rW	rW							rW	rW

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SWSTART** : レギュラチャンネルの変換開始

このビットは、変換を開始するためにソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態

1 : レギュラチャンネルの変換を開始します。

注 : **このビットはADON = 1 のときのみセットできます。それ以外の場合変換は開始されません。**

ビット 29:28 **EXTEN** : レギュラチャンネルの外部トリガイネーブル

これらのビットは、外部トリガ極性を選択しレギュラグループのトリガを有効にするために、ソフトウェアによってセット／クリアされます。

00 : トリガ検出は無効です。

01 : 立ち上がりエッジでトリガを検出します。

10 : 立ち下がりエッジでトリガを検出します。

11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

ビット 27:24 **EXTSEL[3:0]** : レギュラグループの外部イベント選択

これらのビットは、レギュラグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 CC1 イベント
0001 : タイマ 1 CC2 イベント
0010 : タイマ 1 CC3 イベント
0011 : タイマ 2 CC2 イベント
0100 : タイマ 2 CC3 イベント
0101 : タイマ 2 CC4 イベント
0110 : タイマ 2 TRGO イベント
0111 : タイマ 3 CC1 イベント
1000 : タイマ 3 TRGO イベント
1001 : タイマ 4 CC4 イベント
1010 : タイマ 5 CC1 イベント
1011 : タイマ 5 CC2 イベント
1100 : タイマ 5 CC3 イベント
1101 : タイマ 8 CC1 イベント
1110 : タイマ 8 TRGO イベント
1111 : EXTI ライン 11

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **JSWSTART** : インジェクトチャネルの変換開始

このビットは、ソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態

1 : インジェクトチャネルの変換を開始します。

注 : このビットは **ADON = 1** のときのみセットできます。それ以外の場合変換は開始されません。

ビット 21:20 **JEXTEN** : インジェクトチャネルの外部トリガイネーブル

これらのビットは、外部トリガ極性を選択しインジェクトグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : トリガ検出は無効です。

01 : 立ち上がりエッジでトリガを検出します。

10 : 立ち下がりエッジでトリガを検出します。

11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

ビット 19:16 JEXTSEL[3:0] : インジェクトグループの外部イベント選択

これらのビットは、インジェクトグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 CC4 イベント
0001 : タイマ 1 TRGO イベント
0010 : タイマ 2 CC1 イベント
0011 : タイマ 2 TRGO イベント
0100 : タイマ 3 CC2 イベント
0101 : タイマ 3 CC4 イベント
0110 : タイマ 4 CC1 イベント
0111 : タイマ 4 CC2 イベント
1000 : タイマ 4 CC3 イベント
1001 : タイマ 4 TRGO イベント
1010 : タイマ 5 CC4 イベント
1011 : タイマ 5 TRGO イベント
1100 : タイマ 8 CC2 イベント
1101 : タイマ 8 CC3 イベント
1110 : タイマ 8 CC4 イベント
1111 : EXTI ライン 15

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 ALIGN : データの配置

このビットは、ソフトウェアによってセット／クリアされます。[図 63](#) および [図 64](#) を参照してください。

0 : 右詰め
1 : 左詰め

ビット 10 EOCS : 変換選択終了

このビットは、ソフトウェアによってセット／クリアされます。

0 : EOC ビットはレギュラ変換シーケンスが終了するたびにセットされます。オーバーラン検出は DMA = 1 のときのみ有効です。

1 : EOC ビットはレギュラ変換が終了するたびにセットされます。オーバーラン検出は有効です。

ビット 9 DDS : DMA ディセーブル選択 (シングル ADC モード)

このビットは、ソフトウェアによってセット／クリアされます。

0 : 最後の転送後新たな DMA リクエストは発行されません (DMA コントローラの設定どおり)

1 : データが変換され DMA=1 である限り DMA リクエストは発行されます。

ビット 8 DMA : ダイレクトメモリアクセスモード (シングル ADC モード)

このビットは、ソフトウェアによってセット／クリアされます。詳細については、DMA コントローラの章を参照してください。

0 : DMA モードは無効です。
1 : DMA モードは有効です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 CONT : 連続変換

このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされた場合、それがクリアされるまで連続的に変換が行われます。

0 : シングル変換モード
1 : 連続変換モード

ビット 0 ADON : A/D コンバータのオン／オフ

このビットは、ソフトウェアによってセット／クリアされます。

注 : **0 : ADC の変換を無効にして、パワーダウンモードに移行します。**
 1 : ADC を有効にします。

13.12.4 ADC サンプル時間レジスタ 1 (ADC_SMPR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0		SMP14[2:0]		SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 : 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:0 **SMPx[2:0]** : チャネル x サンプル時間選択

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャネル選択ビットを変更することはできません。

注 : **000 : 3 サイクル**
 001 : 15 サイクル
 010 : 28 サイクル
 011 : 56 サイクル
 100 : 84 サイクル
 101 : 112 サイクル
 110 : 144 サイクル
 111 : 480 サイクル

13.12.5 ADC サンプル時間レジスタ 2 (ADC_SMPR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5_0		SMP4[2:0]		SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:0 **SMPx[2:0]** : チャネル x サンプル時間選択

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャネル選択ビットを変更することはできません。

注 : **000 : 3 サイクル**
 001 : 15 サイクル
 010 : 28 サイクル
 011 : 56 サイクル
 100 : 84 サイクル
 101 : 112 サイクル
 110 : 144 サイクル
 111 : 480 サイクル



13.12.6 ADC インジェクトチャネルデータオフセットレジスタ x (ADC_JOFRx)
(x=1..4)

アドレスオフセット : 0x14-0x20

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 JOFFSETx[11:0] : インジェクトチャネル x のデータオフセット

これらのビットは、インジェクトチャネルを変換するときに生変換データから引かれるオフセットを定義するために、ソフトウェアによって書き込まれます。変換結果は、ADC_JDRx レジスタで読み出すことができます。

13.12.7 ADC ウォッチドッグ高閾値レジスタ (ADC_HTR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0FFF

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 HT[11:0] : アナログウォッチドッグ高閾値

これらのビットは、アナログウォッチドッグの高閾値を定義するために、ソフトウェアによって書き込まれます。

注 : ソフトウェアは、ADC 変換の実行中にこれらのレジスタに書き込むことができます。プログラムされた値は、次の変換完了時に有効になります。このレジスタへの書き込みは、書き込み遅延を伴って実行されます。そのため、プログラムされた新しい値が有効になる時間は確定的ではありません。

13.12.8 ADC ウォッチドッグ低閾値レジスタ (ADC_LTR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **LT[11:0]** : アナログウォッチドッグ低閾値

これらのビットは、アナログウォッチドッグの低閾値を定義するために、ソフトウェアによって書き込まれます。

注 : ソフトウェアは、ADC 変換の実行中にこれらのレジスタに書き込むことができます。プログラムされた値は、次の変換完了時に有効になります。このレジスタへの書き込みは、プログラムされた新しい値が有効になる時間の不確実性が生じてもいいように、書き込み遅延を伴って実行されます。

13.12.9 ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L[3:0]				SQ16[4:1]			
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **L[3:0]** : レギュラチャネルシーケンス長

これらのビットは、レギュラチャネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

0000 : 1 変換
0001 : 2 変換
...
1111 : 16 変換

ビット 19:15 **SQ16[4:0]** : レギュラシーケンスの 16 番目の変換

これらのビットには、変換シーケンス内の 16 番目として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 **SQ15[4:0]** : レギュラシーケンスの 15 番目の変換

ビット 9:5 **SQ14[4:0]** : レギュラシーケンスの 14 番目の変換

ビット 4:0 **SQ13[4:0]** : レギュラシーケンスの 13 番目の変換



13.12.10 ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:26 **SQ12[4:0]** : レギュラシーケンスの 12 番目の変換
これらのビットには、シーケンス内の 12 番目の変換として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 24:20 **SQ11[4:0]** : レギュラシーケンスの 11 番目の変換

ビット 19:15 **SQ10[4:0]** : レギュラシーケンスの 10 番目の変換

ビット 14:10 **SQ9[4:0]** : レギュラシーケンスの 9 番目の変換

ビット 9:5 **SQ8[4:0]** : レギュラシーケンスの 8 番目の変換

ビット 4:0 **SQ7[4:0]** : レギュラシーケンスの 7 番目の変換

13.12.11 ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:25 **SQ6[4:0]** : レギュラシーケンスの 6 番目の変換
これらのビットには、変換シーケンスで変換される 6 番目として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 24:20 **SQ5[4:0]** : レギュラシーケンスの 5 番目の変換

ビット 19:15 **SQ4[4:0]** : レギュラシーケンスの 4 番目の変換

ビット 14:10 **SQ3[4:0]** : レギュラシーケンスの 3 番目の変換

ビット 9:5 **SQ2[4:0]** : レギュラシーケンスの 2 番目の変換

ビット 4:0 **SQ1[4:0]** : レギュラシーケンスの 1 番目の変換

13.12.12 ADC インジェクトシーケンスレジスタ (ADC_JSQR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JL[1:0]		JSQ4[4:1]			
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 JL[1:0] : インジェクトシーケンス長

これらのビットは、インジェクトチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

- 00 : 1 変換
- 01 : 2 変換
- 10 : 3 変換
- 11 : 4 変換

ビット 19:15 JSQ4[4:0] : インジェクトシーケンスの 4 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

これらのビットには、変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 JSQ3[4:0] : インジェクトシーケンスの 3 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 9:5 JSQ2[4:0] : インジェクトシーケンスの 2 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 4:0 JSQ1[4:0] : インジェクトシーケンスの 1 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

注 : JL[1:0]=3 のとき (シーケンサには 4 つのインジェクト変換)、ADC はチャンネルを次の順に変換します。JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0]。
JL=2 のとき (シーケンサには 3 つのインジェクト変換)、ADC はチャンネルを次の順に変換します。JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0]。
JL=1 のとき (シーケンサには 2 つのインジェクト変換)、ADC はチャンネルを JSQ3[4:0]、JSQ4[4:0] の順に変換します。
JL=0 のとき (シーケンサには 1 つのインジェクト変換)、ADC は JSQ4[4:0] チャンネルだけ変換します。

13.12.13 ADC インジェクトデータレジスタ x (ADC_JDRx) (x= 1..4)

アドレスオフセット : 0x3C - 0x48

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **JDATA[15:0]** : インジェクトデータ

これらのビットは読み出し専用です。これらのビットには、インジェクトチャネル x の変換結果が格納されます。[図 63](#) および [図 64](#) に示すように、データは左詰めまたは右詰めされています。

13.12.14 ADC レギュラデータレジスタ (ADC_DR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **DATA[15:0]** : レギュラデータ

これらのビットは読み出し専用です。これらのビットには、レギュラチャネルの変換結果が格納されます。[図 63](#) および [図 64](#) に示すように、データは左詰めまたは右詰めされています。

13.12.15 ADC 共通ステータスレジスタ (ADC_CSR)

アドレスオフセット : 0x00 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

このレジスタにより、ADC1 のステータスビットの状態を確認することができます。ただし、これは読み出し専用で、別の ADC のステータスビットをクリアすることはできません。その代わり、対応する ADC_SR レジスタに 0 を書き込むことによって各ステータスビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR1	STRT1	JSTRT1	JEOC 1	EOC1	AWD1
										r	r	r	r	r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **OVR1** : ADC1 オーバーランフラグ

このビットは ADC1_SR レジスタの OVR ビットのコピーです。

ビット 4 **STRT1** : ADC1 レギュラチャネル開始フラグ

このビットは ADC1_SR レジスタの STRT ビットのコピーです。

- ビット 3 **JSTRT1** : ADC1 インジェクトチャネル開始フラグ
このビットは ADC1_SR レジスタの JSTRT ビットのコピーです。
- ビット 2 **JEOC1** : ADC1 インジェクトチャネル変換終了フラグ
このビットは ADC1_SR レジスタの JEOC ビットのコピーです。
- ビット 1 **EOC1** : ADC1 変換終了フラグ
このビットは ADC1_SR レジスタの EOC ビットのコピーです。
- ビット 0 **AWD1** : ADC1 アナログウォッチドッグフラグ
このビットは ADC1_SR レジスタの AWD ビットのコピーです。

13.12.16 ADC 共通制御レジスタ (ADC_CCR)

アドレスオフセット : 0x04 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

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

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23 **TSVREFE** : 温度センサと V_{REFINT} イネーブル
このビットは、温度センサおよび V_{REFINT} チャネルを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : 温度センサおよび V_{REFINT} チャネルは無効です。
1 : 温度センサおよび V_{REFINT} チャネルは有効です。
注 : **TSVREFE がセットされているときは VBATE を無効にする必要があります。両方のビットがセットされている場合、VBAT 変換のみ行われます。**
- ビット 22 **VBATE** : V_{BAT} イネーブル
このビットは、V_{BAT} チャネルを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : V_{BAT} チャネルは無効です。
1 : V_{BAT} チャネルは有効です。
- ビット 21:18 予約済みであり、リセット値に保持する必要があります。
- ビット 17:16 **ADCPRE** : ADC プリスケアラ
ADC へのクロックの周波数を選択するために、ソフトウェアによってセット／クリアされます。クロックはすべての ADC に共通です。
注 : **00 : PCLK2 は 2 分周されます。**
01 : PCLK2 は 4 分周されます。
10 : PCLK2 は 6 分周されます。
11 : PCLK2 は 8 分周されます。
- ビット 15:0 予約済みであり、リセット値に保持する必要があります。

13.12.17 ADC レジスタマップ

次の表は ADC レジスタの一覧です。

表 79. ADC グローバルレジスタマップ

オフセット	レジスタ
0x000 - 0x04C	ADC1
0x050 - 0x2FC	予約済み
0x300 - 0x308	共通レジスタ

表 80. ADC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD
	リセット値																											0	0	0	0	0	0
0x04	ADC_CR1	Res.	Res.	Res.	Res.	Res.	OVRIE	RES[1:0]	AWDEN	JAWDEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DISC NUM [2:0]	JDISEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]						
	リセット値						0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_CR2	Res.	SWSTART	EXTEN[1:0]	EXTSEL[3:0]			Res.	JSWSTART	JEXTEN[1:0]	JEXTSEL [3:0]			Res.	Res.	Res.	Res.	Res.	ALIGN	EOCS	DDS	DMA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONT	ADON	
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	ADC_SMPR1	サンプル時間ビット SMPx_x																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	ADC_SMPR2	サンプル時間ビット SMPx_x																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_JOFR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JOFFSET1[11:0]												
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	ADC_JOFR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JOFFSET2[11:0]												
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	ADC_JOFR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JOFFSET3[11:0]												
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	ADC_JOFR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JOFFSET4[11:0]												
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	ADC_HTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT[11:0]												
	リセット値																				1	1	1	1	1	1	1	1	1	1	1	1	1
0x28	ADC_LTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT[11:0]												
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	ADC_SQR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L[3:0]	レギュラチャネルシーケンス SQx_x ビット																						
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	ADC_SQR2	Res.	Res.	レギュラチャネルシーケンス SQx_x ビット																													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	ADC_SQR3	Res.	Res.	レギュラチャネルシーケンス SQx_x ビット																													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



表 80. ADC レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38	ADC_JSQR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	JL[1:0]		インジェクトチャネルシーケンス JSQx_x ビット																			
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	ADC_JDR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	ADC_JDR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	ADC_JDR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	ADC_JDR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	ADC_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		レギュラ DATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 81. ADC レジスタマップとリセット値 (共通 ADC レジスタ)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD
	リセット値																										0	0	0	0	0	0	0
0x04	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSVREFE	VBATE	Res.	Res.	Res.	Res.	ADCPRE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値									0	0																						

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

14 デルタシグマモジュレータのデジタルフィルタ (DFSDM)

14.1 概要

デルタシグマモジュレータのデジタルフィルタ (DFSDM) は、外部デルタシグマモジュレータをマイクロコントローラにインタフェース接続するための専用ハイパフォーマンスモジュールです。最大 4 つまでの外部デジタルシリアルインタフェース (チャネル) と、柔軟性の高いデルタシグマストリームデジタル処理オプションを持つ最大 2 つのデジタルフィルタの機能があり、最終的に最大 24 ビットの ADC の分解能を提供します。DFSDM は、マイクロコントローラメモリからのオプションの並列データストリーム入力も備えています。

外部デルタシグマモジュレータにより、外部デルタシグマモジュレータのアナログ入力から変換されたアナログ値のデジタルデータストリームが利用できます。このデジタルデータストリームは、シリアルインタフェースを介して DFSDM 入力チャネルに送信されます。DFSDM は、いくつかの規格をサポートしており、次のさまざまなデルタシグマモジュレータ出力に接続します。SPI インタフェースおよびマンチェスタ符号化 1 線インタフェース (ともに調整可能なパラメータあり)。DFSDM モジュールは、最大 4 つの多重化入力デジタルシリアルチャネルの接続をサポートしており、最大 2 つの DFSDM モジュールで接続を共有します。また、DFSDM モジュールは最大 4 つの内部 16 ビットデータチャネルからの代替並列データ入力にもサポートしています (マイクロコントローラメモリから)。

DFSDM は入力データストリームを最終デジタルデータワードに変換して、デルタシグマモジュレータのアナログ入力のアナログ入力値を表します。変換は、入力シリアルデータストリームのデジタルフィルタリングおよびデシメーションといった設定可能なデジタル処理に基づいて行われます。

変換速度および分解能は、フィルタタイプ、フィルタ段数、フィルタの長さ、積分器の長さといったデジタル処理に関する設定可能なパラメータに合わせて調整できます。最大出力データの分解能は 24 ビットです。変換モードには、シングル変換モードと連続モードの 2 種類があります。データは、DMA を介してシステム RAM バッファに自動的に保存され、ソフトウェアの負荷を軽減できます。

柔軟性の高いタイマリガシステムは、DFSDM の変換開始を制御するために使用できます。このタイミング制御により、同時変換のトリガや、変換の間にプログラム可能な遅延の挿入が可能です。

DFSDM にはアナログウォッチドック機能があります。入力チャネルデータストリームや最終出力データに、アナログウォッチドックを割り当てることができます。アナログウォッチドックには、独自の入力データストリームのデジタルフィルタリングがあり、監視するデータに必要な速度および分解能を満たすことができます。

制御アプリケーションにおける短絡を検出するために、短絡検出回路があります。このブロックは、各入力チャネルデータストリームにおいて一定時間内の安定したデータの発生 (入力データストリームでの 0 または 1 について) を監視します。

極値検出回路ブロックは、最終出力データを監視して、出力データ値の最大値と最小値を保存します。保存された極値は、ソフトウェアによって復元することができます。

通常モードと STOP モードの 2 種類の電力モードがあります。

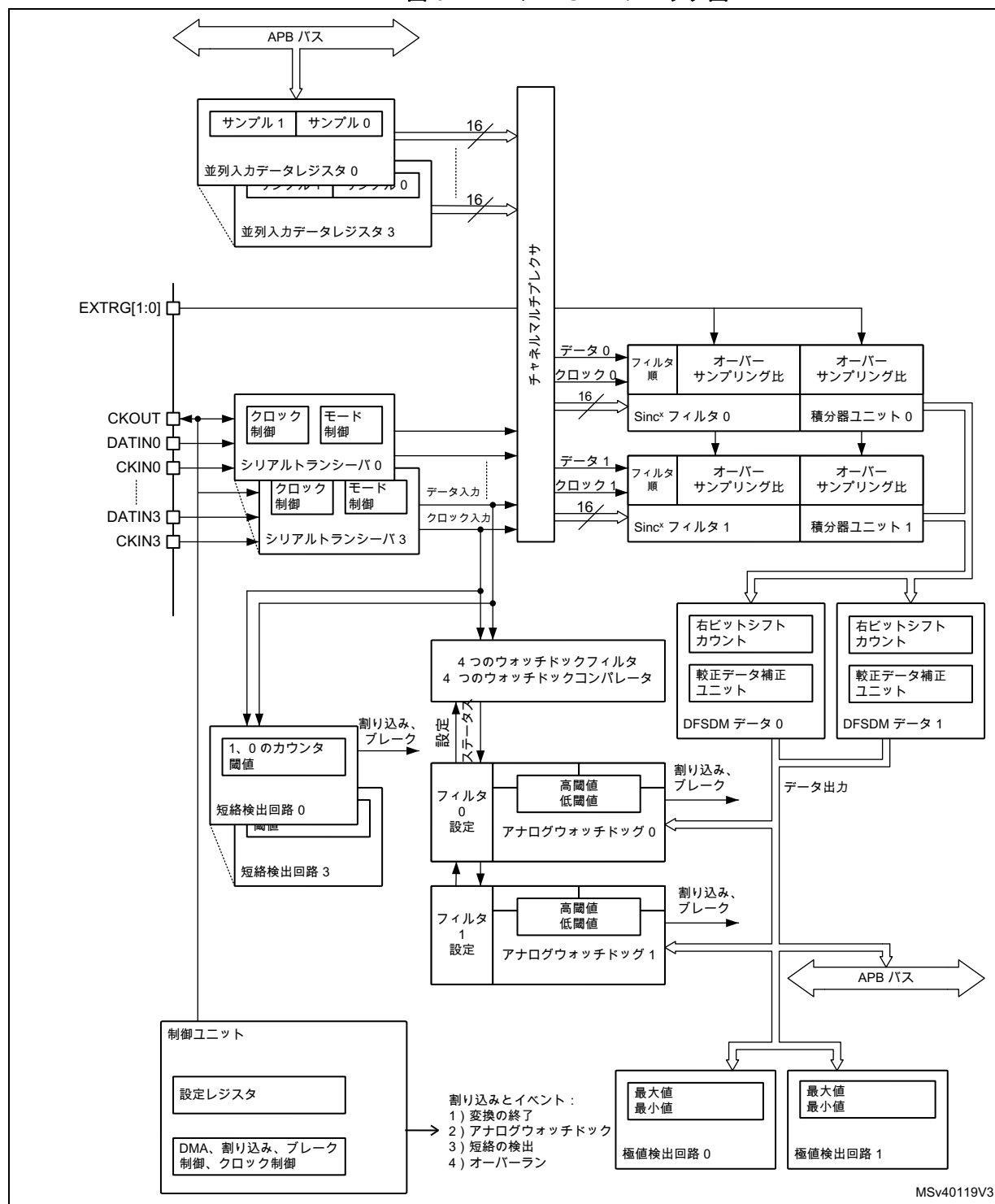
14.2 DFSDM の主な機能

- 最大 4 つの多重化入力デジタルシリアルチャネル：
 - － さまざまなデルタシグマモジュレータを接続するために設定可能な SPI インタフェース
 - － 設定可能なマンチェスタ符号化 1 線インタフェースのサポート
 - － デルタシグマモジュレータのクロック出力
- 最大 4 つの内部デジタル並列チャネルからの代替入力：
 - － 最大 16 ビットの分解能を持つ入力
 - － 内部ソース：メモリ（CPU/DMA 書き込み）データストリーム
- 調整可能なデジタル信号処理：
 - － Sinc^x フィルタ：フィルタ段数／タイプ（1～5）、オーバーサンプリング比（最大 1～1024）
 - － 積分器：オーバーサンプリング比（1～256）
- 最大 24 ビット出力データの分解能：
 - － 最終データの右ビットシフト（0～31 ビット）
- 符号付き出力データフォーマット
- 自動データオフセット補正（オフセットはユーザによってレジスタに保存）
- 連続またはシングル変換
- 変換開始の同期：
 - － ソフトウェアトリガ
 - － 内部タイマ
 - － 外部イベント
 - － 最初の DFSDM フィルタ（DFSDM_FLT0）と同期した変換開始
- アナログウォッチドッグの機能：
 - － データの低閾値および高閾値レジスタ
 - － 独自の設定可能な Sinc^x デジタルフィルタ（段数 = 1～3、オーバーサンプリング比 = 1～32）
 - － 出力データレジスタまたは 1 つ以上の入力デジタルシリアルチャネルからの入力
 - － 標準変換から独立した連続監視
- 飽和したアナログ入力値（下限と上限）を検出する短絡検出回路：
 - － 入力データストリームで 1～256 の連続した 0 と 1 を検出する最大 8 ビットカウンタ
 - － 各チャネルの連続監視（4 つのシリアルチャネルのトランシーバ出力）
- アナログウォッチドックイベントや短絡検出回路イベントでのブレイク生成
- 極値検出回路：
 - － 出力データの最小値と最大値の保存
 - － ソフトウェアによるリフレッシュ
- DMA を使用した変換データ読み出し可能
- 割り込み：変換終了、オーバーラン、アナログウォッチドック、短絡、チャネルクロック無
- 「レギュラ」または「インジェクト」変換：
 - － 「レギュラ」変換は「インジェクト」変換のタイミングに影響しない限り、連続モードでもいつでもリクエスト可能

14.3 DFSDM の機能詳細

14.3.1 DFSDM ブロック図

図 67. 1 つの DFSDM ブロック図



MSv40119V3

注： この例では、2 つの DFSDM フィルタと 4 つの入力チャネルを示します。

14.3.2 DFSDM ピンおよび内部信号

表 82. DFSDM 外部ピン

名前	信号タイプ	説明
VDD	電源	デジタル電源供給。
VSS	電源	デジタルグランド電源供給。
CKIN[3:0]	クロック入力	外部デルタシグマモジュレータから供給されるクロック信号。FT 入力。
DATIN[3:0]	データ入力	外部デルタシグマモジュレータから供給されるデータ信号。FT 入力。
CKOUT	クロック出力	外部デルタシグマモジュレータにクロック信号を供給するためのクロック出力。
EXTRG[1:0]	外部トリガ信号	アナログ変換を開始するための 2 つの EXTI 信号からの入力トリガ (GPIO : EXTI11、EXTI15)。

表 83. DFSDM 内部信号

名前	信号タイプ	説明
dfsdm_jtrg[10:0]	内部／外部トリガ信号	アナログ変換を開始するための内部／外部トリガソースからの入力トリガ。詳細は、表 84 を参照してください。
dfsdm_break[3:0]	ブレーク信号出力	アナログウォッチドックや短絡検出回路からのブレーク信号イベント生成。
dfsdm_dma[1:0]	DMA リクエスト信号	各 DFSDM_FLTx (x=0~1) からの DMA リクエスト信号：インジェクト変換イベントの終了。
dfsdm_it[1:0]	割り込みリクエスト信号	各 DFSDM_FLTx の割り込み信号 (x = 0~1)

表 84. DFSDM トリガ接続

トリガ名	トリガソース
dmfsm_jtrg0	TIM1_TRGO
dmfsm_jtrg1	TIM3_TRGO
dmfsm_jtrg2	TIM8_TRGO
dmfsm_jtrg3	TIM10_OC1
dmfsm_jtrg4	N/A
dmfsm_jtrg5	TIM4_TRGO
dmfsm_jtrg6	N/A
dmfsm_jtrg7	TIM6_TRGO
dmfsm_jtrg8	N/A
dmfsm_jtrg9	EXTI11
dmfsm_jtrg10	EXTI15

14.3.3 DFSDM のリセットおよびクロック

DFSDM のオン/オフ制御

DFSDM インタフェースは、DFSDM_CH0CFGR1 レジスタで DFSDMEN=1 をセットすることによって全体的に有効になります。DFSDM が全体的に有効になると、すべての入力チャネル (y=0~3) およびデジタルフィルタ DFSDM_FLTx (x=0~1) はイネーブルビット (DFSDM_CHyCFGR1 のチャネルイネーブルビット CHEN および DFSDM_FLTxCR1 の DFSDM_FLTx イネーブルビット DFEN) がセットされた場合に動作を開始します。

デジタルフィルタ x DFSDM_FLTx (x=0~1) は、DFSDM_FLTxCR1 レジスタで DFEN=1 をセットすることによって有効になります。DFSDM_FLTx が有効 (DFEN=1) になると、Sinc^x デジタルフィルタユニットおよび積分器ユニット両方が再初期化されます。

DFEN をクリアすると、変換が進行中であった場合、即座に停止し、DFSDM_FLTx が STOP モードに入ります。DFSDM_FLTxAWSR および DFSDM_FLTxISR (リセット済み) を除き、すべてのレジスタ設定は変更されません。

チャネル y (y=0~3) は、DFSDM_CHyCFGR1 レジスタで CHEN=1 をセットすることによって有効になります。チャネルが有効になると、外部デルタシグマモジュレータまたは並列内部データソース (メモリからの CPU/DMA ワイヤ) からシリアルデータを受信します。

DFSDM は、デバイスが STOP モードに入るためにシステムクロックを停止する前に全体的に無効 (DFSDM_CH0CFGR1 の DFSDMEN=0) にしておく必要があります。

DFSDM クロック

内部 DFSDM クロック $f_{DFSDMCLK}$ は、チャネルトランシーバ、デジタル処理ブロック (デジタルフィルタ、積分器) および次に追加されるブロック (アナログウォッチドック、短絡検出回路、極値検出回路、制御ブロック) の駆動に使用され、RCC ブロックにより生成され、システムクロック SYSCLK (最大 $f_{SYSCLK} = 100$ MHz まで) またはペリフェラルクロック PCLK2 ([セクション 6.3.24 : RCC 専用クロック設定レジスタ \(RCC_DCKCFGR\)](#)) の DFSDMSEL ビットの説明を参照) から生成されます。DFSDM クロックは STOP モード (すべての DFSDM_FLTx, x=0~1 で DFEN = 0 の場合) で自動的に停止します。

DFSDM シリアルチャネルトランシーバは、外部シリアルクロックを受信して、外部シリアルデータストリームをサンプリングできます。内部 DFSDM クロックは、標準 SPI 符号化を使用する場合、外部シリアルクロックの 4 倍以上、マンチェスタ符号化を使用する場合、6 倍以上の速さが必要になります。

DFSDM は外部出力クロック信号を供給して、外部デルタシグマモジュレータのクロック入力を駆動できます。CKOUT ピンから供給されます。この出力クロック信号は、特定のデバイスのデータシートに指定された範囲内である必要があり、DFSDM クロックまたはオーディオクロック (DFSDM_CH0CFGR1 レジスタの CKOUTSRC ビットを参照) からプログラム可能な分周器によって範囲 2~256 (DFSDM_CH0CFGR1 レジスタの CKOUTDIV) で生成されます。オーディオクロックソースは、RCC 設定の SAI1SEL[1:0] フィールドで選択された SAI1 クロックです ([セクション 6.3.24 : RCC 専用クロック設定レジスタ \(RCC_DCKCFGR\)](#) を参照)。

14.3.4 シリアルチャネルトランシーバ

アナログウォッチドックまたは短絡検出回路の各フィルタによる変換で選択できる 4 つの多重化シリアルデータチャネルがあります。これらのシリアルトランシーバは、外部デルタシグマモジュレータからデータストリームを受信します。データストリームは SPI フォーマットまたはマンチェスタ符号化フォーマットで送信できます (DFSDM_CHyCFGR1 レジスタの SITP[1:0] ビットを参照)。チャネルは、DFSDM_CHyCFGR1 レジスタで CHEN=1 をセットすることによって動作が有効になります。

チャンネル入力を選択

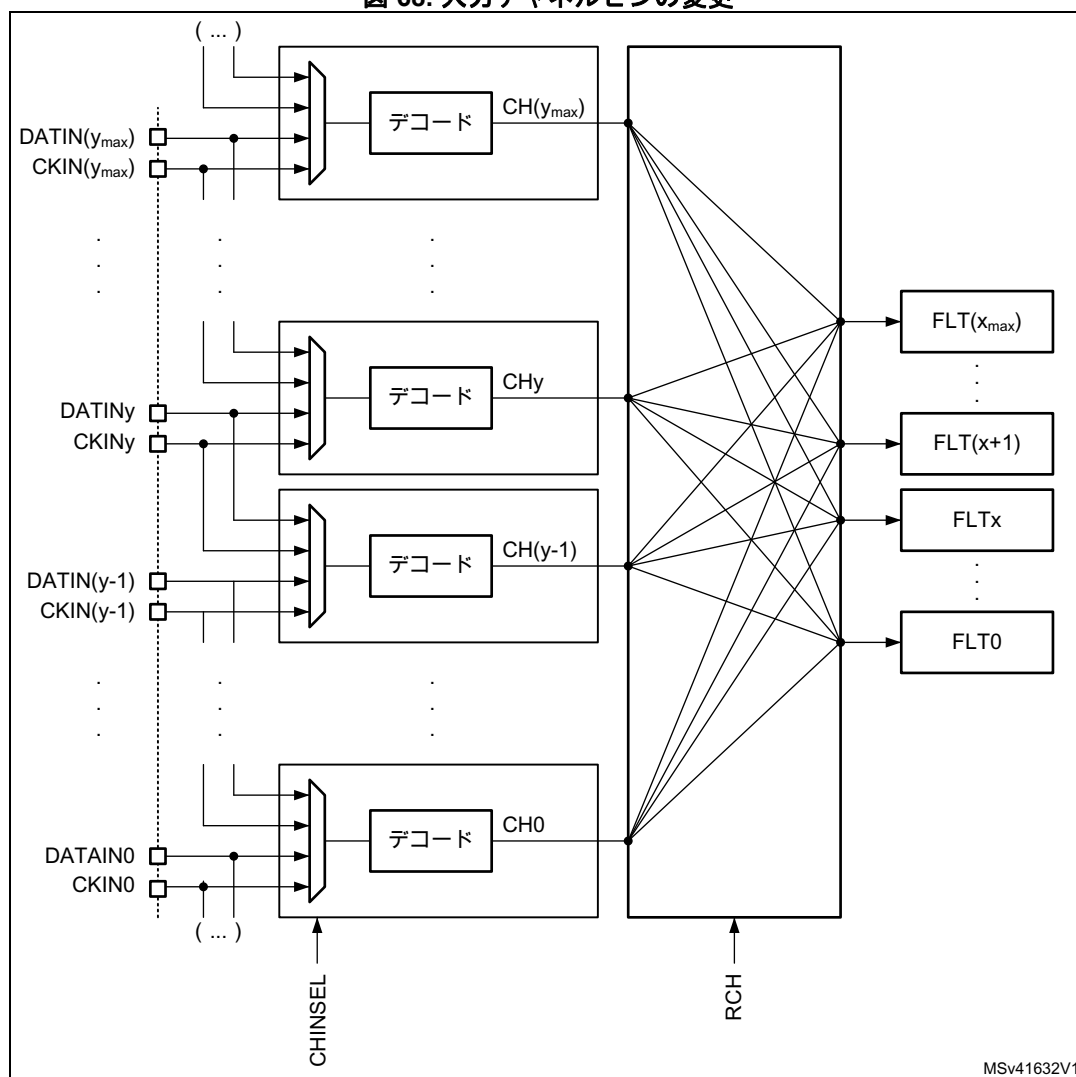
DATINy ピンおよび CKINy ピンからのシリアル入力（データおよびクロック信号）は、次のチャンネルピンから変更できます。このシリアル入力チャンネル変更は DFSDM_CHyCFGR1 レジスタの CHINSEL ビットでセットされます。

チャンネル変更は PDM（パルス密度変調）ステレオマイクロフォンタイプからのオーディオデータの収集に使用できます。PDM ステレオマイクロフォンにはデータ信号とクロック信号が 1 つずつあります。データ信号は、左右のオーディオチャンネルの情報を供給します（左チャンネルは立ち上がりクロックエッジサンプル、右チャンネルは立ち下がりクロックエッジサンプル）。

PDM マイクロフォン入力のシリアルチャンネルの設定：

- PDM マイクロフォン信号（データ、クロック）は、DFSDM 入力シリアルチャンネル y（DATINy、CKOUT）ピンに接続されます。
- チャンネル y は次のように設定されます。CHINSEL = 0（特定のチャンネルピンからの入力：DATINy、CKINy）。
- チャンネル (y-1)（モジュロ 4）は次のように設定されます。次のチャンネル ((y-1) + 1) ピンからの CHINSEL = 1 入力：同様に DATINy、CKINy から。
- チャンネル y : SITP[1:0] = 0（データストローブのための立ち上がりエッジ）=> チャンネル y の左オーディオチャンネル。
- チャンネル (y-1) : SITP[1:0] = 1（データストローブのための立ち下がりエッジ）=> チャンネル y-1 の右オーディオチャンネル。
- 2 つの DFSDM フィルタがチャンネル y およびチャンネル (y-1) に割り当てられます（PDM マイクロフォンからの左右チャンネルをフィルタするため）。

図 68. 入力チャネルピンの変更



MSv41632V1

出力クロック生成

クロック信号は CKOUT ピンで供給され、外部デルタシグマモジュレータのクロック入力を駆動できます。CKOUT 信号の周波数は、DFSDM クロックまたはオーディオクロック (DFSDM_CH0CFGR1 レジスタの CKOUTSRC ビットを参照) から前置分周器 (DFSDM_CH0CFGR1 レジスタの CKOUTDIV) によって生成されます。出力クロックを停止すると、CKOUT 信号はロー状態にセットされます (出力クロックは DFSDM_CHyCFGR1 レジスタの CKOUTDIV=0 または DFSDM_CH0CFGR1 レジスタの DFSDMEN=0 で停止できます)。出力クロック停止は次に対して実行されます。

- DFSDMEN クリア後の 4 つのシステムクロック (CKOUTSRC=0 の場合)
- DFSDMEN クリア後の 1 つのシステムクロックおよび 3 つのオーディオクロック (CKOUTSRC=1 の場合)

CKOUTSRC を変更する前に、ソフトウェアは CKOUT ピンのグリッチを回避するために CKOUT の停止を待つ必要があります。出力クロック信号の周波数は、0~20 MHz の範囲でセットする必要があります。

SPI データ入力フォーマットの操作

SPI フォーマットでは、データストリームはデータ信号およびクロック信号を介してシリアルフォーマットで送信されます。データ信号は常に DATINy ピン から供給されます。クロック信号は外部では CKINy ピンから、内部では CKOUT 信号ソースから生成される信号で供給できます。

外部クロックソースを選択する場合 (SPICKSEL[1:0]=0)、データ信号 (DATINy ピン) は立ち上がりクロックエッジまたは立ち下がりクロックエッジ (CKINy ピン) で SITP[1:0] ビットの設定 (DFSDM_CHyCFGR1 レジスタ) に応じてサンプルされます。

内部クロックソース - DFSDM_CHyCFGR1 レジスタの SPICKSEL[1:0] を参照：

- CKOUT 信号：
 - － クロック入力 (CKOUT から) に直接使用している外部デルタシグマモジュレータに接続して出力シリアル通信クロックを生成する場合
 - － サンプルングポイント：SITP[1:0] 設定に応じた立ち上がり／立ち下がりエッジ
- CKOUT/2 信号 (CKOUT 立ち上がりエッジで生成)：
 - － クロック入力 (CKOUT から) を 2 つに分周する外部デルタシグマモジュレータに接続して出力シリアル通信クロックを生成する場合 (また、この出力クロック変更が各クロック入力の立ち上がりエッジで有効である場合)
 - － サンプルングポイント：毎秒の CKOUT の立ち下がりエッジ
- CKOUT/2 信号 (CKOUT 立ち下がりエッジで生成)：
 - － クロック入力 (CKOUT から) を 2 つに分周する外部デルタシグマモジュレータに接続して出力シリアル通信クロックを生成する場合 (また、この出力クロック変更が各クロック入力の立ち下がりエッジで有効である場合)
 - － サンプルングポイント：毎秒の CKOUT の立ち上がりエッジ

注： 内部クロックソースは、外部デルタシグマモジュレータが CKOUT 信号をクロック入力として (同期クロックおよびデータ操作のために) 使用する場合のみ使用できます。

内部クロックソースの使用で、CKINy ピンを接続しないで済みます (CKINy ピンは他の目的に使用できます)。

クロックソースの信号の周波数は、SPI 符号化では 0~20 MHz の範囲で $f_{DFSDMCLK}/4$ 未満にセットする必要があります。

マンチェスタ符号化データ入力フォーマットの操作

マンチェスタ符号化フォーマットでは、データストリームは DATINy ピンのみを介してシリアルフォーマットで送信されます。デコードされたデータおよびクロック信号は、マンチェスタデコードの後、シリアルストリームから復元されます。マンチェスタ符号化で可能な設定は 2 つあります (DFSDM_CHyCFGR1 レジスタの SITP[1:0] ビットを参照)。

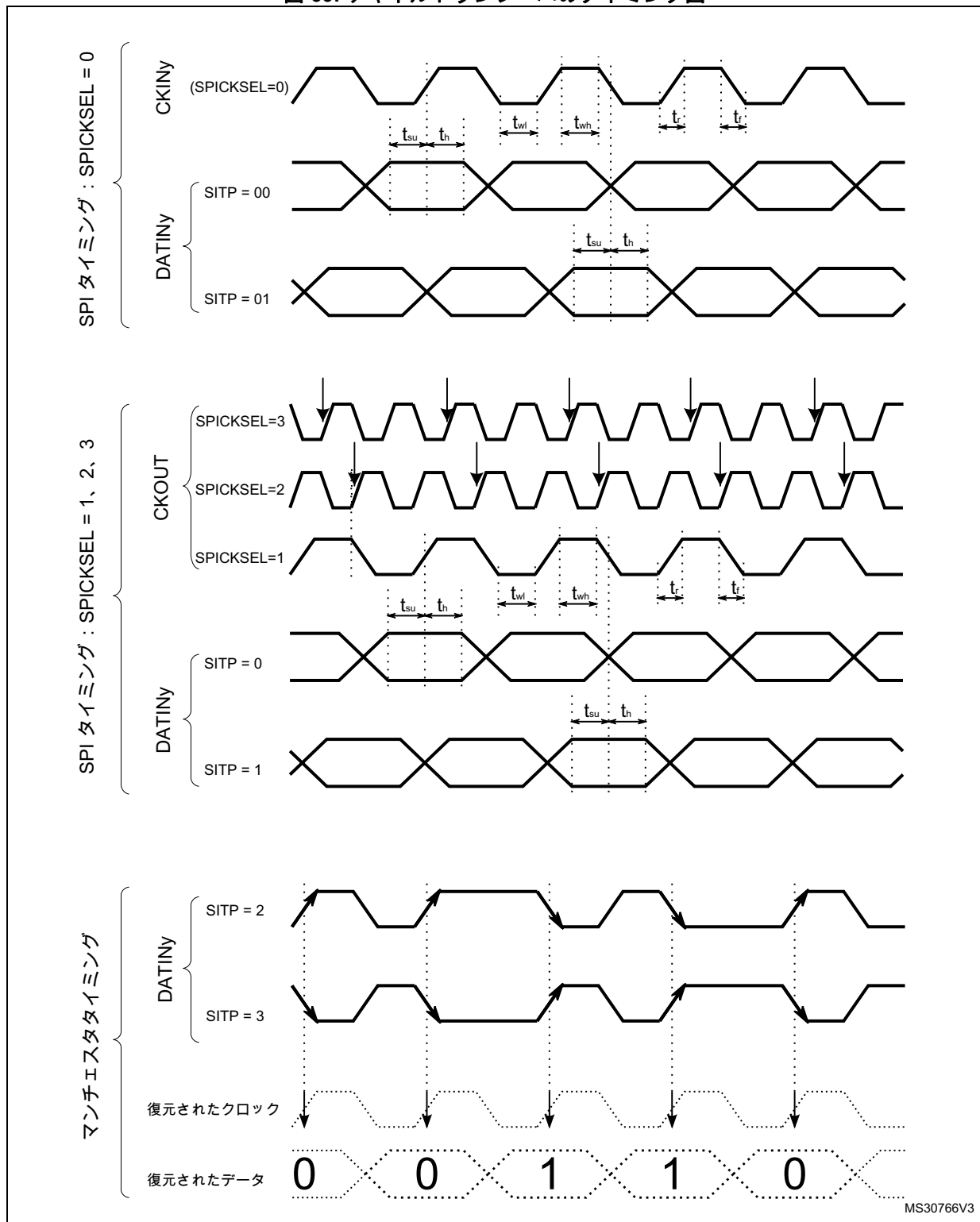
- 信号立ち上がりエッジ = log 0; 信号立ち下がりエッジ = log 1
- 信号立ち上がりエッジ = log 1; 信号立ち下がりエッジ = log 0

マンチェスタ符号化の復元されたクロック信号周波数は、0~10 MHz および $f_{DFSDMCLK}/6$ 未満の範囲でなければなりません。

マンチェスタ符号化データを正しく受信するには、CKOUTDIV 分周器 (DFSDM_CH0CFGR1 レジスタ) が、次の式に応じて想定されるマンチェスタデータレートでセットされる必要があります。

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

図 69. チャネルトランシーバのタイミング図



クロック無検出

チャンネルシリアルクロック入力は、クロック有無のチェックが可能で、変換およびエラーレポートの正しい動作を保証します。クロック無検出は、DFSDM_CHyCFGR1 レジスタの CKABEN ビットによって、各入力チャンネル y で有効/無効にできます。有効にした場合、このクロック無検出は特定のチャンネルで連続して実行されます。クロック無フラグがセット (CKABF[y] = 1) され、入力クロックエラーの際に割り込み (CKABIE = 1 の場合) を呼び出すことができます (DFSDM_FLT0ISR レジスタの CKABF[3:0] および DFSDM_CHyCFGR1 の CKABEN 参照)。クロック無フラグが (DFSDM_FLT0ICR レジスタの CLRCKABF によって) クリアされると、クロック無フラグはリフレッシュされます。クロック無ステータスビット CKABF[y] は、対応チャンネル y が無効である場合、ハードウェアでもセットされます (CHEN[y] = 0 の場合、CKABF[y] はセットで保持されます)。

クロック無イベントが発生した場合、データ変換 (またはアナログウォッチドックおよび短絡検出回路) は誤ったデータを提供します。ユーザはこのイベントを管理し、クロック無がレポートされた際に特定のデータを破棄する必要があります。

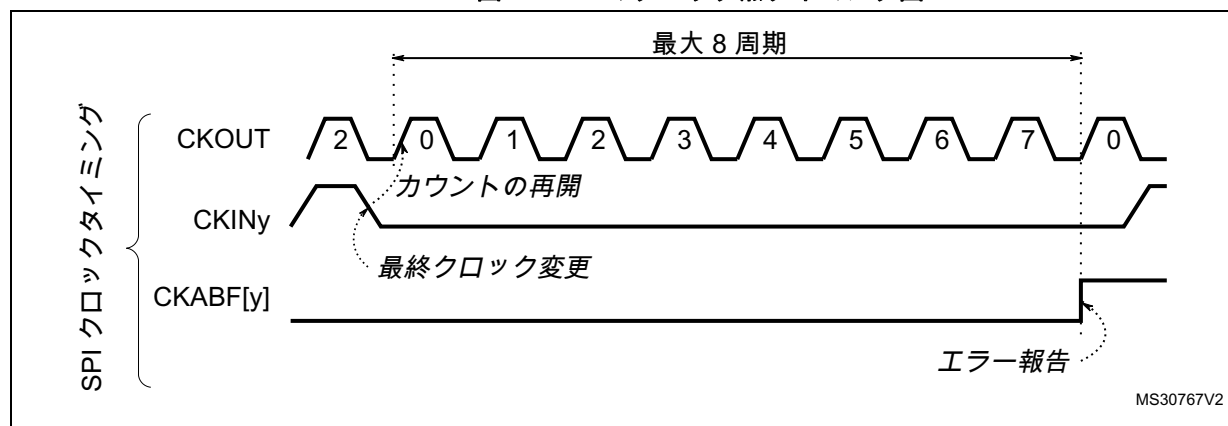
クロック無機能は、システムクロックが CKOUT 信号 (DFSDM_CH0CFGR1 レジスタの CKOUTSRC=0) に使用されている場合のみ使用可能です。

トランシーバがまだ同期されていないときは、クロック無フラグがセットされ、CLRCKABF[y] ビット (DFSDM_FLT0ICR レジスタ) でクリアできません。クロック無検出機能を含むソフトウェアシーケンスは次のとおりにする必要があります。

- CHEN = 1 で特定のチャンネルを有効化します。
- クロック無フラグが実際にクリア (CKABF = 0) されるまで、(CLRCKABF = 1 によって) クロック無フラグのクリアを試みます。この時に、トランシーバが同期され (信号クロックが有効になり)、データを受信できます。
- クロック無機能 CKABEN = 1 と関連する割り込み CKABIE = 1 を有効にし、SPI クロックが損失したか、マンチェスタデータエッジが不足しているかを検出します。

SPI データフォーマットを使用している場合、クロック無検出は出力クロック生成 (CKOUT 信号) のある外部入力クロックの比較に基づきます。入力チャンネルへの外部入力クロック信号は、CKOUT 信号 (DFSDM_CH0CFGR1 レジスタの CKOUTDIV フィールドによって制御) の 8 信号周期に最低 1 回変更する必要があります。

図 70. SPI のクロック無タイミング図



マンチェスタデータフォーマットを使用している場合、クロック無はクロック復元がマンチェスタ符号化信号から実行できないことを示します。正しいクロック復元には、まず 1 から 0 または 0 から 1 の遷移を持つデータを受信する必要があります (マンチェスタ同期については、[図 72](#) を参照)。

マンチェスタ符号化でのクロック無の検出(最初に成功した同期の後)は、出力クロック生成 (CKOUT 信号) のある符号化されたシリアルデータ入力信号の変更比較に基づきます。CKOUT 信号 (DFSDM_CH0CFGR1 レジスタの CKOUTDIV ビットによって制御) の 2 周期間に DATINy ピンの電圧レベル変更が必要になります。この条件は、正しくマンチェスタ符号化データとクロック信号を復元できるように最低データレートも定義します。

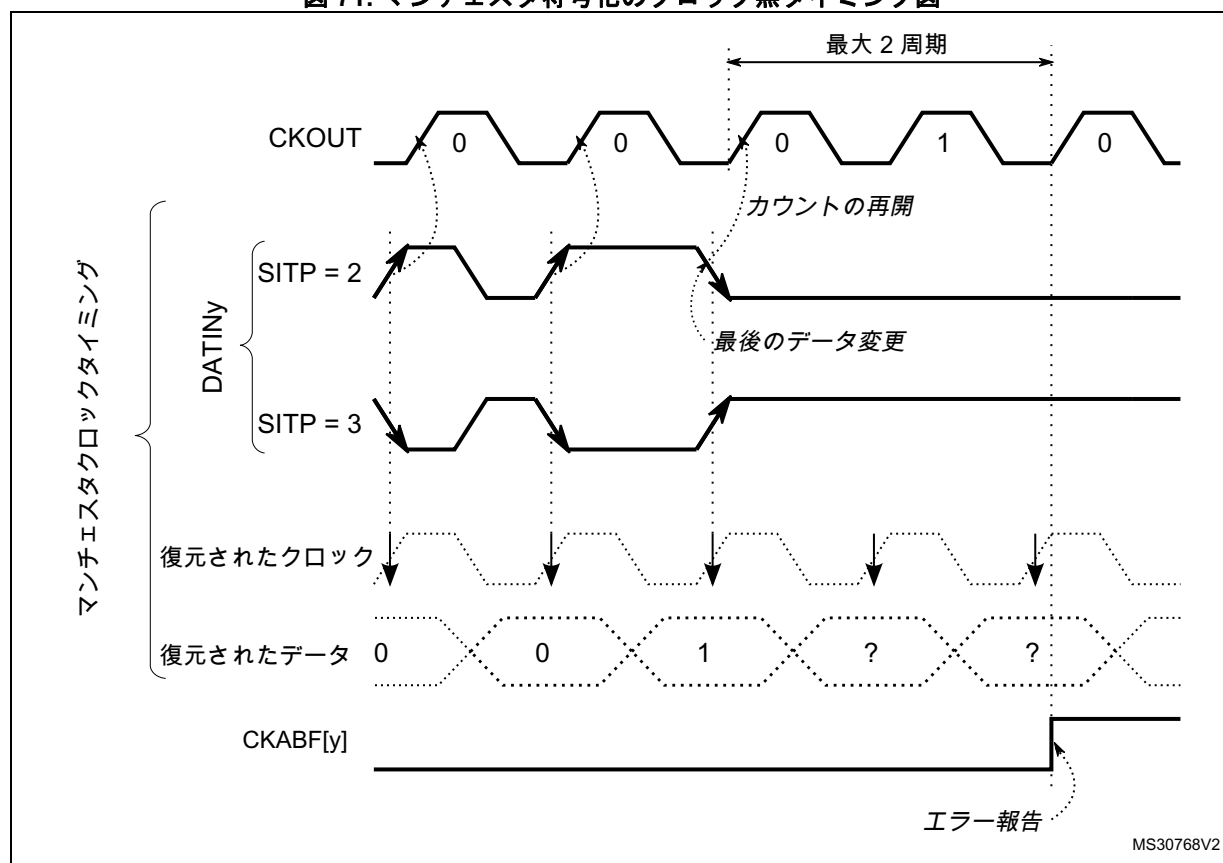
マンチェスタ符号化データの最大データレートは、CKOUT 信号未滿とする必要があります。

そのため、マンチェスタ符号化データを正しく受信するには、CKOUTDIV 分周器は、次の式に応じてセットされる必要があります。

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

クロック無フラグがセット (CKABF[y] = 1) され、入力クロック復元エラーの際に割り込み (CKABIE = 1 の場合) を呼び出すことができます (DFSDM_FLT0ISR レジスタの CKABF[3:0] および DFSDM_CHyCFGR1 の CKABEN)。クロック無フラグが (DFSDM_FLT0ICR レジスタの CLRCKABF によって) クリアされると、クロック無フラグはリフレッシュされます。

図 71. マンチェスタ符号化のクロック無タイミング図



マンチェスタ/SPI のコード同期

チャンネルを有効にした直後、マンチェスタ符号化ストリームを同期する必要があります (DFSDM_CHyCFGR1 レジスタの CHEN=1)。0 から 1 または 1 から 0 のデータ遷移を (有効なデータエッジを検出可能にするために) 受信した時に同期が終了します。同期の終了は、以下に詳述しているソフトウェアシーケンスに従って DFSDM_FLT0ICR の CLRCKABF[y] によってクリアされた後、特定のチャンネルの CKABF[y]=0 をポーリングしてチェックできます。

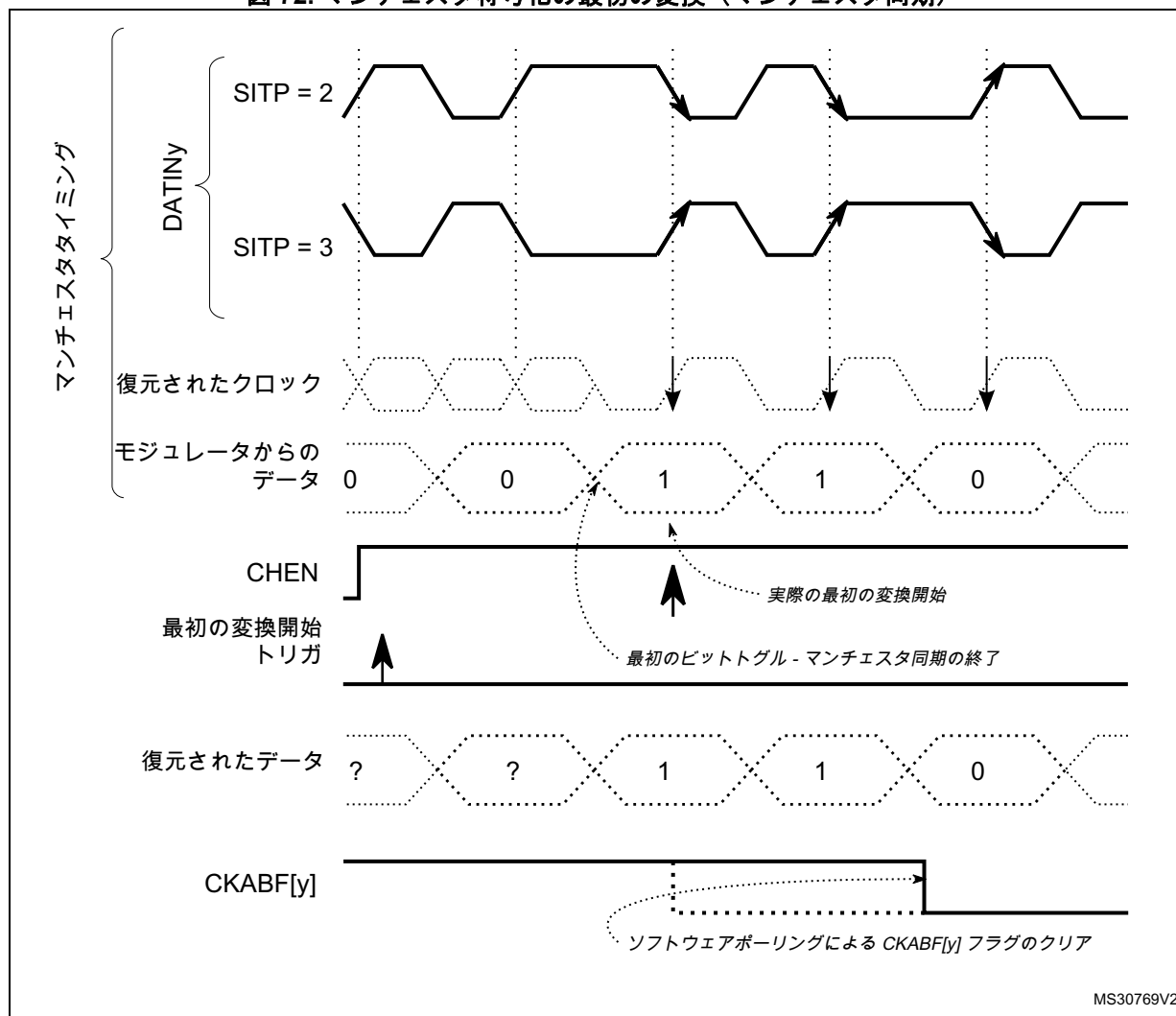
CLRCKABF[y] ビットをセットすることによって、CKABF[y] フラグはクリアされます。チャンネル y がまだ同期されていない場合、ハードウェアは即座に CKABF[y] フラグをセットします。ソフトウェアは CKABF[y] フラグを読み出してセットされている場合、CLRCKABF[y] ビットをセットすることによってこのフラグのクリアを再実行します。このソフトウェアシーケンス (CKABF[y] フラグのポーリング) は、CKABF[y] フラグがセットされる (マンチェスタストリームが同期されていることを通知する) まで続行します。マンチェスタ符号化データを同期/受信できるようにするには、CKOUTDIV 分周器 (DFSDM_CH0CFGR1 レジスタ) が、次の式に応じて想定されるマンチェスタデータレートでセットされる必要があります。

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

SPI 符号化ストリームはクロック入力信号 (有効な立ち上がり/立ち下がりエッジ) を最初に検出した後に同期されます。

注 : **トランシーバがまだ同期されていないときは、クロック無フラグがセットされ、CLRCKABF[y] ビット (DFSDM_FLT0ICR レジスタ) でクリアできません。**

図 72. マンチェスタ符号化の最初の変換 (マンチェスタ同期)



MS30769V2

外部シリアルクロック周波数の測定

チャネルシリアルクロック入力周波数の測定で、アプリケーションに重要な外部デルタシグマモジュレータの実際のデータレートを確認できます。

外部シリアルクロック入力周波数は、1 回の変換時間内にタイマカウンタ DFSDM クロック ($f_{DFSDMCLK}$) で測定できます。カウントは、変換開始 (レギュラまたはインジェクト) 後の最初の入力データクロックで開始し、変換終了 (変換フラグの終了がセット) 前の直前の入力データクロックまでに終了します。変換が終了 (JEOCF=1 または REOCF=1) したときに、各変換時間 (最初から最後のシリアルサンプルまでの時間) が DFSDM_FLTxCNVTIMR レジスタのカンタ CNVCNT[27:0] で更新されます。その後、ユーザはデジタルフィルタ設定 (FORD、FOSR、IOSR、FAST) に応じてデータレートの計算ができます。フィルタがバイパスされた場合 (FOSR=0、積分器のみ有効、DFSDM_FLTxCNVTIMR レジスタで CNVCNT[27:0]=0) のみ、外部シリアル周波数の測定は停止します。

並列データ入力 ([セクション 14.3.6 : 並列データ入力](#)) の場合、測定された周波数は、1 回の変換中の平均入力データレートです。

注： 変換が割り込まれた場合（例：選択したチャンネルの無効化／有効化）、割り込み時間も CNVCNT[27:0] でカウントされます。そのため、正しい変換時間結果を得るには変換に割り込みが発生しないようにすることを推奨します。

変換時間：

FAST = 0 であるインジェクト変換またはレギュラ変換（または FAST=1 の場合の最初の変換）：

Sinc^x フィルタ（x = 1～5）の場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (I_{\text{OSR}} - 1 + F_{\text{ORD}}) + F_{\text{ORD}}] / f_{\text{CKIN}}$$

FastSinc フィルタの場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (I_{\text{OSR}} - 1 + 4) + 2] / f_{\text{CKIN}}$$

FAST = 1 であるレギュラ変換（最初の変換を除く）：

Sinc^x フィルタおよび FastSinc フィルタの場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * I_{\text{OSR}}] / f_{\text{CKIN}}$$

F_{OSR} = FOSR[9:0]+1 = 1（フィルタバイパス、積分器のみ有効）の場合：

$$t = I_{\text{OSR}} / f_{\text{CKIN}} \text{ (... ただし、CNVCNT=0)}$$

ここで、

- f_{CKIN} は、チャンネル入力クロック周波数（特定のチャンネル CKINy ピン）または入力データレート（並列データ入力の場合）です。
- F_{OSR} は、フィルタのオーバーサンプリング比です。 $F_{\text{OSR}} = \text{FOSR}[9:0]+1$ （DFSDM_FLTxFCR レジスタを参照）
- I_{OSR} は、積分器のオーバーサンプリング比です。 $I_{\text{OSR}} = \text{IOSR}[7:0]+1$ （DFSDM_FLTxFCR レジスタを参照）
- F_{ORD} は、フィルタ段数です。 $F_{\text{ORD}} = \text{FORD}[2:0]$ （DFSDM_FLTxFCR レジスタを参照）

チャンネルのオフセット設定

各チャンネルには独自のオフセット設定（レジスタ）があり、特定のチャンネルの各変換結果（インジェクトまたはレギュラ）から最終的に減算されます。オフセット補正は、データの右ビットシフトの後に実行されます。オフセットは、DFSDM_CHyCFGR2 レジスタの OFFSET[23:0] フィールドに 24 ビット符号値として保存されます。

データの右ビットシフト

24 ビット値に整列された結果を得るために、各チャンネルは多くの右ビットシフトを定義し、特定のチャンネルの各変換結果（インジェクトまたはレギュラ）に適用します。データのビットシフト数は、DFSDM_CHyCFGR2 レジスタの DTRBS[4:0] ビットで保存されます。

右ビットシフトは、結果を最も近い整数値に丸めます。シフトされた結果の符号は、結果データの有効な 24 ビット符号フォーマットを持つために維持されます。

14.3.5 入力シリアルインタフェースの設定

次のパラメータは入力シリアルインタフェースに合わせて設定する必要があります。

- **出力クロックの前置分周器。** DFSDM クロック (2~256) から出力クロックを生成するためのプログラム可能な前置分周器があります。これは、DFSDM_CH0CFGR1 レジスタの CKOUTDIV[7:0] ビットによって定義されます。
- **シリアルインタフェースのタイプおよび入力クロックの位相。** SPI 符号化またはマンチェスタ符号化の選択および入力クロックのサンプリングエッジ。これは、DFSDM_CHyCFGR1 レジスタの SITP [1:0] ビットによって定義されます。
- **入力クロックソース。** CKINy ピンからの外部ソースまたは CKOUT ピンからの内部ソース。これは、DFSDM_CHyCFGR1 レジスタの SPICKSEL[1:0] フィールドによって定義されます。
- **最終データの右ビットシフト。** 24 ビット値に整列された結果を得るために最終データの右ビットシフトを定義します。これは、DFSDM_CHyCFGR2 レジスタの DTRBS[4:0] によって定義されます。
- **チャンネルあたりのチャンネルオフセット。** 特定のシリアルチャンネルのアナログオフセット (接続された外部デルタシグマモジュレータのオフセット) を定義します。これは、DFSDM_CHyCFGR2 レジスタの OFFSET[23:0] ビットによって定義されます。
- **チャンネルごとの短絡検出回路およびクロック無の有効化。** DFSDM_CHyCFGR1 レジスタの特定のシリアルチャンネルで短絡検出回路 (SCDEN ビット) およびクロック無監視 (CKABEN ビット) を有効化/無効化します。
- **アナログウォッチドックフィルタや短絡検出回路の閾値設定。** アナログウォッチドックフィルタのパラメータおよびチャンネル短絡検出回路パラメータを設定します。設定は、DFSDM_CHyAWSCDR レジスタで定義されます。

14.3.6 並列データ入力

各入力チャンネルには、16 ビット並列データ入力 (シリアルデータ入力を除く) 用のレジスタがあります。各 16 ビット並列入力は、次の内部データソースからのみ転送できます。

- 直接 CPU/DMA 書き込み。

特定のチャンネルのシリアルデータ入力または並列データ入力の選択は、DFSDM_CHyCFGR1 レジスタの DATMPX[1:0] フィールドで行われます。DATMPX[1:0] は次の並列データソースも定義します。CPU/DMA による直接書き込み。

各チャンネルには、32 ビットデータ入力レジスタ DFSDM_CHyDATINR があり、16 ビットデータで書き込むことができます。データは 16 ビット符号フォーマットです。これらのデータは 16 ビット並列データを受け取るデジタルフィルタへの入力として使用できます。

シリアルデータ入力を選択されている場合 (DATMPX[1:0] = 0)、DFSDM_CHyDATINR レジスタは書き込み保護されます。

メモリからの入力 (直接 CPU/DMA 書き込み)

CPU または DMA (DATMPX[1:0]=2) による DFSDM_CHyDATINR レジスタへの直接データ書き込みは、メモリまたはペリフェラルからのデジタルデータストリームを処理するためのデータ入力に使用できます。

データは、CPU または DMA によって DFSDM_CHyDATINR レジスタに書き込むことができます。

1. CPU データ書き込み:

入力データは、CPU によって DFSDM_CHyDATINR レジスタに直接書き込まれます。

2. DMA データ書き込み:

DFSDM_CHyDATINR レジスタにメモリバッファからデータを転送するには、DMA をメモリ間転送モードに設定する必要があります。転送先メモリアドレスは DFSDM_CHyDATINR レジスタのアドレスです。データは DMA の転送速度でメモリから DFSDM 並列入力に転送されます。

この DMA 転送は、DFSDM 変換結果の読み出しに使用される DMA とは異なります。入力データ書き込み用の最初の DMA (メモリ間転送として設定) と、データ結果読み出し用の 2 つ目の DMA (ペリフェラルからメモリへの転送として設定) は、同時に使用可能です。

DFSDM_CHyDATINR へのアクセスは、16 ビット幅または 32 ビット幅のいずれかで、1 回の書き込み操作で 1 つまたは 2 つのサンプルをそれぞれロードすることができます。32 ビット入力データレジスタ (DFSDM_CHyDATINR) は、DFSDM_CHyCFGR1 レジスタの DATPACK[1:0] フィールドで定義されるデータパッキング動作モードに応じて、1 つまたは 2 つの 16 ビットデータサンプルで満たすことができます。

1. 標準モード (DATPACK[1:0]=0) :

チャンネル y の入力データとして使用される DFSDM_CHyDATINR レジスタの INDAT0[15:0] フィールドにサンプルが 1 つだけ保存されます。上位 16 ビット (INDAT1[15:0]) は無視され、書き込み保護されます。デジタルフィルタは、CPU/DMA によって満たされた後にデータレジスタを空にするために 1 回の入力サンプリング (INDAT0[15:0] から) を実行する必要があります。このモードは書き込み動作ごとに 1 つのサンプルをロードするために DFSDM_CHyDATINR レジスタへの 16 ビット CPU/DMA アクセスとともに使用されます。

2. インタリーブモード (DATPACK[1:0]=1) :

DFSDM_CHyDATINR レジスタは、2 つのサンプルバッファとして使用されます。最初のサンプルは INDAT0[15:0] に、2 つ目のサンプルは INDAT1[15:0] に保存されます。デジタルフィルタは、DFSDM_CHyDATINR レジスタを空にするためにチャンネル y から 2 回の入力サンプリングを実行する必要があります。このモードは書き込み動作ごとに 2 つのサンプルをロードするために DFSDM_CHyDATINR レジスタへの 32 ビット CPU/DMA アクセスとともに使用されます。

3. デュアルモード (DATPACK[1:0]=2) :

2 つのサンプルが、DFSDM_CHyDATINR レジスタに書き込まれます。INDAT0[15:0] のデータはチャンネル y 用、INDAT1[15:0] のデータはチャンネル y+1 用です。INDAT1[15:0] のデータは、次の (y+1) チャンネルデータレジスタ DFSDM_CH[y+1]DATINR の INDAT0[15:0] に自動的にコピーされます。デジタルフィルタは、DFSDM_CHyDATINR レジスタを空にするためにチャンネル y から 1 回、チャンネル (y+1) から 1 回の 2 回のサンプリングを実行する必要があります。

デュアルモード設定 (DATPACK[1:0]=2) は、偶数チャンネル数 (y = 0, 2) でのみ使用できます。奇数チャンネル (y = 1, 3) をデュアルモードにセットすると、このチャンネルの INDAT0[15:0] および INDAT1[15:0] 両方が書き込み保護されます。偶数チャンネルをデュアルモードにセットすると、偶数チャンネルと正しく動作するように次の奇数チャンネルを標準モード (DATPACK[1:0]=0) にセットする必要があります。

DFSDM_CHyDATINR レジスタデータモードおよびチャンネルへのデータサンプルの割り当てについては、[図 73](#) を参照してください。

図 73. DFSDM_CHyDATINR レジスタの動作モードおよび割り当て

標準モード			インタリーブモード			デュアルモード			
31	16 15	0	31	16 15	0	31	16 15	0	
使用されない		Ch0 (サンプル 0)	Ch0 (サンプル 1)		Ch0 (サンプル 0)	Ch1 (サンプル 0)		Ch0 (サンプル 0)	y = 0
使用されない		Ch1 (サンプル 0)	Ch1 (サンプル 1)		Ch1 (サンプル 0)	使用されない		Ch1 (サンプル 0)	y = 1
使用されない		Ch2 (サンプル 0)	Ch2 (サンプル 1)		Ch2 (サンプル 0)	Ch3 (サンプル 0)		Ch2 (サンプル 0)	y = 2
使用されない		Ch3 (サンプル 0)	Ch3 (サンプル 1)		Ch3 (サンプル 0)	使用されない		Ch3 (サンプル 0)	y = 3

MSv40123V1

MSv40123V1

1 つまたは 2 つのサンプルをロードするための DFSDM_CHyDATINR レジスタへの書き込みは、選択した入力チャンネル (チャンネル y) のデータ収集を有効 (チャンネル y の変換開始) にしてから実行される必要があります。そうしないと、書き込まれたデータは次の処理で失われます。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

たとえば、シングル変換モードおよびインタリーブモードの場合、シングル変換開始前に DFSDM_CHyDATINR へのデータサンプルの書き込みを開始しないでください（変換開始前の DFSDM_CHyDATINR に存在するすべてのデータが破棄されます）。

14.3.7 チャネル選択

インジェクトチャネルグループおよびレギュラチャネルによる変換で選択できる 4 つの多重化チャネルがあります。

インジェクトチャネルグループは、4 チャネルのいずれかまたはすべてを選択したものを指します。DFSDM_FLTxJCHGR レジスタの JCHG[3:0] は、JCHG[y]=1 がチャネル y を選択していることを示す場合、インジェクトグループのチャネルを選択します。

インジェクト変換はスキャンモード（JSCAN=1）またはシングルモード（JSCAN=0）で動作可能です。スキャンモードでは、選択された各チャネルが次々と変換されます。最下位チャネル（選択されている場合チャネル 0）が最初に変換され、JCHG[3:0] によって選択されたすべてのチャネルが変換されるまで、次に上位のチャネルを変換し続けます。シングルモード（JSCAN=0）では、選択されたチャネルの 1 つだけを変換し、チャネル選択が次のチャネルに移ります。JSCAN=0 の場合、JCHG[3:0] に書き込むと、チャネル選択が選択された最下位のチャネルにリセットされます。

インジェクト変換は、ソフトウェアまたはトリガから開始できます。レギュラ変換によって割り込まれることはありません。

レギュラチャネルは、4 チャネルのうち 1 つだけ選択したものを指します。DFSDM_FLTxCR1 レジスタの RCH[1:0] は、選択されたチャネルを示します。

レギュラ変換は、ソフトウェア（またはトリガ）から開始できます。連続したレギュラ変換のシーケンスは、インジェクト変換がリクエストされた際に一時的に割り込みが発生します。

無効化されたチャネル（DFSDM_CHyCFGR1 レジスタの CHEN=0）の変換を実行すると、供給される入力データ（クロック信号）がないため変換が終了してしまいます。この場合、特定のチャネル（DFSDM_CHyCFGR1 レジスタの CHEN=1）を有効化するか、DFSDM_FLTxCR1 レジスタを DFEN=0 にして変換を停止する必要があります。

14.3.8 デジタルフィルタ設定

DFSDM には、Sinc^x タイプのデジタルフィルタが実装されています。この Sinc^x フィルタは入力デジタルデータストリームフィルタリングを実行し、出力データレートの減少（デシメーション）および出力データ分解能の向上につながります。Sinc^x フィルタは設定可能で、必要な出力データレートおよび出力データ分解能を満たすことができます。設定可能なパラメータは次のとおりです。

- フィルタ段数／タイプ：（DFSDM_FLTxFCR レジスタの FORD[2:0] ビットを参照）：
 - FastSinc
 - Sinc¹
 - Sinc²
 - Sinc³
 - Sinc⁴
 - Sinc⁵
- オーバーサンプリング比／デシメーション率（DFSDM_FLTxFCR レジスタの FOSR[9:0] ビットを参照）
 - FOSR = 1-1024 - FastSinc フィルタおよび Sinc^x フィルタ x = F_{ORD} = 1~3 の場合
 - FOSR = 1-215 - Sinc^x フィルタ x = F_{ORD} = 4 の場合
 - FOSR = 1-73 - Sinc^x フィルタ x = F_{ORD} = 5 の場合

フィルタには次の転送機能 (H ドメインのインパルスレスポンス) があります。

- Sinc^x フィルタタイプ : $H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^x$
- FastSinc フィルタタイプ : $H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$

図 74. 例 : Sinc³ フィルタレスポンス :

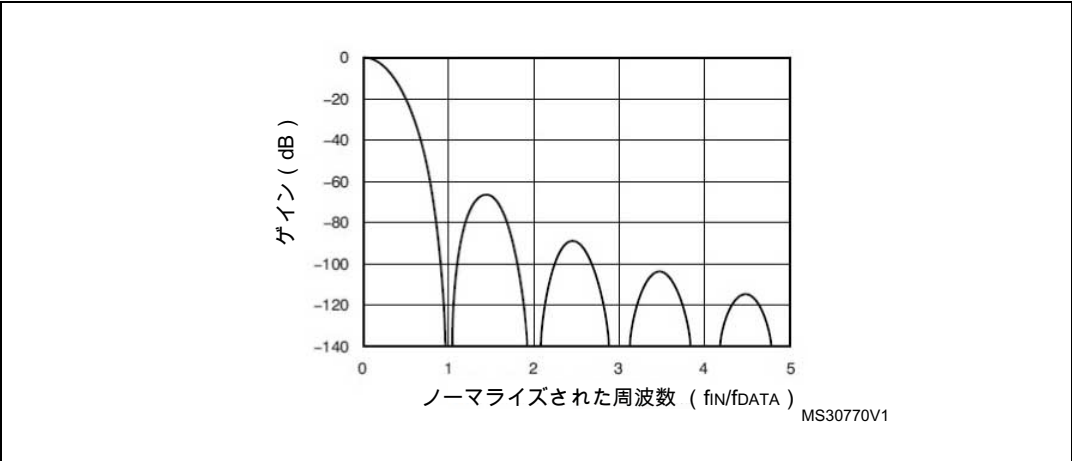


表 85. フィルタの最大出力分解能 (フィルタ出力のピークデータ値) 一部の FOSR 値

FOSR	Sinc ¹	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	+/- x	+/- x ²	+/- 2x ²	+/- x ³	+/- x ⁴	+/- x ⁵
4	+/- 4	+/- 16	+/- 32	+/- 64	+/- 256	+/- 1024
8	+/- 8	+/- 64	+/- 128	+/- 512	+/- 4096	-
32	+/- 32	+/- 1024	+/- 2048	+/- 32768	+/- 1048576	+/- 33554432
64	+/- 64	+/- 4096	+/- 8192	+/- 262144	+/- 16777216	+/- 1073741824
128	+/- 128	+/- 16384	+/- 32768	+/- 2097152	+/- 268435456	結果はフルスケール入力 (> 32 ビット符号整数) で オーバーフローできます
256	+/- 256	+/- 65536	+/- 131072	+/- 16777216		
1024	+/- 1024	+/- 1048576	+/- 2097152	+/- 1073741824		

Sinc フィルタタイプのプロパティおよび使用の詳細については、デジタルフィルタに関する理論を学習されることを推奨します (さらに多くの資料をインターネットからダウンロードできます)。

14.3.9 積分器ユニット

積分器は、デジタルフィルタからのデータにおける追加のデシメーションおよび分解能向上を実行します。積分器は、単純にフィルタからの特定のデータサンプル数に対してデジタルフィルタからのデータの合計を実行します。

積分器のオーバーサンプリング比パラメータは、積分器からの 1 回のデータ出力に合計されるデータカウント数を定義します。IOSR は、1~256 の範囲でセットできます (DFSDM_FLTxFCR レジスタの IOSR[7:0] ビットの説明を参照)

**表 86. 積分器の最大出力分解能 (積分器出力のピークデータ値)
(一部の IOSR 値、FOSR = 256 および Sinc³ フィルタタイプ (最大データ) の場合)**

IOSR	Sinc ¹	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	+/- FOSR. x	+/- FOSR ² . x	+/- 2.FOSR ² . x	+/- FOSR ³ . x	+/- FOSR ⁴ . x	+/- FOSR ⁵ . x
4	-	-	-	+/- 67 108 864	-	-
32	-	-	-	+/- 536 870 912	-	-
128	-	-	-	+/- 2 147 483 648	-	-
256	-	-	-	+/- 2 ³²	-	-

14.3.10 アナログウォッチドッグ

アナログウォッチドッグの目的は、アナログ信号が特定の最大および最低閾値に達した場合に、外部信号 (ブレイクまたは割り込み) をトリガすることです。この時、割り込み/イベント/ブレイク生成が呼び出されます。

各アナログウォッチドッグは、AWFSEL ビット設定 (DFSDM_FLTxCR1 レジスタ) に応じて、シリアルデータレシーバ出力 (各チャンネルでアナログウォッチドッグフィルタ後) またはデータ出力レジスタ (現在のインジェクト変換またはレギュラ変換の結果) を監視します。アナログウォッチドッグ x での監視対象入力チャンネルかどうかは、DFSDM_FLTxCR2 レジスタの AWDCH[3:0] で選択されます。

入力チャンネルのアナログウォッチドッグ変換は、標準変換から独立します。この場合、アナログウォッチドッグはメインのインジェクト変換またはレギュラ変換から独立した各入力チャンネルで独自のフィルタと信号処理を使用します。アナログウォッチドッグ変換は、メインのインジェクト変換またはレギュラ変換が一時停止 (RCIP = 0、JCIP = 0) 状態でもチャンネルを監視するために、選択された入力チャンネルで連続モードで実行されます。

特定のデータ値 (DFSDM_FLTxAWHTR レジスタの AWHT[23:0] ビットおよび DFSDM_FLTxAWLTR レジスタの AWLT[23:0] ビットでセット) と比較される高閾値および低閾値があります。

データ値を持つ閾値レジスタを比較する 2 つのオプションがあります。

- オプション 1: この場合、入力データは最終出力データレジスタ (AWFSEL=0) から取得されます。このオプションは次の特徴があります。
 - 入力データの高分解能 (最大 24 ビット)
 - スローな応答時間 - 過電流検出のような高速応答アプリケーションには不適切です。
 - 比較のために最終データはビットシフトおよびオフセットデータ補正後に取得されます。
 - メインのレギュラ変換またはインジェクト変換が実行された後のみ最終データを利用できます。
 - 並列入力データソース (DFSDM_CHyCFGR1 レジスタで DATMPX[1:0] ≠ 0) の場合に使用可能です。
- オプション 2: この場合、入力データはあらゆるシリアルデータレシーバ出力 (AWFSEL=1) から取得されます。このオプションは次の特徴があります。
 - 入力シリアルデータは、設定可能なオーバーサンプリング比 (1 ~ 32) およびフィルタ段数 (1~3) がある専用のアナログウォッチドック Sinc^x チャネルフィルタによって処理されます (DFSDM_CHyAWSCDR レジスタの AWFOSR[4:0] および AWFORD[1:0] ビットの設定を参照)。
 - 低分解能 (最大 16 ビット)
 - 高速な応答時間 - 過電流/過電圧検出のような高速応答が必要なアプリケーションに適切です。
 - データは、メインのレギュラ変換またはインジェクト変換アクティビティとは独立した連続モードで利用できます。

入力チャネルを監視している場合 (AWFSEL=1)、閾値と比較するデータは、AWDCH[3:0] フィールド (DFSDM_FLTxCR2 レジスタ) で選択されたチャネルから取得されます。選択されたチャネルフィルタの各結果は、1 つの閾値ペア (AWHT[23:0] / AWLT[23:0]) と比較されます。この場合、アナログウォッチドックフィルタからのデータは最大 16 ビットの分解能となるため、上位の 16 ビット (AWHT[23:8] / AWLT[23:8]) のみが、アナログウォッチドックフィルタ出力と比較される 16 ビット閾値を定義します。AWHT[7:0] / AWLT[7:0] ビットは、この場合 (AWFSEL=1) の比較では取得されません。

各入力チャネルのアナログウォッチドックフィルタ設定のパラメータは、DFSDM_CHyAWSCDR レジスタ (フィルタ段数 AWFORD[1:0] およびフィルタオーバーサンプリング比 AWFOSR[4:0]) でセットされます。

各入力チャネルには、アナログウォッチドックデータ (アナログウォッチドックフィルタから) をアナログウォッチドックの閾値 (AWHT/AWLT) を比較する独自のコンパレータがあります。いくつかのチャネル (DFSDM_FLTxCR2 レジスタの AWDCH[3:0] フィールド) が選択されている場合、複数の比較リクエストを同時に受信する場合があります。この場合、最低値のチャネルリクエストがまず管理され、選択されたチャネルを昇順に管理していきます。チャネルごとに、結果を個別のフラグ (DFSDM_FLTxAWSR レジスタの AWHTF[3:0]、AWLTF[3:0] フィールド) で記録できます。各チャネルリクエストは 8 DFSDM クロックサイクルで実行されます。そのため、各チャネルの帯域幅は 8 DFSDM クロックサイクルに制限されます (AWDCH[3:0] = 0x0F の場合)。最大入力チャネルのサンプリングクロック周波数は、4 で割った DFSDM クロック周波数となるため、設定 AWFOSR = 0 (アナログウォッチドックフィルタがバイパス) はこの入力クロック速度でアナログウォッチドック機能に使用できません。そのため、ユーザは入力サンプリングクロック速度および DFSDM 周波数の点で、監視されるチャネルの数とアナログウォッチドックフィルタのパラメータを適切に設定する必要があります。

特定のチャンネル y のアナログウォッチドックフィルタデータは、DFSDM_CHyWDATR レジスタの WDATA[15:0] フィールドでファームウェアによる準備に使用できます。そのアナログウォッチドックフィルタデータは (DFSDM_CHyCFGR1 レジスタが CHEN=1 の場合)、アナログウォッチドックフィルタ設定およびチャンネル入力クロック周波数によって提供されるデータレートで連続して変換されます。

アナログウォッチドックフィルタ変換は、積分器のないレギュラ高速連続変換のように機能します。アナログウォッチドックフィルタ出力からの 1 つの結果に必要なシリアルサンプル数 (チャンネル入力クロック周波数 f_{CKIN} において) :

最初の変換 :

Sinc^x フィルタ ($x = 1 \sim 5$) の場合 : サンプル数 = $[F_{OSR} * F_{ORD} + F_{ORD} + 1]$

FastSinc フィルタの場合 : サンプル数 = $[F_{OSR} * 4 + 2 + 1]$

次の変換 :

Sinc^x フィルタおよび FastSinc フィルタの場合 : サンプル数 = $[F_{OSR} * IOSR]$

ここで、

F_{OSR} フィルタのオーバーサンプリング比 : $F_{OSR} = AWFOSR[4:0] + 1$ (DFSDM_CHyAWSCDR レジスタを参照)

F_{ORD} フィルタ段数 : $F_{ORD} = AWFORD[1:0]$ (DFSDM_CHyAWSCDR レジスタを参照)

出力データレジスタを監視する場合 (AWFSEL=0)、最終データの右ビットシフトおよびオフセット補正 (DFSDM_CHyCFGR2 レジスタの OFFSET[23:0] および DTRBS[4:0] フィールドを参照) 後に実行されます。比較は、各インジェクト変換またはレギュラ変換の終了後に AWDCH[3:0] フィールド (DFSDM_FLTxCR2 レジスタ) で選択されたチャンネルに実行されます。

アナログウォッチドックイベントのステータスは、特定のイベントがラッチされる DFSDM_FLTxAWSR レジスタで信号化されます。AWHTF[y]=1 フラグは、チャンネル y を跨ぐ AWHT[23:0] 値を信号化します。AWLTF[y]=1 フラグは、チャンネル y を跨ぐ AWLT[23:0] 値を信号化します。DFSDM_FLTxAWSR レジスタでラッチされたイベントは、DFSDM_FLTxAWCFR レジスタの対応するクリアリングビット CLRAWHTF[y] または CLRAWLTF[y] に“1”を書き込むことによってクリアされます。

アナログウォッチドックのグローバルステータスは、DFSDM_FLTxISR レジスタの AWDF フラグビットによって信号化されます (割り込みソースの高速検出に使用されます)。AWDF=1 は、最低 1 つのウォッチドックイベントが発生すると信号化します (最低 1 チャンネルに対して AWHTF[y]=1 または AWLTF[y]=1)。AWDF ビットはすべての AWHTF[3:0] および AWLTF[3:0] がクリアされたときにクリアされます。

アナログウォッチドックイベントはブレイク出力信号に割り当てることができます。高閾値または低閾値に跨るイベントに割り当て 4 つのブレイク出力があります (dfsdm_break[3:0])。特定のアナログウォッチドックイベントへのブレイク信号割り当ては、DFSDM_FLTxAWHTR レジスタおよび DFSDM_FLTxAWLTR レジスタの BKAWH[3:0] および BKAWL[3:0] フィールドで実行されます。

14.3.11 短絡検出回路

短絡検出回路の目的は、アナログ信号が飽和した値（フルスケール範囲外）に達してこの値が一定時間残った場合に、非常に高速な応答時間で信号化することです。この挙動は、短絡を検出したり、回路のエラー（例：過電流または過電圧）を明らかにできます。割り込み／イベント／ブレイク生成が呼び出されます。

短絡検出回路への入力データは、チャンネルトランシーバ出力から取得されます。

各入力チャンネルにアップカウントするカウンタがあり、シリアルデータレシーバ出力で連続した 0 と 1 をカウントしています。受信したデータストリームの変更（データ信号の 1 から 0、0 から 1 への変更）がある場合に、カウンタが再開します。このカウンタは短絡閾値レジスタ値 (DFSDM_CHyAWSCDR レジスタの SCDT[7:0] ビット) に達すると、短絡イベントが呼び出されます。各入力チャンネルには、短絡検出回路があります。SCDEN ビット (DFSDM_CHyCFGR1 レジスタ) をセットすることで、すべてのチャンネルを継続的な監視対象として選択でき、それぞれ独自の短絡検出回路設定があります (SCDT[7:0] ビットの閾値、ステータスビット SCDF[3:0]、ステータスクリアリングビット CLRSCDF[3:0])。ステータスフラグ SCDF[y] は、対応チャンネル y が無効である場合 (CHEN[y] = 0)、ハードウェアでもクリアされます。

各チャンネルで、短絡検出回路イベントはブレイク出力信号 dfsdm_break[3:0] に割り当てることができます。短絡検出回路イベントに割り当てる 4 つのブレイク出力があります。特定のチャンネル短絡検出回路イベントへのブレイク信号割り当ては、DFSDM_CHyAWSCDR レジスタの BKSCD[3:0] フィールドで実行されます。

並列入力データチャンネル選択 (DFSDM_CHyCFGR1 レジスタで DATMPX[1:0] ≠ 0) の場合に短絡検出回路は使用できません。

4 つのブレイク出力がすべて使用できます (アナログウォッチドック機能と共用)。

14.3.12 極値検出回路

極値検出回路の目的は、最終出力データワードの最小値と最大値（ピークトゥピーク値）を収集することです。

出力データワードが、極値検出回路の最大レジスタ (DFSDM_FLTxEXMAX レジスタの EXMAX[23:0] ビット) に保存された値を超える場合、このレジスタは現在の出力データワード値およびデータが EXMAXCH[1:0] ビット (DFSDM_FLTxEXMAX レジスタ) に保存されるチャンネルとともに更新されます。

出力データワードが、極値検出回路の最小レジスタ (DFSDM_FLTxEXMIN レジスタの EXMIN[23:0] ビット) に保存された値を下回る場合、このレジスタは現在の出力データワード値およびデータが EXMINCH[1:0] ビット (DFSDM_FLTxEXMIN レジスタ) に保存されるチャンネルとともに更新されます。

最小および最大レジスタ値は (特定の DFSDM_FLTxEXMAX または DFSDM_FLTxEXMIN の読み出しで) ソフトウェアによってリフレッシュできます。リフレッシュ後、極値検出回路の最小データレジスタ DFSDM_FLTxEXMIN には 0x7FFFFFFF (正の最大値) が、最大レジスタ DFSDM_FLTxEXMAX には 0x800000 (負の最小値) が入ります。

右ビットシフトおよびオフセットデータ補正後、極値検出回路は比較を実行します。各極値検出回路に対して、極値の計算に考慮される入力チャンネルが EXCH[3:0] ビット (DFSDM_FLTxCR2 レジスタ) で選択されます。

14.3.13 データユニットブロック

データユニットブロックは全体の処理パスの最後のブロックです。外部デルタシグマモジュレータ - シリアルトランシーバ - Sinc フィルタ - 積分器 - データユニットブロック。

出力データレートは、シリアルデータストリームレート、フィルタ、積分器の設定に依存します。最大出力データレートは次のとおりです。

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DFSDM_CKIN}}}{F_{\text{OSR}} \cdot (I_{\text{OSR}} - 1 + F_{\text{ORD}}) + (F_{\text{ORD}} + 1)} \quad \dots \text{FAST} = 0, \text{Sincx filter}$$

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DFSDM_CKIN}}}{F_{\text{OSR}} \cdot (I_{\text{OSR}} - 1 + 4) + (2 + 1)} \quad \dots \text{FAST} = 0, \text{FastSinc filter}$$

または

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DFSDM_CKIN}}}{F_{\text{OSR}} \cdot I_{\text{OSR}}} \quad \dots \text{FAST} = 1$$

並列データ入力の場合の最大出力データレート：

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DATAIN_RATE}}}{F_{\text{OSR}} \cdot (I_{\text{OSR}} - 1 + F_{\text{ORD}}) + (F_{\text{ORD}} + 1)} \quad \dots \text{FAST} = 0, \text{Sincx filter}$$

または

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DATAIN_RATE}}}{F_{\text{OSR}} \cdot (I_{\text{OSR}} - 1 + 4) + (2 + 1)} \quad \dots \text{FAST} = 0, \text{FastSinc filter}$$

または

$$\text{Datarate}[\text{サンプル /s}] = \frac{f_{\text{DATAIN_RATE}}}{F_{\text{OSR}} \cdot I_{\text{OSR}}} \quad \dots \text{FAST}=1 \text{ またはあらゆるフィルタバイパスの場合 } (F_{\text{ORD}} = 1)$$

where: $f_{\text{DATAIN_RATE}}$...input data rate from CPU/DMA

最終データ幅は 24 ビットとなり、処理パスからのデータは最大 32 ビットまで可能であるため、最終データの右ビットシフトはこのモジュールで実行されます。この右ビットシフトは選択された各入力チャネルに対して 0~31 ビットの範囲で設定可能です (DFSDM_CHyCFGR2 レジスタの DTRBS[4:0] ビットを参照)。右ビットシフトは、結果を最も近い整数値に丸めます。シフトされた結果の符号は、結果データの有効な 24 ビット符号フォーマットを持つために維持されます。

次のステップでは、結果のオフセット補正が実行されます。オフセット補正值 (レジスタ DFSDM_CHyCFGR2 に保存された OFFSET[23:0]) は、特定のチャネルの出力データから減算されます。OFFSET[23:0] フィールドのデータは、適切な較正ルーチンでソフトウェアによってセットされます。

デジタル処理のすべての動作は 32 ビット符号レジスタで実行されるため、結果がオーバーフローしないように次の条件を満たす必要があります。

$$\begin{aligned} F_{\text{OSR}}^{F_{\text{ORD}}} \cdot I_{\text{OSR}} &\leq 2^{31} \quad \dots \text{Sinc}^x \text{ フィルタ、} x = 1 \sim 5 \text{ の場合} \\ 2 \cdot F_{\text{OSR}}^2 \cdot I_{\text{OSR}} &\leq 2^{31} \quad \dots \text{FastSinc フィルタの場合} \end{aligned}$$

注： フィルタおよび積分器バイパスの場合 ($IOSR[7:0]=0$ 、 $FOSR[9:0]=0$)、すべての出力データを読み出せるように入力データレート (f_{DATAIN_RATE}) を制限する必要があります。
 $f_{DATAIN_RATE} \leq f_{APB}$
 ここで、 f_{APB} は DFSDM ペリフェラルが接続されているバス周波数です。

14.3.14 符号付きデータフォーマット

DFSDM 入力シリアルチャネルにつき、1 つの外部デルタシグマモジュレータに接続できます。外部デルタシグマモジュレータには、2 つの異なる入力 (正と負) を持つことができ、差分またはシングルエンド信号測定に使用できます。

デルタシグマモジュレータ出力は常に符号付きフォーマットで想定されています (デルタシグマモジュレータからの 0 と 1 のデータストリームは -1 および +1 の値を表します)。

レジスタの符号付きデータフォーマット： データは、最終出力データ、アナログウォッチドック、極値検出回路、オフセット補正のレジスタで符号付きフォーマットになります。出力データワードの最上位ビットは、値の符号を表します (2 の補数フォーマット)。

14.3.15 変換の開始

インジェクト変換は、次の方法で開始できます。

- ソフトウェア：DFSDM_FLTxCR1 レジスタの JSWSTART に“1”を書き込みます。
- トリガ：JEXTSEL[2:0] は、同時に JEXTEN がトリガを有効化してアクティブエッジを選択する際にトリガ信号を選択します (DFSDM_FLTxCR1 レジスタを参照)。
- JSYNC=1 の場合の DFSDM_FLT0 との同期：DFSDM_FLTx ($x>0$) の場合、インジェクト変換は自動的に開始します。DFSDM_FLT0 の場合、インジェクト変換はソフトウェアによって開始します (DFSDM_FLT0CR2 レジスタで JSWSTART=1)。DFSDM_FLTx ($x>0$) での各インジェクト変換は、常にローカル設定 (JSCAN、JCHG など) に応じて実行されます。

スキャン変換が有効である場合 (ビット JSCAN=1)、インジェクト変換がトリガされるたびに、インジェクトグループで選択されたすべてのチャネル (DFSDM_FLTxJCHGR レジスタの JCHG[3:0] ビット) が最下位のチャネル (選択されている場合チャネル 0) から順番に変換されます。

スキャン変換が無効である場合 (ビット JSCAN=0)、インジェクト変換がトリガされるたびに、インジェクトグループで選択されたチャネル (DFSDM_FLTxJCHGR レジスタの JCHG[3:0] ビット) が 1 つだけ変換されると、チャネル選択が次に選択されたチャネルに移ります。JSCAN=0 の場合、JCHG[3:0] に書き込むと、チャネル選択が選択された最下位のインジェクトチャネルにセットされます。

指定した時間で実行できるインジェクト変換は 1 回だけです。このため、インジェクト変換の他のリクエストがすでに発行されており、未完である場合、インジェクト変換を開始するあらゆるリクエストは無視されます。

レギュラ変換は、次の方法で開始できます。

- ソフトウェア：DFSDM_FLTxCR1 レジスタの RSWSTART に“1”を書き込みます。
- RSYNC=1 の場合の DFSDM_FLT0 との同期：DFSDM_FLTx ($x>0$) の場合、レギュラ変換は自動的に開始します。DFSDM_FLT0 の場合、レギュラ変換はソフトウェアによって開始します (DFSDM_FLT0CR2 レジスタで RSWSTART=1)。DFSDM_FLTx ($x>0$) での各レギュラ変換は、常にローカル設定 (RCONT、RCH など) に応じて実行されます。

指定した時間で保留または実行できるレギュラ変換は 1 回だけです。このため、レギュラ変換の他のリクエストがすでに発行されており、未完である場合、レギュラ変換を開始するあらゆるリクエストは無視されます。レギュラ変換は、インジェクト変換による割り込み、またはインジェクト変換進行中に開始が発生した場合、保留できます。この保留中のレギュラ変換は延期され、すべてのインジェクト変換終了後に実行されます。延期されたレギュラ変換は、DFSDM_FLTxRDATAR レジスタの RPEND ビットで信号化されます。



14.3.16 連続モードおよび高速連続モード

DFSDM_FLTxCR1 レジスタの RCONT をセットすると、レギュラ変換が連続モードで実行されます。RCONT=1 は、RSWSTART に“1”が書き込まれた後に RCH[1:0] によって選択されたチャンネルが繰り返し変換されることを示します。

連続モードで実行されるレギュラ変換は、RCONT に“0”を書き込むことで停止できます。RCONT をクリアすると、進行中の変換は即座に停止します。

連続モードでは、DFSDM_FLTxCR1 レジスタの FAST ビットをセットすることによって、データレートを上げることができます。この場合、1 つのチャンネルから連続して変換する場合、フィルタ内のデータは前にサンプルした連続データから有効になるため、フィルタに新しいデータを入れる必要はありません。速度の増加は、選択したフィルタ段数に依存します。RSWSTART=1 による連続変換の開始後、高速モード (FAST=1) での最初の変換は、フルタイム (FAST=0 の場合と同じくらい) かかりますが、後続の変換はより短い間隔で終了します。

連続モードでの変換時間：

FAST = 0 (または FAST=1 の場合の最初の変換) の場合：

Sinc^x フィルタの場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (I_{\text{OSR}} - 1 + F_{\text{ORD}}) + F_{\text{ORD}}] / f_{\text{CKIN}}$$

FastSinc フィルタの場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (I_{\text{OSR}} - 1 + 4) + 2] / f_{\text{CKIN}}$$

FAST = 1 (最初の変換を除く) の場合：

Sinc^x フィルタおよび FastSinc フィルタの場合：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * I_{\text{OSR}}] / f_{\text{CKIN}}$$

$F_{\text{OSR}} = \text{FOSR}[9:0] + 1 = 1$ (フィルタバイパス、積分器のみ有効) の場合：

$$t = I_{\text{OSR}} / f_{\text{CKIN}} \text{ (... ただし、CNVCNT=0)}$$

連続モードはインジェクト変換では使用できません。インジェクト変換はタイマトリガによって開始でき、正確なタイミングで連続モードをエミュレートします。

レギュラ連続変換が進行中の場合 (RCONT=1) およびレギュラ連続変換をリクエストする DFSDM_FLTxCR1 レジスタに書き込みアクセス (RCONT=1) が実行された場合、レギュラ連続変換は次の変換サイクルから再開されます (DFSDM_FLTxCR1 レジスタに変更がない場合でも、新しいレギュラ連続変換が新しいチャンネル選択に適用されるのと同様)。

14.3.17 リクエスト優先度

インジェクト変換は、レギュラ変換より優先されます。すでに進行中のレギュラ変換は、インジェクト変換のリクエストによって即座に割り込まれます。このレギュラ変換は、インジェクト変換終了後に再開されます。

他のインジェクト変換が保留中またはすでに進行中の場合、インジェクト変換は開始できません。インジェクト変換を開始するあらゆるリクエスト (JSWSTART またはトリガいずれかによるもの) は、ビット JCIP (DFSDM_FLTxISR レジスタ) が“1”である限り、無視されます。

同様に、他のレギュラ変換が保留中またはすでに進行中の場合、レギュラ変換は開始できません。レギュラ変換を開始するあらゆるリクエスト (RSWSTART によるもの) は、ビット RCIP (DFSDM_FLTxISR レジスタ) が“1”である限り、無視されます。

しかし、レギュラ変換がすでに進行中の状態でインジェクト変換がリクエストされた場合、レギュラ変換は即座に停止し、インジェクト変換が開始されます。その後、レギュラ変換は再開し、この延期された開始はビット RPEND で信号化されます。

インジェクト変換は一時的に連続レギュラ変換のシーケンスに割り込める点で、レギュラ変換より優先されます。インジェクト変換のシーケンスが終了すると、RCONT がまだセットされている場合、連続レギュラ変換が再開します（また、RPEND ビットは最初のレギュラ変換結果において延期した開始を信号化します）。

優先度には、DFSDM に同じ書き込みによっていつアクションが開始されたか、他のアクションの最後に複数のアクションが保留されているかどうかの影響します。たとえば、インジェクト変換の処理中（JCIP=1）に、DFSDM_FLTxCR1 へのシングル書き込み操作があり、RSWSTART に“1”が書き込まれ、レギュラ変換をリクエストしたとします。インジェクト変換シーケンスが終了すると、優先度からレギュラ変換を次に実行して、延期された開始が RPEND ビットで信号化されることとなります。

14.3.18 RUN モードでの省電力

消費電力を低減するために、DFSDM フィルタおよび積分器は、変換で使用していない場合に自動的にアイドル状態に入ります（RCIP=0、JCIP=0）。

14.4 DFSDM 割り込み

CPU 性能を向上させるため、CPU イベント発生に関連する割り込みが実装されています。

- インジェクト変換終了割り込み：
 - DFSDM_FLTxCR2 レジスタの JEOCIE ビットによって有効になります。
 - DFSDM_FLTxISR レジスタの JEOCF ビットで示されます。
 - DFSDM_FLTxJDATAR レジスタの読み出しによってクリアされます（インジェクトデータ）。
 - 変換終了が発生したチャンネルを示します。DFSDM_FLTxJDATAR レジスタの JDATACH[1:0] ビットでレポートされます。
- レギュラ変換終了割り込み：
 - DFSDM_FLTxCR2 レジスタの REOCIE ビットによって有効になります。
 - DFSDM_FLTxISR レジスタの REOCF ビットで示されます。
 - DFSDM_FLTxRDATAR レジスタの読み出しによってクリアされます（レギュラデータ）。
 - 変換終了が発生したチャンネルを示します。DFSDM_FLTxRDATAR レジスタの RDATACH[1:0] ビットでレポートされます。
- インジェクト変換のデータオーバーラン割り込み：
 - インジェクト変換データが（CPU または DMA によって）DFSDM_FLTxJDATAR レジスタから読み出されず、新しいインジェクト変換によって上書きされた場合に発生します。
 - DFSDM_FLTxCR2 レジスタの JOVRIE ビットによって有効になります。
 - DFSDM_FLTxISR レジスタの JOVRF ビットで示されます。
 - DFSDM_FLTxICR レジスタの CLRJOVRF ビットに“1”を書き込むことによってクリアされます。
- レギュラ変換のデータオーバーラン割り込み：
 - レギュラ変換データが（CPU または DMA によって）DFSDM_FLTxRDATAR レジスタから読み出されず、新しいレギュラ変換によって上書きされた場合に発生します。
 - DFSDM_FLTxCR2 レジスタの ROVRIE ビットによって有効になります。
 - DFSDM_FLTxISR レジスタの ROVRF ビットで示されます。
 - DFSDM_FLTxICR レジスタの CLRROVRF ビットに“1”を書き込むことによってクリアされます。

- アナログウォッチドッグ割り込み：
 - － 変換されたデータ (DFSDM_FLTxCR1 レジスタの AWFSEL ビット設定に応じた出力データまたはアナログウォッチドックフィルタからのデータ) が DFSDM_FLTxAWHTR / DFSDM_FLTxAWLTR レジスタの高／低閾値に跨る場合に発生します。
 - － DFSDM_FLTxCR2 レジスタの AWDIE ビット (選択されたチャネル AWDCH[3:0]) によって有効になります。
 - － DFSDM_FLTxISR レジスタの AWDF ビットで示されます。
 - － DFSDM_FLTxAWSR レジスタの AWHTF[3:0] フィールドおよび AWLTF[3:0] フィールドによってアナログウォッチドックの高閾値または低閾値を個別に示します。
 - － DFSDM_FLTxAWCFR レジスタの対応する CLRAWHTF[3:0] ビットまたは CLRAWLTF[3:0] ビットに“1”を書き込むことによってクリアされます。
- 短絡検出回路割り込み：
 - － 安定したデータ数が DFSDM_CHyAWSCDR レジスタの閾値に跨る場合に発生します。
 - － DFSDM_FLTxCR2 レジスタの SCDIE ビット (DFSDM_CHyCFGR1 レジスタの SCDEN ビットによって選択されたチャネル) によって有効になります。
 - － DFSDM_FLTxISR レジスタの SCDF[3:0] ビットで示されます (短絡検出回路イベントが発生したチャネルもレポート)。
 - － DFSDM_FLTxICR レジスタの対応する CLRSCDF[3:0] ビットに“1”を書き込むことによってクリアされます。
- チャネルクロック無割り込み：
 - － CKINy ピンにクロックがない場合に発生します ([セクション 14.3.4: シリアルチャネルトランシーバのクロック無検出](#)を参照)
 - － DFSDM_FLTxCR2 レジスタの CKABIE ビット (DFSDM_CHyCFGR1 レジスタの CKABEN ビットによって選択されたチャネル) によって有効になります。
 - － DFSDM_FLTxISR レジスタの CKABF[y] ビットで示されます。
 - － DFSDM_FLTxICR レジスタの CLRCKABF[y] ビットに“1”を書き込むことによってクリアされます。

表 87. DFSDM 割り込みリクエスト

割り込みイベント	イベントフラグ	イベント／割り込みのクリア方法	割り込み有効制御ビット
インジェクト変換終了	JEOCF	DFSDM_FLTxJDATAR を読み出す	JEOCIE
レギュラ変換終了	REOCF	DFSDM_FLTxRDATAR を読み出す	REOCIE
インジェクトデータオーバーラン	JOVRF	CLRJOVRF = 1 を書き込む	JOVRIE
レギュラデータオーバーラン	ROVRF	CLRROVRF = 1 を書き込む	ROVRIE
アナログウォッチドッグ	AWDF、 AWHTF[3:0]、 AWLTF[3:0]	CLRAWHTF[3:0] = 1 を書き込む CLRAWLTF[3:0] = 1 を書き込む	AWDIE、 (AWDCH[3:0])
短絡検出回路	SCDF[3:0]	CLRSCDF[3:0] = 1 を書き込む	SCDIE、 (SCDEN)
チャネルクロック無	CKABF[3:0]	CLRCKABF[3:0] = 1 を書き込む	CKABIE、 (CKABEN)

14.5 DFSDM の DMA 転送

CPU 介入を低減するために、変換を DMA 転送を使用してメモリに転送することができます。インジェクト変換の DMA 転送は、DFSDM_FLTxCR1 レジスタの JDMAEN=1 ビットをセットすることで有効になります。レギュラ変換の DMA 転送は、DFSDM_FLTxCR1 レジスタの RDMAEN=1 ビットをセットすることで有効になります。

注： DMA 転送で、DMA が DFSDM_FLTxJDATAR レジスタまたは DFSDM_FLTxRDATAR レジスタを読み出すため、割り込みフラグはインジェクト変換またはレギュラ変換 (DFSDM_FLTxISR レジスタの JEOCF または REOCF ビット) の最後に自動的にクリアされます。

14.6 DFSDM チャネル y レジスタ (y=0~3)

14.6.1 DFSDM チャネル設定 y レジスタ (DFSDM_CHyCFGR1) (y=0~3)

このレジスタは、チャネル y (y = 0~3) によって使用されるパラメータを指定します。

アドレスオフセット : $0x00 + 0x20 * y$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSDM EN	CKOUT SRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]							
r/w	r/w							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHIN SEL	CHEN	CKAB EN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
r/w	r/w	r/w	r/w				r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w

ビット 31 **DFSDMEN** : DFSDM インタフェースのグローバル有効ビット

0 : DFSDM インタフェース無効

1 : DFSDM インタフェース有効

DFSDM インタフェースが有効である場合、有効化された y チャネルおよび x フィルタ設定 (DFSDM_CHyCFGR1 の CHEN ビットおよび DFSDM_FLTxCR1 の DFEN ビット) に応じて動作が開始します。DFSDMEN=0 をセットすることでクリアされるデータ :

– すべてのレジスタ DFSDM_FLTxISR (x = 0~1) がリセット状態にセットされます。

– すべてのレジスタ DFSDM_FLTxAWSR (x = 0~1) がリセット状態にセットされます。

注： DFSDMEN は、DFSDM_CH0CFGR1 レジスタ (チャネル y=0) にのみ存在します。

ビット 30 **CKOUTSRC** : 出力シリアルクロックソースの選択

0 : システムクロックからの出力クロックソース

1 : オーディオクロックからの出力クロックソース

– CKDFSDMASEL ビットによって選択された I2S クロック (セクション 6.3.24 : RCC 専用クロック設定レジスタ (RCC_DCKCFGR))

この値は、DFSDMEN=0 (DFSDM_CH0CFGR1 レジスタ) の場合にのみ修正できます。

注： CKOUTSRC は、DFSDM_CH0CFGR1 レジスタ (チャネル y=0) にのみ存在します。

ビット 29:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **CKOUTDIV[7:0]** : 出力シリアルクロック分周回路

- 0 : 出力クロック生成は無効です (CKOUT 信号はロー状態にセットされます)。
 - 1~255 : 2 ~ 256 の範囲で CKOUT 信号のシリアルクロック出力のシステムクロックの分周を定義します (分周回路 = CKOUTDIV+1)。
- CKOUTDIV はクロック無検出の閾値も定義します。
- この値は、DFSDMEN=0 (DFSDM_CH0CFGR1 レジスタ) の場合にのみ修正できます。
- DFSDMEN=0 (DFSDM_CH0CFGR1 レジスタ) の場合、CKOUT 信号はロー状態にセットされます (設定は DFSDMEN=0 の後の 1 DFSDM クロックサイクルで実行されます)。

注: **CKOUTDIV は、DFSDM_CH0CFGR1 レジスタ (チャンネル y=0) にのみ存在します。**

ビット 15:14 **DATPACK[1:0]** : DFSDM_CHyDATINR レジスタのデータパッキングモード

- 0 : 標準 : DFSDM_CHyDATINR レジスタの入力データは INDAT0[15:0] にのみ保存されます。DFSDM_CHyDATINR レジスタを空にするには、1 つのサンプルをチャンネル y から DFSDM フィルタで読み出す必要があります。
 - 1 : インタリーブ : DFSDM_CHyDATINR レジスタの入力データは次の 2 つのサンプルとして保存されます。
 - INDAT0[15:0] の最初のサンプル (チャンネル y に割り当て)
 - INDAT1[15:0] の 2 つ目のサンプル (チャンネル y に割り当て)DFSDM_CHyDATINR レジスタを空にするには、2 つのサンプルをチャンネル y からデジタルフィルタで読み出す必要があります (INDAT0[15:0] パートは最初のサンプルとして、INDAT1[15:0] パートは次のサンプルとして読み出されます)。
 - 2 : デュアル : DFSDM_CHyDATINR レジスタの入力データは次の 2 つのサンプルとして保存されます。
 - INDAT0[15:0] の最初のサンプル (チャンネル y に割り当て)
 - INDAT1[15:0] の 2 つ目のサンプル (チャンネル y+1 に割り当て)DFSDM_CHyDATINR レジスタを空にするには、最初のサンプルをチャンネル y からデジタルフィルタで、2 つ目のサンプルをチャンネル y+1 から別のデジタルフィルタで読み出す必要があります。デュアルモードは、偶数チャンネル数 (y = 0, 2) で使用でき、奇数チャンネル数 (y = 1, 3) の場合、DFSDM_CHyDATINR は書き込み保護されます。偶数チャンネルをデュアルモードにセットすると、偶数チャンネルと正しく動作するように次の奇数チャンネルを標準モード (DATPACK[1:0]=0) にセットする必要があります。
 - 3 : 予約済み
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

ビット 13:12 **DATMPX[1:0]** : チャンネル y の入力データマルチプレクサ

- 0 : チャンネル y へのデータは外部シリアル入力から 1 ビット値で取得されます。DFSDM_CHyDATINR レジスタは書き込み保護されています。
 - 1 : 予約済み
 - 2 : チャンネル y へのデータは内部 DFSDM_CHyDATINR レジスタから直接 CPU/DMA 書き込みで取得されます。DATPACK[1:0] ビットフィールド設定に応じて 1 つまたは 2 つの 16 ビットデータサンプルを書き込むことができます。
 - 3 : 予約済み
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

ビット 11:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **CHINSEL** : チャンネル入力の選択

- 0 : チャンネル入力と同じチャンネル y のピンから取得されます。
 - 1 : チャンネル入力があるチャンネル (チャンネル (y+1) モジュール 8) のピンから取得されます。
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

ビット 7 **CHEN** : チャンネル y 有効化

- 0 : チャンネル y 無効
 - 1 : チャンネル y 有効
- チャンネル y が有効である場合、シリアルデータの受信は特定のチャンネル設定に応じて開始されます。

ビット 6 **CKABEN** : チャンネル y のクロック無検出回路有効化

- 0 : チャンネル y のクロック無検出回路無効
- 1 : チャンネル y のクロック無検出回路有効

- ビット 5 **SCDEN** : チャネル y の短絡検出回路有効化
- 0 : 入力チャネル y が、短絡検出回路によって保護されません。
 - 1 : 入力チャネル y が、短絡検出回路によって継続的に保護されます。

ビット 4 予約済みであり、リセット値に保持する必要があります。

- ビット 3:2 **SPICKSEL[1:0]** : チャネル y の SPI クロック選択
- 0 : 外部 CKIN y 入力からのクロック - SITP[1:0] に応じたサンプリングポイント
 - 1 : 内部 CKOUT 出力からのクロック - SITP[1:0] に応じたサンプリングポイント
 - 2 : 内部 CKOUT からクロック - 毎秒の CKOUT 立ち下がりエッジでのサンプリングポイント
クロック入力 (CKOUT から) を 2 つに分周する外部デルタシグマモジュレータに接続して出力シリアル通信クロックを生成する場合 (また、この出力クロック変更が各クロック入力の立ち上がりエッジで有効である場合)
 - 3 : 内部 CKOUT 出力からのクロック - 毎秒の CKOUT 立ち上がりエッジでのサンプリングポイント
クロック入力 (CKOUT から) を 2 つに分周する外部デルタシグマモジュレータに接続して出力シリアル通信クロックを生成する場合 (また、この出力クロック変更が各クロック入力の立ち下がりエッジで有効である場合)
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

- ビット 1:0 **SITP[1:0]** : チャネル y のシリアルインタフェースのタイプ
- 00 : データをストローブする立ち上がりエッジのある SPI
 - 01 : データをストローブする立ち下がりエッジのある SPI
 - 10 : DATIN y ピンでのマンチェスタ符号化入力: 立ち上がりエッジ = ロジック 0、立ち下がりエッジ = ロジック 1
 - 11 : DATIN y ピンでのマンチェスタ符号化入力: 立ち上がりエッジ = ロジック 1、立ち下がりエッジ = ロジック 0
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

14.6.2 DFSDM チャネル設定 y レジスタ (DFSDM_CHyCFGR2) ($y=0\sim3$)

このレジスタは、チャネル y ($y=0\sim3$) によって使用されるパラメータを指定します。

アドレスオフセット : $0x04 + 0x20 * y$

リセット値 : $0x0000\ 0000$

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

- ビット 31:8 **OFFSET[23:0]** : チャネル y の 24 ビット 較正オフセット
- チャネル y では、OFFSET がチャネルからの各変換結果に適用されます。
- この値はソフトウェアでセットされます。

- ビット 7:3 **DTRBS[4:0]** : チャネル y のデータ右ビットシフト
- 0~31 : 積分器からのデータ結果のシフト (最終結果を得るために右に何ビットシフトするか) を定義します。
- ビットシフトは、オフセット較正前に実行されます。データシフトは、結果を最も近い整数値に丸めます。シフトされた結果の符号は、(結果データの有効な 24 ビット符号フォーマットを持つために) 維持されます。
- この値は、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。



14.6.3 DFSDM チャネルアナログウォッチドックおよび短絡検出回路レジスタ (DFSDM_CHyAWSCDR) (y=0~3)

チャネル y (y = 0~3) の短絡検出回路およびアナログウォッチドック設定

アドレスオフセット : 0x08 + 0x20 * y

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]				
								r/w	r/w		r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
r/w	r/w	r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **AWFORD[1:0]** : チャネル y のアナログウォッチドッグ Sinc フィルタ段数

0 : FastSinc フィルタタイプ

1 : Sinc¹ フィルタタイプ

2 : Sinc² フィルタタイプ

3 : Sinc³ フィルタタイプ

Sinc^x フィルタタイプの転送機能 :

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^x$$

FastSinc フィルタタイプの転送機能 :

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

このビットは、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20:16 **AWFOSR[4:0]** : チャネル y のアナログウォッチドッグフィルタのオーバーサンプリング比 (デシメーション率)

0~31 : Sinc タイプフィルタの長さを 1~32 の範囲で定義します (AWFOSR + 1)。この数字は、アナログデータレートのデシメーション率でもあります。

このビットは、CHEN=0 (DFSDM_CHyCFGR1 レジスタ) の場合にのみ修正できます。

注 : **AWFOSR = 0 の場合、フィルタに効果はありません (フィルタバイパス)。**

ビット 15:12 **BKSCD[3:0]** : チャネル y の短絡検出回路のブレーク信号割り当て

BKSCD[i] = 0 : チャネル y の短絡検出回路に割り当てられていないブレーク i 信号

BKSCD[i] = 1 : チャネル y の短絡検出回路に割り当てられたブレーク i 信号

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **SCDT[7:0]** : チャネル y の短絡検出回路の閾値

これらのビットは、短絡検出回路の閾値カウンタを定義するために、ソフトウェアによって書き込まれます。この値に達すると、特定のチャネルで短絡検出回路イベントが発生します。

14.6.4 DFSDM チャネルウォッチドックフィルタデータレジスタ
(DFSDM_CHyWDATR) (y=0~3)

このレジスタは、入力チャネル y (y = 0~3) に関連付けされたアナログウォッチドックフィルタからのデータを含みます。

アドレスオフセット : 0x0C + 0x20 * y

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **WDATA[15:0]** : 入力チャネル y ウォッチドックデータ

入力チャネル y のアナログウォッチドックフィルタによって変換されたデータです。このデータは、このチャネルに対して制限付き分解能（OSR=1~32/Sinc 段数 = 1~3）で連続して変換されます（トリガなし）。

14.6.5 DFSDM チャネルデータ入力レジスタ（DFSDM_CHyDATINR）（y=0～3）

このレジスタには、16 ビットの入力データが含まれており、DFSDM フィルタモジュールによって処理されます。

アドレスオフセット：0x10 + 0x20 * y

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDAT1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDAT0[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **INDAT0[15:0]**：チャネル y またはチャネル y+1 の入力データ

DATMPX[1:0]=1 または DATMPX[1:0]=2 の場合にデジタルフィルタによって処理される入力並列チャネルデータです。

データは、CPU/DMA によって書き込むことができます（DATMPX[1:0]=2 の場合）。

DATPACK[1:0]=0（標準モード）の場合

INDAT0[15:0] は書き込み保護されています（入力サンプルには使用しません）。

DATPACK[1:0]=1（インタリーブモード）の場合

2 つ目のチャネル y データサンプルは、INDAT1[15:0] に保存されます。最初のチャネル y データサンプルは、INDAT0[15:0] に保存されます。両方のサンプルは 2 つのチャネル y データサンプルとして DFSDM_FLTx フィルタによって順番に読み出されます。

DATPACK[1:0]=2（デュアルモード）の場合

偶数 y チャネルの場合：INDAT1[15:0] のサンプルは、チャネル (y+1) の INDAT0[15:0] に自動的にコピーされます。

奇数 y チャネルの場合：INDAT1[15:0] は書き込み保護されています。

詳細は、[セクション 14.3.6：並列データ入力](#) を参照してください。

INDAT0[15:1] は 16 ビット符号フォーマットです。

ビット 15:0 **INDAT0[15:0]**：チャネル y の入力データ

DATMPX[1:0]=1 または DATMPX[1:0]=2 の場合にデジタルフィルタによって処理される入力並列チャネルデータです。

データは、CPU/DMA によって書き込むことができます（DATMPX[1:0]=2 の場合）。

DATPACK[1:0]=0（標準モード）の場合

チャネル y データサンプルは、INDAT0[15:0] に保存されます。

DATPACK[1:0]=1（インタリーブモード）の場合

最初のチャネル y データサンプルは、INDAT0[15:0] に保存されます。2 つ目のチャネル y データサンプルは、INDAT1[15:0] に保存されます。両方のサンプルは 2 つのチャネル y データサンプルとして DFSDM_FLTx フィルタによって順番に読み出されます。

DATPACK[1:0]=2（デュアルモード）の場合

偶数 y チャネルの場合：チャネル y データサンプルは、INDAT0[15:0] に保存されます。

奇数 y チャネルの場合：INDAT0[15:0] は書き込み保護されています。

詳細は、[セクション 14.3.6：並列データ入力](#) を参照してください。

INDAT0[15:0] は 16 ビット符号フォーマットです。

14.7 DFSDM フィルタ x モジュールレジスタ (x=0~1)

14.7.1 DFSDM 制御レジスタ 1 (DFSDM_FLTxCR1)

アドレスオフセット : $0x100 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWFSEL	FAST	Res.	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	RSYNC	RCON T	RSW START	Res.
	rw	rw				rw	rw			rw		rw	rw	r0w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	JEXTEN[1:0]		Res.	Res.	JEXTSEL[2:0]			Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	rw	rw			rw	rw	rw			rw	rw	rw		r0w	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **AWFSEL** : アナログウォッチドッグ高速モード選択

0 : データ出力値のアナログウォッチドッグ (デジタルフィルタ後)。オフセット補正およびシフト後に比較が実行されます

1 : チャネルトランシーバ値のアナログウォッチドッグ (ウォッチドッグフィルタ後)

ビット 29 **FAST** : レギュラ変換の高速変換モード選択

0 : 高速変換モード無効

1 : 高速変換モード有効

連続モードでレギュラ変換を行うとき、高速モードを有効にすると、各変換 (最初を除く) を標準モードより速く実行できます。このビットは、連続していない変換では効果がありません。

このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

FAST = 0 (または FAST=1 の場合の連続モードにおける最初の変換) の場合 :

$$t = [F_{OSR} * (I_{OSR} - 1 + F_{ORD}) + F_{ORD}] / f_{CKIN} \dots \text{Sinc}^x \text{ フィルタの場合}$$

$$t = [F_{OSR} * (I_{OSR} - 1 + 4) + 2] / f_{CKIN} \dots \text{FastSinc フィルタの場合}$$

連続モードの FAST = 1 (最初の変換を除く) の場合 :

$$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$$

$F_{OSR} = F_{OSR}[9:0] + 1 = 1$ (フィルタバイパス、積分器のみ有効) の場合 :

$$t = I_{OSR} / f_{CKIN} \dots \text{ただし、CNVCNT=0}$$

ここで、 f_{CKIN} は、チャネル入力クロック周波数 (特定のチャネル CKINy ピン) または並列データ入力の場合の入力データレートです。

ビット 28:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **RCH[1:0]** : レギュラチャネル選択

0 : チャネル 0 がレギュラチャネルとして選択されます。

1 : チャネル 1 がレギュラチャネルとして選択されます。

RCIP=1 のときにこのビットに書き込むと、次のレギュラ変換開始時に有効になります。これは、連続モードで特に役立ちます (RCONT=1 の場合)。保留中のレギュラ変換にも影響します (進行中のインジェクト変換のため)。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **RDMAEN** : DMA チャネルのレギュラ変換のデータ読み出し有効化

0 : DMA チャネルがレギュラデータの読み出し有効になりません。

1 : DMA チャネルがレギュラデータの読み出し有効になります。

このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

ビット 20 予約済みであり、リセット値に保持する必要があります。



ビット 19 **RSYNC** : DFSDM_FLT0 と同期したレギュラ変換開始

- 0 : DFSDM_FLT0 と同期してレギュラ変換を開始しません。
- 1 : DFSDM_FLT0 でレギュラ変換が開始された時点で、この DFSDM_FLTx でレギュラ変換を開始します。
このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

ビット 18 **RCONT** : レギュラ変換の連続モード選択

- 0 : レギュラチャネルが、各変換リクエストにつき 1 回だけ変換されます。
- 1 : レギュラチャネルが、各変換リクエストの後に繰り返し変換されます。
連続レギュラ変換がすでに進行中の場合にこのビットに“0”を書き込むと、即座に連続モードを停止します。

ビット 17 **RSWSTART** : ソフトウェアのレギュラチャネルの変換開始

- 0 : “0”を書き込んでも、ビットの値は変化しません。
- 1 : “1”を書き込むと、レギュラ変換開始のリクエストをして、RCIP が“1”になります。すでに RCIP=1 の場合、RSWSTART に書き込んでも変化はありません。RSYNC=1 の場合、“1”を書き込んでも、ビットの値は変化しません。
このビットは常に“0”として読み出されます。

ビット 16:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:13 **JEXTEN[1:0]** : インジェクト変換のトリガ有効化およびトリガエッジ選択

- 00 : トリガ検出は無効です。
- 01 : 選択されたトリガの各立ち上がりエッジがインジェクト変換の開始をリクエストします。
- 10 : 選択されたトリガの各立ち下がりエッジがインジェクト変換の開始をリクエストします。
- 11 : 選択されたトリガの立ち上がりエッジおよび立ち下がりエッジ両方がインジェクト変換の開始をリクエストします。
このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

ビット 12:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **JEXTSEL[2:0]** : インジェクト変換開始のトリガ信号選択

- 0x0~0x7 : 次の表で選択されるトリガ入力
このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

	DFSDM_FLT0	DFSDM_FLT1
0x00	dfsdm_jtrg0	dfsdm_jtrg0
0x01	dfsdm_jtrg1	dfsdm_jtrg1
0x02	dfsdm_jtrg2	dfsdm_jtrg2
0x03	dfsdm_jtrg3	dfsdm_jtrg3
0x04	dfsdm_jtrg5	dfsdm_jtrg5
0x05	dfsdm_jtrg7	dfsdm_jtrg7
0x06	dfsdm_jtrg9	dfsdm_jtrg9
0x07	dfsdm_jtrg10	dfsdm_jtrg10

表 84 : DFSDM トリガ接続を参照してください。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **JDMAEN** : DMA チャネルのインジェクトチャネルグループのデータ読み出し有効化

- 0 : DMA チャネルがインジェクトデータの読み出し有効になりません。
- 1 : DMA チャネルがインジェクトデータの読み出し有効になります。
このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

ビット 4 **JSCAN** : インジェクト変換のスキャン変換モード

- 0 : 1 つのチャネル変換がチャネルグループから実行されてから、このグループで選択されたチャネルが選択されます。
- 1 : インジェクトグループチャネルの一連の変換が実行され、選択された最下位のチャネルから開始します。
このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。
JSCAN=0 の場合、JCHG に書き込むと、チャネル選択が選択された最下位のチャネルにリセットされます。

- ビット 3 **JSYNC** : DFSDM_FLT0 JSWSTART トリガと同期したインジェクト変換開始
- 0 : DFSDM_FLT0 と同期してインジェクト変換を開始しません。
 - 1 : JSWSTART トリガによって DFSDM_FLT0 でインジェクト変換が開始された時点で、この DFSDM_FLTx でインジェクト変換を開始します。
- このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **JSWSTART** : インジェクトチャネルグループの変換開始
- 0 : “0”を書き込んでも、ビットの値は変化しません。
 - 1 : “1”を書き込むと、インジェクト変換グループでのチャネル変換のリクエストをして、同時に JCIP が“1”になります。すでに JCIP=1 の場合、JSWSTART に書き込んでも変化はありません。JSYNC=1 の場合、“1”を書き込んでも、ビットの値は変化しません。
- このビットは常に“0”として読み出されます。
- ビット 0 **DFEN** : DFSDM_FLTx 有効化
- 0 : DFSDM_FLTx は無効です。特定の DFSDM_FLTx のすべての変換が即座に停止し、すべての DFSDM_FLTx 機能が停止します。
 - 1 : DFSDM_FLTx は有効です。DFSDM_FLTx が有効である場合、DFSDM_FLTx は設定に応じて動作を開始します。
- DFEN=0 をセットすることでクリアされるデータ :
- レジスタ DFSDM_FLTxISR は、リセット状態にセットされます。
 - レジスタ DFSDM_FLTxAWSR は、リセット状態にセットされます。

14.7.2 DFSDM 制御レジスタ 2 (DFSDM_FLTxCR2)

アドレスオフセット : $0x104 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	CKAB IE	SCDIE	AWDIE	ROVR IE	JOVR IE	REOC IE	JEOC IE
				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

- ビット 31:20 予約済みであり、リセット値に保持する必要があります。
- ビット 19:16 **AWDCH[3:0]** : アナログウォッチドッグチャネル選択
- これらのビットは、アナログウォッチドッグによって継続的に保護される入力チャネルを選択します。
- AWDCH[y] = 0 : チャネル y のアナログウォッチドッグは無効です。
- AWDCH[y] = 1 : チャネル y のアナログウォッチドッグは有効です。
- ビット 15:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11:8 **EXCH[3:0]** : 極値検出回路チャネル選択
- これらのビットは、極値検出回路によって取得される入力チャネルを選択します。
- EXCH[y] = 0 : 極値検出回路はチャネル y からデータを受け取りません。
- EXCH[y] = 1 : 極値検出回路はチャネル y からデータを受け取ります。
- ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CKABIE** : クロック無割り込み有効化

- 0 : チャネル入カクロック無割り込みの検出を無効にします。
 - 1 : チャネル入カクロック無割り込みの検出を有効にします。
- CKABF[3:0] (DFSDM_FLTxISR) の説明を参照してください。

注: **CKABIE** は、**DFSDM_FLT0CR2 レジスタ (フィルタ x=0) にのみ存在します。**

ビット 5 **SCDIE** : 短絡検出回路割り込み有効化

- 0 : 短絡検出回路割り込み無効
 - 1 : 短絡検出回路割り込み有効
- SCDF[3:0] (DFSDM_FLTxISR) の説明を参照してください。

注: **SCDIE** は、**DFSDM_FLT0CR2 レジスタ (フィルタ x=0) にのみ存在します。**

ビット 4 **AWDIE** : アナログウォッチドッグ割り込みイネーブル

- 0 : アナログウォッチドッグ割り込みは無効です。
 - 1 : アナログウォッチドッグ割り込みは有効です。
- AWDF (DFSDM_FLTxISR) の説明を参照してください。

ビット 3 **ROVRIE** : レギュラデータオーバーラン割り込み有効化

- 0 : レギュラデータオーバーラン割り込み無効
 - 1 : レギュラデータオーバーラン割り込み有効
- ROVRF (DFSDM_FLTxISR) の説明を参照してください。

ビット 2 **JOVRIE** : インジェクトデータオーバーラン割り込み有効化

- 0 : インジェクトデータオーバーラン割り込み無効
 - 1 : インジェクトデータオーバーラン割り込み有効
- JOVRF (DFSDM_FLTxISR) の説明を参照してください。

ビット 1 **REOCIE** : レギュラ変換終了割り込み有効化

- 0 : レギュラ変換終了割り込み無効
 - 1 : レギュラ変換終了割り込み有効
- REOCF (DFSDM_FLTxISR) の説明を参照してください。

ビット 0 **JEOCIE** : インジェクト変換終了割り込み有効化

- 0 : インジェクト変換終了割り込み無効
 - 1 : インジェクト変換終了割り込み有効
- JEOCF (DFSDM_FLTxISR) の説明を参照してください。

14.7.3 DFSDM 割り込みおよびステータスレジスタ (DFSDM_FLTxISR)

アドレスオフセット : $0x108 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SCDF[3:0]				Res.	Res.	Res.	Res.	CKABF[3:0]			
				r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	r	r									r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **SCDF[3:0]** : 短絡検出回路フラグ

SDCF[y]=0 : チャンネル y の短絡検出回路イベントが発生していません。
SDCF[y]=1 : チャンネル y で短絡検出回路カウンタが DFSDM_CHyAWSCDR レジスタにプログラムされた値に達しています。
このビットは、ハードウェアによってセットされます。DFSDM_FLTxICR レジスタの対応する CLRSCDF[y] ビットで、ソフトウェアによってクリアできます。SCDF[y] は、CHEN[y] = 0 (特定のチャンネルが無効) の場合、ハードウェアでもクリアされます。

注 : **SCDF[3:0]** は、**DFSDM_FLT0ISR** レジスタ (フィルタ x=0) にのみ存在します。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **CKABF[3:0]** : クロック無フラグ

CKABF[y]=0 : チャンネル y のクロック信号があります。
CKABF[y]=1 : チャンネル y のクロック信号がありません。
特定の y ビットは、クロック無がチャンネル y で検出された際にハードウェアによってセットされます。CHEN=0 (DFSDM_CHyCFGR1 レジスタを参照) の場合、ハードウェアによって CKABF[y]=1 状態で保持されます。トランシーバがまだ同期されていない場合、CKABF[y]=1 状態で保持されます。DFSDM_FLTxICR レジスタの対応する CLRCKABF[y] ビットで、ソフトウェアによってクリアできます。

注 : **CKABF[3:0]** は、**DFSDM_FLT0ISR** レジスタ (フィルタ x=0) にのみ存在します。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **RCIP** : 進行中のレギュラ変換ステータス

0 : 発行されたレギュラチャンネルの変換のリクエストはありません。
1 : レギュラチャンネルの変換が進行中またはレギュラ変換のリクエストが保留中です。
RCIP=1 の場合、レギュラ変換開始のリクエストは無視されます。

ビット 13 **JCIP** : 進行中のインジェクト変換ステータス

0 : (ソフトウェアあるいはトリガによって) 発行されたインジェクトチャンネルグループの変換のリクエストはありません。
1 : JSWSTART に“1”が書き込まれているか、トリガを検出したため、インジェクトチャンネルグループの変換が進行中またはインジェクト変換のリクエストが保留中です。
JCIP=1 の場合、インジェクト変換開始のリクエストは無視されます。

ビット 12:5 予約済みであり、リセット値に保持する必要があります。



ビット 4 **AWDF** : アナログウォッチドッグ

0 : アナログウォッチドッグイベントは発生していません。

1 : アナログウォッチドッグブロックが、DFSDM_FLTxAWLTR レジスタまたは DFSDM_FLTxAWHTR レジスタでプログラムされた値に跨る電圧を検出しました。

このビットは、ハードウェアによってセットされます。DFSDM_FLTxAWSR レジスタのすべてのソースフラグビット AWHTF[3:0] および AWLTF[3:0] をクリアする (DFSDM_FLTxAWCFR レジスタのクリアビットに“1”を書き込む) ことで、ソフトウェアによってクリアされます。

ビット 3 **ROVRF** : レギュラ変換オーバーランフラグ

0 : レギュラ変換オーバーランは発生していません。

1 : レギュラ変換オーバーランが発生し、REOCF が“1”のときにレギュラ変換が終了したことを示します。RDATAR は、オーバーランの影響を受けません。

このビットは、ハードウェアによってセットされます。DFSDM_FLTxICR レジスタの CLRROVRF ビットで、ソフトウェアによってクリアできます。

ビット 2 **JOVRF** : インジェクト変換オーバーランフラグ

0 : インジェクト変換オーバーランは発生していません。

1 : インジェクト変換オーバーランが発生し、JEOCF が“1”のときにインジェクト変換が終了したことを示します。JDATAR は、オーバーランの影響を受けません。

このビットは、ハードウェアによってセットされます。DFSDM_FLTxICR レジスタの CLRJOVRF ビットで、ソフトウェアによってクリアできます。

ビット 1 **REOCF** : レギュラ変換終了フラグ

0 : レギュラ変換が完了していません。

1 : レギュラ変換が完了し、データが読み出されます。

このビットは、ハードウェアによってセットされます。ソフトウェアまたは DMA が DFSDM_FLTxRDATAR を読み出す際にクリアされます。

ビット 0 **JEOCF** : インジェクト変換終了フラグ

0 : インジェクト変換が完了していません。

1 : インジェクト変換が完了し、データが読み出されます。

このビットは、ハードウェアによってセットされます。ソフトウェアまたは DMA が DFSDM_FLTxJDATAR を読み出す際にクリアされます。

注 : 各フラグビットで、DFSDM_FLTxCR2 の対応するビットをセットすることによって、割り込みを有効にできます。割り込みが呼び出された場合、割り込みサービスルーチンが終了する前にフラグがクリアされる必要があります。

DFEN=0 の場合、DFSDM_FLTxISR のすべてのビットは自動的にリセットされます。

14.7.4 DFSDM 割り込みフラグクリアレジスタ (DFSDM_FLTxCr)

アドレスオフセット : $0x10C + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CLRSCDF[3:0]				Res.	Res.	Res.	Res.	CLRCKABF[3:0]			
				rc_w1	rc_w1	rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRR OVRF	CLR J OVRF	Res.	Res.
												rc_w1	rc_w1		

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **CLRSCDF[3:0]** : 短絡検出回路フラグのクリア

CLRSCDF[y]=0 : “0”を書き込んでも、ビットの値は変化しません。

CLRSCDF[y]=1 : 位置 y に“1”を書き込むと DFSDM_FLTxCr レジスタの対応する SCDF[y] ビットをクリアします。

注 : **CLRSCDF[3:0]** は、**DFSDM_FLT0ICr** レジスタ (フィルタ x=0) にのみ存在します。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **CLRCKABF[3:0]** : クロック無フラグのクリア

CLRCKABF[y]=0 : “0”を書き込んでも、ビットの値は変化しません。

CLRCKABF[y]=1 : 位置 y に“1”を書き込むと DFSDM_FLTxCr レジスタの対応する CKABF[y] ビットをクリアします。トランシーバがまだ同期されていないときは、クロック無フラグがセットされ、CLRCKABF[y] ビットでクリアできません。

注 : **CLRCKABF[3:0]** は、**DFSDM_FLT0ICr** レジスタ (フィルタ x=0) にのみ存在します。

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CLRRVRF** : レギュラ変換オーバーランフラグのクリア

0 : “0”を書き込んでも、ビットの値は変化しません。

1 : “1”を書き込むと DFSDM_FLTxCr レジスタの RVRF ビットをクリアします。

ビット 2 **CLRJVRF** : インジェクト変換オーバーランフラグのクリア

0 : “0”を書き込んでも、ビットの値は変化しません。

1 : “1”を書き込むと DFSDM_FLTxCr レジスタの JVRF ビットをクリアします。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

注 : **DFSDM_FLTxCr** のビットは常に 0 として読み出されます。

14.7.5 DFSDM インジェクトチャネルグループ選択レジスタ (DFSDM_FLTxJCHGR)

アドレスオフセット : $0x110 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0001

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

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **JCHG[3:0]** : インジェクトチャネルグループ選択

JCHG[y]=0 : チャネル y はインジェクトグループの一部ではありません。

JCHG[y]=1 : チャネル y はインジェクトグループの一部です。

JSCAN=1 の場合、選択された各チャネルが次々と変換されます。最下位チャネル（選択されている場合チャネル 0）が最初に変換され、選択された最上位のチャネルでシーケンスが終了します。

JSCAN=0 の場合、選択されたチャネルから 1 つだけを変換し、チャネル選択が次のチャネルに移ります。

JSCAN=0 の場合、JCHG に書き込むと、チャネル選択が選択された最下位のチャネルにリセットされます。

最低 1 チャネルは、必ずインジェクトグループに選択されている必要があります。すべての JCHG ビットを 0 にする書き込みは無視されます。

14.7.6 DFSDM フィルタ制御レジスタ (DFSDM_FLTFCR)

アドレスオフセット : $0x114 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : $0x0000\ 0000$

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

ビット 31:29 **FORD[2:0]** : Sinc フィルタ段数

- 0 : FastSinc フィルタタイプ
- 1 : Sinc¹ フィルタタイプ
- 2 : Sinc² フィルタタイプ
- 3 : Sinc³ フィルタタイプ
- 4 : Sinc⁴ フィルタタイプ
- 5 : Sinc⁵ フィルタタイプ
- 6~7 : 予約済み

Sinc^x フィルタタイプの転送機能 :
$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^x$$

FastSinc フィルタタイプの転送機能 :
$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}} \right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

ビット 28:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:16 **FOSR[9:0]** : Sinc フィルタのオーバーサンプリング比 (デシメーション率)

0~1023 : Sinc タイプフィルタの長さを 1~1024 の範囲で定義します ($F_{OSR} = FOSR[9:0] + 1$)。この数字は、フィルタからの出力データレートのデシメーション率でもあります。

このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

注 : *FOSR = 0 の場合、フィルタに効果はありません (フィルタバイパス)。*

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **IOSR[7:0]** : 積分器のオーバーサンプリング比 (平均長)

0~255 : 1~256 の積分器の長さ (IOSR + 1) 積分器からの 1 つの出力データサンプルに合計する Sinc フィルタからのサンプル数を定義します。積分器からの出力データレートは、この数字 (追加のデータデシメーション率) によって減少します。

このビットは、DFEN=0 (DFSDM_FLTxCR1) の場合にのみ修正できます。

注 : *IOSR = 0 の場合、積分器に効果はありません (積分器バイパス)。*

14.7.7 インジェクトグループの DFSDM データレジスタ
 (DFSDM_FLTxJDATAR)

アドレスオフセット：0x118 + 0x80 * x、x = 0~1
リセット値：0x0000 0000

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

- ビット 31:8 **JDATA[23:0]**：インジェクトグループ変換データ
インジェクトグループチャンネルの各変換が終了すると、結果データがこのフィールドに保存されます。データは、JEOCF=1 のときに有効です。このレジスタを読み出すと、対応する JEOCF がクリアされます。
- ビット 7:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1:0 **JDATACH[1:0]**：最近変換したインジェクトチャンネル
インジェクトグループチャンネルの各変換が終了すると、どのチャンネルを変換したかを示すために JDATACH[1:0] が更新されます。そのため、JDATA[23:0] は JDATACH[1:0] によって示されたチャンネルに対応するデータを保持します。

注： **DMA を使用して、このレジスタからデータを読み出せます。ハーフワードアクセスを使用して、変換データのMSBのみを読み出せます。**
このレジスタを読み出すと、DFSDM_FLTxISR の JEOCF がクリアされます。そのため、DMA が有効でこのレジスタからデータを読み出している場合、ファームウェアがこのレジスタを読み出さないようにしてください。

14.7.8 レギュラチャネルの DFSDM データレジスタ (DFSDM_FLTxRDATAR)

アドレスオフセット : $0x11C + 0x80 * x$, $x = 0 \sim 1$

リセット値 : $0x0000\ 0000$

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

ビット 31:8 **RDATA[23:0]** : レギュラチャネル変換データ

各レギュラ変換が終了すると、データがこのレジスタに保存されます。データは、REOCF=1 のときに有効です。このレジスタを読み出すと、対応する REOCF がクリアされます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **RPEND** : レギュラチャネル保留データ

RDATA[23:0] のレギュラデータが、変換中のインジェクトチャネルトリガによって遅延しました。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **RDATACH[1:0]** : 最近変換したレギュラチャネル

各レギュラチャネルが終了すると、どのチャネルを変換したかを示すために RDATACH[1:0] が更新されます (DFSDM_FLTxCR1 レジスタのレギュラチャネル選択 RCH[1:0] がレギュラ変換中に更新できるため)。そのため、RDATA[23:0] は RDATACH[1:0] によって示されたチャネルに対応するデータを保持します。

注 : **ハーフワードアクセスを使用して、変換データのMSBのみを読み出せます。**
 このレジスタを読み出すと、DFSDM_FLTxISR のREOCF がクリアされます。

14.7.9 DFSDM アナログウォッチドッグ高閾値レジスタ (DFSDM_FLTxAWHTR)

アドレスオフセット : $0x120 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : $0x0000\ 0000$

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

ビット 31:8 **AWHT[23:0]** : アナログウォッチドッグ高閾値

これらのビットは、アナログウォッチドッグの高閾値を定義するために、ソフトウェアによって書き込まれます。

注： チャンネルトランシーバが監視する場合（AWFSEL=1）、（アナログウォッチドッグフィルタからのデータは最大 16 ビットの分解能となるため）上位の 16 ビット（AWHT[23:8]）が、アナログウォッチドッグフィルタ出力と比較される 16 ビット閾値を定義します。AWHT[7:0] ビットは、この場合の比較では取得されません。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **BKAWH[3:0]** : アナログウォッチドッグ高閾値イベントへのブレイク信号割り当て

BKAWH[i] = 0 : ブレイク i 信号がアナログウォッチドッグ高閾値イベントに割り当てられません。

BKAWH[i] = 1 : ブレイク i 信号がアナログウォッチドッグ高閾値イベントに割り当てられます。

14.7.10 DFSDM アナログウォッチドッグ低閾値レジスタ（DFSDM_FLTxAWLTR）

アドレスオフセット : $0x124 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

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

ビット 31:8 **AWLT[23:0]** : アナログウォッチドッグ低閾値

これらのビットは、アナログウォッチドッグの低閾値を定義するために、ソフトウェアによって書き込まれます。

注： チャンネルトランシーバが監視する場合（AWFSEL=1）、（アナログウォッチドッグフィルタからのデータは最大 16 ビットの分解能となるため）上位の 16 ビット（AWLT[23:8]）のみが、アナログウォッチドッグフィルタ出力と比較される 16 ビット閾値を定義します。AWLT[7:0] ビットは、この場合の比較では取得されません。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **BKAWL[3:0]** : アナログウォッチドッグ低閾値イベントへのブレイク信号割り当て

BKAWL[i] = 0 : ブレイク i 信号がアナログウォッチドッグ低閾値イベントに割り当てられません。

BKAWL[i] = 1 : ブレイク i 信号がアナログウォッチドッグ低閾値イベントに割り当てられます。

14.7.11 DFSDM アナログウォッチドックステータスレジスタ (DFSDM_FLTxAWSR)

アドレスオフセット : $0x128 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **AWHTF[3:0]** : アナログウォッチドッグ高閾値フラグ

AWHTF[y]=1 は、チャンネル y での高閾値エラーを示します。ハードウェアによってセットされます。
DFSDM_FLTxAWCFR レジスタの対応する CLRAWHTF[y] ビットで、ソフトウェアによってクリアできます。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **AWLTF[3:0]** : アナログウォッチドッグ低閾値フラグ

AWLTF[y]=1 は、チャンネル y での低閾値エラーを示します。ハードウェアによってセットされます。
DFSDM_FLTxAWCFR レジスタの対応する CLRAWLTF[y] ビットで、ソフトウェアによってクリアできます。

注 : *DFEN=0 の場合、DFSDM_FLTxAWSR のすべてのビットは自動的にリセットされます。*

14.7.12 DFSDM アナログウォッチドッグクリアフラグレジスタ (DFSDM_FLTxAWCFR)

アドレスオフセット : $0x12C + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CLRAWHTF[3:0]				Res.	Res.	Res.	Res.	CLRAWLTF[3:0]			
				rc_w1	rc_w1	rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **CLRAWHTF[3:0]** : アナログウォッチドッグ高閾値フラグのクリア

CLRAWHTF[y]=0 : “0”を書き込んでも、ビットの値は変化しません。
CLRAWHTF[y]=1 : 位置 y に“1”を書き込むと DFSDM_FLTxAWSR レジスタの対応する AWHTF[y] ビットをクリアします。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **CLRAWLTF[3:0]** : アナログウォッチドッグ低閾値フラグのクリア

CLRAWLTF[y]=0 : “0”を書き込んでも、ビットの値は変化しません。
CLRAWLTF[y]=1 : 位置 y に“1”を書き込むと DFSDM_FLTxAWSR レジスタの対応する AWLTF[y] ビットをクリアします。



14.7.13 DFSDM 極値検出回路最大値レジスタ (DFSDM_FLTxEEXMAX)

アドレスオフセット : $0x130 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMAX[23:8]															
r1	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMAX[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	EXMAXCH[1:0]	
r0	r0	r0	r0	r0	r0	r0	r0							r	r

ビット 31:8 **EXMAX[23:0]** : 極値検出回路最大値

これらのビットは、ハードウェアによってセットされ、DFSDM_FLTxによって変換された最高値を示します。
EXMAX[23:0] ビットは、このレジスタの読み出しによって値 (0x800000) にリセットされます。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **EXMAXCH[1:0]** : 極値検出回路最大データチャンネル

これらのビットは、データが EXMAX[23:0] に保存されているチャンネルについての情報が含まれています。ビットは、このレジスタの読み出しによってクリアされます。

14.7.14 DFSDM 極値検出回路最小値レジスタ (DFSDM_FLTxEXMIN)

アドレスオフセット : $0x134 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x7FFF FF00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMIN[23:8]															
r0	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMIN[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	EXMINCH[1:0]	
r1	r1	r1	r1	r1	r1	r1	r1							r	r

ビット 31:8 **EXMIN[23:0]** : 極値検出回路最小値

これらのビットは、ハードウェアによってセットされ、DFSDM_FLTxによって変換された最低値を示します。
EXMIN[23:0] ビットは、このレジスタの読み出しによって値 (0x7FFFFFFF) にリセットされます。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **EXMINCH[1:0]** : 極値検出回路最小データチャンネル

これらのビットは、データが EXMIN[23:0] に保存されているチャンネルについての情報が含まれています。ビットは、このレジスタの読み出しによってクリアされます。

14.7.15 DFSDM 変換タイマレジスタ (DFSDM_FLTxCNVTIMR)

アドレスオフセット : $0x138 + 0x80 * x$, $x = 0 \sim 1$

リセット値 : 0x0000 0000

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

ビット 31:4 **CNVCNT[27:0]** : 28 ビットタイマカウント変換時間 $t = \text{CNVCNT}[27:0] / f_{\text{DFSDMCLK}}$

タイマには、DFSDM クロック (システムクロック f_{DFSDMCLK}) からの入力クロックがあります。変換時間測定は、各変換の開始時に開始し、変換終了時に停止します (最初から最後のシリアルサンプルまでの間隔)。フィルタバイパス ($\text{FOSR}[9:0] = 0$) の場合のみ、変換時間測定が停止し、 $\text{CNVCNT}[27:0] = 0$ となります。カウント時間を次に示します。

$\text{FAST} = 0$ (または $\text{FAST} = 1$ の場合の連続モードにおける最初の変換) の場合 :

$t = [\text{FOSR} * (\text{IOSR} - 1 + \text{FORD}) + \text{FORD}] / f_{\text{CKIN}} \dots \text{Sinc}^x$ フィルタの場合

$t = [\text{FOSR} * (\text{IOSR} - 1 + 4) + 2] / f_{\text{CKIN}} \dots \text{FastSinc}$ フィルタの場合

連続モードの $\text{FAST} = 1$ (最初の変換を除く) の場合 :

$t = [\text{FOSR} * \text{IOSR}] / f_{\text{CKIN}}$

$\text{FOSR} = \text{FOSR}[9:0] + 1 = 1$ (フィルタバイパス、積分器のみ有効) の場合 :

$\text{CNVCNT} = 0$ (カウント停止、変換時間 : $t = \text{IOSR} / f_{\text{CKIN}}$)

ここで、 f_{CKIN} は、チャンネル入力クロック周波数 (特定のチャンネル CKINy ピン) または並列データ入力の場合の入力データレート (CPU/DMA 書き込みからの) です。

注 : 変換が割り込まれた場合 (例 : 選択したチャンネルの無効化/有効化)、タイマはこの割り込み時間もカウントします。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

14.8 DFSDM レジスタマップ

次の表に DFSDM レジスタの一覧を示します。

表 88. DFSDM レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	DFSDM_ CH0CFGR1	DFSDMEN	CKOUTSRC	Res	Res	Res	Res	Res	Res	CKOUTDIV[7:0]								DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res	SPICKSEL [1:0]		SITP[1:0]		
	リセット値	0	0							0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0		0	0	0	0
0x04	DFSDM_ CH0CFGR2	OFFSET[23:0]																									DTRBS[4:0]				Res	Res	Res	
	リセット値	0																									0							
0x08	DFSDM_ CH0AWSCDR	Res	Res	Res	Res	Res	Res	Res	Res	AWFORD [1:0]	Res	AWFOSR[4:0]				BKSCD[3:0]			Res.	Res.	Res.	Res.	SCDT[7:0]											
	リセット値									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	
0x0C	DFSDM_ CH0WDATR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDATA[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	DFSDM_ CH0DATINR	INDAT1[15:0]										INDAT0[15:0]																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14 - 0x1C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x20	DFSDM_ CH1CFGR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res	SPICKSEL [1:0]		SITP[1:0]		
	リセット値																	0	0	0	0				0	0	0	0		0	0	0	0	
0x24	DFSDM_ CH1CFGR2	OFFSET[23:0]																									DTRBS[4:0]				Res	Res	Res	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x28	DFSDM_ CH1AWSCDR	Res	Res	Res	Res	Res	Res	Res	Res	AWFORD[1:0]	Res	AWFOSR[4:0]				BKSCD[3:0]			Res.	Res.	Res.	Res.	SCDT[7:0]											
	リセット値									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	
0x2C	DFSDM_ CH1WDATR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDATA[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	DFSDM_ CH1DATINR	INDAT1[15:0]										INDAT0[15:0]																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x34 - 0x3C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	

表 88. DFSDM レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x40	DFSDM_ CH2CFGR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]				DATMPX[1:0]				Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]				SITP[1:0]	
	リセット値																	0	0	0	0				0	0	0	0		0	0	0	0					
0x44	DFSDM_ CH2CFGR2	OFFSET[23:0]																								DTRBS[4:0]				Res.	Res.	Res.						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
0x48	DFSDM_ CH2AWSCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]				BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]													
	リセット値									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0					
0x4C	DFSDM_ CH2WDATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]																					
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x50	DFSDM_ CH2DATINR	INDAT1[15:0]															INDAT0[15:0]																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x54 - 0x5C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.						
0x60	DFSDM_ CH3CFGR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]				SITP[1:0]			
	リセット値																	0	0	0	0				0	0	0	0		0	0	0	0					
0x64	DFSDM_ CH3CFGR2	OFFSET[23:0]																								DTRBS[4:0]				Res.	Res.	Res.						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
0x68	DFSDM_ CH3AWSCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]				BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]													
	リセット値									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0					
0x6C	DFSDM_ CH3WDATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]																					
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x70	DFSDM_ CH3DATINR	INDAT1[15:0]															INDAT0[15:0]																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x74 - 0xFC	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.						
0x100	DFSDM_ FLT0CR1	Res.	AWFSEL	FAST	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		Res.	Res.	JEXTSEL[2:0]		Res.	Res.	Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN				
	リセット値		0	0			0	0			0			0	0	0			0	0			0	0	0			0	0	0		0	0					

表 88. DFSDM レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																		
0x104	DFSDM_ FLT0CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]				Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE																		
	リセット値													0	0	0	0					0	0	0	0		0	0	0	0	0	0	0																		
0x108	DFSDM_ FLT0ISR	Res.	Res.	Res.	Res.	SCDF[3:0]				Res.	Res.	Res.	Res.	CKABF[3:0]				Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF																		
	リセット値					0	0	0	0					1	1	1	1		0	0								0	0	0	0	0	0																		
0x10C	DFSDM_ FLT0ICR	Res.	Res.	Res.	Res.	CLRSCDF [3:0]				Res.	Res.	Res.	Res.	CLRCKABF [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF																					
	リセット値					0	0	0	0					0	0	0	0												0	0																					
0x110	DFSDM_ FLT0JCHGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[3:0]																					
	リセット値																												0	0	0	1																			
0x114	DFSDM_ FLT0FCR	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]																									
	リセット値	0	0	0				0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0																	
0x118	DFSDM_ FLT0JDATAR	JDATA[23:0]																						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATACH [1:0]						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0																		
0x11C	DFSDM_ FLT0RDATAR	RDATA[23:0]																						Res.	Res.	Res.	Res.	RPEND	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATACH[1:0]						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0				0	0																		
0x120	DFSDM_ FLT0AWHTR	AWHT[23:0]																						Res.	Res.	Res.	Res.	BKAWH[3:0]																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0																					
0x124	DFSDM_ FLT0AWLTR	AWLT[23:0]																						Res.	Res.	Res.	Res.	BKAWL[3:0]																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0																				
0x128	DFSDM_ FLT0AWSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]					AWLTF[7:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																		
0x12C	DFSDM_ FLT0AWCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
	リセット値																				0	0	0	0				0	0	0	0	0	0																		
0x130	DFSDM_ FLT0EXMAX	EXMAX[23:0]																						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	

表 88. DFSDM レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
0x134	DFSDM_ FLT0EXMIN	EXMIN[23:0]																									Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		EXMINCH[1:0]
	リセット値	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							0	0																				
0x138	DFSDM_ FLT0CNVTIMR	CNVCNT[27:0]																									Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.		Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																							
0x13C - 0x17C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																			
0x180	DFSDM_ FLT1CR1	Res.	Res.	AWFSEL	FAST	Res.	Res.	RCH[1:0]		Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		Res.	Res.	JEXTSEL[2:0]		Res.	Res.	Res.	Res.	Res.	Res.	JDMAEN	JSCAN	JSYNC	JSW START	DFEN																		
	リセット値		0	0				0	0			0		0	0	0			0	0			0	0	0			0	0	0	0	0	0	0																			
0x184	DFSDM_ FLT1CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]			Res.	Res.	Res.	Res.	EXCH[3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																			
	リセット値													0	0	0	0					0	0	0	0					0	0	0	0	0																			
0x188	DFSDM_ FLT1ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																			
	リセット値																		0	0										0	0	0	0	0																			
0x18C	DFSDM_ FLT1ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																			
	リセット値																														0	0																					
0x190	DFSDM_ FLT1JCHGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[3:0]																						
	リセット値																													0	0	0	1																				
0x194	DFSDM_ FLT1FCR	FORD[2:0]		Res.	Res.	Res.	FOSR[9:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]																											
	リセット値	0	0	0			0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	0																			
0x198	DFSDM_ FLT1JDATAR	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATAH[1:0]		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0																				
0x19C	DFSDM_ FLT1RDATAR	RDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDATAH[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	Res.	0			0	0																		
0x1A0	DFSDM_ FLT1AWHTR	AWHT[23:0]																								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKAWH[3:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	Res.	Res.	Res.	0	0	0	0																			

表 88. DFSDM レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1A4	DFSDM_ FLT1AWLTR	AWLT[23:0]																								Res	Res	Res	Res	BKAWL[3:0]			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x1A8	DFSDM_ FLT1AWSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AWHTF[7:0]							AWLTF[7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1AC	DFSDM_ FLT1AWCFR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRAWHTF [3:0]				CLRAWLTF [3:0]								
	リセット値																					0	0	0	0					0	0	0	0
0x1B0	DFSDM_ FLT1EXMAX	EXMAX[23:0]																								Res	Res	Res	Res	EXMAXCH[1:0]			
	リセット値	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0
0x1B4	DFSDM_ FLT1EXMIN	EXMIN[23:0]																								Res	Res	Res	Res	EXMINCH[1:0]			
	リセット値	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							0	0
0x1B8	DFSDM_ FLT1CNVTIMR	CNVCNT[27:0]																												Res	Res	Res	Res
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			Res
0x1BC - 0x2FC	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

レジスタ境界アドレスについては、[セクション 2.2.2：メモリマップとレジスタ境界アドレス](#) を参照してください。

並行して、アナログシードと専用の RNG_CLK クロックがモニターされます。シードで異常なシーケンスが発生したり、RNG_CLK クロックの周波数が低すぎる場合、ステータスビット (RNG_SR レジスタ内) に示されます。エラーが検出された場合、割り込みを生成することができます。

15.3.1 動作

RNG を作動させるには、次のステップに従います。

1. 必要に応じて割り込みを有効にします (RNG_CR レジスタの IE ビットをセットします)。乱数の準備ができたとき、またはエラーが発生したとき、割り込みが生成されます。
2. RNG_CR レジスタの RNGEN ビットをセットして、乱数発生を有効にします。これによってアナログ部分、RNG_LFSR およびエラー検出回路が作動します。
3. 割り込みのたびに、エラーが発生していないこと (RNG_SR レジスタの SEIS ビットおよび CEIS ビットが“0”)、乱数の準備ができていないこと (RNG_SR レジスタの DRDY ビットが“1”)を確認します。RNG_DR レジスタの内容が読み出せるようになります。

FIPS PUB 140-2 の要件に従い、RNGEN ビット設定後最初に生成された乱数は使用すべきではありませんが、次に生成される乱数との比較のため保存されます。後に生成される各乱数は、それぞれ前に生成された値と比較する必要があります。比較された 2 つの値が同一の場合、試験は不合格となります (連続乱数発生器試験)。

15.3.2 エラー管理

CEIS ビットが“1”として読み出された場合 (クロックエラー)

クロックエラーの場合、RNG_CLK クロックが正しくないため、RNG はそれ以降、乱数を発生することはできません。RNG クロックが供給されるようにクロックコントローラが正しく設定されていることを確認し、CEIS ビットをクリアします。CEIS ビットが“0”になると、RNG が動作できるようになります。クロックエラーはその前に発生した乱数には影響しないため、RNG_DR レジスタの内容は使用できます。

SEIS ビットが“1”として読み出された場合 (シードエラー)

シードエラーの場合、SEIS ビットが“1”の間は乱数発生が中断します。RNG_DR レジスタにおける値が利用可能な場合であっても、エントロピーが十分でない可能性があるため使用することはできません。

ここでは、SEIS ビットをクリアした後、RNGEN ビットをクリアおよびセットし、RNG を再初期化および再起動します。

15.4 RNG レジスタ

RNG は、制御レジスタ、データレジスタ、ステータスレジスタと連動します。これらのレジスタは、ワード（32 ビット）単位でアクセスする必要があります。

15.4.1 RNG 制御レジスタ (RNG_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:4 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **IE** : 割り込みイネーブル

- 0 : RNG 割り込みは無効です。
- 1 : RNG 割り込みは有効です。RNG_SR レジスタで DRDY = 1、SEIS = 1 または CEIS = 1 になると、割り込みは直ちに保留となります。

ビット 2 **RNGEN** : 乱数発生器イネーブル

- 0 : 乱数発生器は無効です。
- 1 : 乱数発生器は有効です。

ビット 1:0 予約済みであり、リセット値のままにしておかなければなりません。

15.4.2 RNG ステータスレジスタ (RNG_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

ビット 31:7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6 **SEIS** : シードエラー割り込みステータス

このビットは SECS と同時にセットされ、“0”を書き込むとクリアされます。

0 : 障害シーケンスは検出されませんでした。

1 : 次の障害シーケンスのうち 1 つが検出されました。

– 64 個以上のビットが連続して同じ値 (0 または 1)

– 0 と 1 の繰り返し (0101010101...01) が 32 回以上連続

RNG_CR レジスタで IE = 1 である場合、割り込みが保留となります。

ビット 5 **CEIS** : クロックエラー割り込みステータス

このビットは CECS と同時にセットされ、“0”を書き込むとクリアされます。

0 : RNG_CLK クロックが正しく検出されました。

1 : RNG_CLK が正しく検出されませんでした ($f_{\text{RNG_CLK}} < f_{\text{HCLK}}/16$)。

RNG_CR レジスタで IE = 1 である場合、割り込みが保留となります。

ビット 4:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 **SECS** : シードエラーの現在のステータス

0 : 障害のあるシーケンスは現在のところ検出されていません。SEIS ビットがセットされている場合、これは障害のあるシーケンスが検出されたが、回復していることを意味します。

1 : 次の障害シーケンスのうち 1 つが検出されました。

– 64 個以上のビットが連続して同じ値 (0 または 1)

– 0 と 1 の繰り返し (0101010101...01) が 32 回以上連続

ビット 1 **CECS** : クロックエラーの現在のステータス

0 : RNG_CLK クロックが正しく検出されました。CEIS ビットがセットされている場合、これはクロックエラーが検出されたが、回復していることを意味します。

1 : RNG_CLK が正しく検出されませんでした ($f_{\text{RNG_CLK}} < f_{\text{HCLK}}/16$)。

ビット 0 **DRDY** : データレディ

0 : RNG_DR レジスタがまだ有効でなく、乱数データは利用できません。

1 : RNG_DR レジスタに有効な乱数データが入っています。

注: RNG_CR レジスタで IE = 1 である場合、割り込みが保留となります。

RNG_DR レジスタがいったん読み出されると、新しい有効な値が計算されるまでこのビットは 0 に戻ります。

15.4.3 RNG データレジスタ (RNG_DR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

RNG_DR レジスタは読み出し専用レジスタであり、読み出されると 32 ビットの乱数値を返します。いったん読み出されると、最長で RNG_CLK クロック 40 周期後に、このレジスタは新しい乱数値を出力します。ソフトウェアは、RNDATA 値を読み出す前に DRDY ビットがセットされていることを確認する必要があります。

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

ビット 31:0 **RNDATA** : 乱数データ
32 ビットの乱数データです。

15.4.4 RNG レジスタマップ

表 89 に、RNG レジスタマップとリセット値を示します。

表 89. RNG レジスタマップとリセット値

オフ セット	レジスタ名 リセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RNG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IE	RNGEN	Res.	Res.
	リセット値																													0	0			
0x04	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	Res.	SECS	CECS	DRDY
	リセット値																									0	0				0	0	0	
0x08	RNG_DR	RNDATA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

16 高機能制御タイマ (TIM1 および TIM8)

16.1 TIM1 および TIM8 の概要

高機能制御タイマ (TIM1 および TIM8) は、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

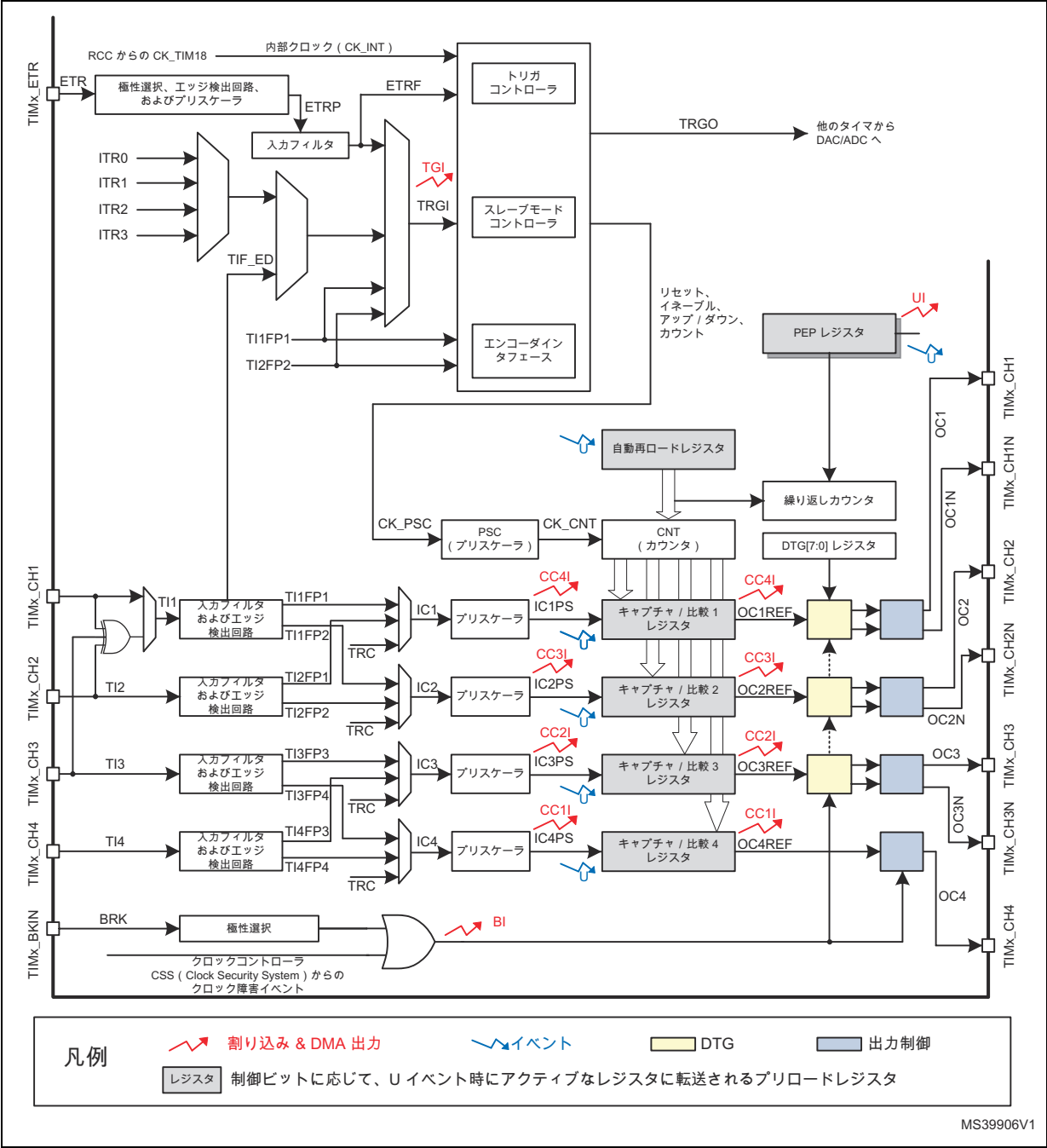
高機能制御タイマ (TIM1 および TIM8) と汎用タイマ (TIMx) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 16.3.20](#) に示すように、相互に同期させることができます。

16.2 TIM1 および TIM8 の主な機能

TIM1 および TIM8 タイマの主な機能は、次のとおりです。

- 16 ビットのアップカウンタ、ダウンカウンタ、アップ/ダウン自動再ロードカウンタ。
- 16 ビットのプログラム可能なプリスケラ。カウンタクロック周波数を 1 から 65536 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 4 つの独立チャネル。
 - － 入力キャプチャ
 - － 出力比較
 - － PWM 生成 (エッジアラインモードとセンターアラインモード)
 - － ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレーク入力。
- 以下のイベント時の割り込み/DMA 生成。
 - － 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - － トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - － 入力キャプチャ
 - － 出力比較
 - － ブレーク入力
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 76. 高機能制御タイマのブロック図



16.3 TIM1 および TIM8 の機能詳細

16.3.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 77](#) と [図 78](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 77. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

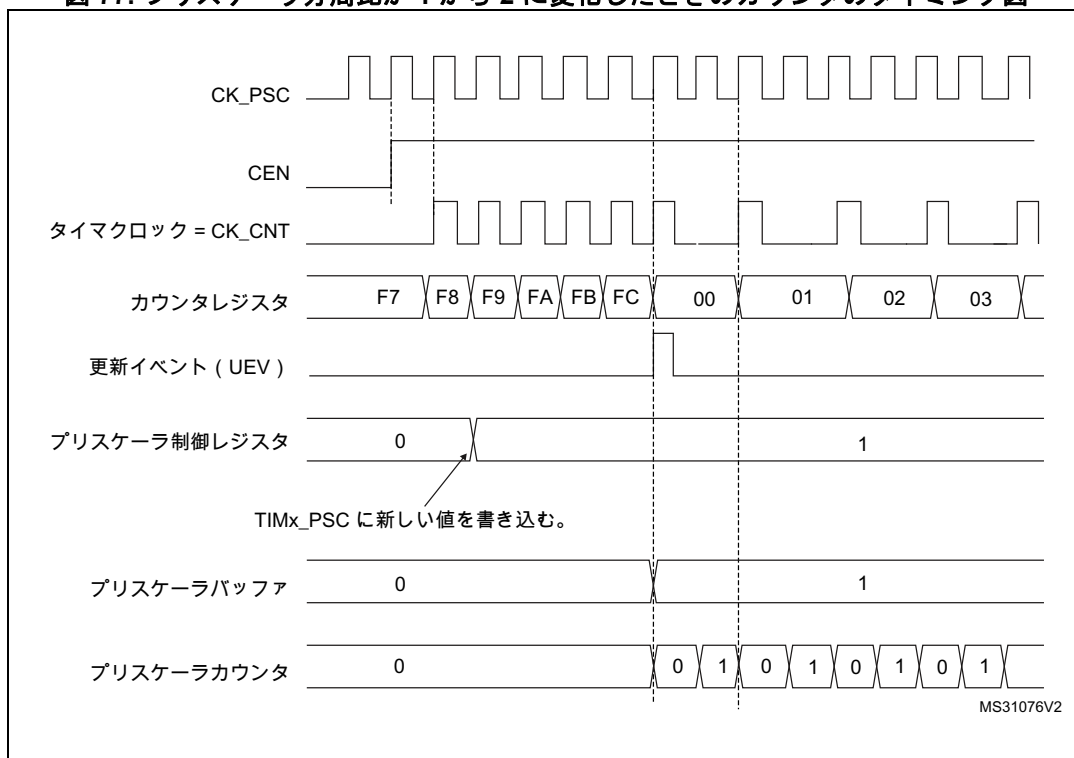
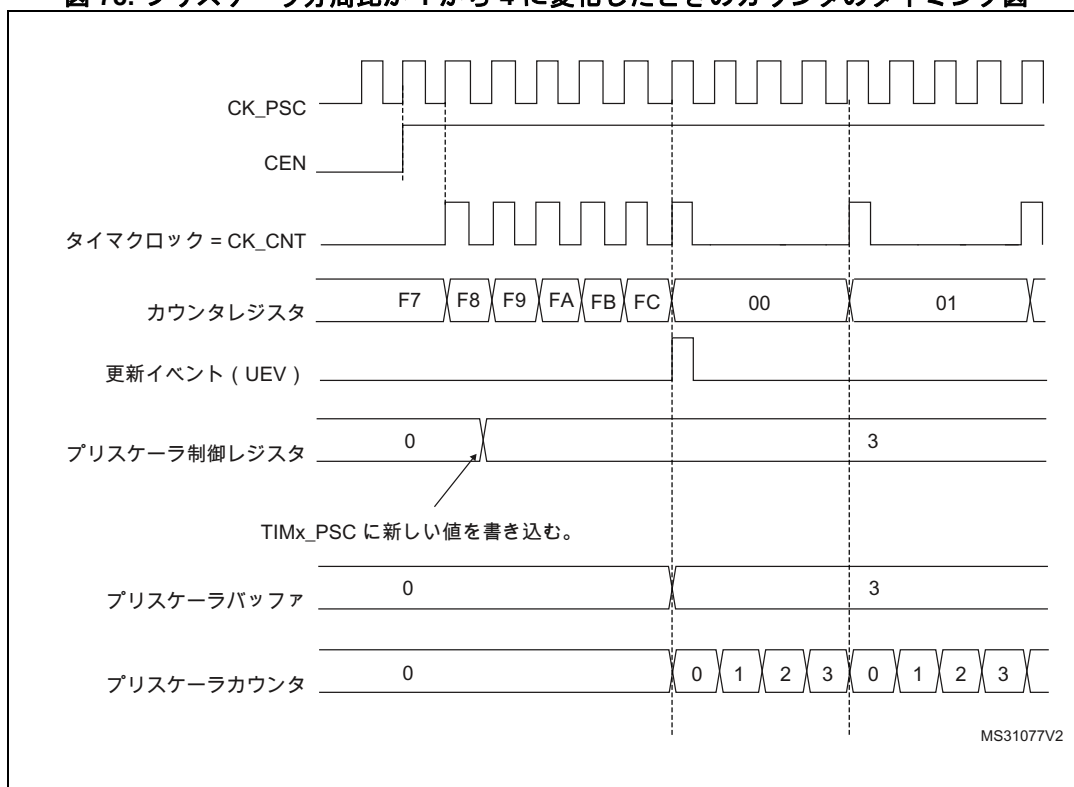


図 78. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



16.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタに設定されている回数プラス 1 (TIMx_RCR+1) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 79. 内部クロック分周比が 1 の場合のカウンタのタイミング図

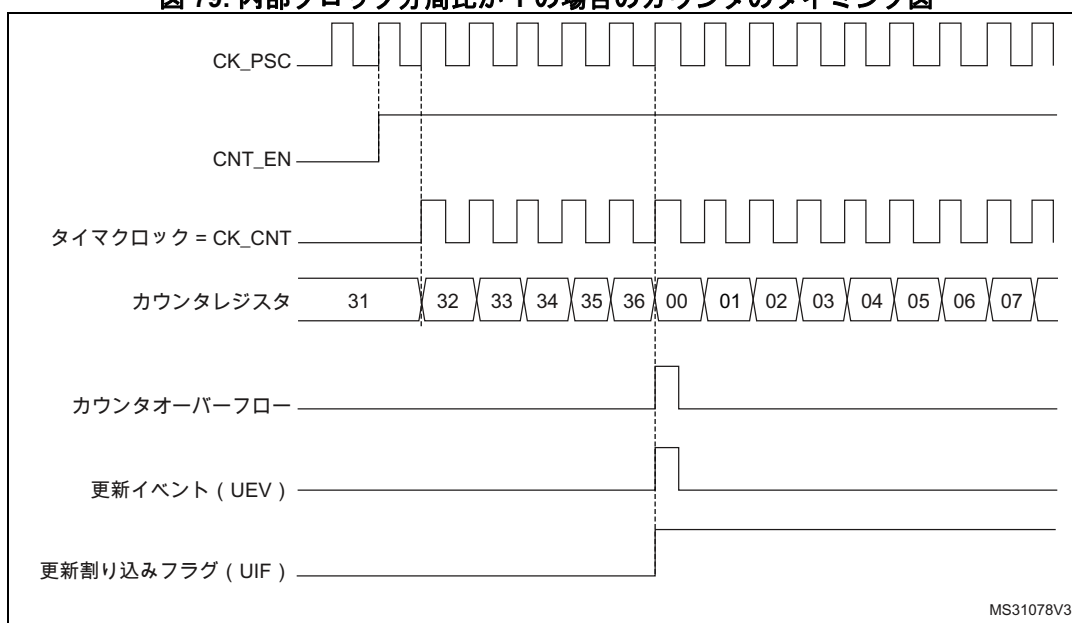


図 80. 内部クロック分周比が 2 の場合のカウンタのタイミング図

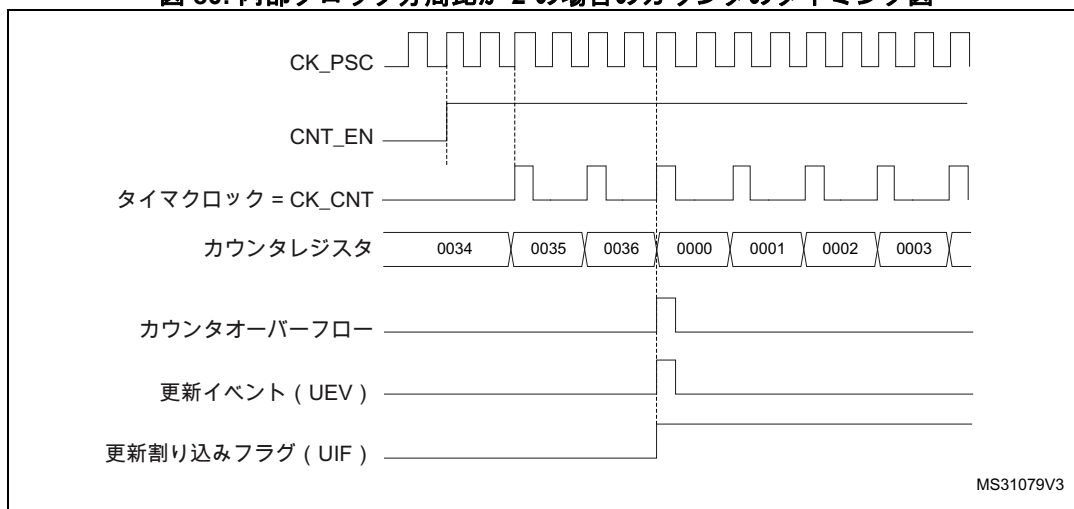


図 81. 内部クロック分周比が 4 の場合のカウンタのタイミング図

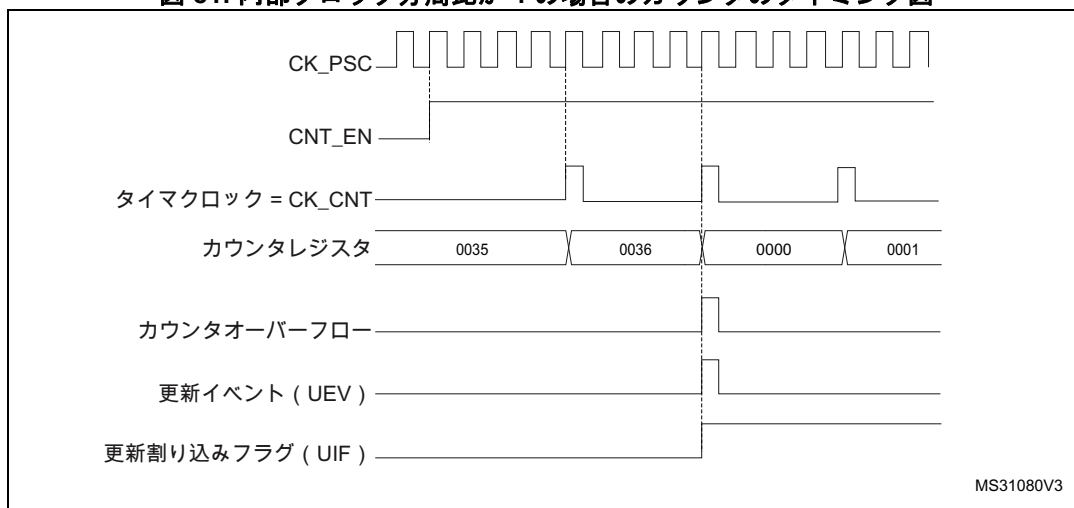


図 82. 内部クロック分周比が N の場合のカウンタのタイミング図

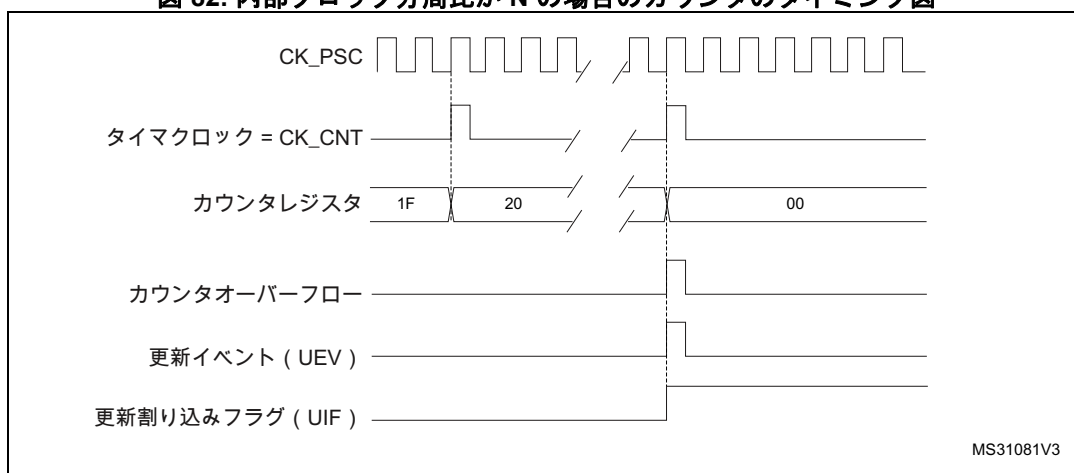


図 83. ARPE=0の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR はプリロードされない)

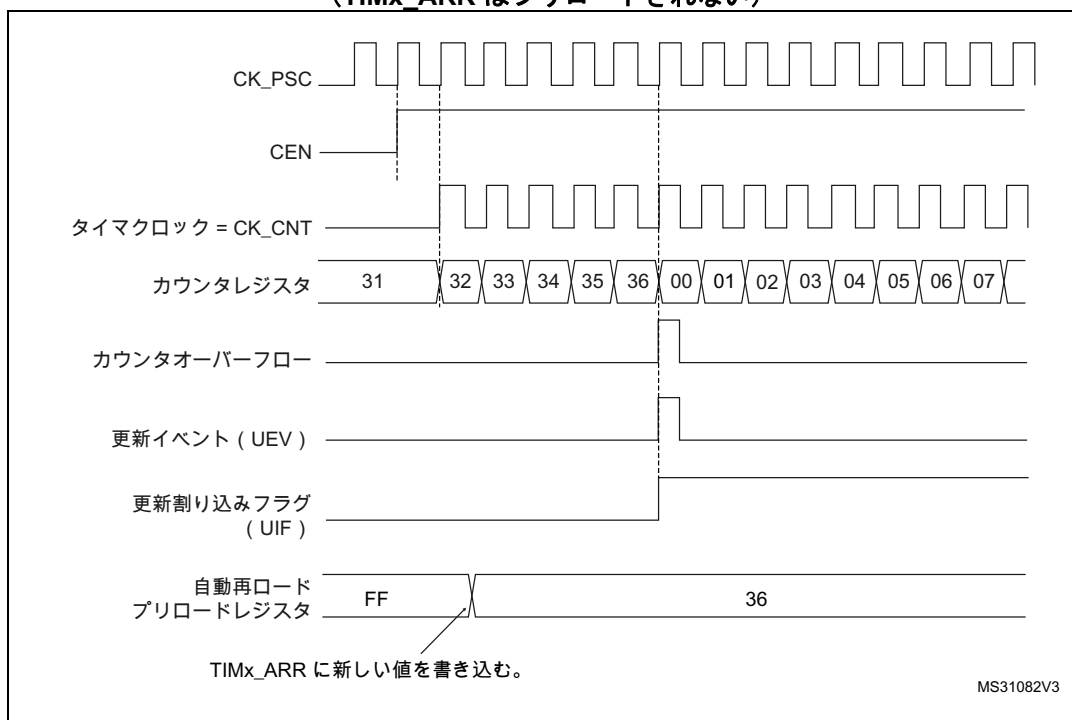
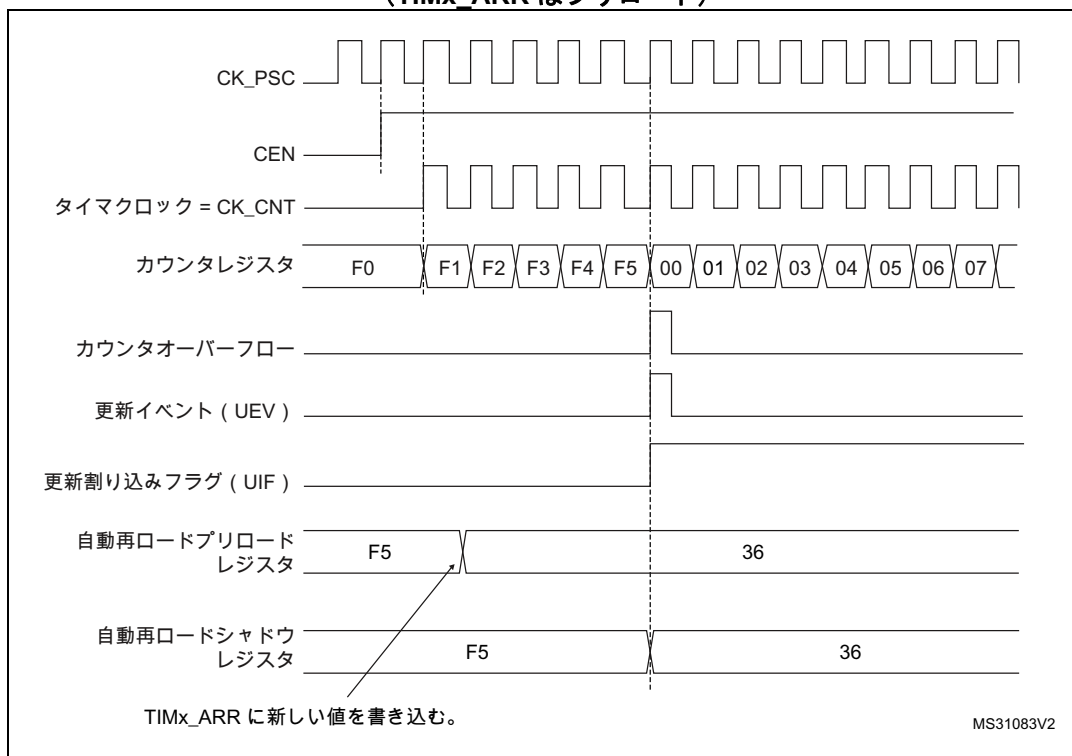


図 84. ARPE=1の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR はプリロード)



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタに設定されている回数プラス 1 (TIMx_RCR+1) までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタが再ロードされる前に自動再ロードが更新されるので、次の周期は予想通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 85. 内部クロック分周比が 1 の場合のカウンタのタイミング図

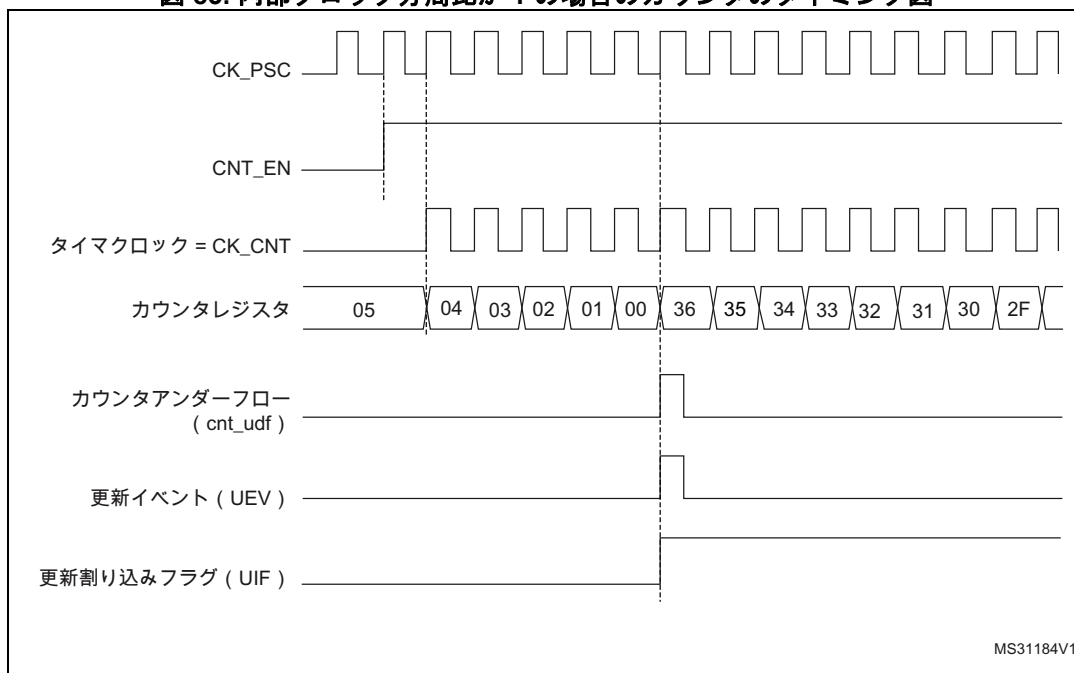


図 86. 内部クロック分周比が 2 の場合のカウンタのタイミング図

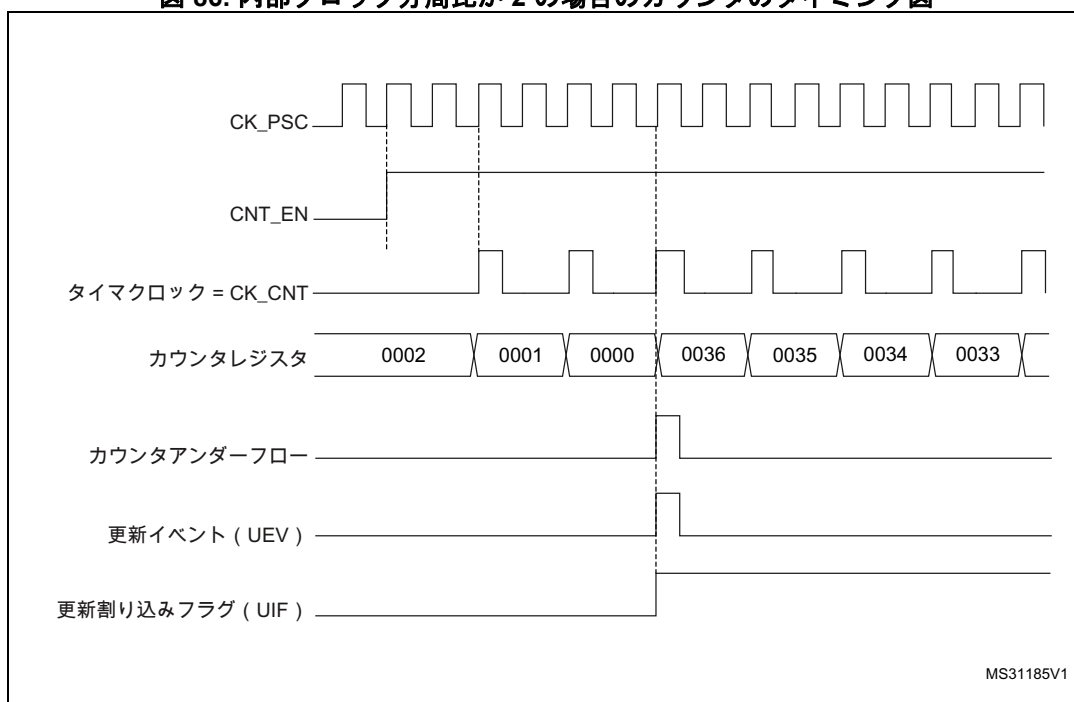


図 87. 内部クロック分周比が 4 の場合のカウンタのタイミング図

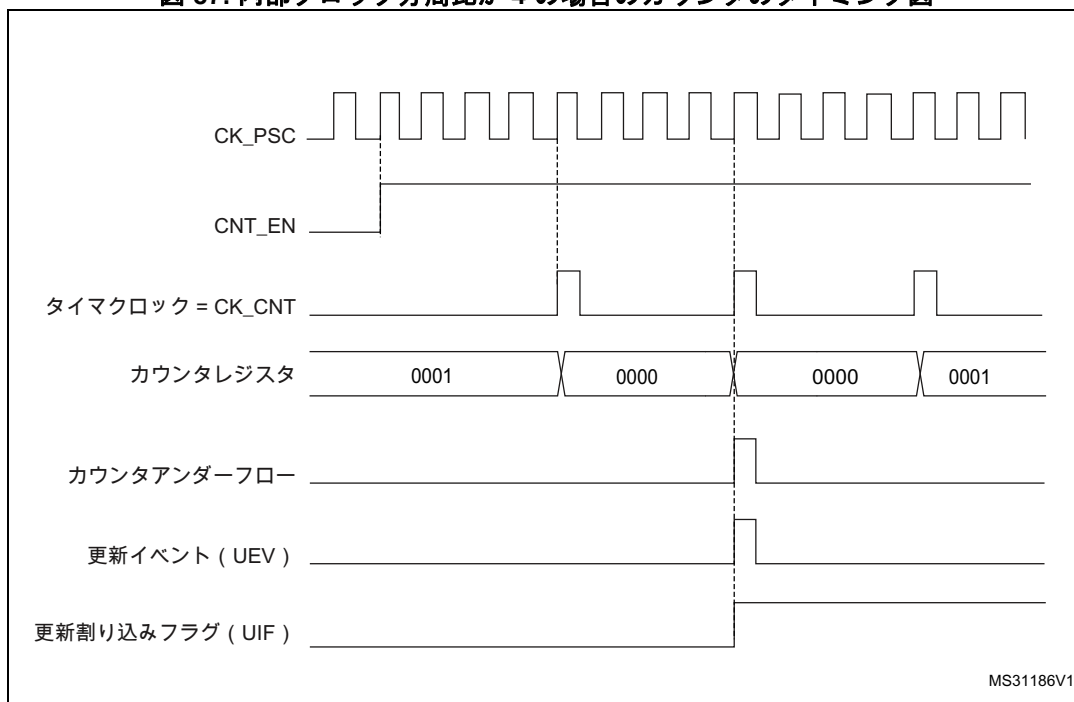


図 88. 内部クロック分周比が N の場合のカウンタのタイミング図

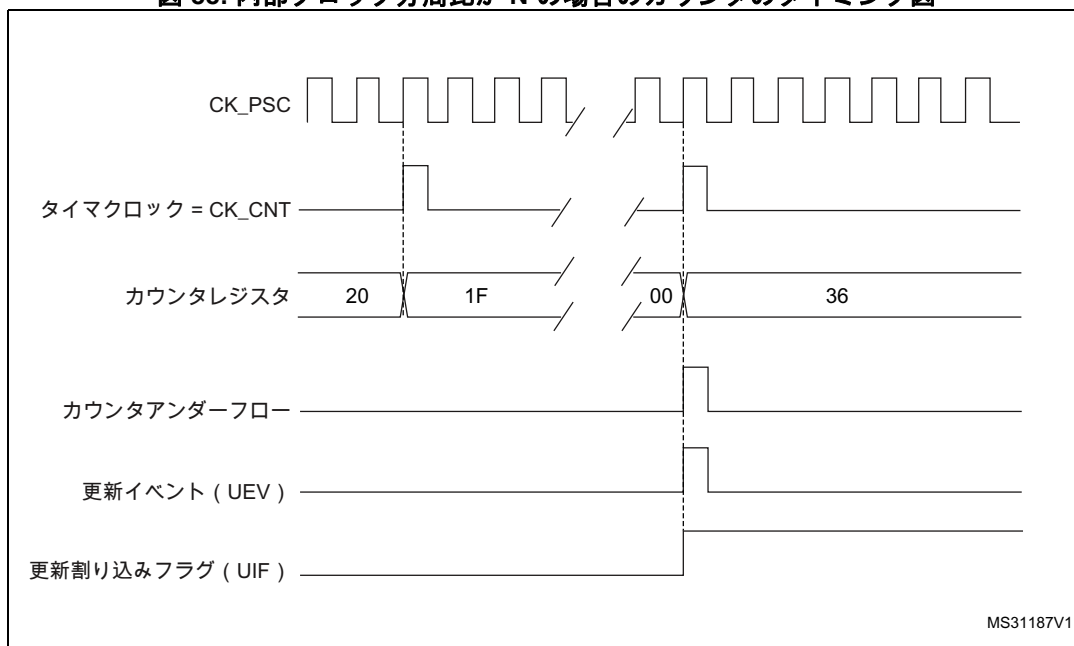
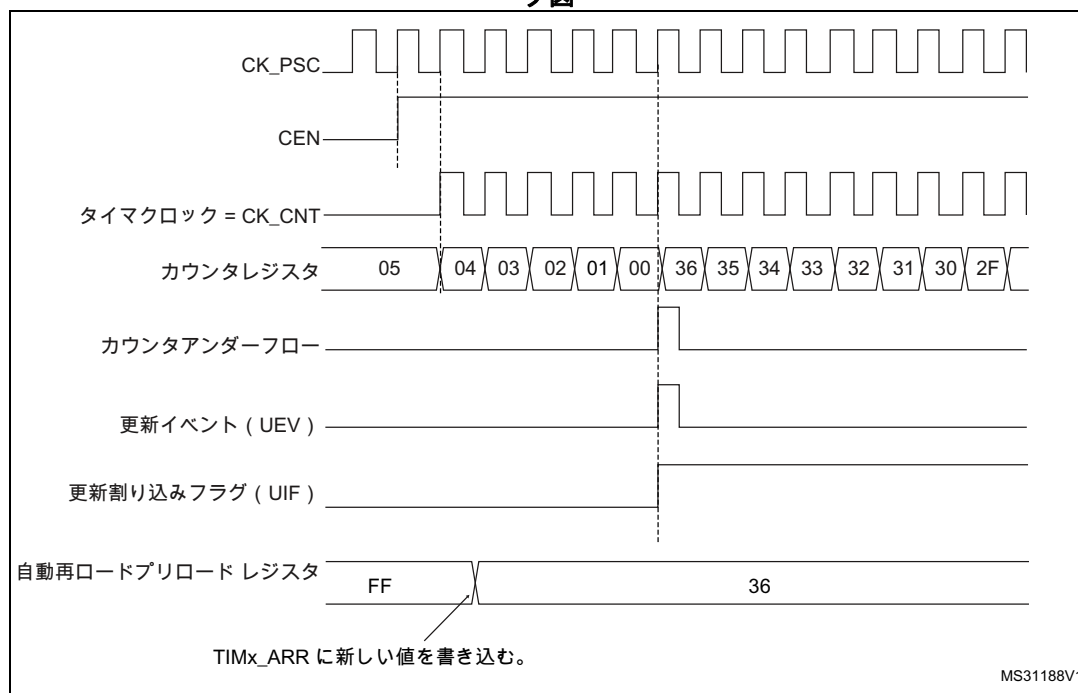


図 89. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図



センタアラインモード (アップ/ダウンカウント)

センタアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センタアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センタアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センタアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センタアラインモード 3、CMS=11) にセットされます。

このモードでは、TIMx_CR1 レジスタの方向ビット (DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

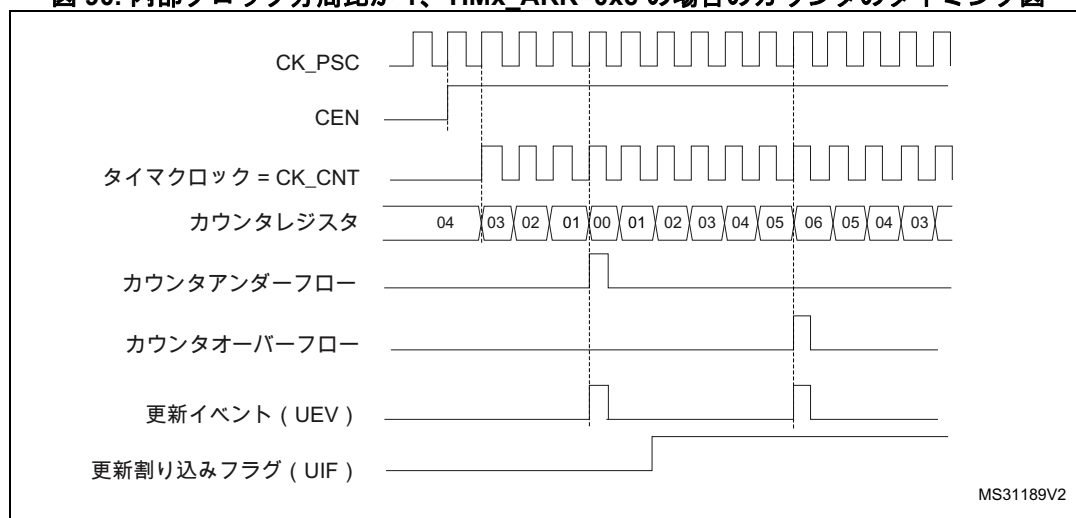
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 90. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 16.4: TIM1 および TIM8 レジスタ](#)を参照)。

図 91. 内部クロック分周比が 2 の場合のカウンタのタイミング図

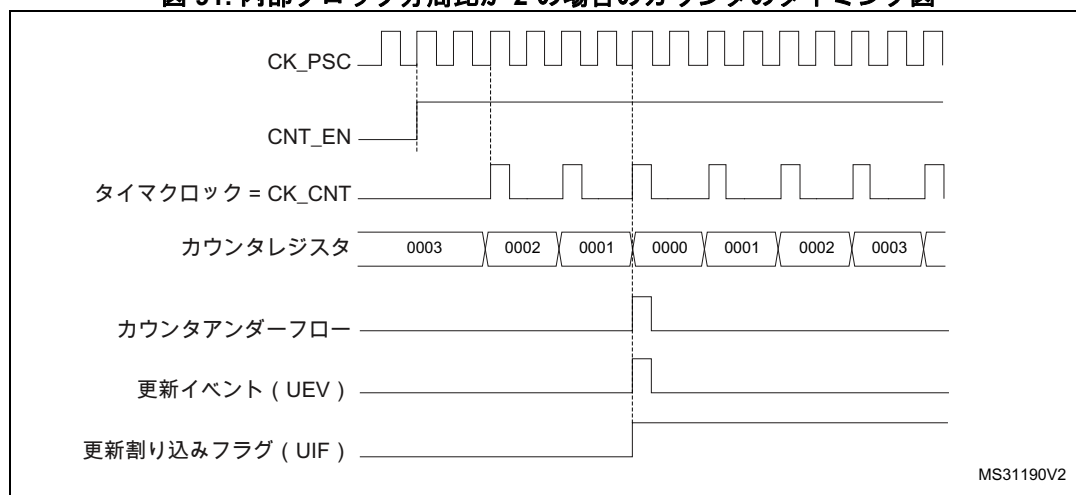
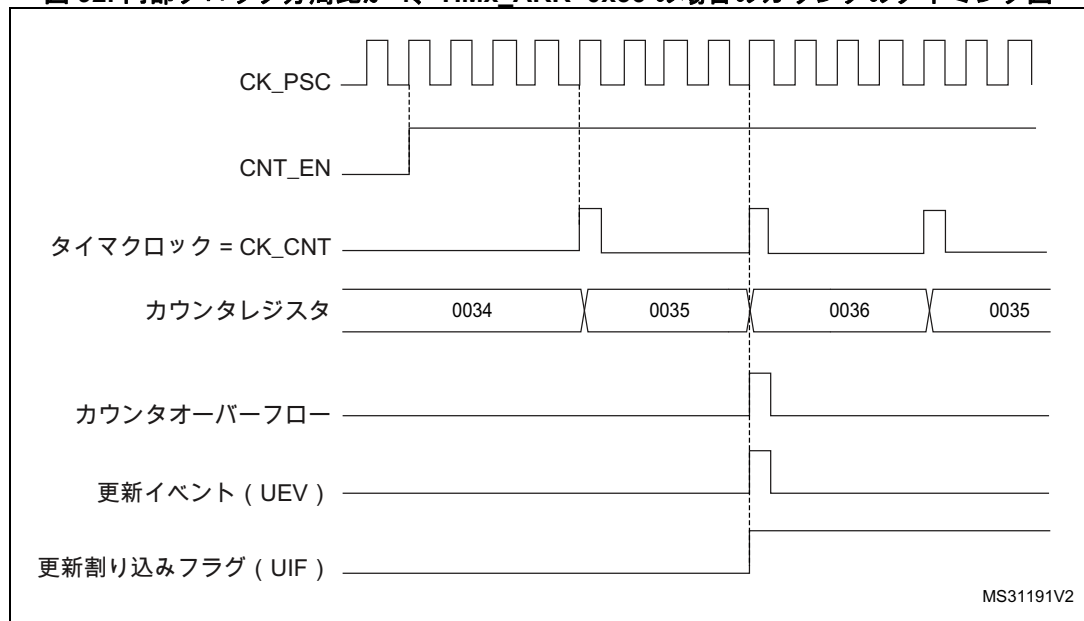


図 92. 内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 93. 内部クロック分周比が N の場合のカウンタのタイミング図

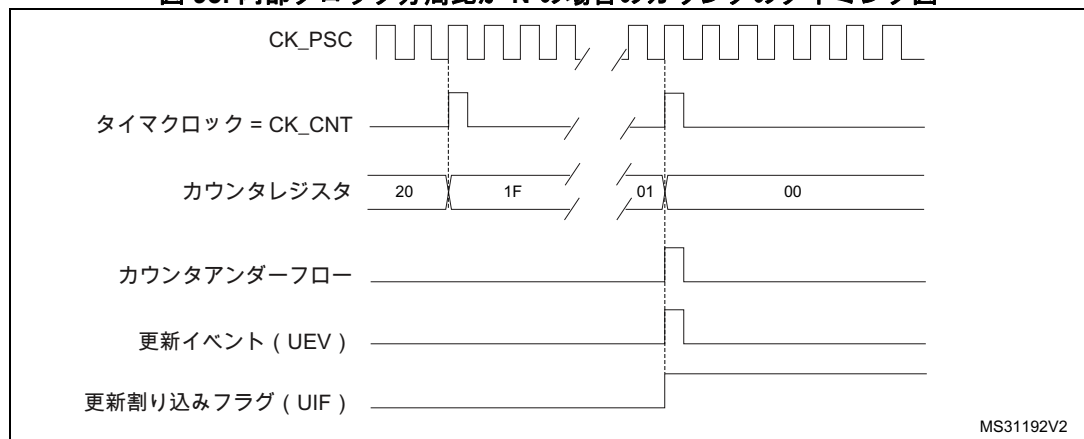


図 94. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

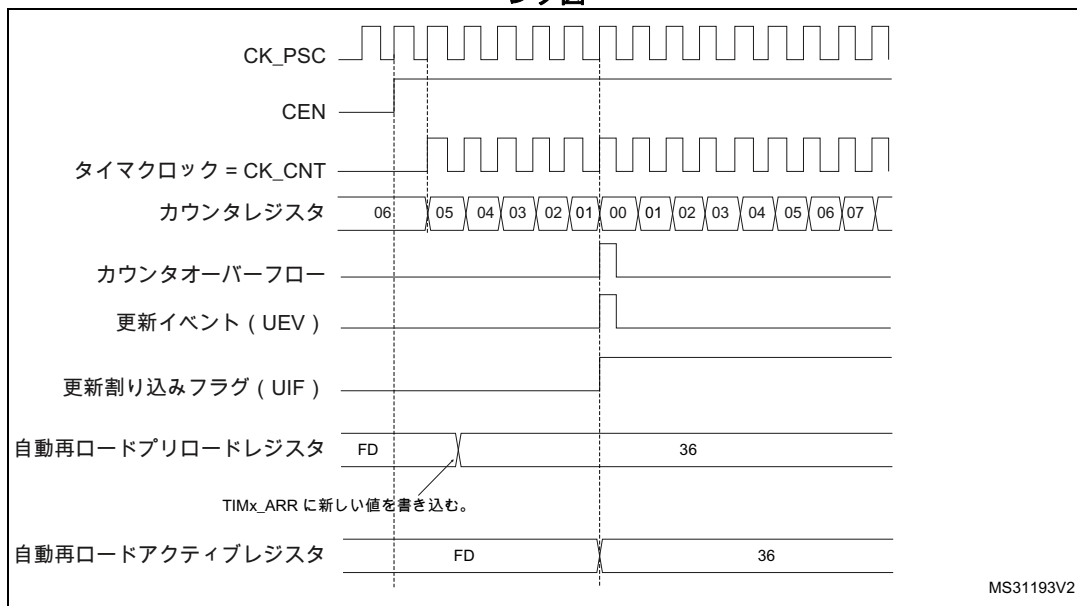
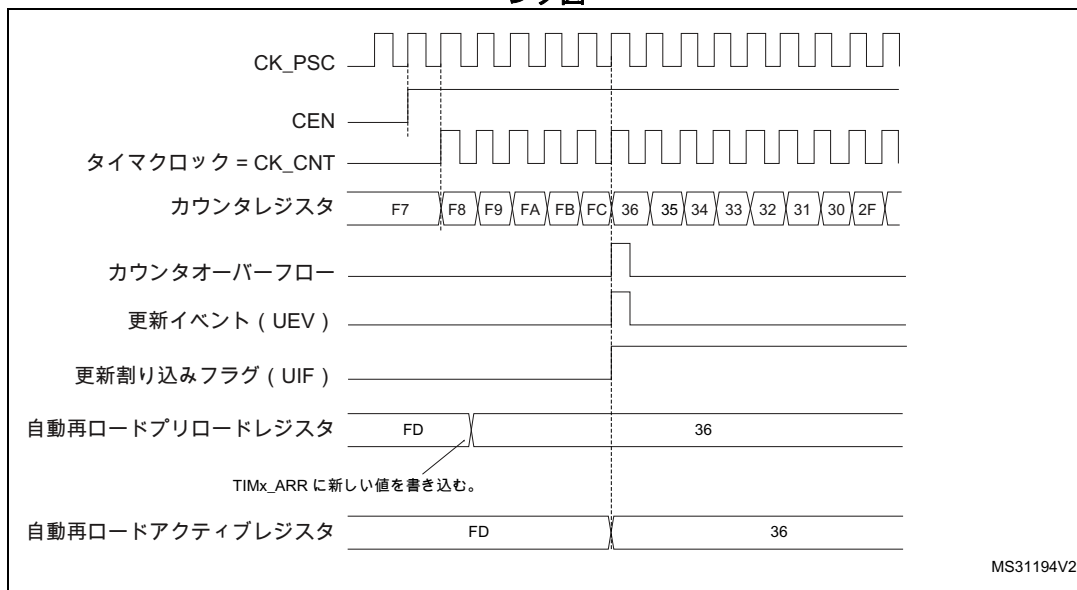


図 95. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



16.3.3 繰り返しカウンタ

セクション 16.3.1: タイムベースユニットに、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx_RCR 繰り返しカウンタレジスタの値を N とすると、N+1 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx_ARR 自動再ロードレジスタ、TIMx_PSC プリスケアラレジスタ、比較モードの TIMx_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。



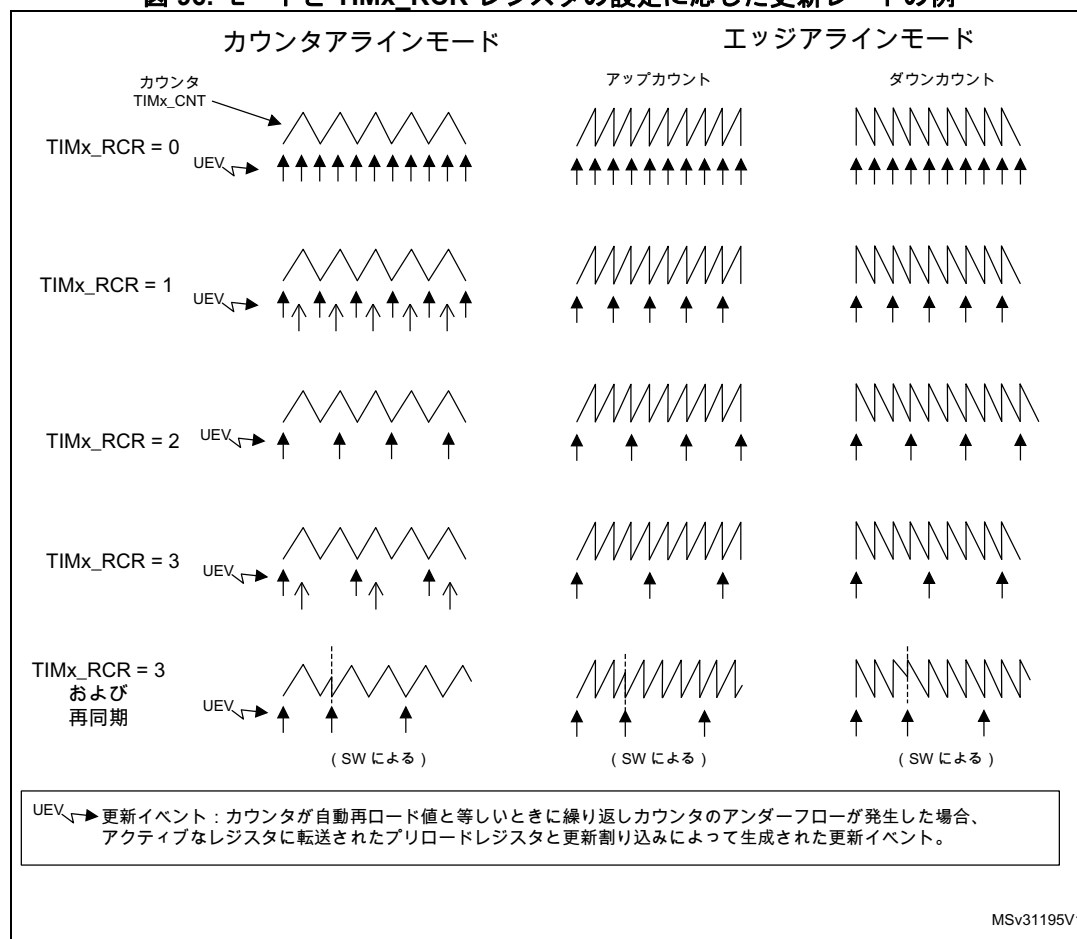
繰り返しカウンタは、次の場合にデクリメントします。

- アップカウントモードで、カウンタオーバーフローごと
- ダウンカウントモードで、カウンタアンダーフローごと
- センターアラインモードで、カウンタオーバーフローとカウンタアンダーフローごと最大繰り返し回数は 128 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センターアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は $2xT_{ck}$ です。

繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx_RCR レジスタの値によって定義されたとおりに維持されます (図 96 を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが発生されると、繰り返しカウンタの値にかかわらず直ちにイベントが発生し、繰り返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

センターアラインモードでは、TIMx_RCR が奇数の場合、TIMx_RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベントが発生します。カウンタの開始前に TIMx_RCR が書き込まれた場合は、オーバーフローで、UEV が発生します。カウンタの開始後に TIMx_RCR が書き込まれた場合は、アンダーフローで UEV が発生します。たとえば、TIMx_RCR = 3 の場合、TIMx_RCR の書き込みタイミングに応じて 4 回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 96. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



MSv31195V1

16.3.4 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

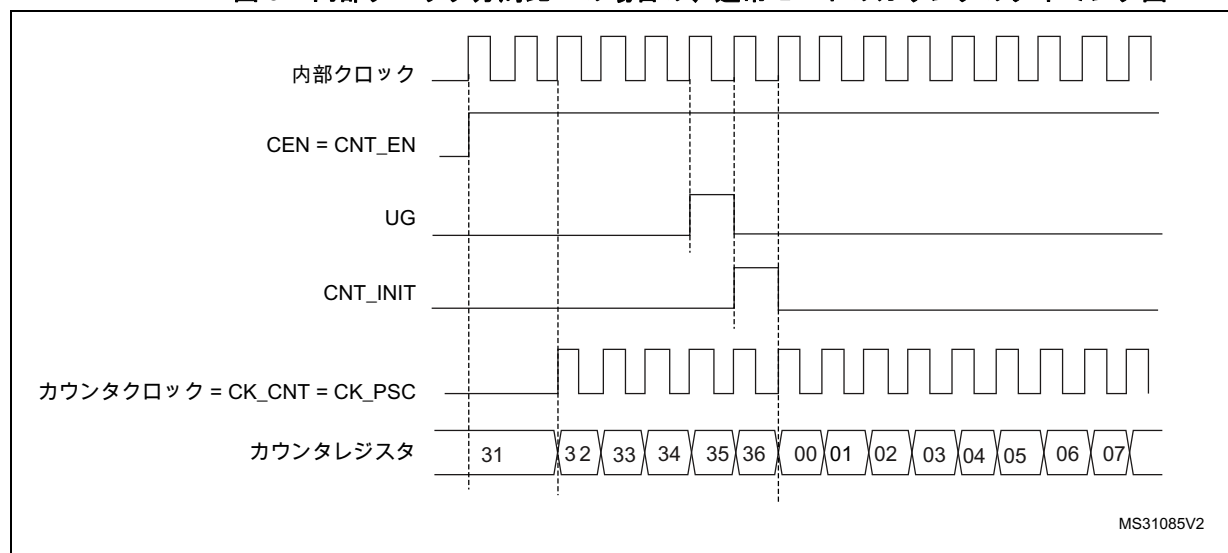
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン
- 外部クロックモード 2: 外部トリガ入力 ETR
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 1 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する](#) を参照してください。

内部クロックソース (CK_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 97](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

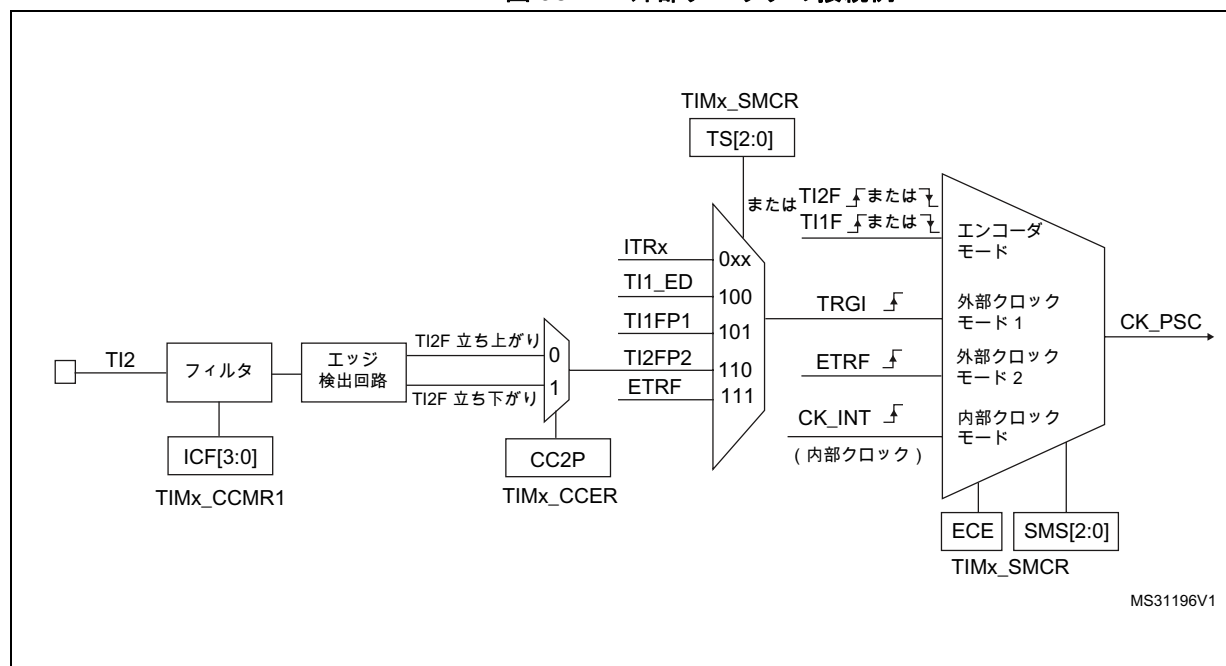
図 97. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 98. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

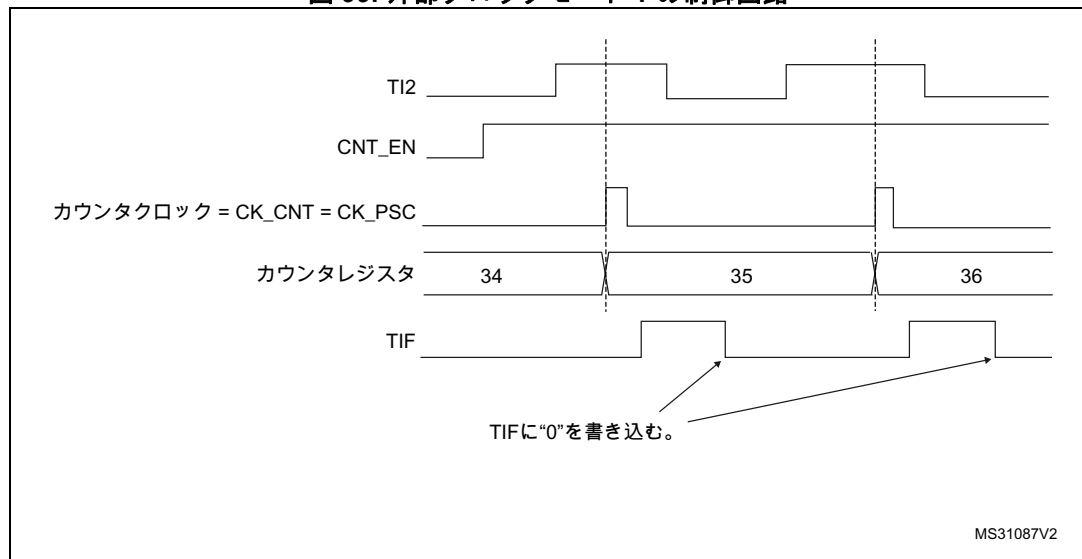
1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。
3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケアラはトリガには使用されないで、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 99. 外部クロックモード 1 の制御回路



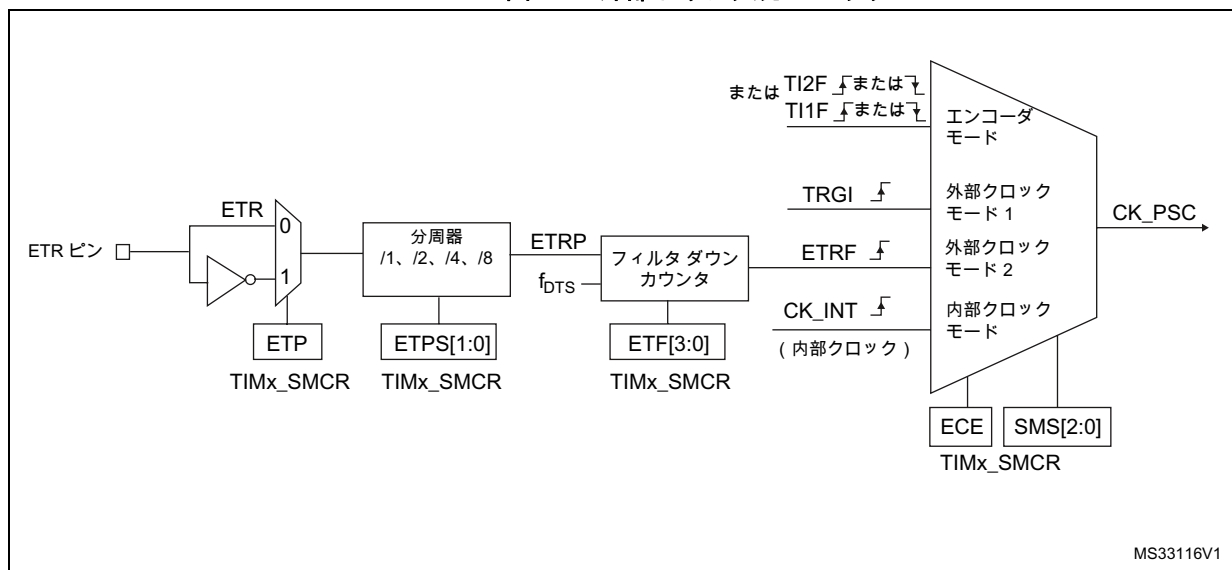
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

[図 100](#) に、外部トリガ入力ブロックの概要を示します。

図 100. 外部トリガ入力ブロック



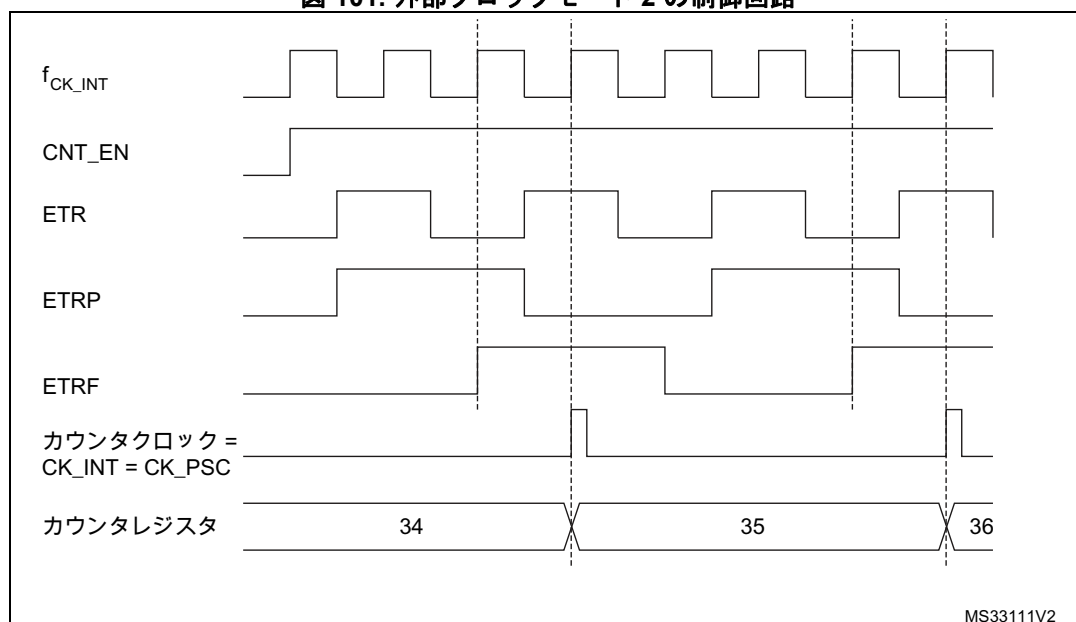
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 101. 外部クロックモード 2 の制御回路



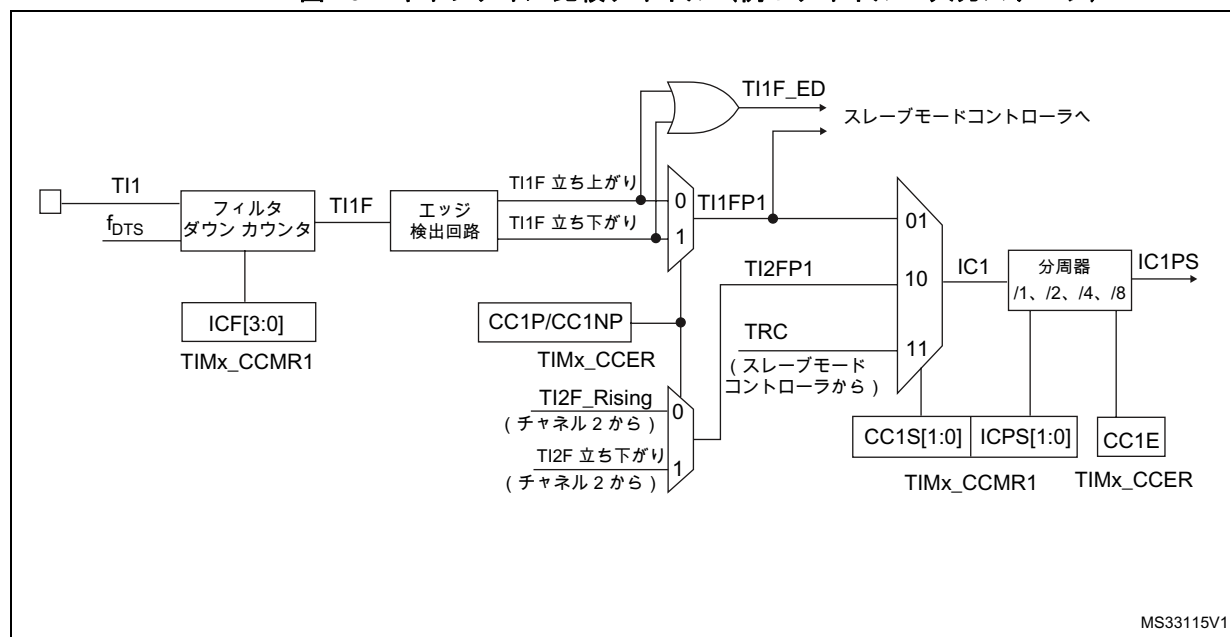
16.3.5 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入力ステージ（デジタルフィルタ、マルチプレクサ、プリスケアラ）、および出力ステージ（比較回路と出力制御）から構成されています。

図 102 から 図 105 に、1 つのキャプチャ／比較チャネルの概要を示します。

入力ステージは、対応する TlxF 入力をサンプリングして、フィルタリングを行った TlxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号（TlxFPx）を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ（ICxPS）に渡されます。

図 102. キャプチャ／比較チャネル（例：チャネル 1 入力ステージ）



出力ステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 103. キャプチャ／比較チャネル 1 メイン回路

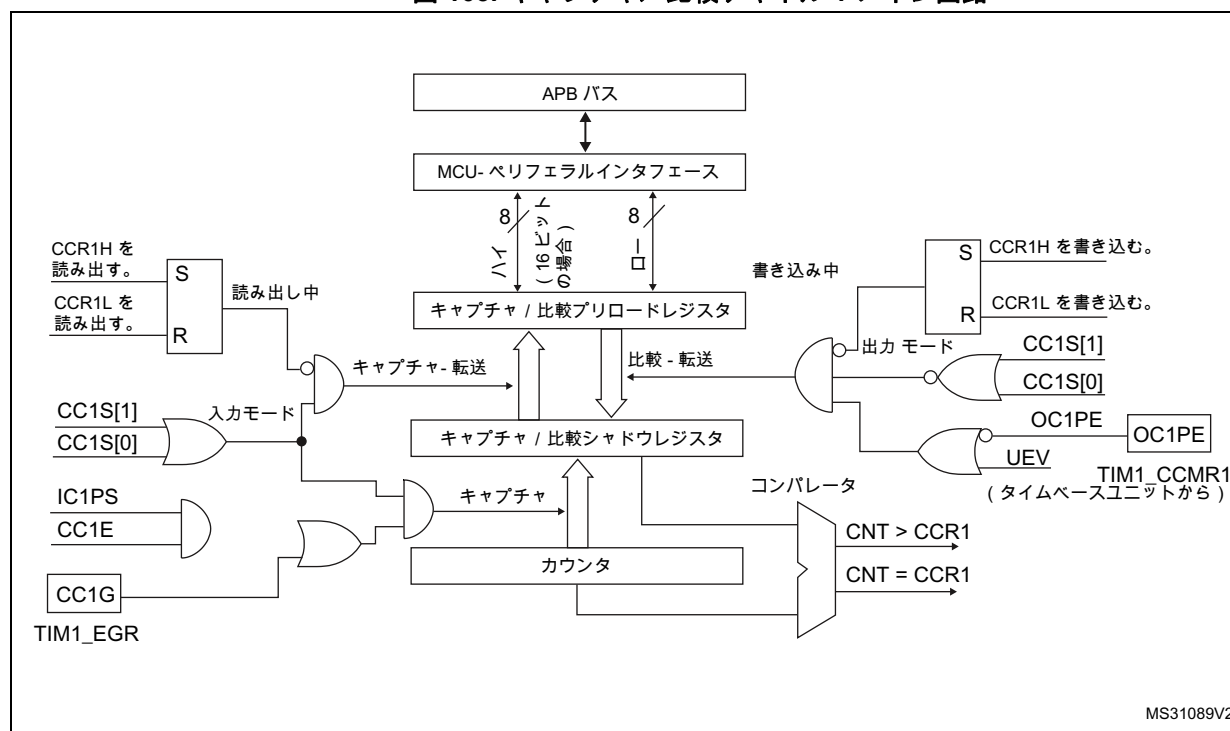


図 104. キャプチャ/比較チャンネル (チャンネル 1 から 3) の出力ステージ

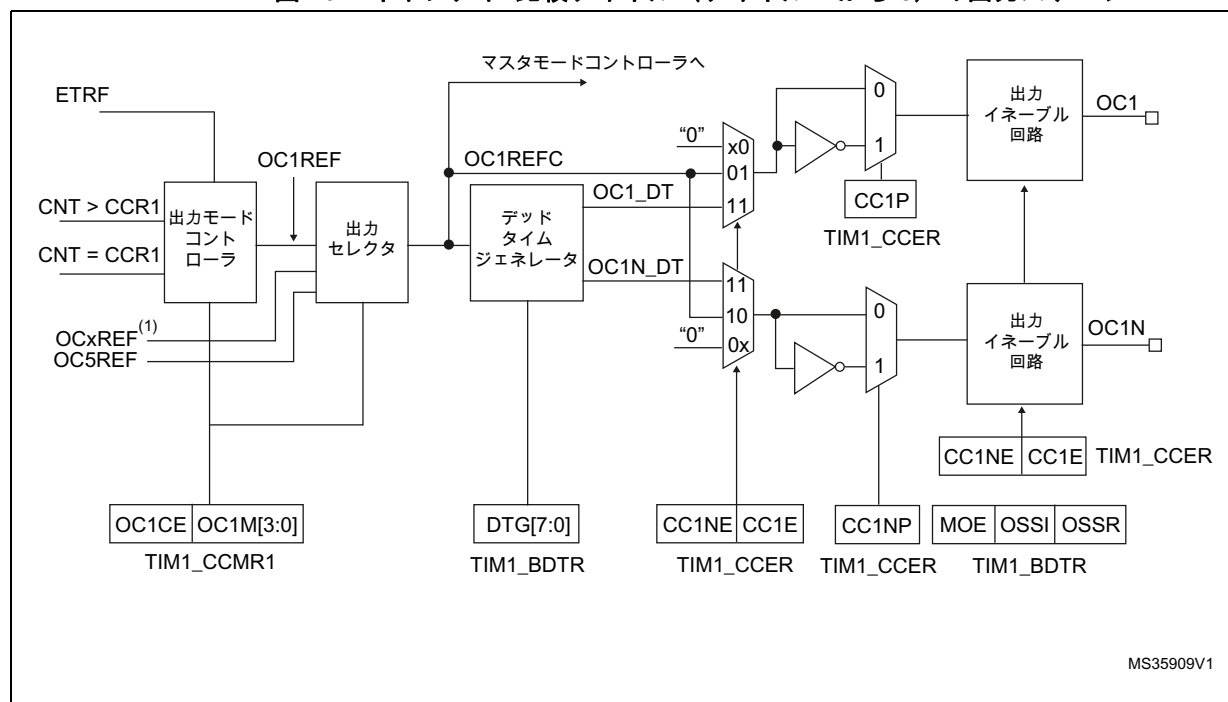
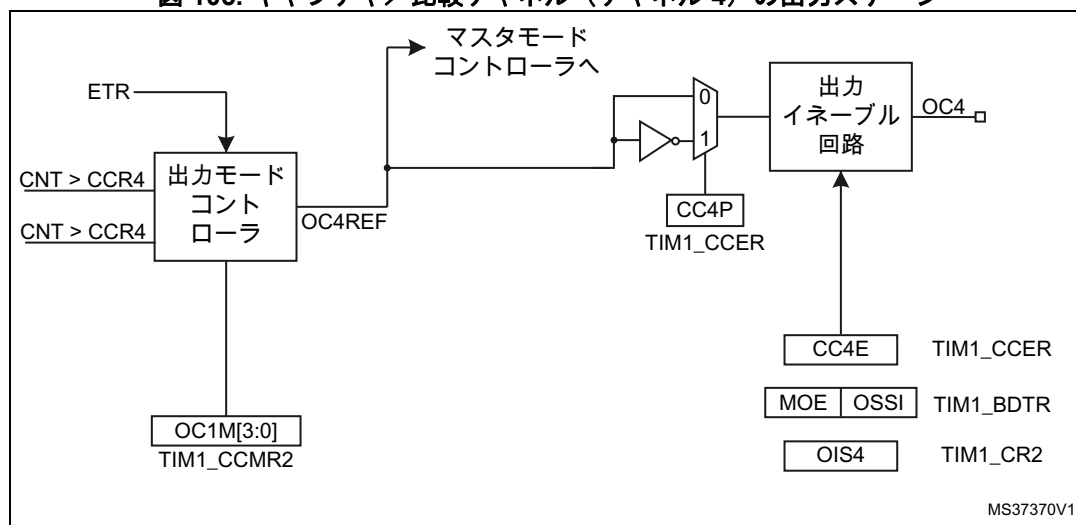


図 105. キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ



キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

16.3.6 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならない、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ変化のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要な場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注： IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

16.3.7 PWM 入力モード

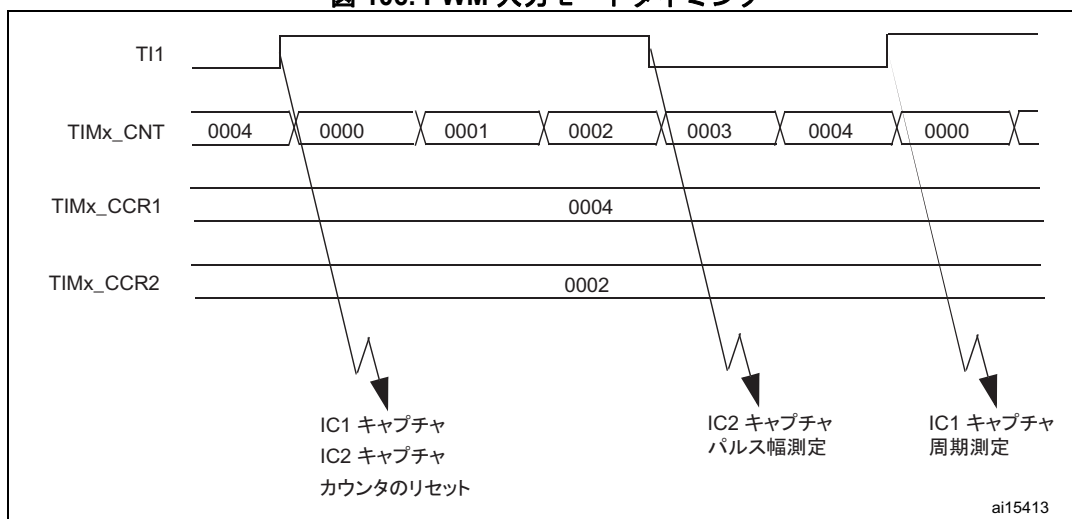
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに“1”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 106. PWM 入力モードタイミング



16.3.8 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx_OC MRx レジスタの OCxM ビットに "101" を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに "100" を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

16.3.9 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できます (ワンパルスモード)。

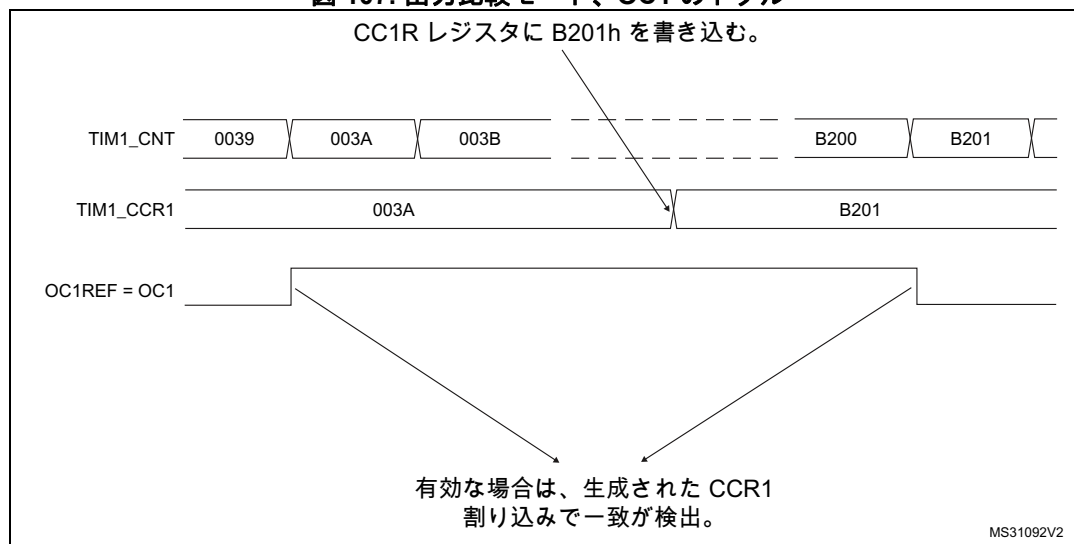
手順:

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例:
 - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 011 を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - 出力を有効にするには、CCxE ビットに 1 を書き込みます。

5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 107](#) に示します。

図 107. 出力比較モード、OC1 のトグル



16.3.10 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx がどうか判断されます (カウントの方向によります)。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

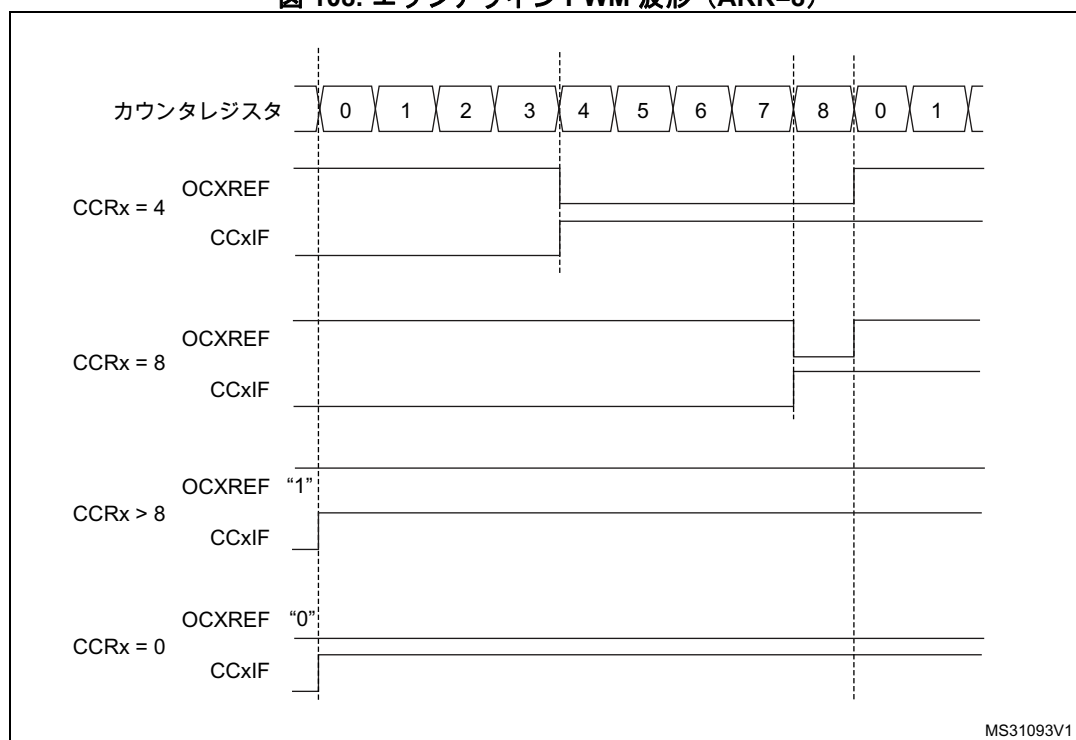
PWM エッジアラインモード

● アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード](#)を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 108](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 108. エッジアライン PWM 波形 (ARR=8)



● ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード](#)を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

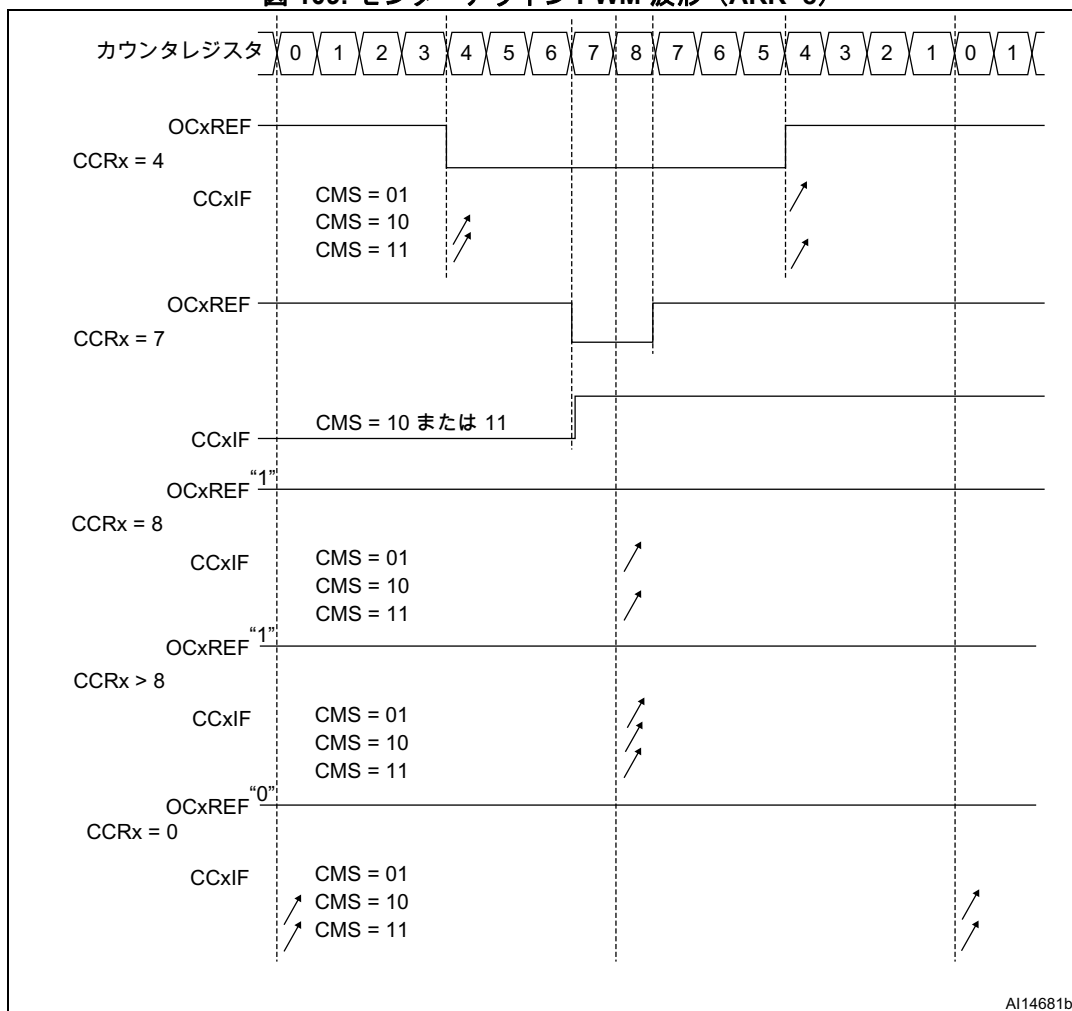
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての構成は、OCxRef/OCx 信号に対して同じ効果を持ちます)。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード \(アップ/ダウンカウント\)](#)を参照してください。

図 109 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 109. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - － 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - － カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

16.3.11 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1 および TIM8) は、2 つの相補信号を出力して、出力時のスイッチオンおよびスイッチオフを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性 (レベルシフタの内在的な遅延、電源スイッチによる遅延など) に応じて調整する必要があります。

出力の極性 (主出力 OCx または補 OCxN) は、出力ごとに独自に選択できます。これは TIMx_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx_BDTR レジスタと TIMx_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、表 92 を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるとき) に、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。TIMx_BDTR レジスタの DTG[7:0] ビットは、全チャネルに対するデッドタイムの生成を制御するために使います。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 110. デッドタイム挿入のある相補出力

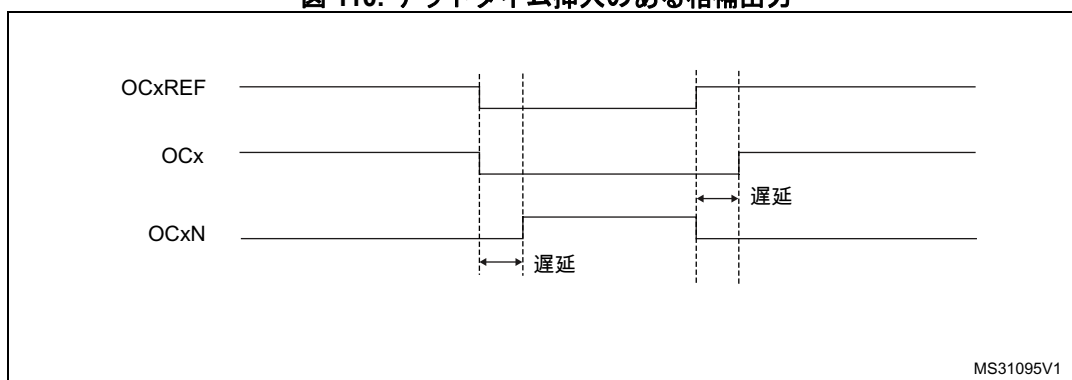
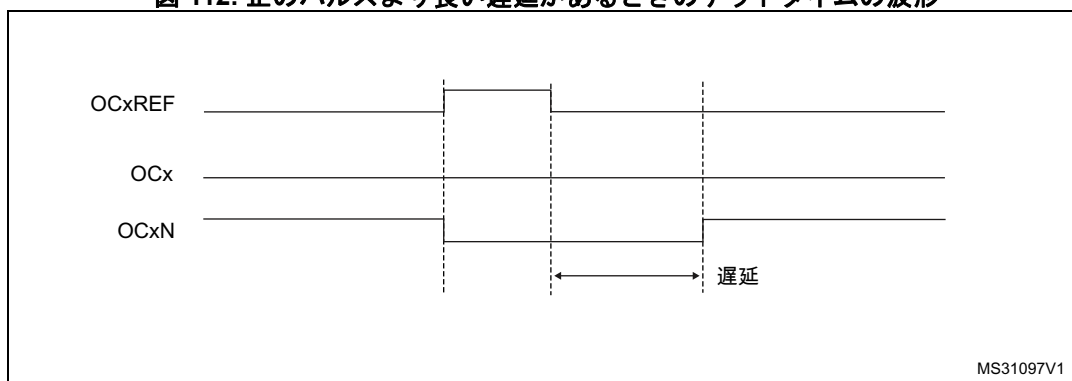


図 111. 負のパルスより長い遅延があるときのデッドタイムの波形



図 112. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャネルで同じであり、TIMx_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 16.4.18 : TIM1 および TIM8 ブレークおよびデッドタイムレジスタ \(TIMx_BDTR\)](#) を参照してください。

OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができます。

注: OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

16.3.12 ブレーク機能の使用

ブレーク機能を使用しているときには、出力イネーブル信号とインアクティブレベルは追加の制御ビット (TIMx_BDTR レジスタの MOE、OSSI、および OSSR ビットと TIMx_CR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、表 92 を参照してください。

ブレークソースは、ブレーク入力ピンか、リセットクロックコントローラ (RCC) のクロックセキュリティシステム (CSS) によって生成されたクロック障害イベントです。クロックセキュリティシステムの詳細については、セクション 6.2.7 を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx_BDTR レジスタの BKE ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がリエッジは非同期のことがあるので、実際の信号 (出力に作用する信号) と同期制御ビット (TIMx_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延 (ダミー命令) を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークが発生すると (ブレーク入力で選択されたレベル)、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態、アイドル状態、またはリセット状態 (OSSI ビットで選択) になります。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャネルは、MOE=0 になったとき、TIMx_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには:
 - 出力は、まずリセット状態のインアクティブ状態に置かれます (極性に依存します)。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください (約 2 CK_TIM クロックサイクル)。
 - OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はそのままか、CCxE または CCxNE ビットがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx_SR レジスタの BIF ビット) がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合は、割り込みを生成できます。TIMx_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。

- TIMx_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント UEV で再び自動的にセットされます。これを使用して、たとえば、レギュレーションを行うことができます。そうでない場合、MOE ビットは、再び 1 が書き込まれるまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

注： **ブレーク入力は、信号レベルに対して動作します。このため、ブレーク入力アクティブな間は、MOE をセットできません（自動的にも、ソフトウェアによっても）。この間、ステータスフラグ BIF をクリアできません。**

ブレークは、BRK 入力によって生成でき、BRK はプログラミング可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。

ブレークの生成には次の2つの方法があります。

- BRK 入力を使用します。BRK はプログラム可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。
- ソフトウェアから生成します。TIMx_EGR レジスタで BG ビットをセットします。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ（デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性）を固定することができます。TIMx_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 16.4.18：TIM1 および TIM8 ブレークおよびデッドタイムレジスタ \(TIMx_BDTR\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。


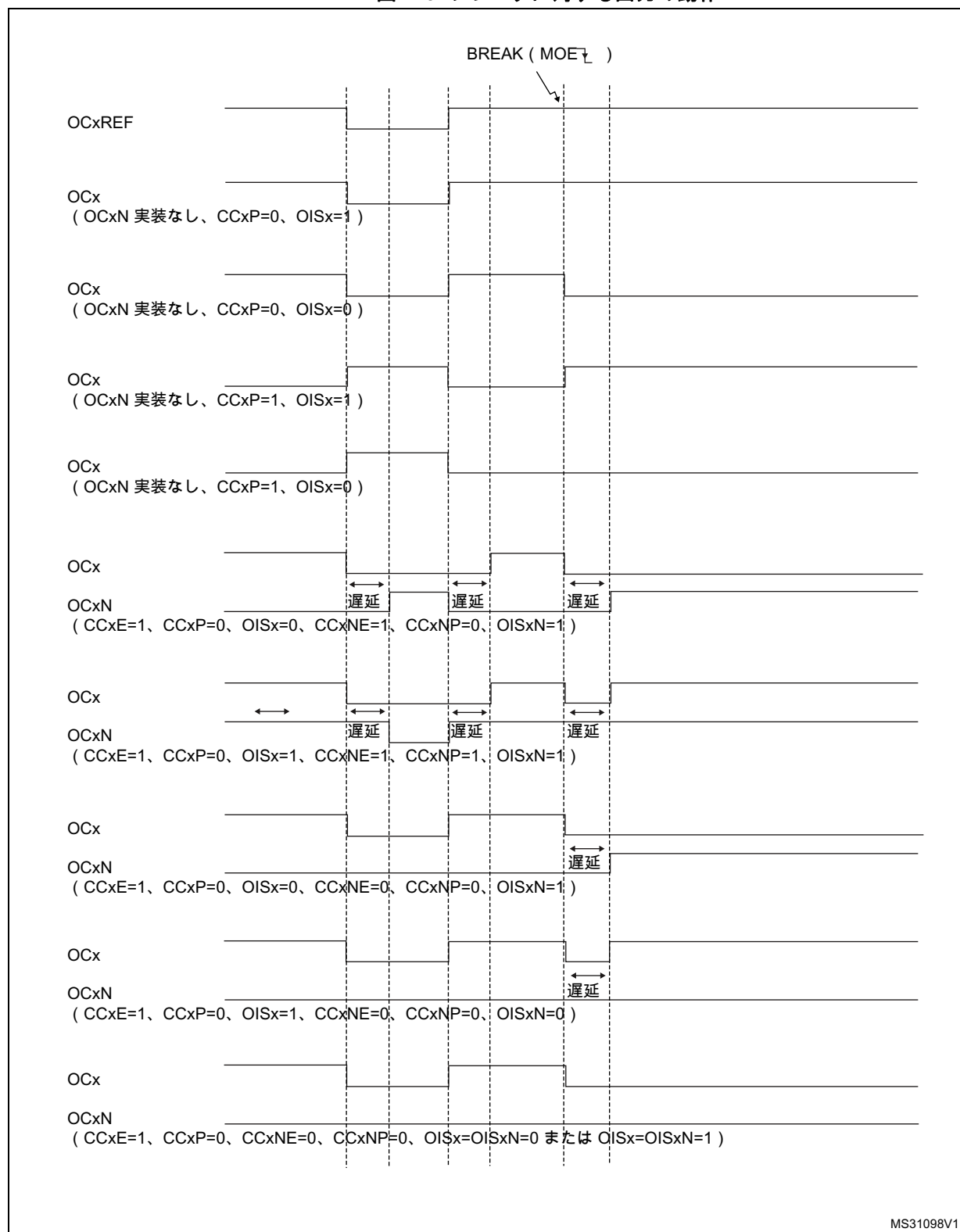
 [113](#) に、ブレークに対する出力の動作例を示します。

図 113. ブレークに対する出力の動作



16.3.13 外部イベントによる OCxREF 信号のクリア

特定のチャネルの OCxREF 信号は、ETRF 入力にハイレベルを適用する（対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする）ことによってローに駆動できます。OCxREF 信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

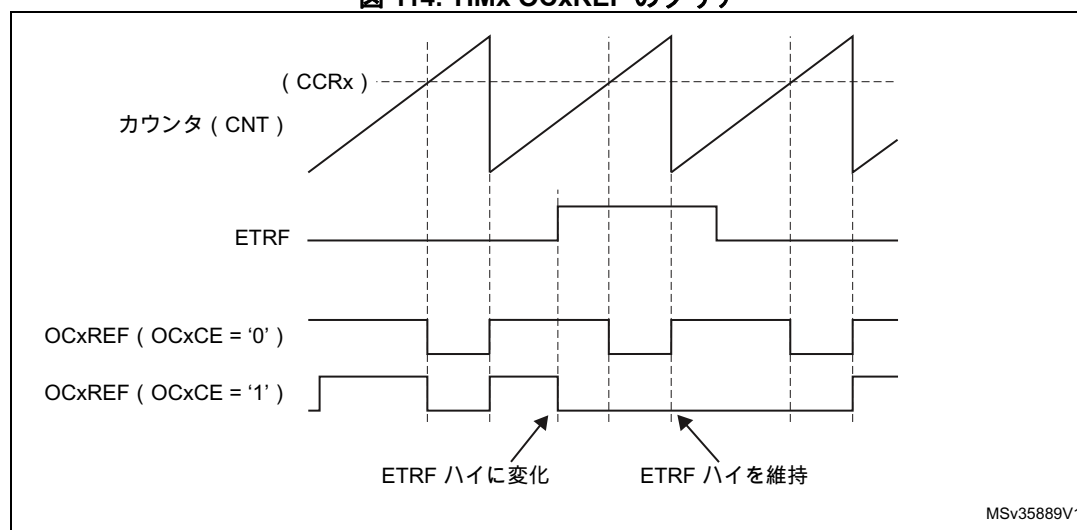
この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは機能しません。

たとえば、ETR 信号を比較回路の出力に接続して、電流処理に使用することができます。この場合、ETR は次のように設定する必要があります。

1. 外部トリガプリスケラはオフ状態に維持します (TIMx_SMCR レジスタの ETPS[1:0] ビットを“00”にセット)。
2. 外部クロックモード 2 を無効にします (TIMx_SMCR レジスタの ECE ビットを“0”にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、ユーザのニーズに応じて設定できます。

図 114 に、イネーブルビット OCxCE の両方の値について、ETRF 入力が高レベルになったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 114. TIMx OCxREF のクリア



MSv35889V1

注： 100% デューティサイクルの PWM の場合 (CCR_x > ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

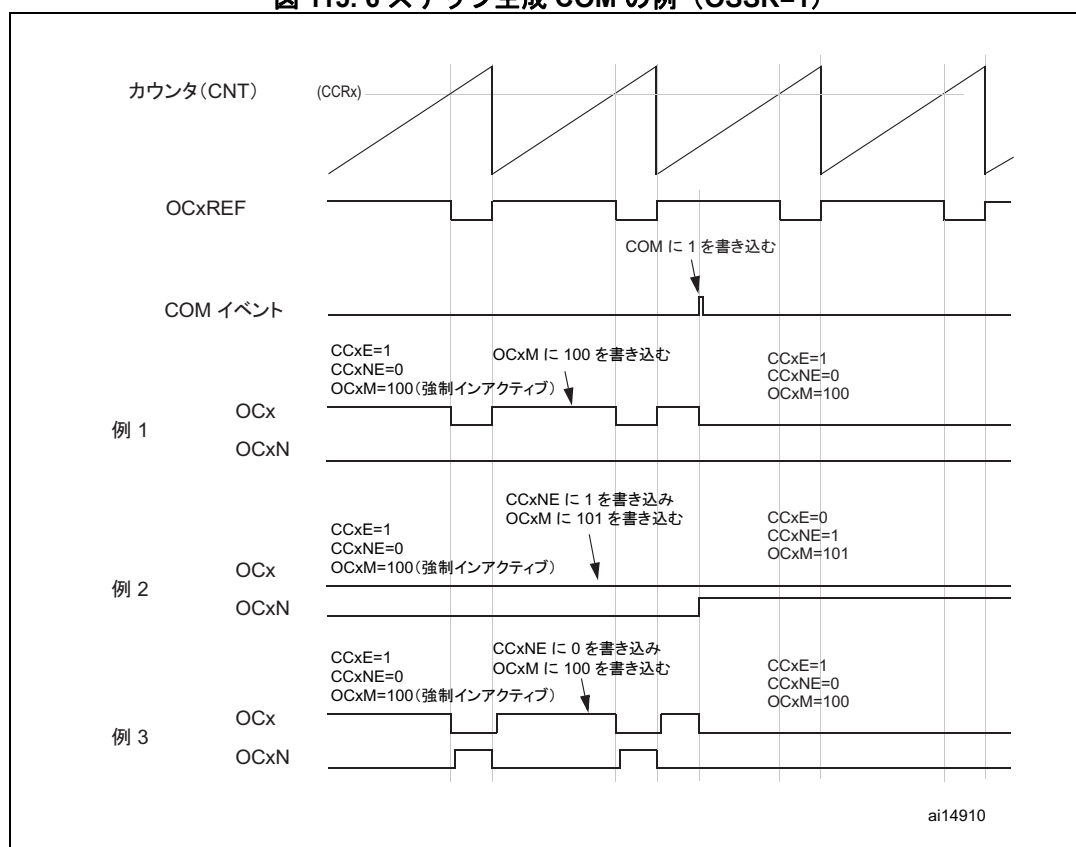
16.3.14 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転換イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (TRGI 立ち上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx_SR レジスタの COMIF ビット)、これによって割り込み (TIMx_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 115 に、COM イベントが発生したときの OCx と OCxN 出力の動作を、3 種類のプログラミング構成の例で示します。

図 115. 6 ステップ生成 COM の例 (OSSR=1)



16.3.15 ワンパルスモード

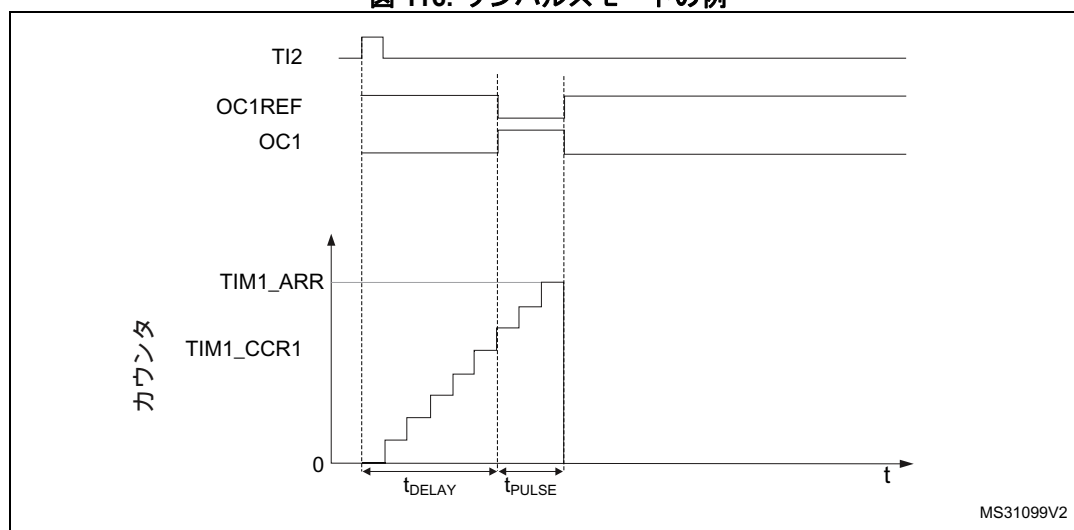
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 116. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで CC2P=“0”と CC2NP=“0”を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに“110”を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース : OCx 高速イネーブル :

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

16.3.16 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS=“001”を、TI1 エッジのみをカウントしている場合は SMS=“010”を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS=“011”を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2 つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。[表 90](#) を参照してください。カウンタのクロック供給は、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択の後) は TI1 と TI2、フィルタされず反転されない場合は TI1FP1=TI1、フィルタされず反転されない場合は TI2FP2=TI2) での有効な遷移ごとに行われます。ただし、カウンタは有効である (TIMx_CR1 レジスタの CEN ビットに“1”が書き込まれている) ことが前提です。2 つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機

能を続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。表 90 は、カウント方向とエンコーダ信号の可能な組み合わせを示します（TI1 と TI2 は同時に切り替わらないと想定しています）。

表 90. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	高	ダウン	アップ	カウントなし	カウントなし
	低	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	高	カウントなし	カウントなし	アップ	ダウン
	低	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の 両方をカウント	高	ダウン	アップ	アップ	ダウン
	低	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 117 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01"（TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置）
- CC2S="01"（TIMx_CCMR2 レジスタ、TI1FP2 は TI2 に配置）
- CC1P="0"、CC1NP ="0"、IC1F ="0000"（TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1）
- CC2P="0"、CC2NP ="0"、IC2F ="0000"（TIMx_CCER レジスタ、TI1FP2 非反転、TI1FP2=TI2）
- SMS="011"（TIMx_SMCR レジスタ、両方の入力立ち上がり立ち下がりの両エッジでアクティブ）
- CEN="1"（TIMx_CR1 レジスタ、カウンタ有効）



図 117. エンコーダインタフェースモードにおけるカウンタの動作例

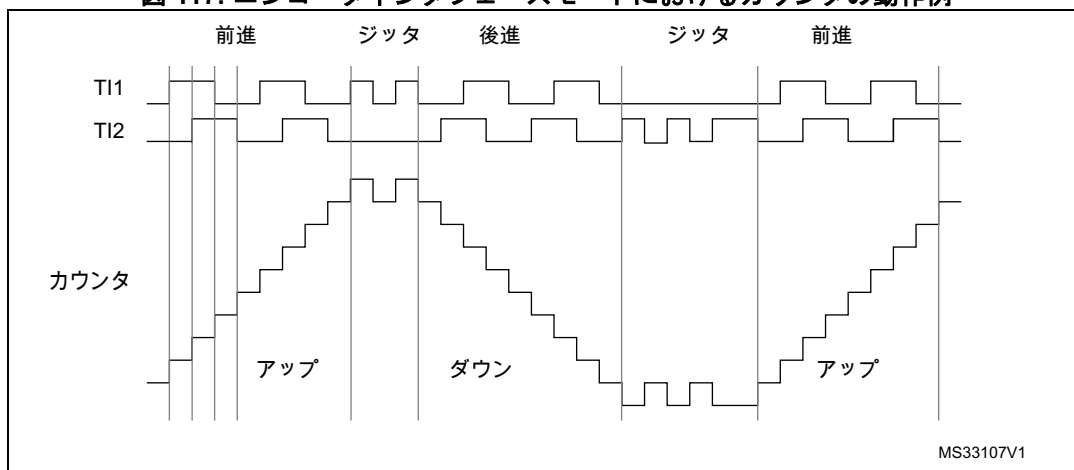
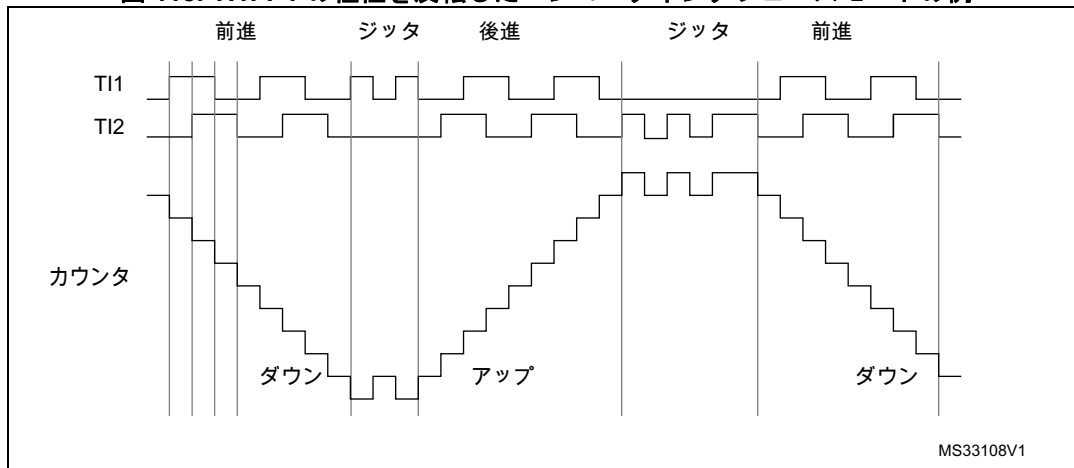


図 118 に、TI1FP1 の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、CC1P=“1”）。

図 118. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

16.3.17 タイマ入力 XOR 機能

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン TIMx_CH1、TIMx_CH2、および TIMx_CH3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。この機能をホールセンサのインタフェースに使用した例を、次の [セクション 16.3.18](#) に示します。

16.3.18 ホールセンサとのインタフェース

これは、高機能制御タイマ (TIM1 または TIM8) を使用して PWM 信号を生成し、モータと [図 119](#) で「インタフェースタイマ」と記されている別のタイマ TIMx (TIM2、TIM3、TIM4 または TIM5) を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて TI1 入力チャンネル (TIMx_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (TIMx_CH1、TIMx_CH2、および TIMx_CH3) をキャプチャします。

スレーブモードコントローラはリセットモードに設定され、スレーブ入力は TI1F_ED です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャンネル 1 がキャプチャモードで設定され、キャプチャ信号は TRC です ([図 102](#) を参照)。キャプチャされた値は、入力の 2 回の変化の間の経過時間に対応し、モータの速度情報を与えます。

「インタフェースタイマ」を出力モードで使用して、(COM イベントをトリガすることで) 高機能制御タイマ (TIM1 または TIM8) のチャンネルの設定を変更するパルスを生成することができます。TIM1 タイマは、モータを駆動する PWM 信号を生成するために使用されます。このためには、プログラミングした遅延の後に正パルスが生成されるように (出力比較モードまたは PWM モードで) インタフェースタイマチャンネルをプログラミングする必要があります。このパルスは、TRGO 出力を通じて高機能制御タイマ (TIM1 または TIM8) に送られます。

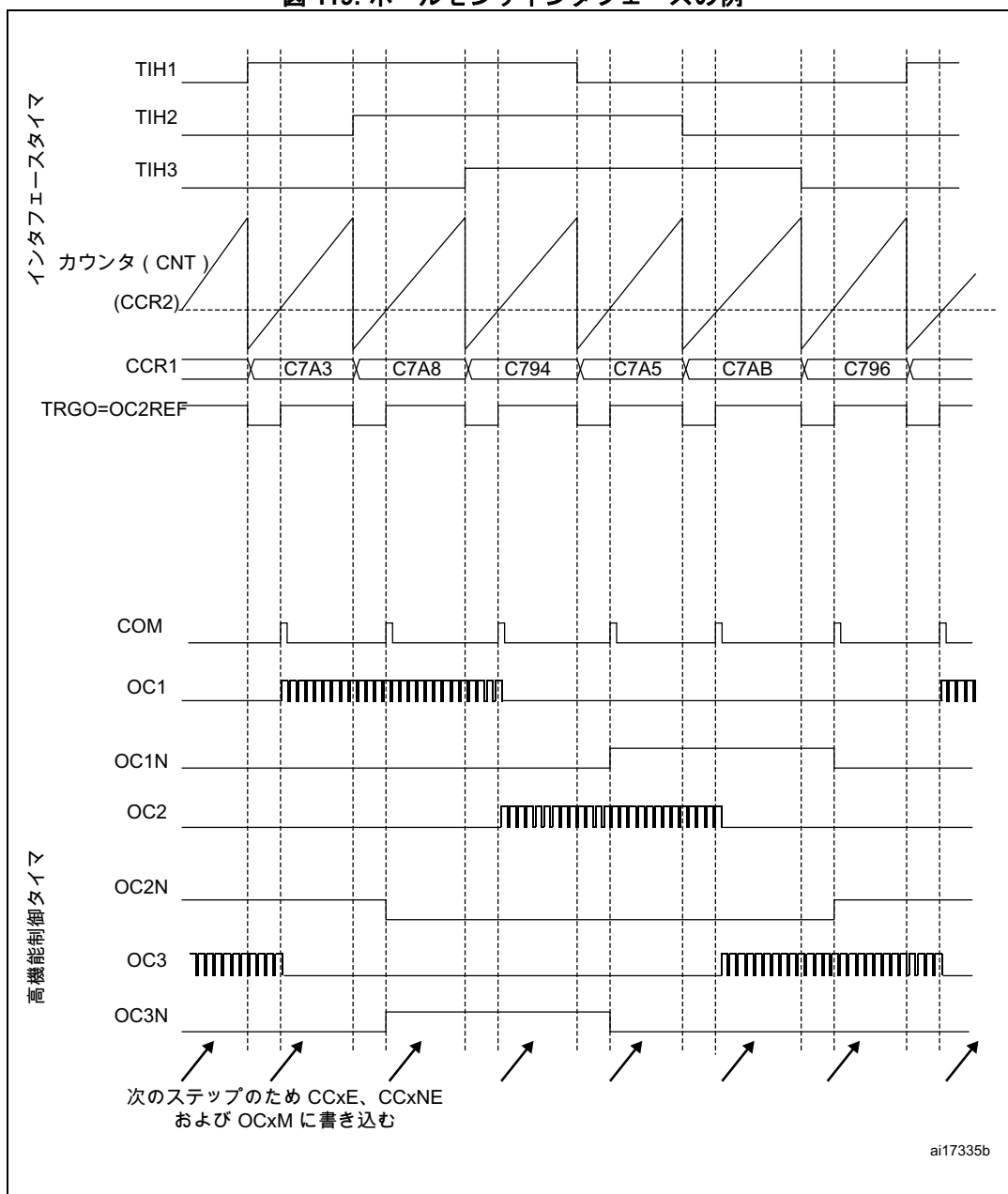
例: TIMx タイマの 1 つに接続されているホール入力の変化するたびに、プログラミングした遅延の後に高機能制御タイマ TIM1 の PWM 設定を変更するとします。

- 3 つのタイマ入力を TI1 入力チャンネルに OR 接続します。このためには、TIMx_CR2 レジスタの TI1S ビットに“1”を書き込みます。
- タイムベースをプログラムします。このためには、TIMx_ARR に最大値を書き込み、TI1 の変化でカウンタがクリアされるようにします。最大カウンタ時間がセンサの 2 回の変化の間の時間より長くなるように、プリスケアラを設定します。
- チャンネル 1 をキャプチャモード (TRC 選択) にプログラムします。すなわち、TIMx_CCMR1 レジスタの CC1S ビットに“11”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャンネル 2 を PWM 2 モードにプログラミングし、希望の遅延を指定します。このためには、TIMx_CCMR1 レジスタの OC2M ビットに“111”を、CC2S ビットに“00”を書き込みます。
- TRGO 上のトリガ出力として OC2REF を選択します。このためには、TIMx_CR2 レジスタの MMS ビットに“101”を書き込みます。

高機能制御タイマ TIM1 で、トリガ入力として適切な ITR 入力を選択する必要があり、タイマが PWM 信号を生成するようにプログラミングし、キャプチャ/比較制御信号がプリロードされ (TIMx_CR2 レジスタの CCPC=1)、COM イベントがトリガ入力によって制御されなければなりません (TIMx_CR2 レジスタの CCUS=1)。PWM 制御ビット (CCxE、OCxM) は、COM イベント後に次のステップのために書き込まれます (これは、OC2REF の立ち上がりエッジによって生成される割り込みサブルーチンで行うことができます)。

図 119 に、この例を示します。

図 119. ホールセンサインタフェースの例



16.3.19 TIMx と外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

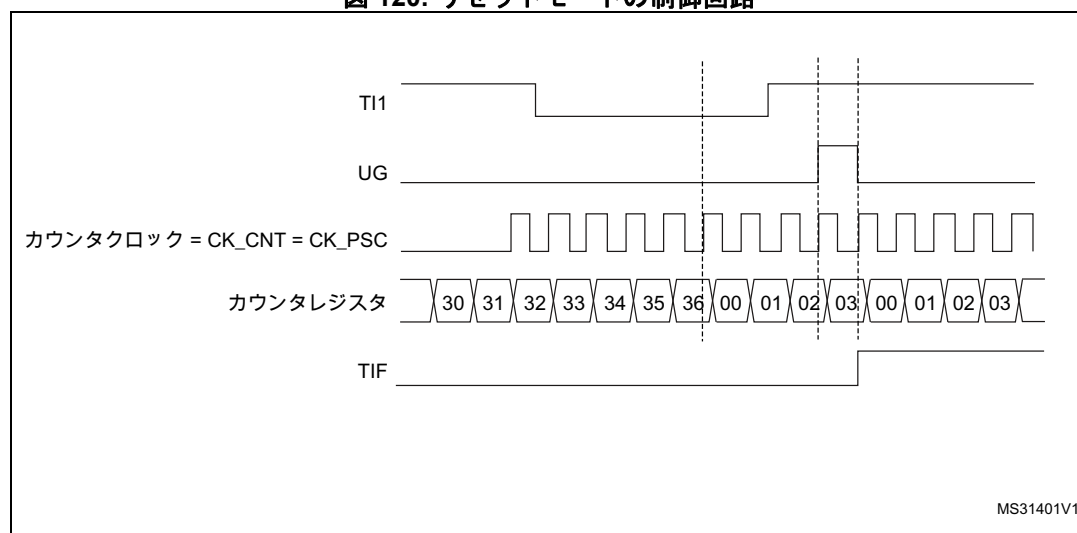
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタに CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 120. リセットモードの制御回路



MS31401V1

スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

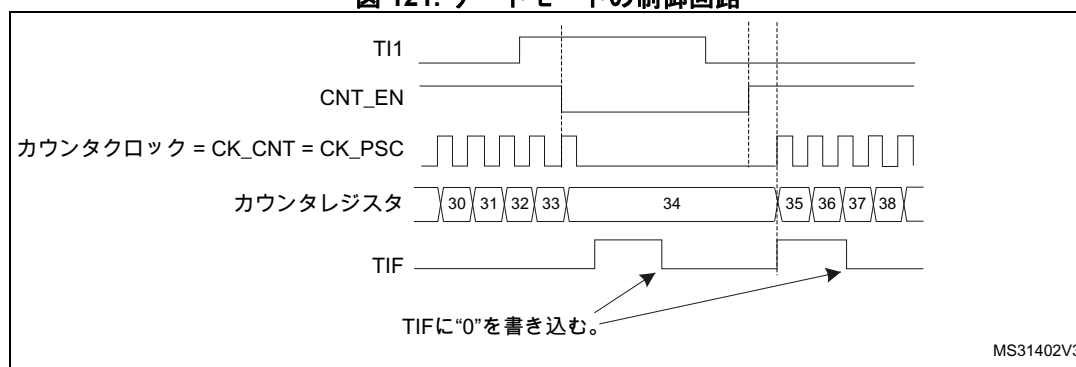
次の例では、アップカウンタは TI1 入力があるときにだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01 ビット)。TIMx_CCER レジスタで CC1P=1 と CC1NP=0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 121. ゲートモードの制御回路



MS31402V3

スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

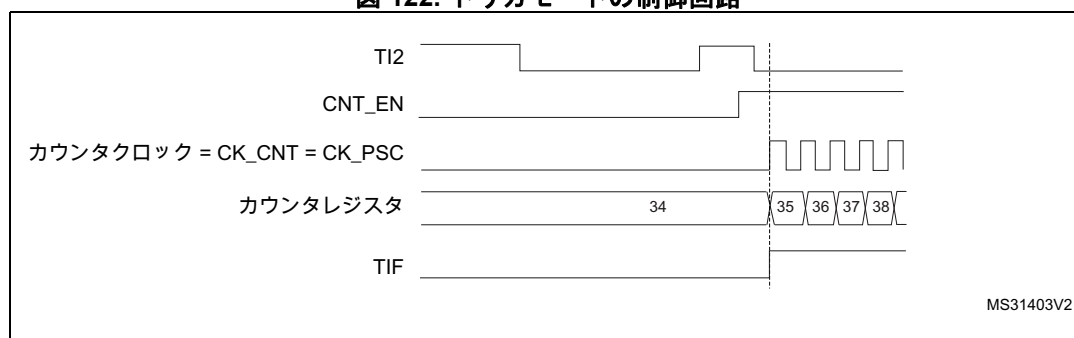
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないの、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 122. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、またはトリガモード）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

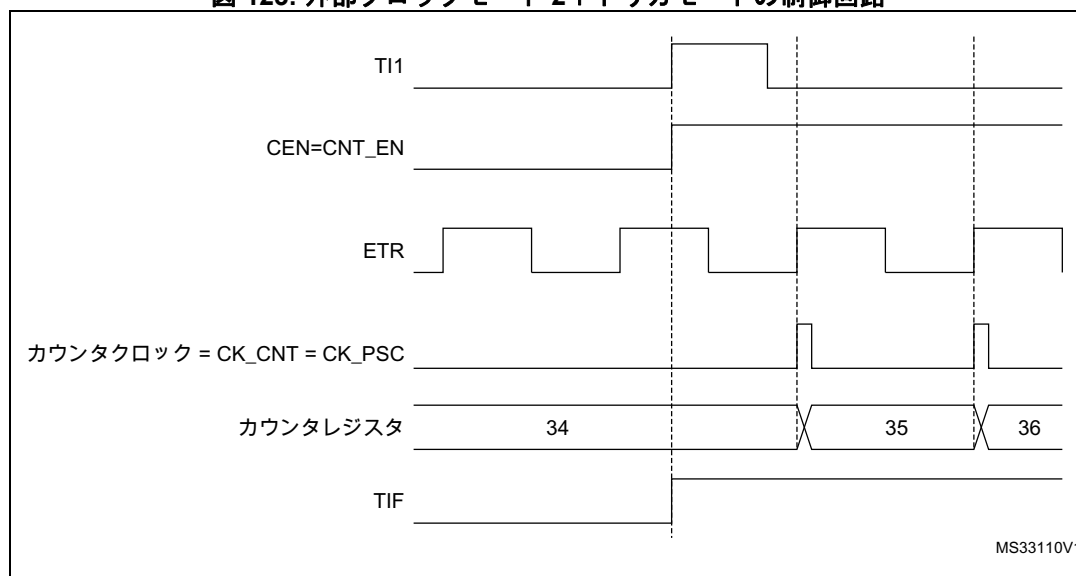
1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000 : フィルタなし
 - ETPS = 00 : プリスケアラは無効
 - ETP = 0 : ETR の立ち上がりエッジを検出。ECE = 1 で外部クロックモード 2 を有効にする。

2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケラはトリガには使用されないので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 123. 外部クロックモード 2+トリガモードの制御回路



16.3.20 タイマの同期

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 17.3.15 : タイマの同期 \(499 ページ\)](#) を参照してください。

16.3.21 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 30.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#) を参照してください。

16.4 TIM1 および TIM8 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1: レジスタに関する略記](#)を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット) またはワード (32 ビット) 単位で行うことができます。

16.4.1 TIM1 および TIM8 制御レジスタ 1 (TIMx_CR1)

アドレスオフセット: 0x00

リセット値: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]**: クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (ETR、Tlx) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

00: $t_{DTS}=t_{CK_INT}$

01: $t_{DTS}=2*t_{CK_INT}$

10: $t_{DTS}=4*t_{CK_INT}$

11: 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE**: 自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。

1: TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]**: センターアラインモード選択

00: エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01: センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10: センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。

11: センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注: カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

- 0 : カウンタはアップカウンタとして使用されます。
- 1 : カウンタはダウンカウンタとして使用されます。

注： このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します（CEN ビットをクリア）。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- － カウンタオーバーフロー／アンダーフロー
- － UG ビットのセット
- － スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- － カウンタオーバーフロー／アンダーフロー
- － UG ビットのセット
- － スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

注： 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

16.4.2 TIM1 および TIM8 制御レジスタ 2（TIM_x_CR2）

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw



ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **OIS4** : 出力アイドル状態 4 (OC4 出力)

OIS1 ビットの説明を参照してください。

ビット 13 **OIS3N** : 出力アイドル状態 3 (OC3N 出力)

OIS1N ビットの説明を参照してください。

ビット 12 **OIS3** : 出力アイドル状態 3 (OC3 出力)

OIS1 ビットの説明を参照してください。

ビット 11 **OIS2N** : 出力アイドル状態 2 (OC2N 出力)

OIS1N ビットの説明を参照してください。

ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)

OIS1 ビットの説明を参照してください。

ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)

0 : MOE=0 のとき、デッドタイム後に OC1N=0

1 : MOE=1 のとき、デッドタイム後に OC1N=0

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)

0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 7 **TI1S** : TI1 選択

0 : TIMx_CH1 ピンが TI1 入力に接続されます。

1 : TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[2:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされるとき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。 (TRGO)。

100 : 比較 - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : 比較 - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

110 : 比較 - OC3REF 信号がトリガ出力 (TRGO) として使用されます。

111 : 比較 - OC4REF 信号がトリガ出力 (TRGO) として使用されます。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

- 0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。
- 1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

- 0 : キャプチャ/比較制御ビットがプリロードされる時には（CCPC=1）、COMG ビットをセットすることによってのみ更新されます。
- 1 : キャプチャ/比較制御ビットがプリロードされる時には（CCPC=1）、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

注： このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

- 0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。
- 1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込みの後、これらのビットは、転換イベント（COM）が発生した時にのみ更新されます（CCUS ビットに応じて、COMG ビットがセットまたは TRGI で立ち上がりエッジが検出されたとき）。

注： このビットは、相補出力を持つチャンネルでのみ機能します。

16.4.3 TIM1 および TIM8 のスレーブモード制御レジスタ（TIMx_SMCR）

アドレスオフセット : 0x08

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW

ビット 15 **ETP** : 外部トリガ極性

- このビットは、ETR と $\overline{\text{ETR}}$ のいずれがトリガ動作に使用されるかを選択します。
- 0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。
- 1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

- このビットは、外部クロックモード 2 を有効にします。
- 0 : 外部クロックモード 2 は無効です。
- 1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

注：

- 1: ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります（SMS=111、TS=111）。
- 2 : 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません（TS ビットが 111 でないことが必要）。
- 3 : 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力 ETRF です。

ビット 13:12 ETPS[1:0] : 外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、TIMxCLK 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

00 : プリスケアラオフ

01 : ETRP 周波数は 2 分周されます。

10 : ETRP 周波数は 4 分周されます。

11 : ETRP 周波数は 8 分周されます。

ビット 11:8 ETF[3:0] : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

ビット 7 MSM : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 TS[2:0] : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

000 : 内部トリガ 0 (ITR0)

001 : 内部トリガ 1 (ITR1)

010 : 内部トリガ 2 (ITR2)

011 : 内部トリガ 3 (ITR3)

100 : TI1 エッジ検出回路 (TI1F_ED)

101 : フィルタタイマ入力 1 (TI1FP1)

110 : フィルタタイマ入力 2 (TI2FP2)

111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、[表 91 : TIMx 内部トリガ接続](#)を参照してください。

注 : 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号（TRGI）のアクティブエッジが外部入力で選択された極性にリンクされます（入力制御レジスタおよび制御レジスタの説明を参照してください）。
000：スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。
001：エンコーダモード 1 - カウンタは、TI1FP1 のレベルに応じて、TI2FP2 のエッジでカウントアップ/ダウンします。
010：エンコーダモード 2 - カウンタは、TI2FP2 のレベルに応じて、TI1FP1 のエッジでカウントアップ/ダウンします。
011：エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。
100：リセットモード - 選択されたトリガ入力（TRGI）の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
101：ゲートモード - カウンタクロックは、トリガ入力（TRGI）がハイのときに有効になります。トリガがローになると、カウンタは停止します（リセットはされません）。カウンタの開始と停止の両方が制御されます。
110：トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します（リセットはされません）。カウンタの開始のみが制御されます。
111：外部クロックモード 1 - 選択されたトリガ（TRGI）の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として **TI1F_ED** が選択されている場合（**TS=100**）、ゲートモードを使用することはできません。**TI1F_ED** は **TI1F** の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

表 91. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

16.4.4 TIM1 および TIM8 DMA／割り込み有効レジスタ（TIMx_DIER）

アドレスオフセット：0x0C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

- 0：トリガ DMA リクエストは無効です。
- 1：トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

- 0：COM DMA リクエストは無効です。
- 1：COM DMA リクエストは有効です。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割り込みイネーブル

0 : ブレーク割り込みは無効です。

1 : ブレーク割り込みは有効です。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5 **COMIE** : COM 割り込みイネーブル

0 : COM 割り込みは無効です。

1 : COM 割り込みは有効です。

ビット 4 **CC4IE** : キャプチャ／比較 4 割り込みイネーブル

0 : CC4 割り込みは無効です。

1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割り込みイネーブル

0 : CC3 割り込みは無効です。

1 : CC3 割り込みは有効です。

ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル

0 : CC2 割り込みは無効です。

1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

16.4.5 TIM1 および TIM8 のステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャネルが入力キャプチャモードに設定されている時のみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
0 : オーバーキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割り込みフラグ
このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。
0 : ブレークイベントは発生していません。
1 : ブレーク入力アクティブレベルが検出されました。

ビット 6 **TIF** : トリガ割り込みフラグ
このフラグは、トリガイイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力アクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。
0 : トリガイイベントは発生していません。
1 : トリガ割り込みが保留中です。

ビット 5 **COMIF** : COM 割り込みフラグ
このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき)。ソフトウェアによってクリアされます。
0 : COM イベントは発生していません。
1 : COM 割り込みがペンディング中です。

ビット 4 **CC4IF** : キャプチャ/比較 4 割り込みフラグ
CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ/比較 3 割り込みフラグ
CC1IF の説明を参照してください。

ビット 2 **CC2IF** : キャプチャ／比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ／比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます（センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください）。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ／ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（IC1 で、選択された極性に一致するエッジが検出されました）。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

– 繰り返しカウンタ値に関するオーバーフローまたはアンダーフロー（繰り返しカウンタ=0 の場合の更新）、および TIMx_CR1 レジスタの UDIS=0 の場合。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき（[セクション 16.4.3 : TIM1 および TIM8 のスレーブモード制御レジスタ \(TIMx_SMCR\)](#) を参照）。

16.4.6 TIM1 および TIM8 のイベント生成レジスタ（TIMx_EGR）

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 TG : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 COMG : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4 CC4G : キャプチャ / 比較 4 イベント生成

CC1G の説明を参照してください。

ビット 3 CC3G : キャプチャ / 比較 3 イベント生成

CC1G の説明を参照してください。

ビット 2 CC2G : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 CC1G : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 UG : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

16.4.7 TIM1 および TIM8 のキャプチャ／比較モードレジスタ 1（TIMx_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]			IC2PSC[1:0]		IC1F[3:0]			IC1PSC[1:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

出力比較モード：

ビット 15 **OC2CE**：出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]**：出力比較 2 モード

ビット 11 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 10 **OC2FE**：出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]**：キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC2 チャンネルは出力として設定されます。

01：CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10：CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11：CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC2S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC2E=0）のときにのみ書き込み可能です。

ビット 7 **OC1CE**：出力比較 1 クリアイネーブル

OC1CE：出力比較 1 クリアイネーブル

0：OC1Ref は ETRF 入力の影響を受けません。

1：OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

- 注 :**
- 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。
 - 2 : PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。
 - 3 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとります。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

- 注 :**
- 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。
 - 2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャネルは出力として設定されます。

01 : CC1 チャネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャネルは出力として設定されます。

01 : CC2 チャネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャネルがオフ (TIMx_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK_INT}$, $N = 2$

0010 : $f_{SAMPLING} = f_{CK_INT}$, $N = 4$

0011 : $f_{SAMPLING} = f_{CK_INT}$, $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$, $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$, $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$, $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$, $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$, $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$, $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$, $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$, $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$, $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$, $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$, $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$, $N = 8$

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。
プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。
00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。
01 : キャプチャは、2 イベントごとに行われます。
10 : キャプチャは、4 イベントごとに行われます。
11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向 (入力 / 出力) と、使用される入力を定義します。
00 : CC1 チャンネルは出力として設定されます。
01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。
10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。
11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

16.4.8 TIM1 および TIM8 のキャプチャ / 比較モードレジスタ 2 (TIMx_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ / 比較 4 選択

このビットフィールドは、チャンネルの方向 (入力 / 出力) と、使用される入力を定義します。
00 : CC4 チャンネルは出力として設定されます。
01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード



ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： **CC3S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。**

入力キャプチャモード

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケール

ビット 9:8 **CC4S** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： **CC4S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。**

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケール

ビット 1:0 **CC3S** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： **CC3S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。**

16.4.9 TIM1 および TIM8 のキャプチャ／比較有効レジスタ（TIMx_CCER）

アドレスオフセット：0x20

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P**：キャプチャ／比較 4 出力極性
CC1P の説明を参照してください。

ビット 12 **CC4E**：キャプチャ／比較 4 出力イネーブル
CC1E の説明を参照してください。

ビット 11 **CC3NP**：キャプチャ／比較 3 相補出力極性
CC1NP の説明を参照してください。

ビット 10 **CC3NE**：キャプチャ／比較 3 相補出力イネーブル
CC1NE の説明を参照してください。

ビット 9 **CC3P**：キャプチャ／比較 3 出力極性
CC1P の説明を参照してください。

ビット 8 **CC3E**：キャプチャ／比較 3 出力イネーブル
CC1E の説明を参照してください。

ビット 7 **CC2NP**：キャプチャ／比較 2 相補出力極性
CC1NP の説明を参照してください。

ビット 6 **CC2NE**：キャプチャ／比較 2 相補出力イネーブル
CC1NE の説明を参照してください。

ビット 5 **CC2P**：キャプチャ／比較 2 出力極性
CC1P の説明を参照してください。

ビット 4 **CC2E**：キャプチャ／比較 2 出力イネーブル
CC1E の説明を参照してください。

ビット 3 **CC1NP**：キャプチャ／比較 1 相補出力極性
CC1 チャンネルが出力として設定されている場合：
0：OC1N はアクティブハイです。
1：OC1N はアクティブローです。
CC1 チャンネルが入力として設定されている場合：
このビットは、TI1FP1とTI2FP1の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。
注： 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。
注： このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつCC1S="00"（チャンネルは出力として設定）になった直後は書き込みできません。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力イネーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

00 : 非反転/立ち上がりエッジ

この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ

この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されます (ゲートモードまたはエンコーダモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 0 **CC1E** : キャプチャ / 比較 1 出力イネーブル

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

表 92. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力
制御ビット機能

制御ビット					出力状態 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	0	0	0	出力無効（タイマによって駆動されない） OCx=0、OCx_EN=0	出力無効（タイマによって駆動されない） OCxN=0、OCxN_EN=0
				1	出力無効（タイマによって駆動されない） OCx=0、OCx_EN=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP、OCxN_EN = 1
			1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	出力無効（タイマによって駆動されない） OCxN=0、OCxN_EN=0
				1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補（OCREF ではなく） + 極性 + デッドタイム OCxN_EN=1
		1	0	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0	出力無効（タイマによって駆動されない） OCxN=CCxNP、OCxN_EN=0
				1	オフ状態（インアクティブ状態で出力有効） OCx=CCxP、OCx_EN=1	OCxREF + 極性 OCxN=OCxREF xor CCxNP、OCxN_EN=1
			1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	オフ状態（インアクティブ状態で出力有効） OCxN=CCxNP、OCxN_EN=1
				1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補（OCREF ではなく） + 極性 + デッドタイム OCxN_EN=1
0	0	X	0	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0	出力無効（タイマによって駆動されない） OCxN=CCxNP、OCxN_EN=0
				1	出力無効（タイマによって駆動されない） 非同期：OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0 クロックが存在する場合：デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
			1	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0	
				1	オフ状態（インアクティブ状態で出力有効） 非同期：OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合：デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
	1		0	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0	出力無効（タイマによって駆動されない） OCxN=CCxNP、OCxN_EN=0
				1	オフ状態（インアクティブ状態で出力有効） 非同期：OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合：デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
			1	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0	
				1	オフ状態（インアクティブ状態で出力有効） 非同期：OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合：デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	

1. チャンネルの両方の出力が使用されないとき（CCxE=CCxNE=0）、OISx、OISxN、CCxP、および CCxNP はクリアされたままでなければなりません。

注： 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OC x および OC xN チャンネルの状態と、GPIO レジスタに依存します。

16.4.10 TIM1 および TIM8 のカウンタ（TIMx_CNT）

アドレスオフセット：0x24

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

16.4.11 TIM1 および TIM8 プリスケアラ（TIMx_PSC）

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数（CK_CNT）は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます（更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます）。

16.4.12 TIM1 および TIM8 自動再ロードレジスタ（TIMx_ARR）

アドレスオフセット：0x2C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

APR の更新と動作の詳細については、[セクション 16.3.1：タイムベースユニット](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

16.4.13 TIM1 および TIM8 繰り返しカウンタレジスタ（TIMx_RCR）

アドレスオフセット：0x30

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]**：繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U_RC でのみ、REP_CNT に REP 値がロードされるので、TIMx_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) は次のことを意味します。

- エッジアラインモードでは、PWM 周期の数
- センターアラインモードでは、PWM の 1/2 周期の数

16.4.14 TIM1 および TIM8 キャプチャ／比較レジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

CC1 チャンネルが入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

16.4.15 TIM1 および TIM8 キャプチャ／比較レジスタ 2（TIMx_CCR2）

アドレスオフセット：0x38

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 CCR2[15:0]：キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合：
CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC2 出力に送信される値を含みます。
CC2 チャンネルが入力として設定されている場合：
CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

16.4.16 TIM1 および TIM8 キャプチャ／比較レジスタ 3（TIMx_CCR3）

アドレスオフセット：0x3C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 CCR3[15:0]：キャプチャ / 比較値

CC3 チャンネルが出力として設定されている場合：
CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC3 出力に送信される値を含みます。
CC3 チャンネルが入力として設定されている場合：
CCR3 は、最後の入力キャプチャ 3 イベント（IC3）によって転送されたカウンタ値です。

16.4.17 TIM1 および TIM8 キャプチャ／比較レジスタ 4（TIMx_CCR4）

アドレスオフセット：0x40

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 15:0 **CCR4[15:0]**：キャプチャ / 比較値

CC4 チャンネルが出力として設定されている場合：

CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC4 出力に送信される値を含みます。

CC4 チャンネルが入力として設定されている場合：

CCR4 は、最後の入力キャプチャ 4 イベント（IC4）によって転送されたカウンタ値です。

16.4.18 TIM1 および TIM8 ブレークおよびデッドタイムレジスタ（TIMx_BDTR）

アドレスオフセット：0x44

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

注： *ビット AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。*

ビット 15 **MOE**：メイン出力イネーブル

このビットは、ブレーク入力がアクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0：OC および OCN 出力は無効か、強制的にアイドル状態になります。

1：OC および OCN 出力は、それぞれのイネーブルビット（TIMx_CCER レジスタの CCxE、CCxNE ビット）がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください（[セクション 16.4.9：TIM1 および TIM8 のキャプチャ／比較有効レジスタ（TIMx_CCER）](#)（455 ページ））。

ビット 14 **AOE**：自動出力イネーブル

0：MOE はソフトウェアによってのみセットできます。

1：MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます（ブレーク入力 that アクティブでない場合）。

注： *このビットは、LOCK レベル 1 がプログラムされている場合（TIMx_BDTR レジスタの LOCK ビット）、変更できません。*

ビット 13 BKP : ブレーク極性

- 0 : ブレーク入力 BRK はアクティブローです。
1 : ブレーク入力 BRK はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CSS クロック障害イベント) は無効です。
1 : ブレーク入力 (BRK および CSS クロック障害イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 16.4.9 : TIM1 および TIM8 のキャプチャ/比較有効レジスタ \(TIMx_CCER\) \(455 ページ\)](#))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります。この場合、OC/OCN イネーブル出力信号 = 1 です。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 16.4.9 : TIM1 および TIM8 のキャプチャ/比較有効レジスタ \(TIMx_CCER\) \(455 ページ\)](#))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 = TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込みできません。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMR レジスタの OCxM および OCxPE ビット) が書き込みできません。

注 : LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間（DT）は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$ 。

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$ 。

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$ 。

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$ 。

例 : $T_{DTS}=125ns$ (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

16.4.19 TIM1 および TIM8 DMA 制御レジスタ (TIMx_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			r/w	r/w	r/w	r/w	r/w				r/w	r/w	r/w	r/w	r/w

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送回数（タイマは、TIMx_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を検出します）を指定します。

TIMx_DMAR アドレス)

00000 : 1 回転送

00001 : 2 回転送

00010 : 3 回転送

...

10001 : 18 回転送

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx_CR1、

00001 : TIMx_CR2、

00010 : TIMx_SMCR

...

例 : 次の転送を考えます : DBL = 7 転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

16.4.20 フル転送のための TIM1 および TIM8 DMA アドレス（TIMx_DMAR）

アドレスオフセット：0x4C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **DMAB[15:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります：

$$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL は TIMx_DCR 内で設定）。

DMA パースト機能の使用例

この例では、タイマ DMA パースト機能を使って CCRx レジスタ値を更新します（x = 2、3、4）。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

- 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
- DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
- TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
- TIMx を有効化
- DMA チャンネルを有効化注：

注： この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

16.4.21 TIM1 および TIM8 レジスタマップ

TIM1 および TIM8 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 93. TIM1 および TIM8 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		CKD [1:0]		ARPE		CMS [1:0]	DIR	OPM	URS	UDIS	CEN
	リセット値																			OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	THS	MMS[2:0]		CCDS	CCUS	Res	CCPC
0x04	TIMx_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	THS	MMS[2:0]		CCDS	CCUS	Res	CCPC
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ETP	ECE	ETPS [1:0]		ETF[3:0]		MSM		TS[2:0]		Res	SMS[2:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	Res	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC2CE	OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]	OC1CE		OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIMx_CCMR1 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]		IC2 PSC [1:0]	CC2S [1:0]	IC1F[3:0]		IC1 PSC [1:0]		CC1S [1:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	TIMx_CCMR2 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC2CE	OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]	OC3CE		OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]		IC4 PSC [1:0]	CC4S [1:0]	IC3F[3:0]		IC3 PSC [1:0]		CC3S [1:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	TIMx_CNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIMx_RCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[7:0]							
	リセット値																									0	0	0	0	0	0	0	0

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 93. TIM1 および TIM8 レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIMx_CCR1																	CCR1[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIMx_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIMx_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIMx_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSRR	OSSI	LOCK[1:0]	DT[7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
	リセット値																				0	0	0	0	0				0	0	0	0	0
0x4C	TIMx_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

17 汎用タイマ（TIM2 から TIM5）

17.1 TIM2 から TIM5 の概要

この汎用タイマは、プログラマブルなプリスケラによって駆動される 16 ビットまたは 32 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定（入力キャプチャ）や出力波形の生成（出力比較と PWM）など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

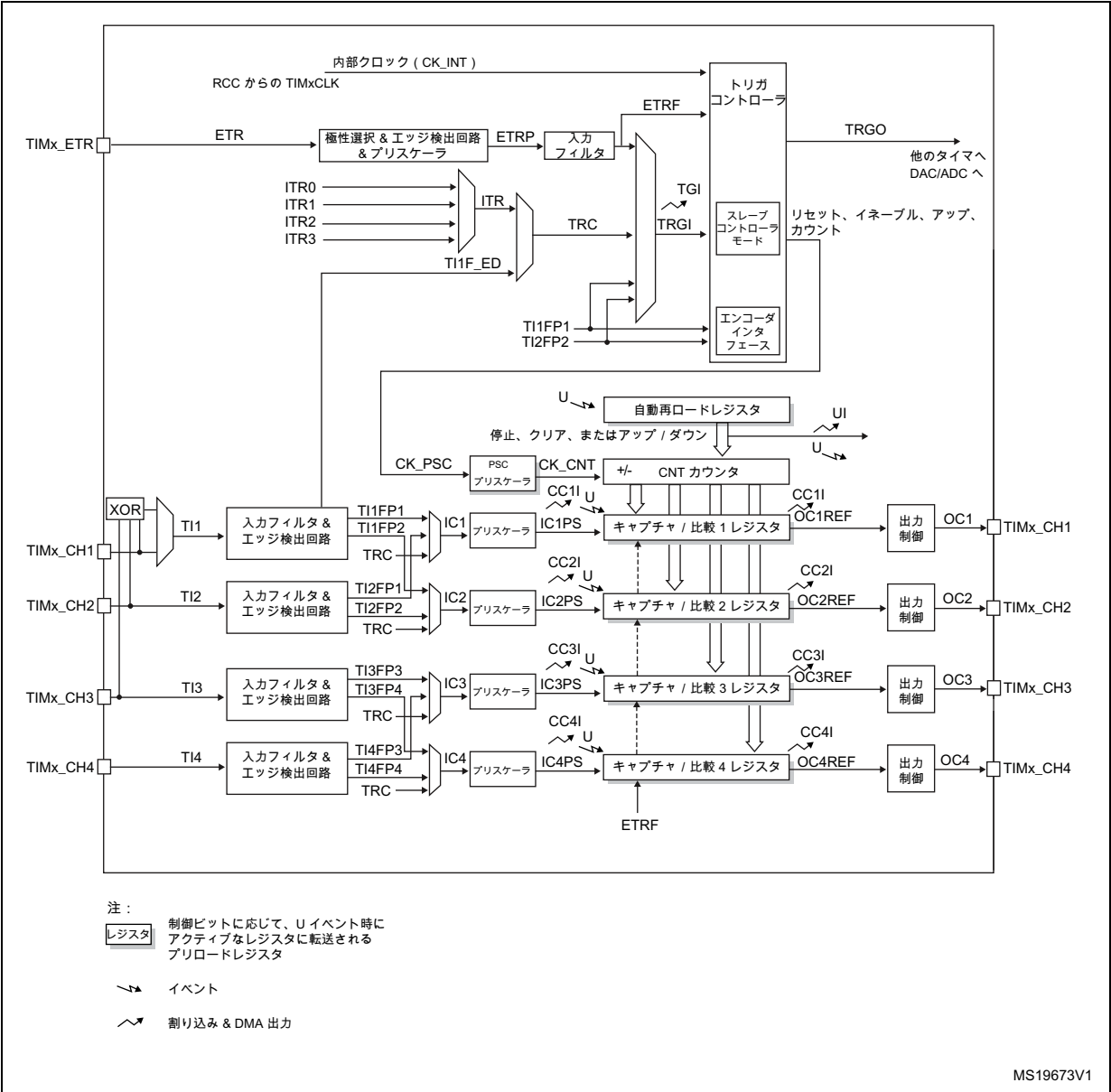
タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 17.3.15](#)に示すように、相互に同期させることができます。

17.2 TIM2 から TIM5 の主な機能

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビット（TIM3 と TIM4）または 32 ビット（TIM2 と TIM5）のアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65536 の間で分周する 16 ビットプログラム可能プリスケラ。
- 次の機能を持つ、最大 4 つの独立チャネル。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモードとセンターアラインモード）
 - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み/DMA 生成。
 - 更新：カウンタオーバーフロー/アンダーフロー、カウンタの初期化（ソフトウェアまたは内部/外部トリガによる）
 - トリガイベント（カウンタの開始、停止、初期化、または内部/外部トリガによるカウント）
 - 入力キャプチャ
 - 出力比較
- 位置決め目的のインクリメンタル（直交）エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 124. 汎用タイマのブロック図



17.3 TIM2 から TIM5 の機能説明

17.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタはカウントアップの動作が可能です。カウンタのクロックは、プリスケラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC) :
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー (またはダウンカウント時はアンダーフロー) に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウントイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

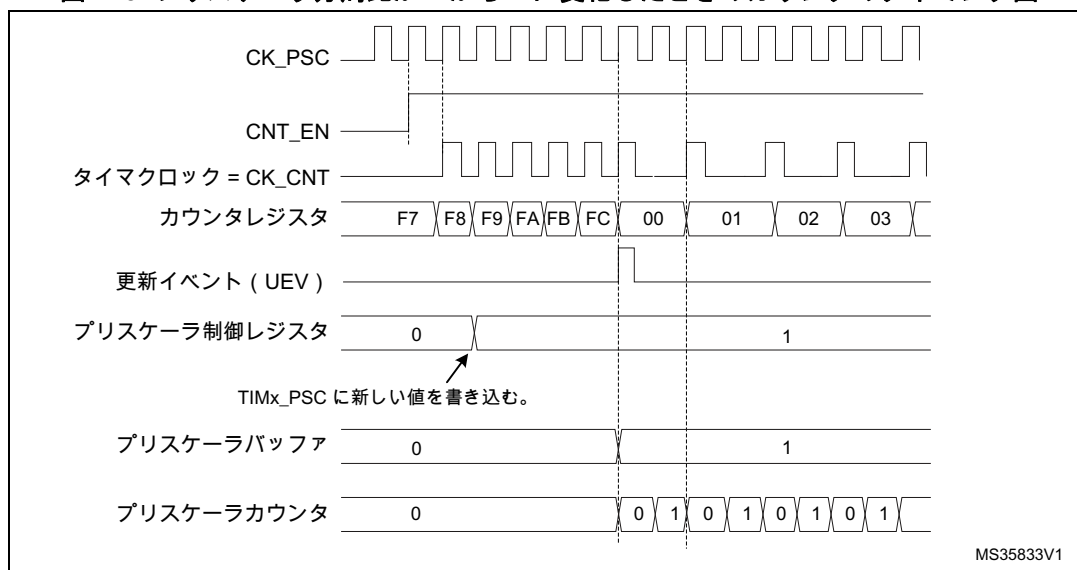
実際のカウンタイネーブル信号 CNT_EN は、CEN の 1 クロックサイクル後にセットされます。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビット/32 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

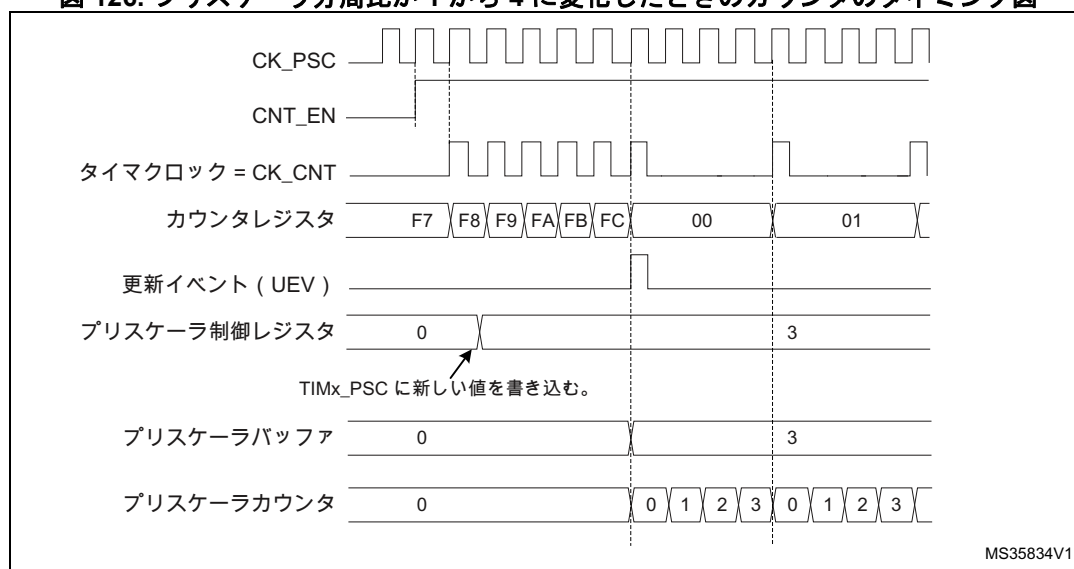
図 125 と 図 126 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 125. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図



MS35833V1

図 126. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



MS35834V1

17.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることで生成できます。

UEV イベントは、TIMx_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 127. 内部クロック分周比が 1 の場合のカウンタのタイミング図

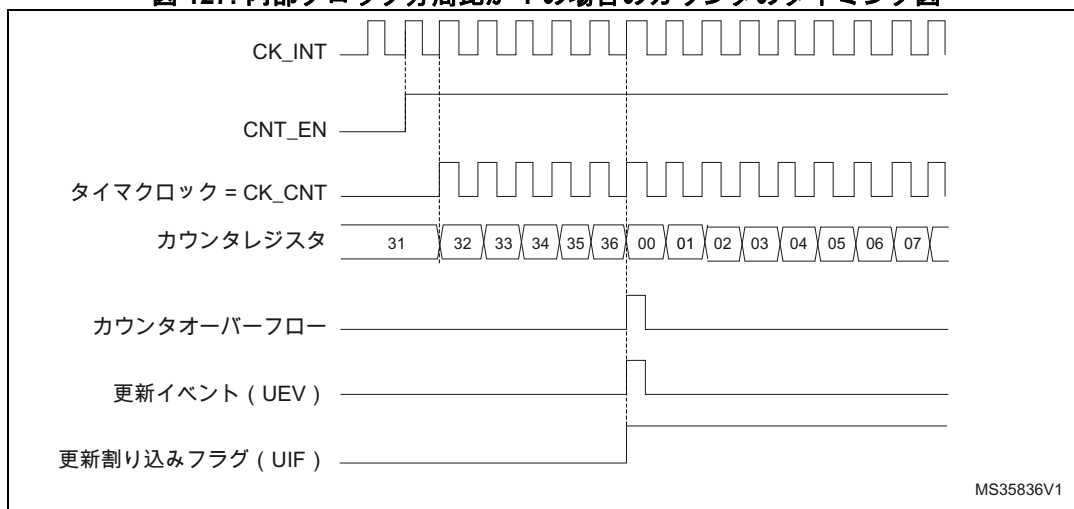


図 128. 内部クロック分周比が 2 の場合のカウンタのタイミング図

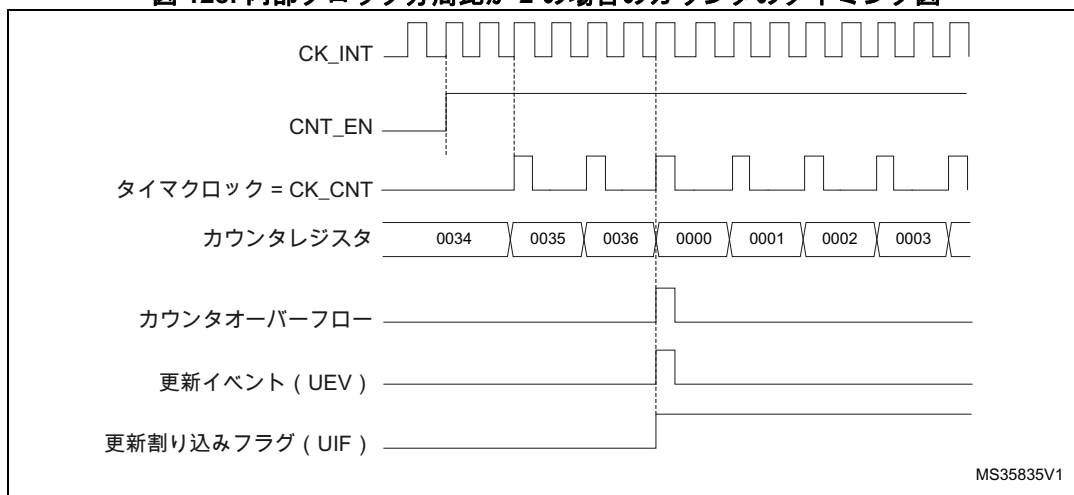


図 129. 内部クロック分周比が 4 の場合のカウンタのタイミング図

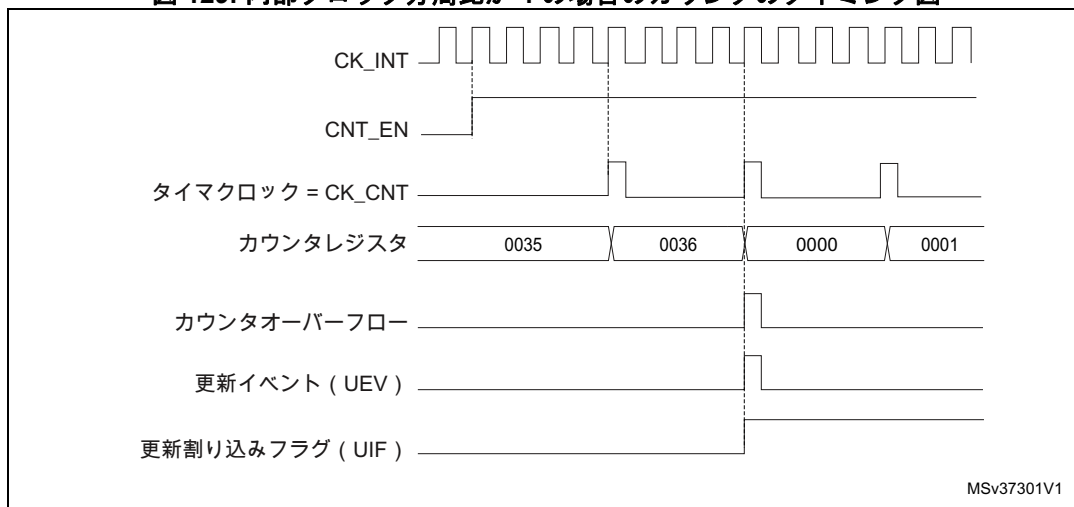


図 130. 内部クロック分周比が N の場合のカウンタのタイミング図

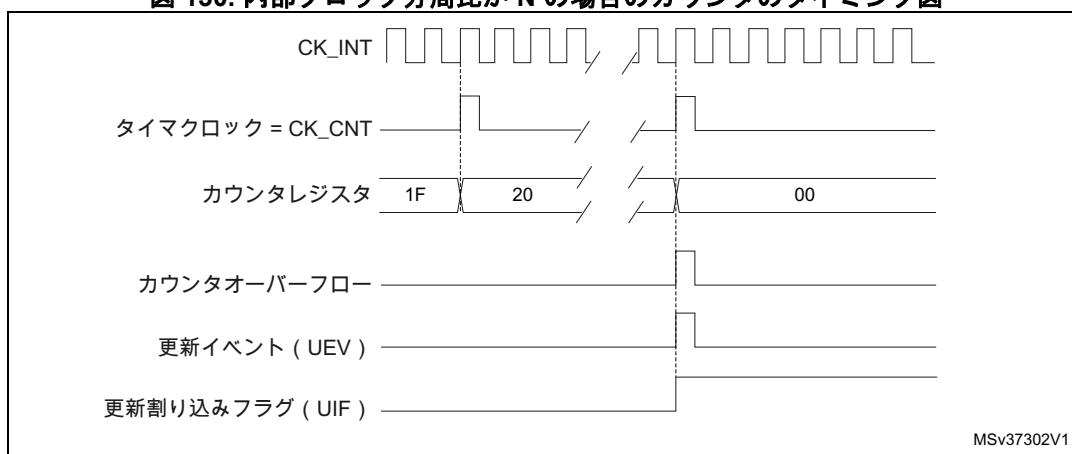


図 131. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

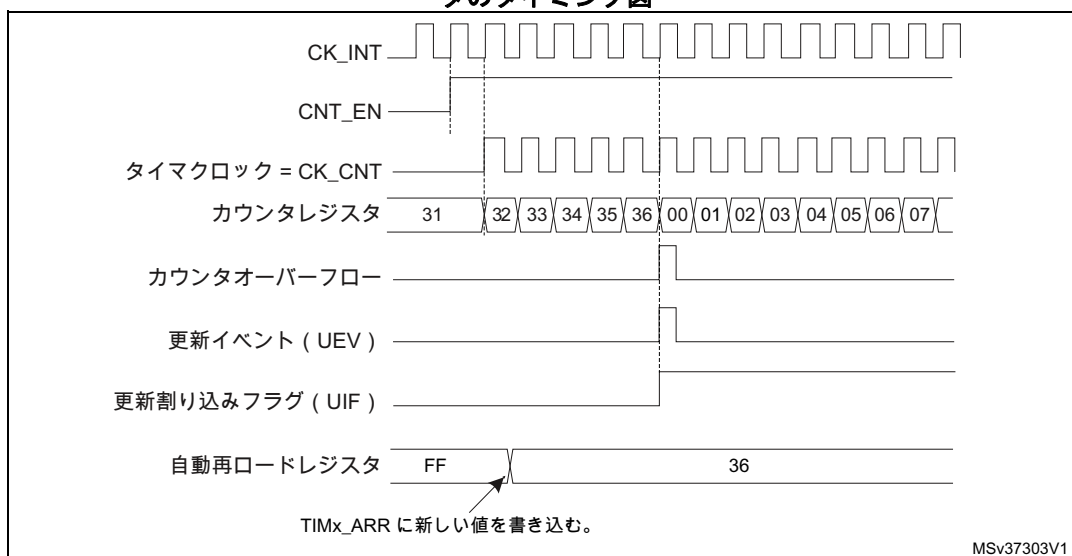
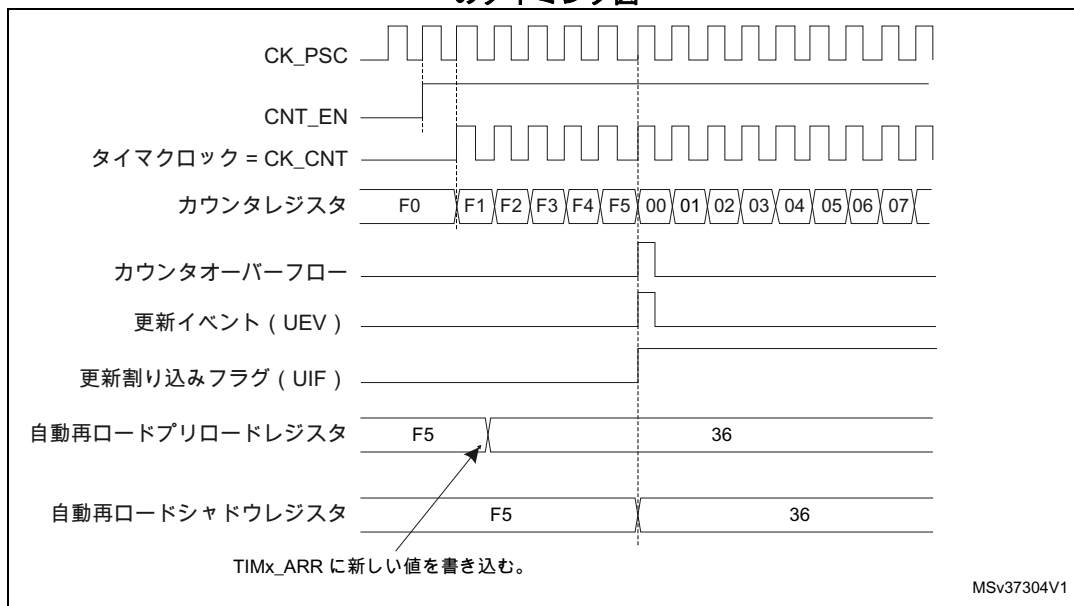


図 132. ARPE=1 (TIMx_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

更新イベントは、カウンタアンダーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることにより生成できます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 133. 内部クロック分周比が 1 の場合のカウンタのタイミング図

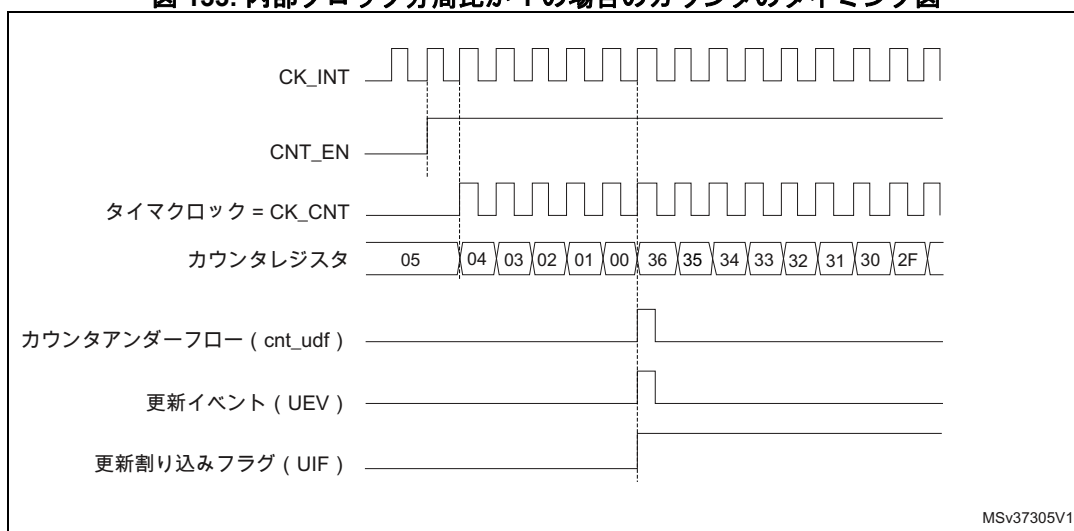


図 134. 内部クロック分周比が 2 の場合のカウンタのタイミング図

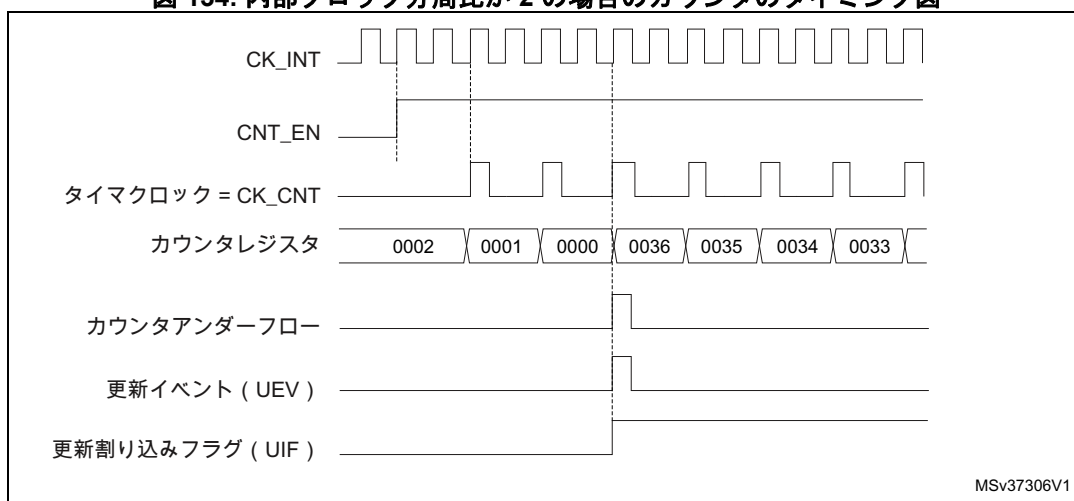


図 135. 内部クロック分周比が 4 の場合のカウンタのタイミング図

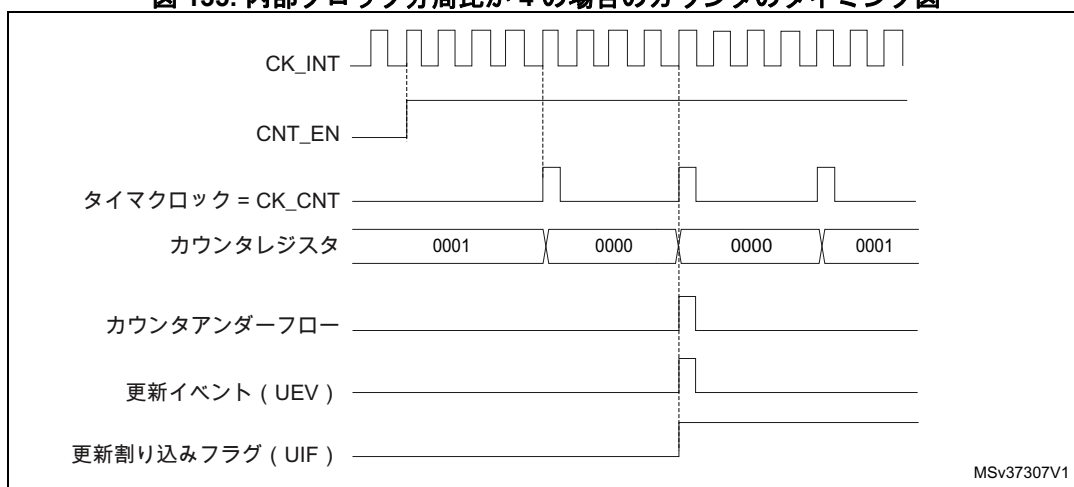


図 136. 内部クロック分周比が N の場合のカウンタのタイミング図

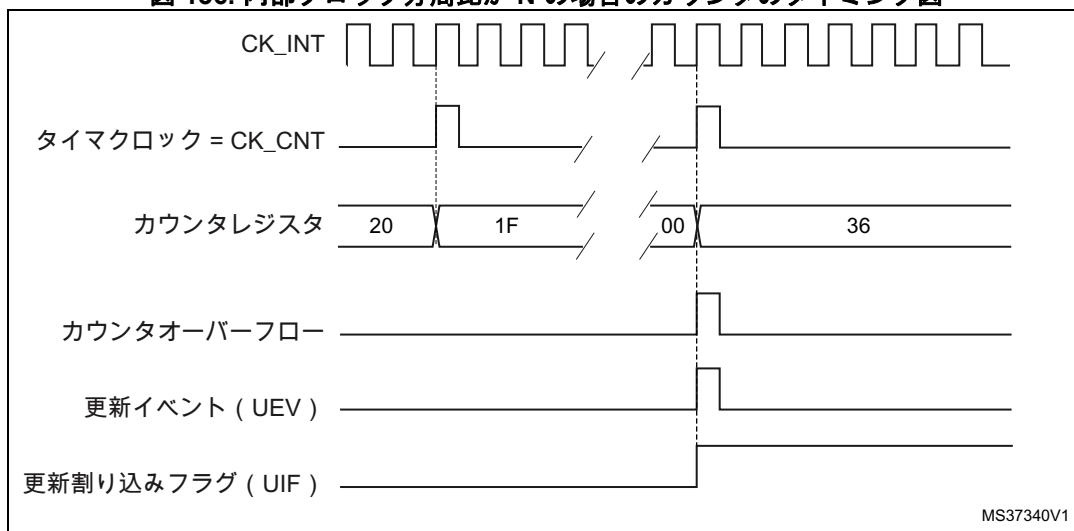
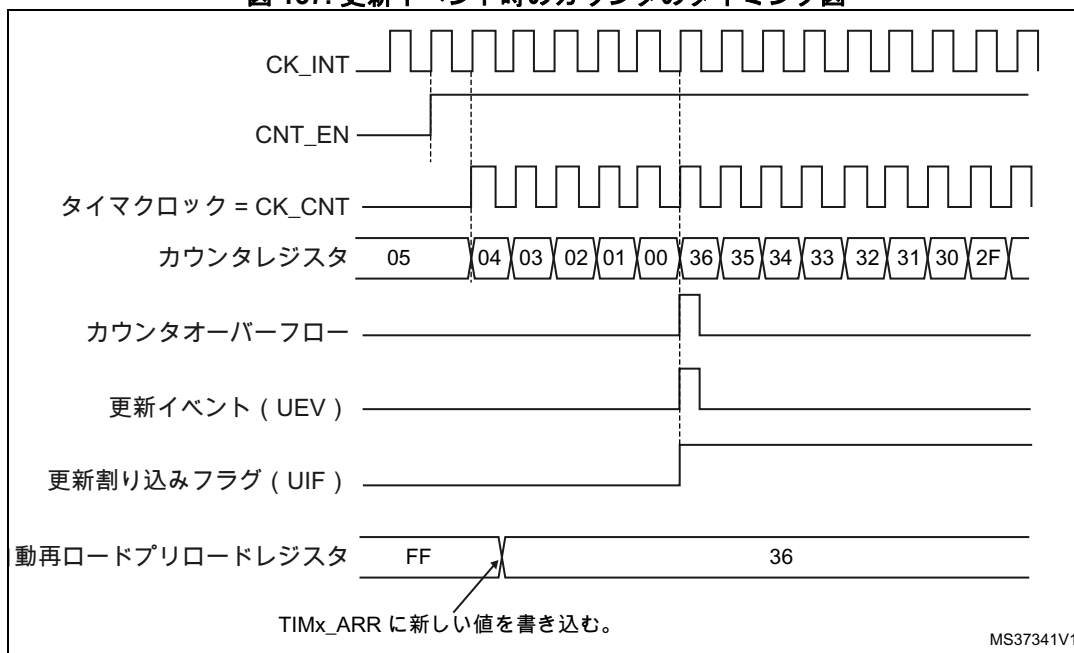


図 137. 更新イベント時のカウンタのタイミング図



センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが "00" に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、方向ビット (TIMx_CR1 レジスタの DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

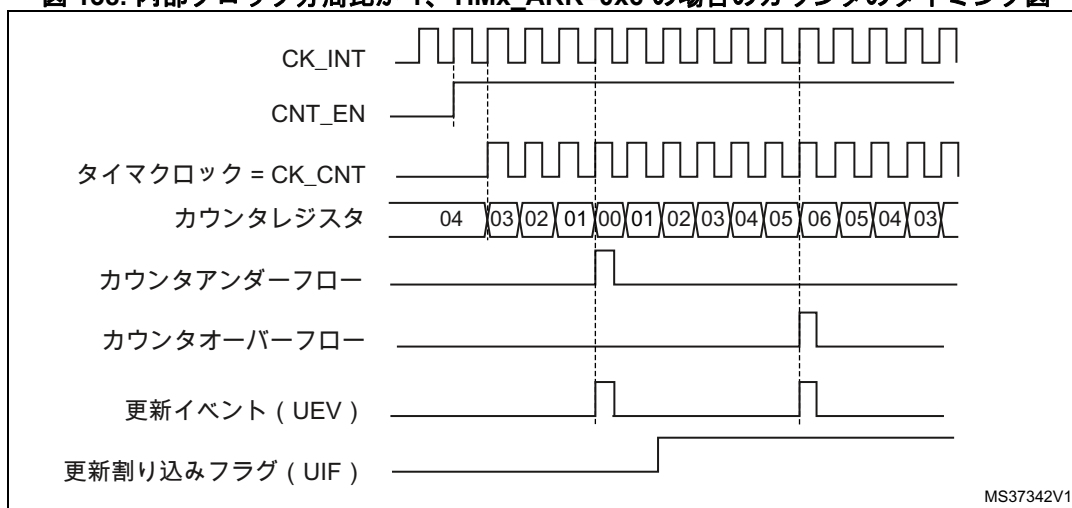
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 138. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 17.4.1 : TIMx 制御レジスタ 1 \(TIMx_CR1\)](#) (505 ページ) を参照してください)。

図 139. 内部クロック分周比が 2 の場合のカウンタのタイミング図

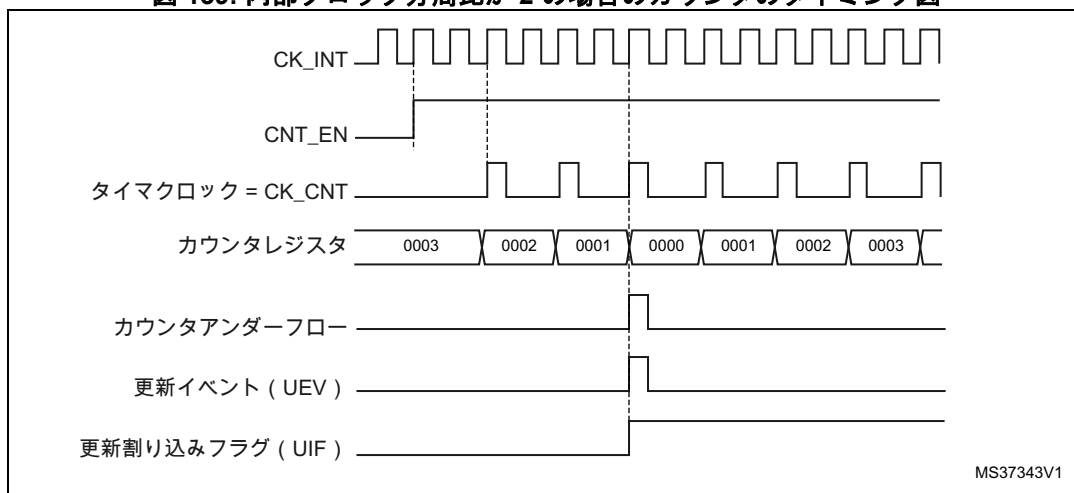
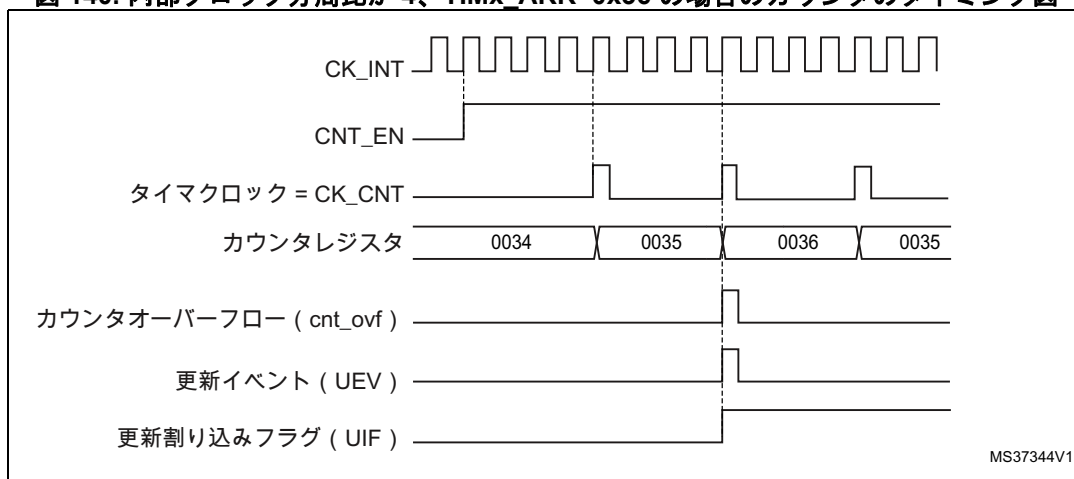


図 140. 内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 141. 内部クロック分周比が N の場合のカウンタのタイミング図

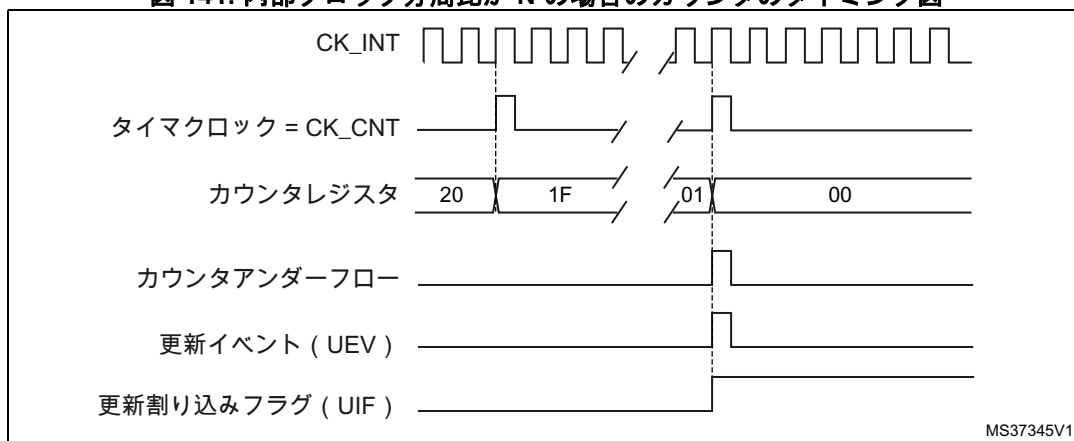


図 142. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図

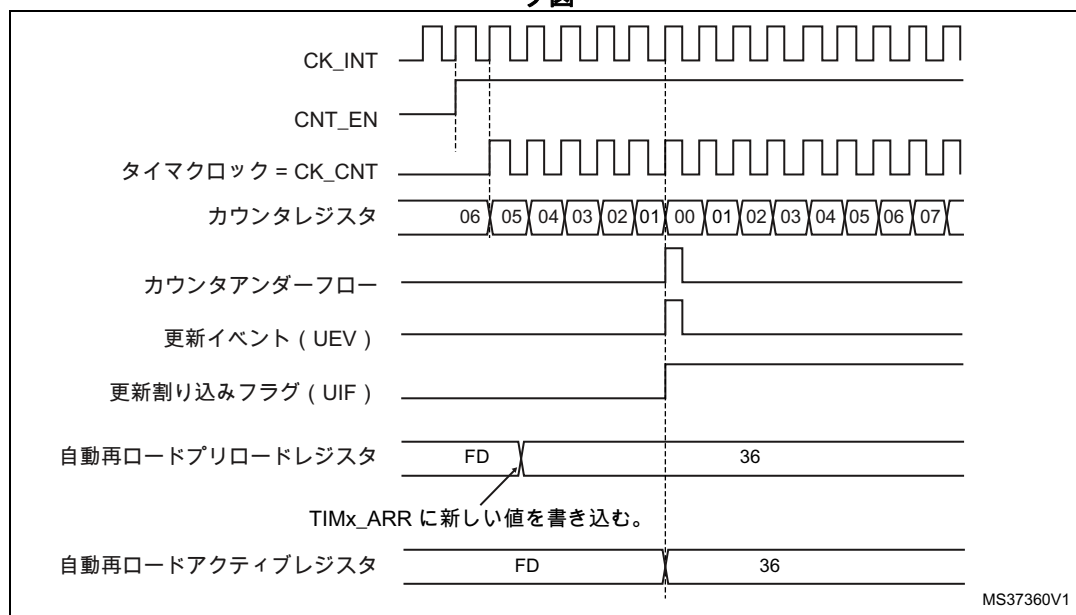
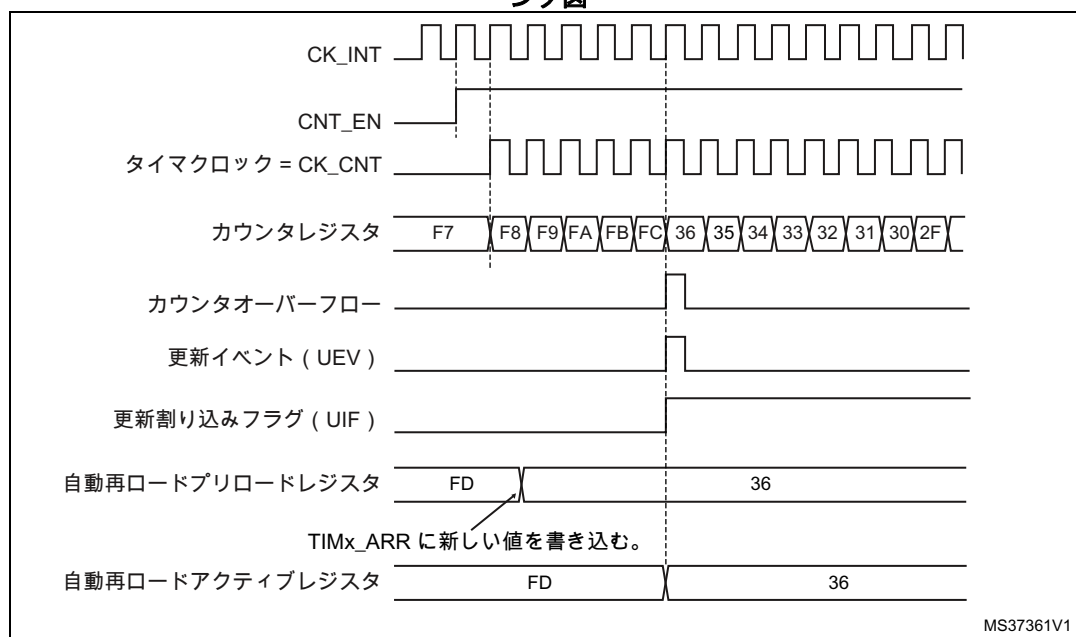


図 143. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



17.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

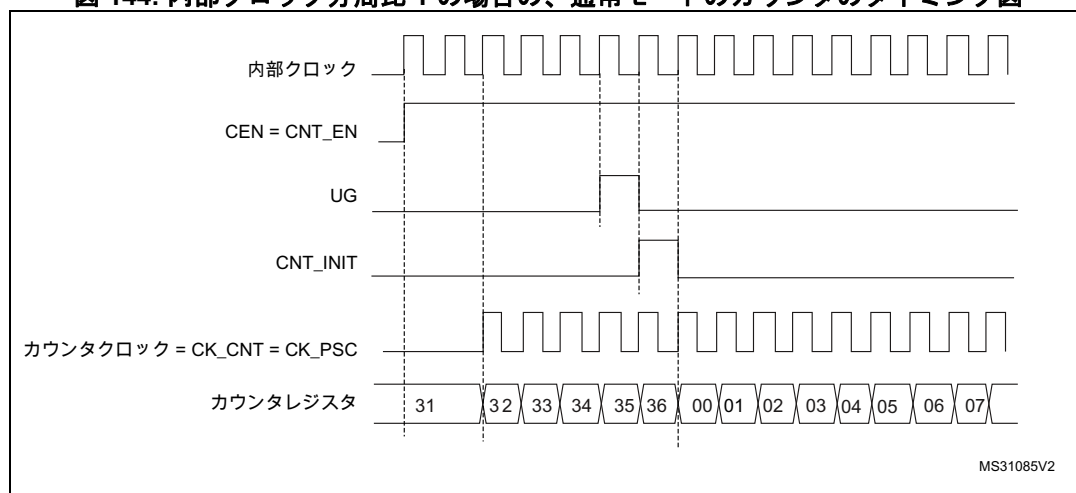
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン (Tix)
- 外部クロックモード 2: 外部トリガ入力 (ETR) は、TIM2、TIM3、TIM4 のみで利用可能。
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマがタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

内部クロックソース (CK_INT)

スレーブモードコントローラが無効の場合 (TIMx_SMCR レジスタの SMS=000)、CEN、DIR ビット (TIMx_CR1 レジスタ) と UG ビット (TIMx_EGR レジスタ) が実際の制御ビットであり、ソフトウェアでのみ変更できます (自動的にクリアされたままの UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 144](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

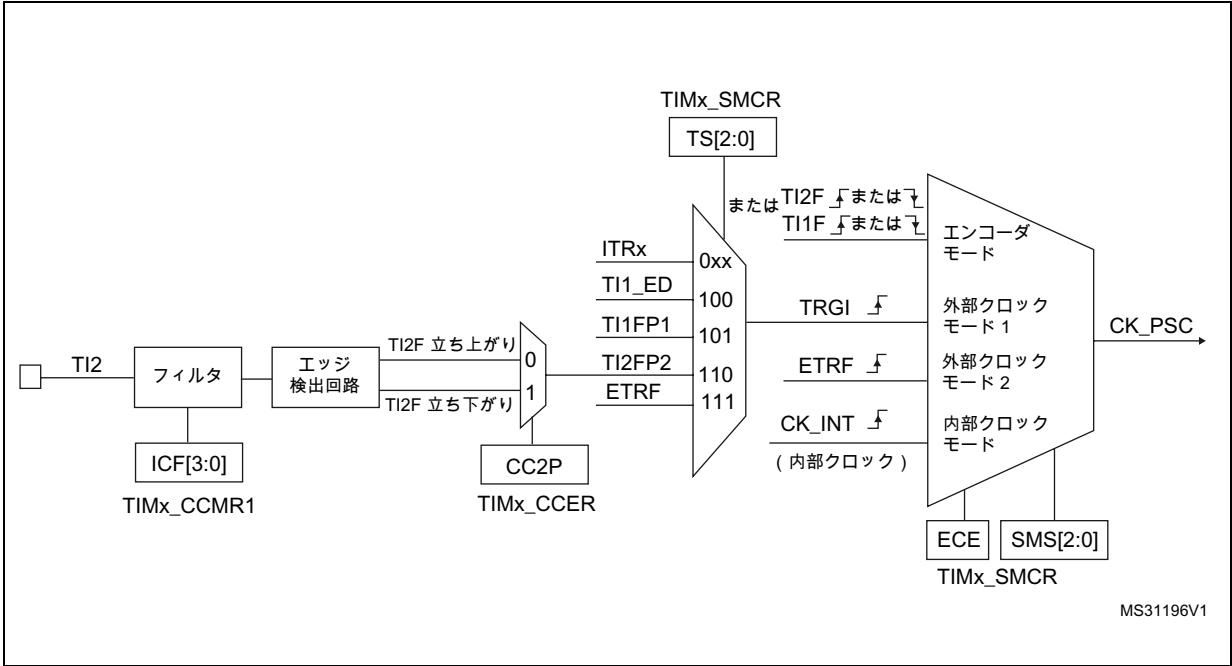
図 144. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 145. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの ICF[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。

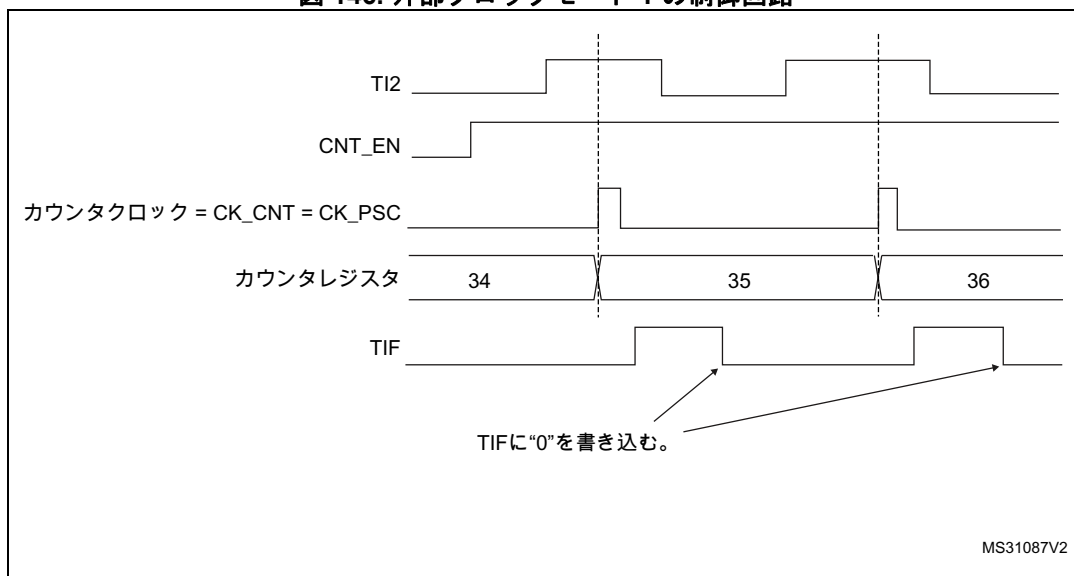
注： キャプチャプリスケアラはトリガには使用されないで、設定は不要です。

3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 146. 外部クロックモード 1 の制御回路



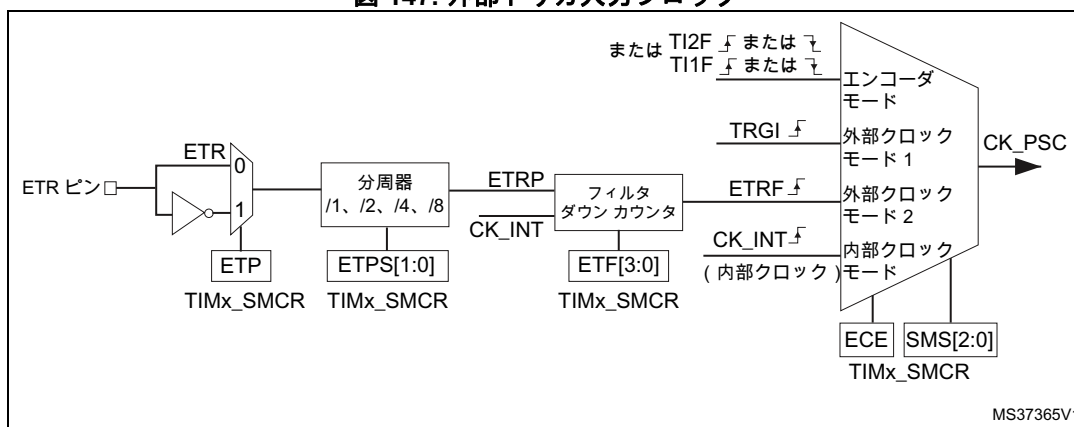
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 147 に、外部トリガ入力ブロックの概要を示します。

図 147. 外部トリガ入力ブロック



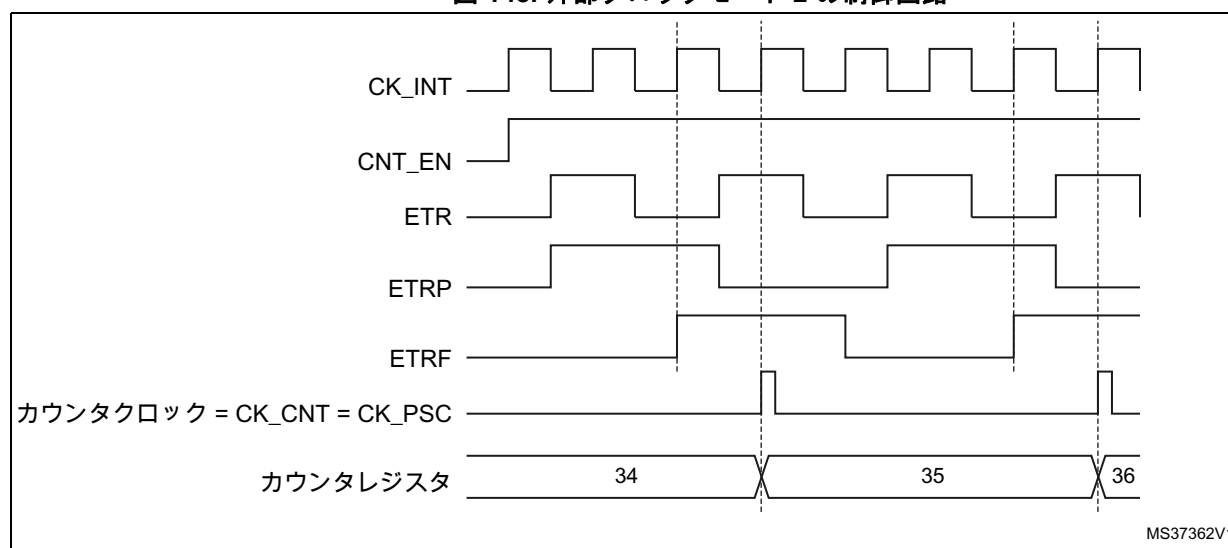
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 148. 外部クロックモード 2 の制御回路



17.3.4 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケラ）、および出力ステージ（比較回路と出力制御）から構成されています。

次の図に、キャプチャ／比較チャネルの概要を示します。

入力ステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 149. キャプチャ／比較チャネル（例：チャネル 1 入力ステージ）

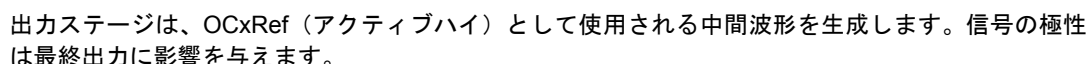
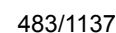
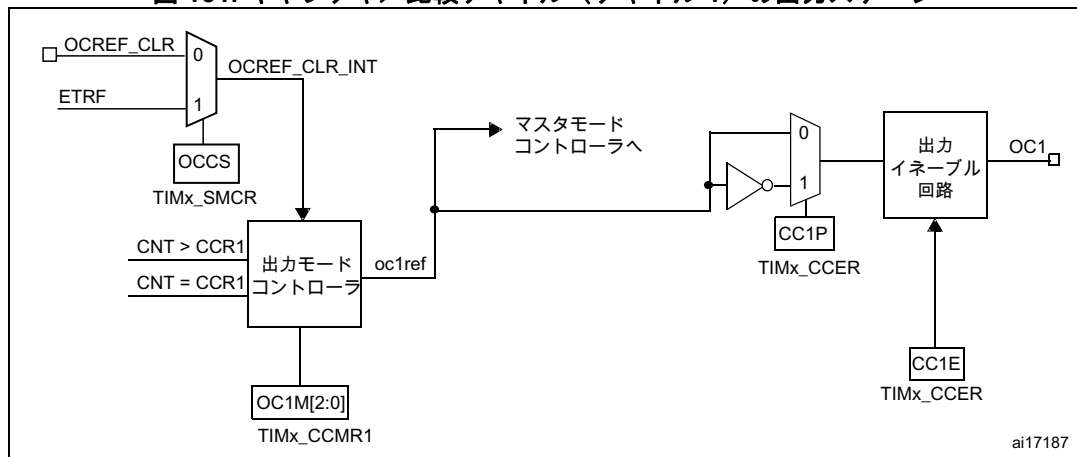


図 150. キャプチャ／比較チャネル 1 メイン回路



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などにつきましては如何なる責任にも負いません。

図 151. キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ／比較ブロックは、1 つのプリロードレジスタと 1 つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

17.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ／比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力が入立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならない、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いたいため、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに 00 を書き込みます)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。

- 必要な場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバーキャプチャの見落としを避けることができます。

注： IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

17.3.6 PWM 入力モード

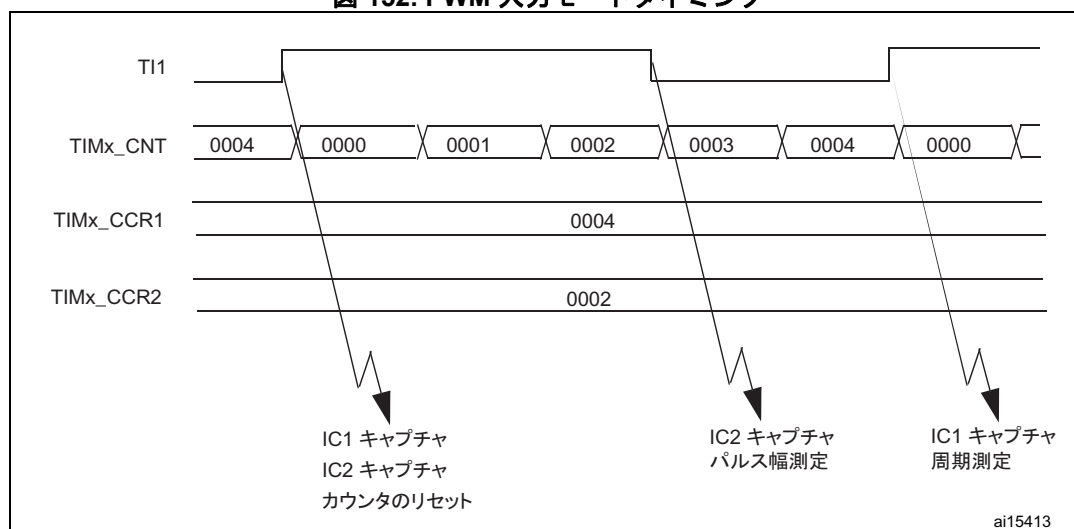
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用します)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに“0”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

図 152. PWM 入力モードタイミング



17.3.7 強制出力モード

このモード (TIMx_CCMRx レジスタの CCxS=00) では、各出力比較信号 (OCxREF、そして OCx) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx_OCMRx レジスタの OCxM ビットに 101 を書き込みます。これにより、OCxREF は強制的にハイレベルになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

TIMx_CCMRx レジスタの OCxM ビットに 100 を書き込むことによって、OCxREF 信号を強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、出力比較モードのセクションで説明します。

17.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

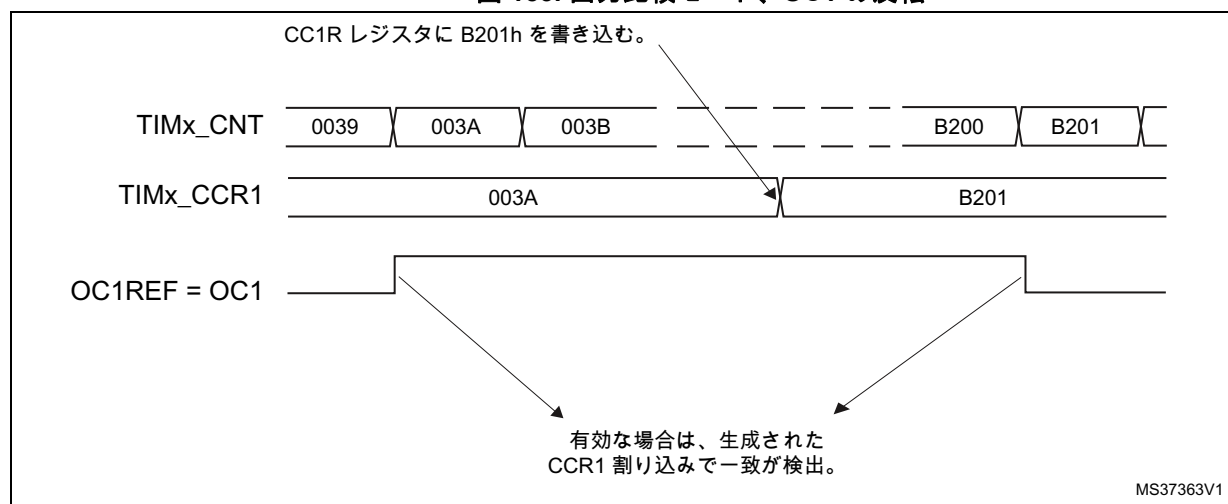
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順：

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
4. 出力モードを選択します。たとえば、CNT が CCRx と一致したときに OCx 出力をトグルし、CCRx プリロードを使用せず、OCx が有効でアクティブハイのときには、OCxM=011、OCxPE=0、CCxP=0、CCxE=1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアで TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 153](#) に示します。

図 153. 出力比較モード、OC1 の反転



17.3.9 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、個々のチャンネル (OCx 出力ごとに PWM 1 波形) で、TIMx_CCMRx レジスタの OCxM ビットに "110" (PWM モード 1) や "111" (PWM モード 2) を書き込むことで、独自に選択できます。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx かどうか判断されます (カウントの方向によります)。ただし、ETRF (OCREF は、次の PWM 周期までは ETR 信号を通じて外部イベントによってクリアできる) に従って、OCREF 信号は次の場合にのみアサートされます。

- 比較結果が変化するとき、または
- 出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。

タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

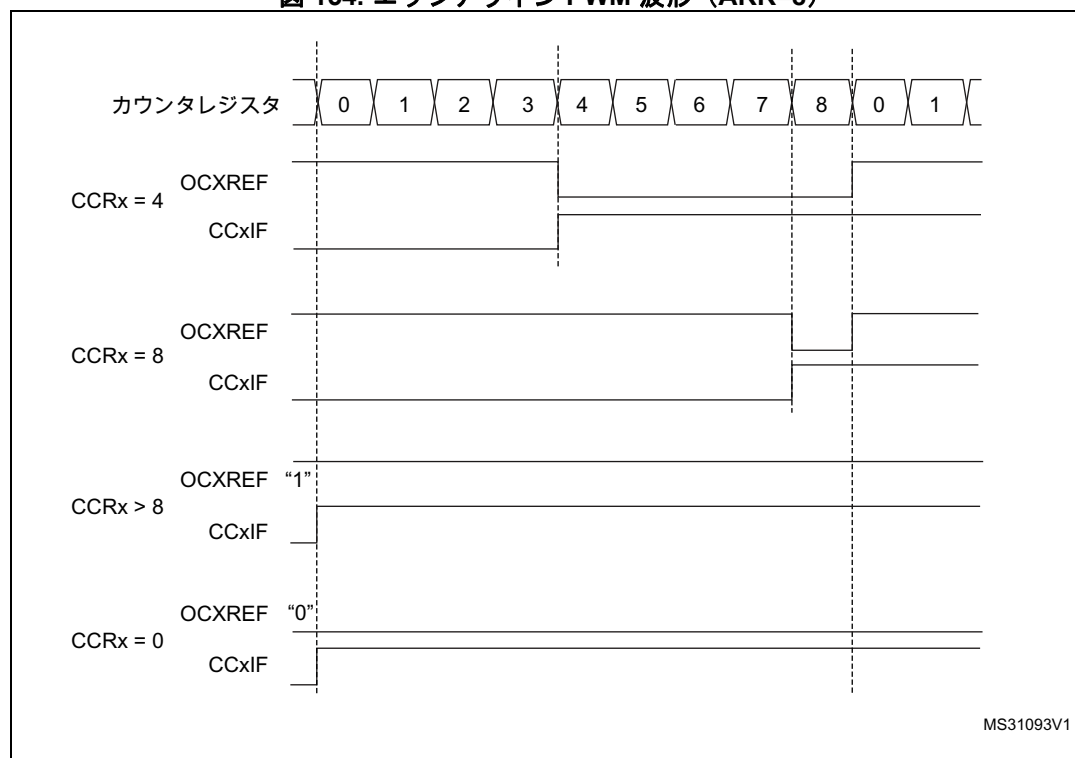
PWM エッジアラインモード

アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(470 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”です。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 154](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 154. エッジアライン PWM 波形 (ARR=8)




ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(473 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

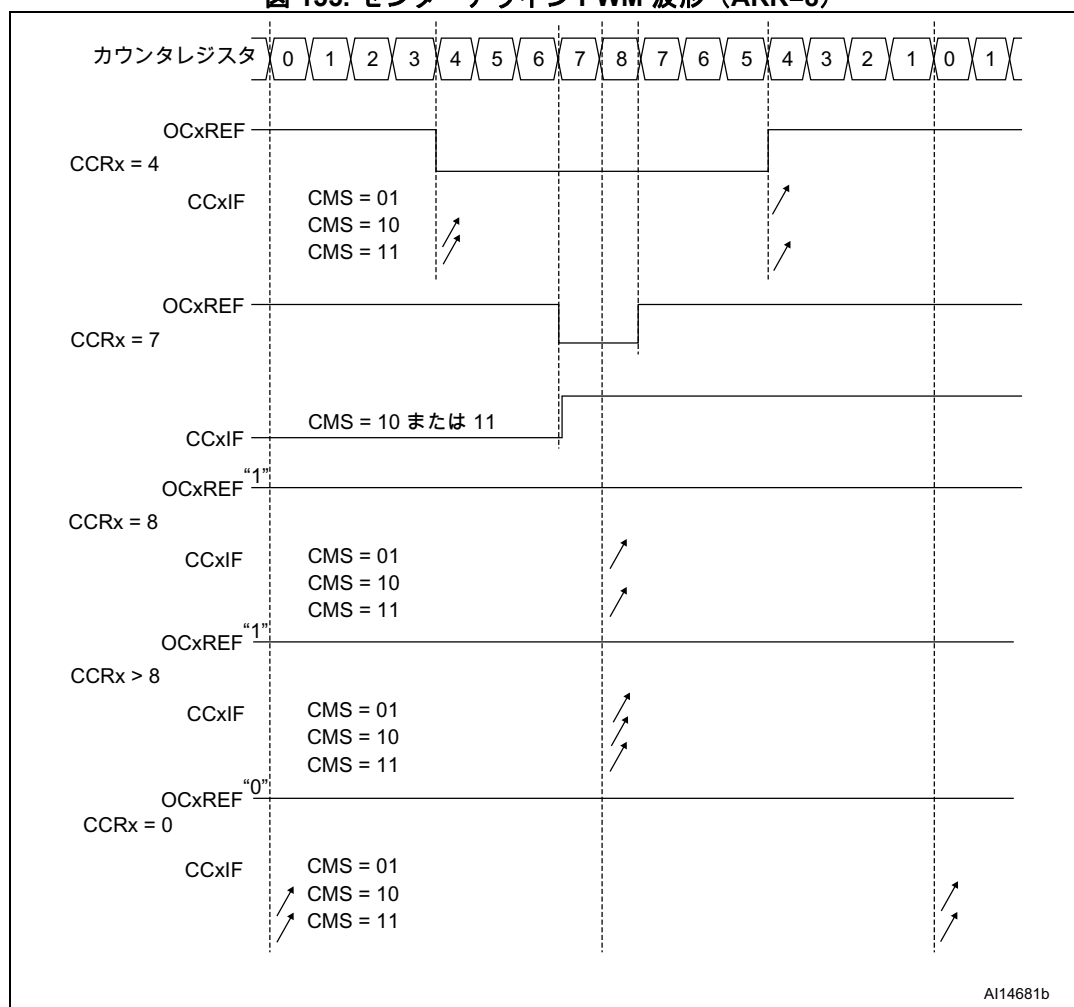
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての設定は、OCxRef/OCx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット（DIR）はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード（アップ／ダウンカウント） \(475 ページ\)](#) を参照してください。

 図 155 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 155. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

17.3.10 ワンパルスモード

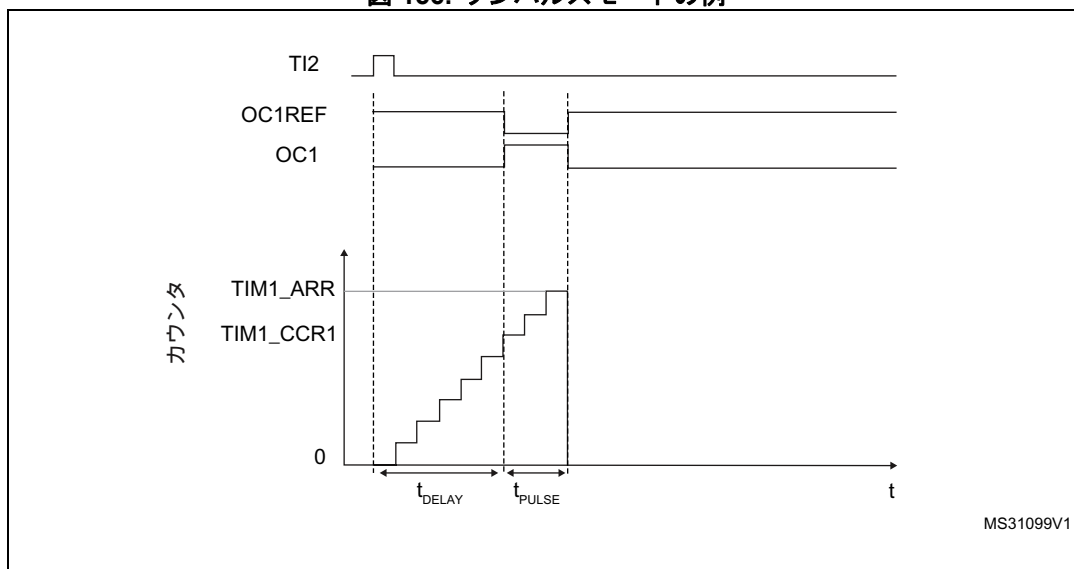
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 156. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタに $CC2S=01$ を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで $CC2P=0$ と $CC2NP=0$ を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として構成するために、TIMx_SMCR レジスタの $TS=110$ を書き込みます。
- TI2FP2 を使用してカウンタを開始するために、TIMx_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR + 1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。オプションで、TIMx_CCMR1 レジスタの OC1PE=1 と TIMx_CR1 レジスタの ARPE=1 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース：OCx 高速イネーブル：

ワンパルスモードでは、TIMx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較を考慮せずにトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

17.3.11 外部イベントによる OCxREF 信号のクリア

指定されたチャンネルの OCxREF 信号は ETRF 入力をハイレベルとする (対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする) ことでローレベルにすることができます。OCxREF 信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは機能しません。

たとえば、ETR 信号を比較回路の出力に接続して、電流処理に使用することができます。この場合、ETR は次のように設定する必要があります。

1. 外部トリガプリスケアラをオフに保つ必要があります。すなわち、TIMx_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。


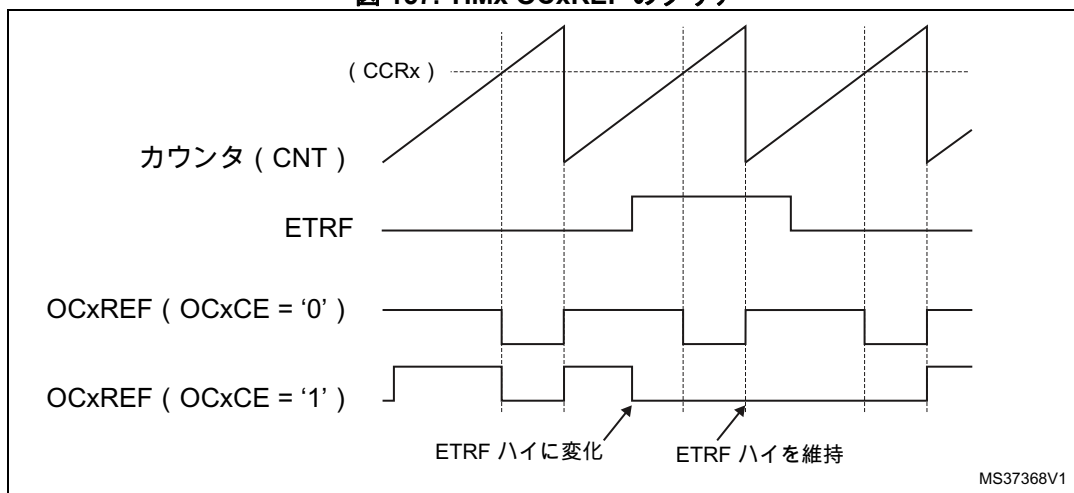
 157 に、OCxCE イネーブルビットの両方の値について、ETRF 入力が高になったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 157. TIMx OCxREF のクリア



1. 100% デューティサイクルの PWM の場合 (CCRx>ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

17.3.12 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001" を、TI1 エッジのみをカウントしている場合は SMS="010" を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011" を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。

2 つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。表 94 を参照してください。カウンタのクロックは、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択後は TI1 および TI2。フィルタされず、反転されない場合は TI1FP1=TI1、フィルタされず、反転されない場合は TI2FP2=TI2) の有効な変化によって駆動されます。ただし、カウンタ有効なことが前提となります (TIMx_CR1 レジスタの CEN ビットが "1")。2 つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 94. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	高	ダウン	アップ	カウントなし	カウントなし
	低	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	高	カウントなし	カウントなし	アップ	ダウン
	低	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の両方 をカウント	高	ダウン	アップ	アップ	ダウン
	低	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 158 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx_CCMR2 レジスタ、TI2FP2 は TI2 に配置)
- CC1P="0"、CC1NP="0"、IC1F="0000" (TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0"、CC2NP="0"、IC2F="0000" (TIMx_CCER レジスタ、TI2FP2 非反転、TI2FP2=TI2)
- SMS="011" (TIMx_SMCR レジスタ、両方の入力が立ち上がりと立ち下りの両エッジでアクティブ)
- CEN=1 (TIMx_CR1 レジスタ、カウンタ有効)

図 158. エンコーダインタフェースモードにおけるカウンタの動作例

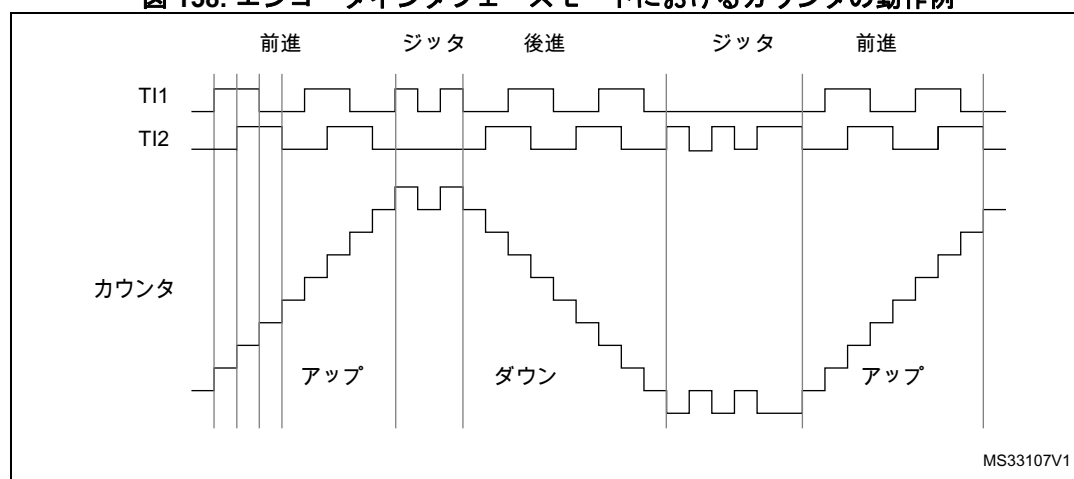
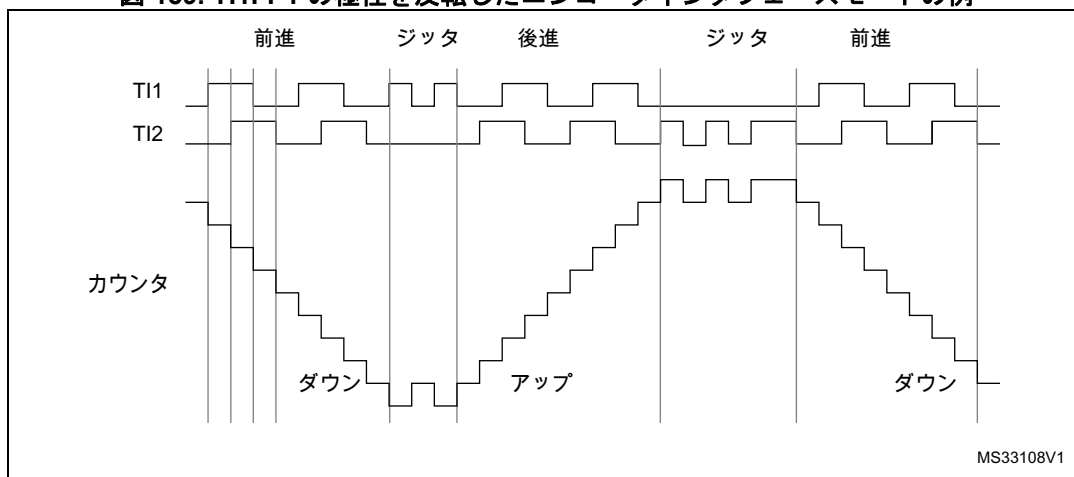


図 159 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P=1)。

図 159. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

17.3.13 タイマ入力 XOR 機能

TIM_CR2 レジスタの TI1S ビットを使用して、チャンネル 1 の入力フィルタを、TIMx_CH1 から TIMx_CH3 までの 3 つの入力ピンを組み合わせた XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

17.3.14 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

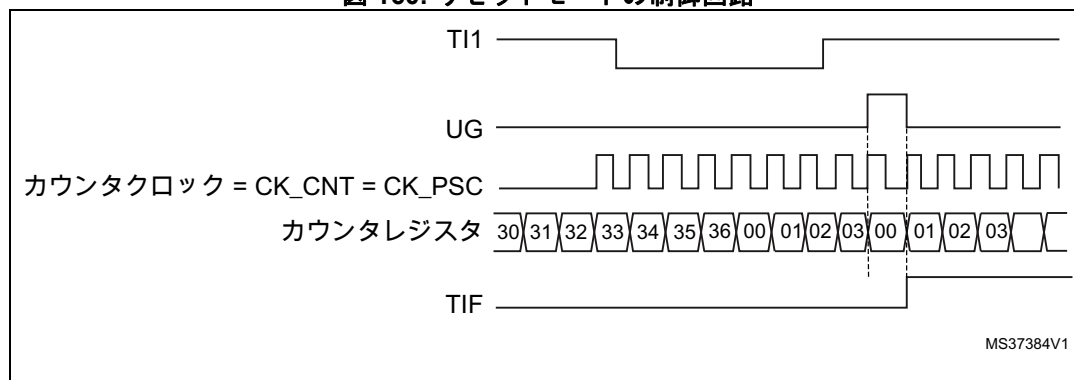
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 160. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

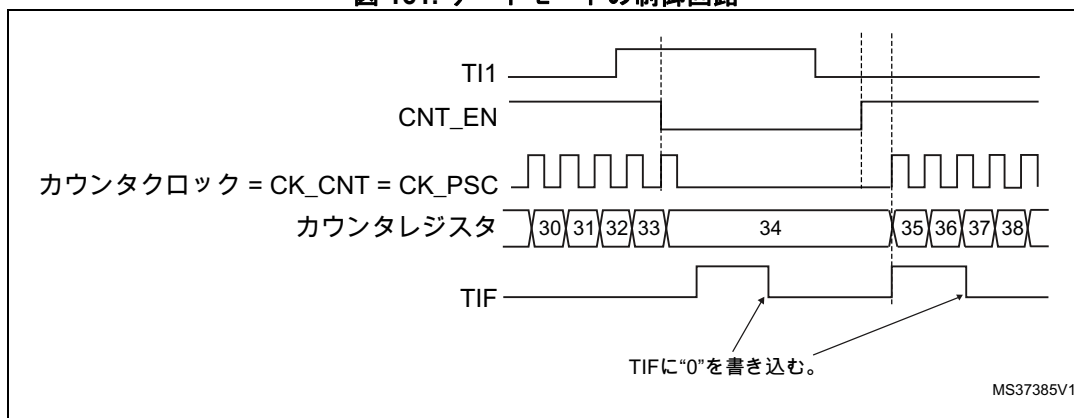
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01 ビット)。TIMx_CCER レジスタに CC1P=1 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 161. ゲートモードの制御回路



1. ゲートモードはエッジではなくレベルに対して動作するため、 $CCxP=CCxNP=1$ の設定（立ち上がりと立ち下がり両エッジの検出）はゲートモードでは意味がありません。

スレープモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

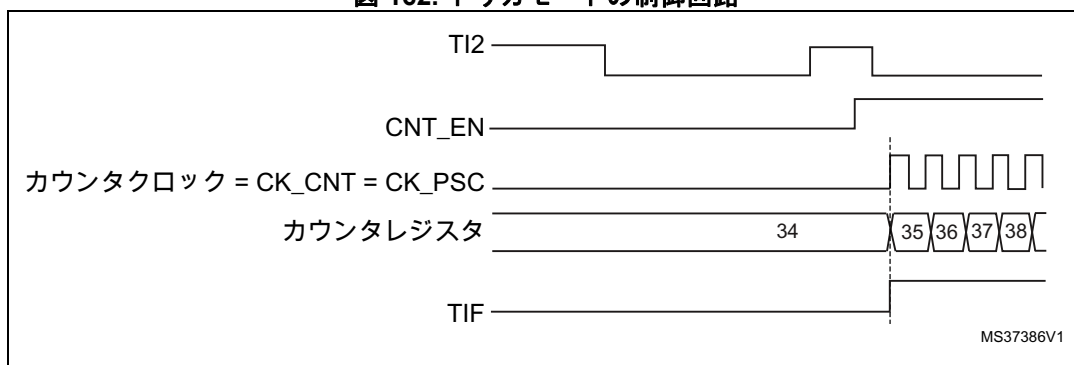
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、 $IC2F=0000$ のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの $CC2S=01$ ）。TIMx_CCER レジスタに $CC2P=1$ を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに $SMS=110$ を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに $TS=110$ を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 162. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、およびトリガモードで動作している場合）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

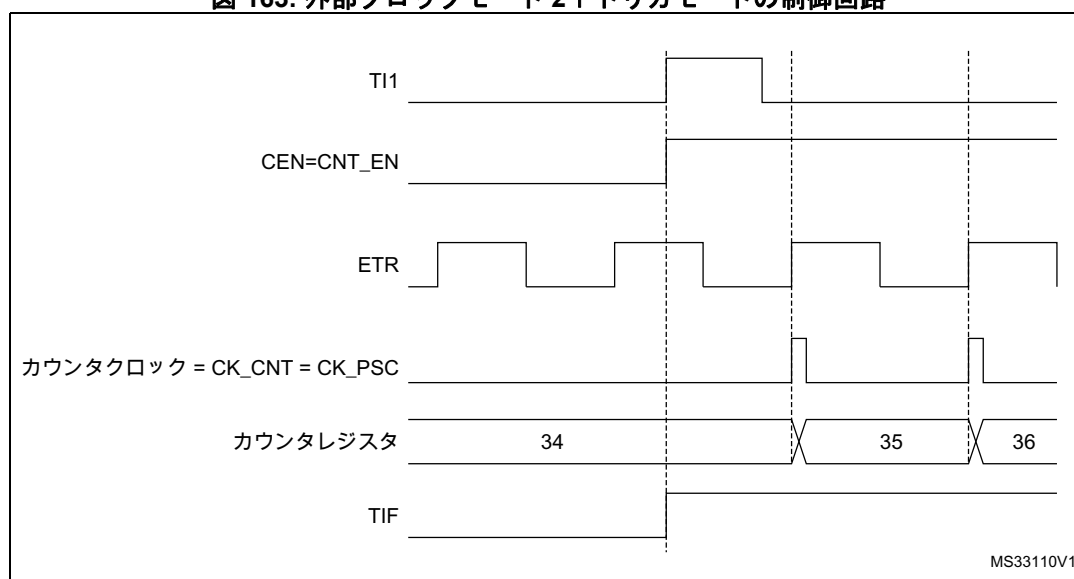
次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000：フィルタなし
 - ETPS = 00：プリスケアラは無効
 - ETP = 0：ETR の立ち上がりエッジを検出。ECE = 1 で外部クロックモード 2 を有効にする。
2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000：フィルタなし。
 - キャプチャプリスケアラはトリガには使用されないで、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタに CC1P=0 を書き込んで、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 163. 外部クロックモード 2+トリガモードの制御回路



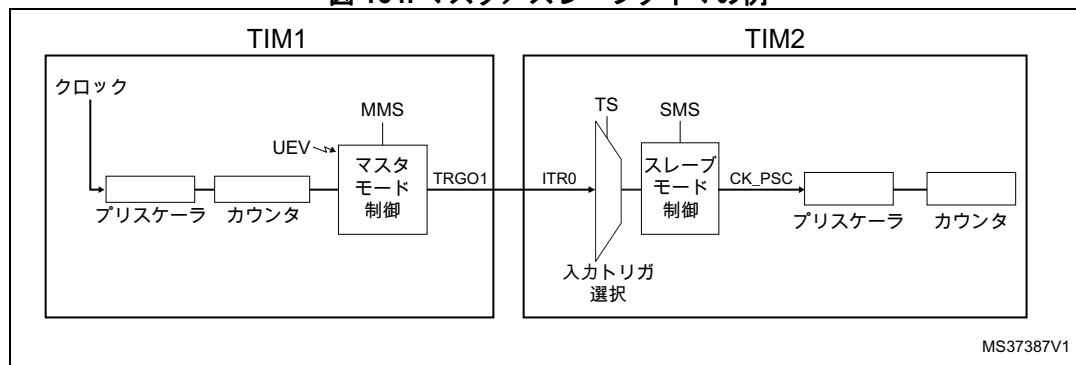
17.3.15 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。マスタモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

図 164 に、トリガ選択およびマスタモード選択ブロックの概要を示します。

タイマを別のタイマのプリスケアラとして使用する

図 164. マスタ/スレーブタイマの例



たとえば、タイマ 1 がタイマ 2 のプリスケアラとして動作するように設定できます。図 164 を参照してください。このためには、次の操作を行います。

- 更新イベント UEV ごとに定期的なトリガ信号を出力するように、タイマ 1 をマスタモードに設定します。TIM1_CR2 レジスタの MMS=010 を書き込んだ場合、更新イベントが生成されるたびに、TRGO1 で立ち上がりエッジが出力されます。
- タイマ 1 の出力 TRGO1 をタイマ 2 に接続し、タイマ 2 が内部トリガ ITR0 を使用するスレーブモードに設定します。このためには、TIM2_SMCR レジスタの TS ビットに 000 を書き込みます。
- 次に、スレーブモードコントローラを外部クロックモード 1 にします (TIM2_SMCR レジスタの SMS=111 を書き込みます)。これによりタイマ 2 は、タイマ 1 の定期的なトリガ信号の立ち上がりエッジ (タイマ 1 カウンタのオーバーフローに対応しています) をクロックとして動作します。
- 最後に、それぞれの CEN ビット (TIMx_CR1 レジスタ) をセットすることによって、両方のタイマを有効にする必要があります。

注： タイマ 1 のトリガ出力として OCx が選択された場合 (MMS=1xx)、その立ち上がりエッジがタイマ 2 カウンタのクロックとして使用されます。

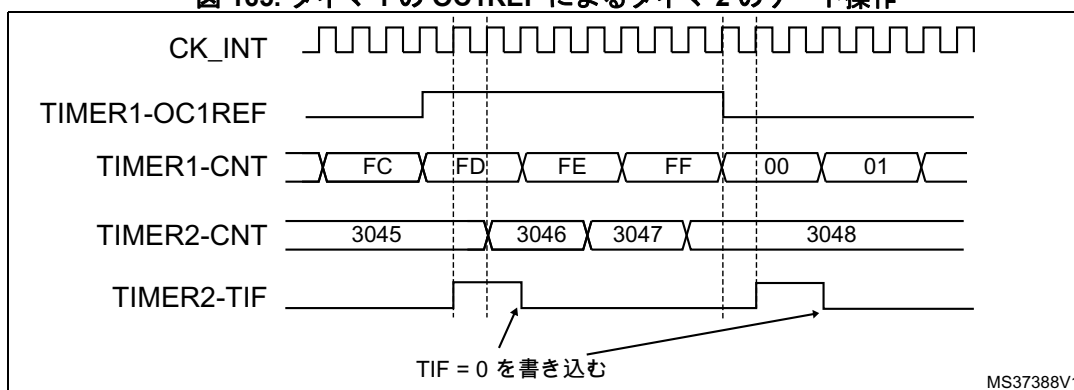
タイマを使用して別のタイマを有効にする

この例では、タイマ 1 の出力比較 1 でタイマ 2 の有効／無効を制御します。接続については、[図 164](#) を参照してください。タイマ 2 は、タイマ 1 の OC1REF がハイのときにのみ、分周された内部クロックをカウントします。両方のカウンタクロック周波数は、CK_INT をプリスケアラで 3 分周したものです ($f_{CK_CNT} = f_{CK_INT}/3$)。

- タイマ 1 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=100)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。

注： タイマ 2 のクロックはカウンタ 1 と同期しないので、このモードはタイマ 2 カウンタのイネーブル信号にのみ影響を与えます。

図 165. タイマ 1 の OC1REF によるタイマ 2 のゲート操作



MS37388V1

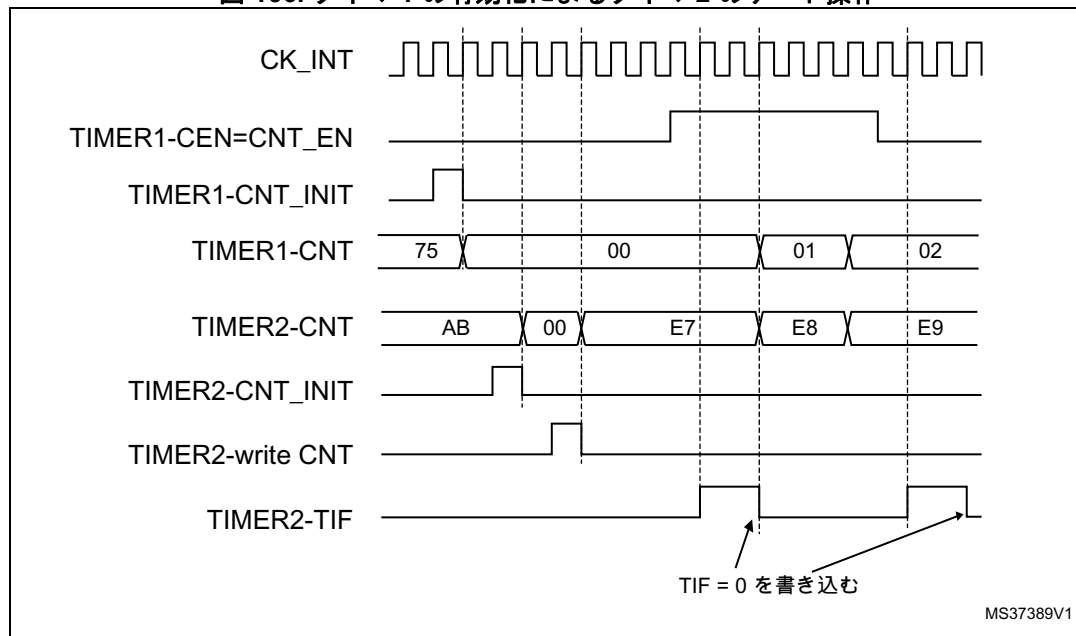
[図 165](#) の例では、タイマ 2 のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。タイマ 1 を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では、タイマ 1 とタイマ 2 を同期させます。タイマ 1 はマスタであり、0 からカウントを開始します。タイマ 2 はスレーブであり、0xE7 から開始します。プリスケアラの分周比は両方のタイマで同じです。TIM1_CR1 レジスタの CEN ビットに“0”を書き込むことによってタイマ 1 を無効にすると、タイマ 2 は停止します。

- タイマ 1 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=100)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- UG ビット (TIM1_EGR レジスタ) に“1”を書き込むことによって、タイマ 1 をリセットします。
- UG ビット (TIM2_EGR レジスタ) に“2”を書き込むことによって、タイマ 1 をリセットします。

- タイマ 2 のカウンタ (TIM2_CNT) に 0xE7 を書き込むことによって、タイマ 2 を 0xE7 に初期化します。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。
- CEN ビット (TIM1_CR1 レジスタ) に“0”を書き込むことによって、タイマ 1 を停止します。

図 166. タイマ 1 の有効化によるタイマ 2 のゲート操作

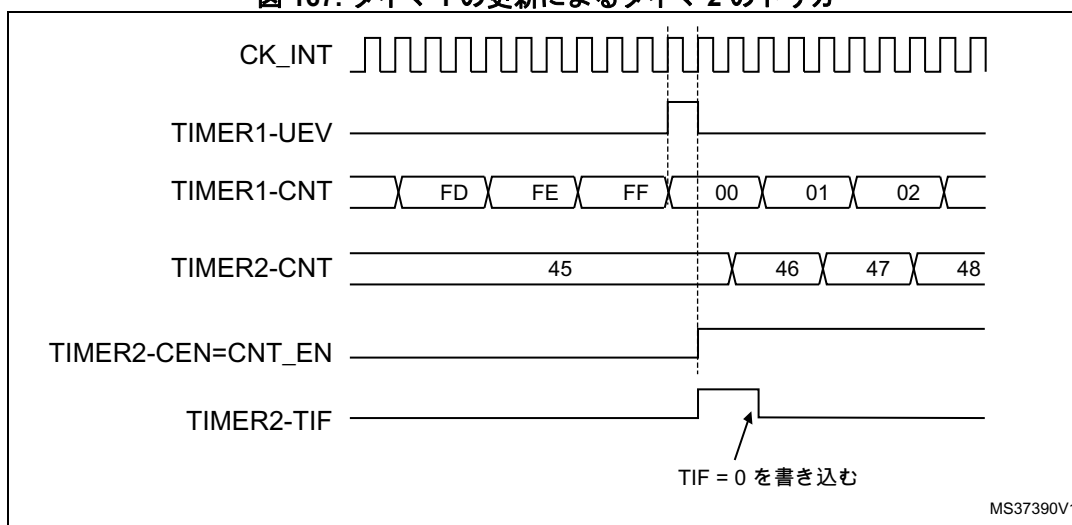


タイマを使用して別のタイマを開始する

この例では、タイマ 1 の更新イベントによってタイマ 2 の有効化を設定します。接続については、[図 164](#) を参照してください。タイマ 1 によって更新イベントが生成されると、タイマ 2 は、分周された内部クロックで現在値 (ゼロである必要はありません) からカウントを開始します。タイマ 2 がトリガ信号を受信すると、その CEN ビットが自動的にセットされ、カウンタは TIM2_CR1 レジスタの CEN ビットに“0”が書き込まれるまでカウントします。両方のカウンタクロック周波数は、CK_INT をプリスケールで 3 分周したものです ($f_{CK_CNT} = f_{CK_INT}/3$)。

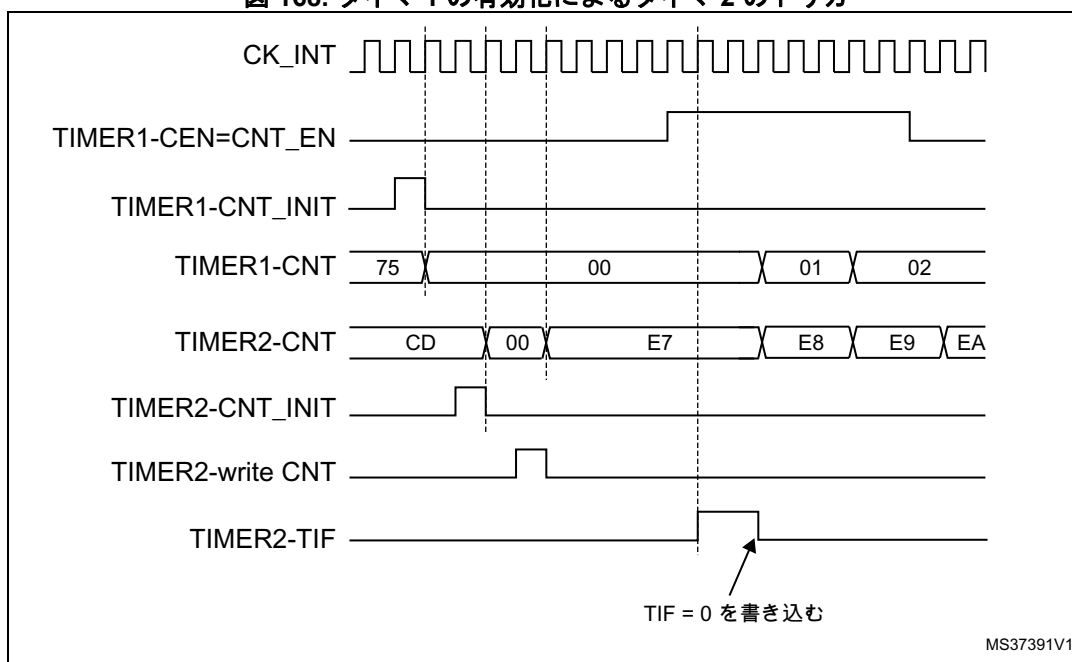
- タイマ 1 をマスタモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=010)。
- タイマ 1 の周期を設定します (TIM1_ARR レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をトリガモードに設定します (TIM2_SMCR レジスタの SMS=110)。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。

図 167. タイマ 1 の更新によるタイマ 2 のトリガ



前の例と同じように、カウントを開始する前に両方のカウンタを初期化することができます。[図 168](#) は [図 165](#) と同じ設定ですが、ゲートモードではなくトリガモードでの動作を示します (TIM2_SMCR レジスタの SMS=110)。

図 168. タイマ 1 の有効化によるタイマ 2 のトリガ



タイマを別のタイマのプリスケアラとして使用する

たとえば、タイマ 1 がタイマ 2 のプリスケアラとして動作するように設定できます。接続については、[図 164](#) を参照してください。このためには、次の操作を行います。

- タイマ 1 をマスタモードに設定して、その更新イベント（UEV）をトリガ出力として送信し（TIM1_CR2 レジスタの MMS=010）、カウンタオーバーフローごとに周期信号を出力します。
- タイマ 1 の周期を設定します（TIM1_ARR レジスタ）。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します（TIM2_SMCR レジスタの TS=000）。
- タイマ 2 を外部クロックモード 1 に設定します（TIM2_SMCR レジスタの SMS=111）。
- CEN ビット（TIM2_CR1 レジスタ）に“2”を書き込んで、タイマ 1 を開始します。
- CEN ビット（TIM1_CR1 レジスタ）に“1”を書き込んで、タイマ 1 を開始します。

外部トリガに対応して 2 つのタイマを同期して開始します。

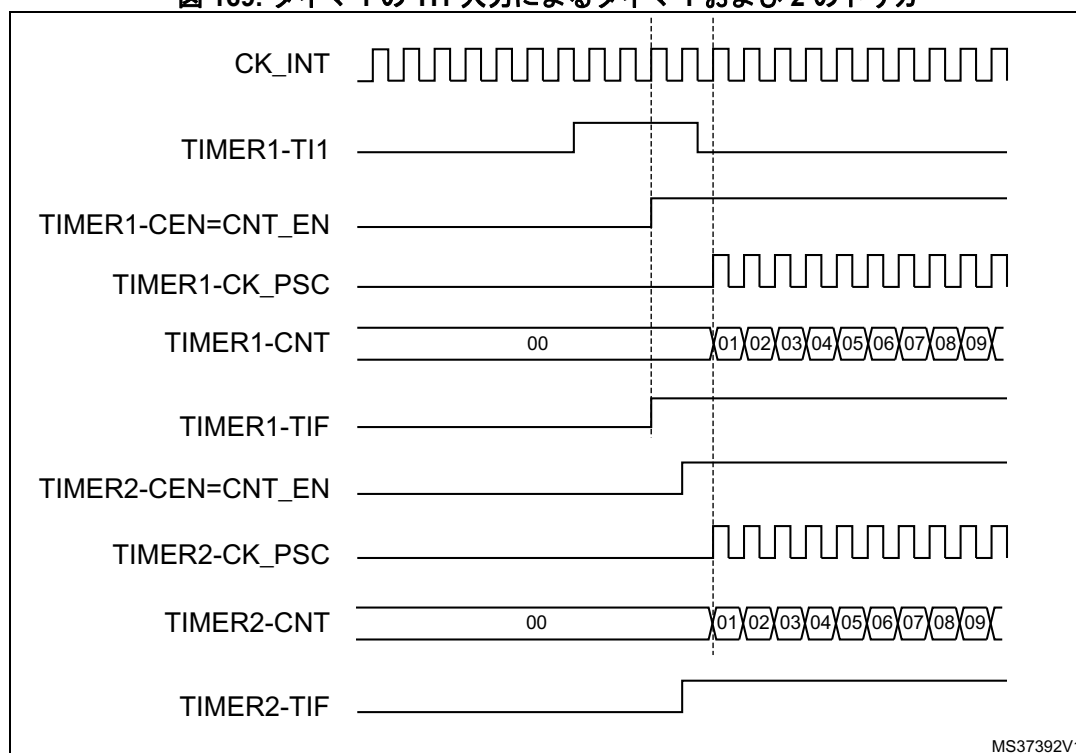
この例では、TI1 入力の立ち上がりエッジでタイマ 1 を有効にし、タイマ 1 が有効になると、タイマ 2 を有効にします。接続については、[図 164](#) を参照してください。カウンタの同時性を確保するため、タイマ 1 はマスタ/スレーブモードに設定する必要があります（TI1 に対してはスレーブ、タイマ 2 に対してはマスタ）。

- タイマ 1 をマスタモードに設定して、その有効化をトリガ出力として送信します（TIM1_CR2 レジスタの MMS=001）。
- タイマ 1 をスレーブモードに設定して、TI1 から入力トリガを受け取るようにします（TIM1_SMCR レジスタの TS=100）。
- タイマ 1 をトリガモードに設定します（TIM1_SMCR レジスタの SMS=110）。
- MSM=1（TIM1_SMCR レジスタ）を書き込むことによって、タイマ 1 をマスタ/スレーブモードに設定します。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します（TIM2_SMCR レジスタの TS=000）。
- タイマ 2 をトリガモードに設定します（TIM2_SMCR レジスタの SMS=110）。

TI1（タイマ 1）で立ち上がりエッジが発生すると、両方のカウンタが同時に内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

注： この例では、両方のタイマが開始前に初期化されます（それぞれの UG ビットをセットすることによって）。両方のカウンタは 0 から開始しますが、カウンタレジスタ（TIMx_CNT）に書き込むことによって、容易にオフセットを挿入できます。マスタ/スレーブモードでは、タイマ 1 の CNT_EN と CK_PSC の間に遅延が挿入されます。

図 169. タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ



17.3.16 デバッグモード

マイクロコントローラがデバッグモードになると（FPU 搭載 Cortex[®]-M4 コアは停止状態）、TIMx カウンタは、DBGMCU モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、停止します。詳細については、[セクション 30.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

17.4 TIM2 から TIM5 のレジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

32 ビットのペリフェラルレジスタには、ワード（32 ビット）単位で書き込む必要があります。他のすべてのペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読み出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

17.4.1 TIMx 制御レジスタ 1 (TIMx_CR1)

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD**：クロック分周

このビットフィールドは、タイマクロック（CK_INT）周波数と、デジタルフィルタ（ETR、Tlx）によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。
1: TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS**：センターアラインモード選択

00: エッジアラインモードカウンタは、方向ビット（DIR）に応じて、カウントアップまたはカウントダウンします。
01: センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。
10: センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。
11: センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注： カウンタが有効（CEN=1）なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR**：方向

0: カウンタはアップカウンタとして使用されます。
1: カウンタはダウンカウンタとして使用されます。

注： このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM**：ワンパルスモード

0: カウンタは更新イベントで停止しません。
1: カウンタは次の更新イベントでカウントを停止します（CEN ビットをクリア）。



ビット 2 URS：更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。

0：次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- － カウンタオーバーフロー／アンダーフロー
- － UG ビットのセット
- － スレープモードコントローラからの更新生成

1：カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 UDIS：更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0：UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- － カウンタオーバーフロー／アンダーフロー
- － UG ビットのセット
- － スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1：UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

ビット 0 CEN：カウンタイネーブル

0：カウンタは無効です。

1：カウンタは有効です。

注： 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

17.4.2 TIMx 制御レジスタ 2（TIMx_CR2）

アドレスオフセット：0x04

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]				CCDS	Res.	Res.
								r/w	r/w	r/w	r/w	r/w			

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S**：TI1 選択

- 0：TIMx_CH1 ピンが TI1 入力に接続されます。
- 1：TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます（XOR 接続）。

ビット 6:4 **MMS[2:0]**：マスタモード選択

- これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます（TRGO）。組み合わせは、次のとおりです。
- 000：リセット- TIMx_EGR レジスタの UG ビットがトリガ出力（TRGO）として使用されます。トリガ入力によってリセットが生成される場合（スレーブモードコントローラがリセットモードに設定されているとき）、TRGO 信号は実際のリセットより遅延します。
 - 001：イネーブル- カウンタイネーブル信号 CNT_EN がトリガ出力（TRGO）として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和（OR）によって生成されます。
 - カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ／スレーブモードが選択されている場合を除き、TRGO には遅延が存在します（TIMx_SMCR レジスタの MSM ビットの説明を参照してください）。
 - 010：更新- 更新イベントがトリガ出力（TRGO）として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。
 - 011：パルス比較- キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされるとき（すでにハイであった場合も）、トリガ出力は正のパルスを送信します。（TRGO）
 - 100：比較- OC1REF 信号がトリガ出力（TRGO）として使用されます。
 - 101：比較- OC2REF 信号がトリガ出力（TRGO）として使用されます。
 - 110：比較- OC3REF 信号がトリガ出力（TRGO）として使用されます。
 - 111：比較- OC4REF 信号がトリガ出力（TRGO）として使用されます。

ビット 3 **CCDS**：キャプチャ／比較 DMA 選択

- 0：CCx DMA リクエストは、CCx イベントが発生すると送信されます。
- 1：CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

17.4.3 TIMx スレーブモード制御レジスタ（TIMx_SMCR）

アドレスオフセット：0x08

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW

ビット 15 **ETP**：外部トリガ極性

このビットは、ETR と $\overline{\text{ETR}}$ のいずれがトリガ動作に使用されるかを選択します。

0：ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブになります。

1：ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブになります。

ビット 14 **ECE**：外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0：外部クロックモード 2 は無効です。

1：外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

1：ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります（SMS=111、TS=111）。

2：外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません（TS ビットが 111 でないことが必要）。

3：外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力 は ETRF です。

ビット 13:12 **ETPS**：外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、最大でも CK_INT 周波数の 1/4 でなければなりません。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

00：プリスケアラオフ

01：ETRP 周波数は 2 分周されます。

10：ETRP 周波数は 4 分周されます。

11：ETRP 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]**：外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプリングする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000：フィルタなし、サンプリングは f_{DTS} で行われます。

0001： $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ 、N=2

0010： $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ 、N=4

0011： $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ 、N=8

0100： $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N=6

0101： $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N=8

0110： $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=6

0111： $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=8

1000： $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N=6

1001： $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ 、N=8

1010： $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N=5

1011： $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N=6

1100： $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ 、N=8

1101： $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N=5

1110： $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N=6

1111： $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ 、N=8

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 7 **MSM** : マスタ/スレーブモード

- 0 : 影響なし。
- 1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS** : トリガ選択

- このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。
- 000 : 内部トリガ 0 (ITR0)
 - 001 : 内部トリガ 1 (ITR1)
 - 010 : 内部トリガ 2 (ITR2)
 - 011 : 内部トリガ 3 (ITR3)
 - 100 : TI1 エッジ検出回路 (TI1F_ED)
 - 101 : フィルタタイマ入力 1 (TI1FP1)
 - 110 : フィルタタイマ入力 2 (TI2FP2)
 - 111 : 外部トリガ入力 (ETRF)
- 各タイマでの ITRx の詳細については、表 95 を参照してください。

注： 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

- 外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。
- 000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。
 - 001 : エンコーダモード 1 - カウンタは、TI1FP1 のレベルに応じて、TI2FP2 のエッジでカウントアップ/ダウンします。
 - 010 : エンコーダモード 2 - カウンタは、TI2FP2 のレベルに応じて、TI1FP1 のエッジでカウントアップ/ダウンします。
 - 011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。
 - 100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
 - 101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。
 - 110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。
 - 111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

表 95. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

17.4.4 TIMx DMA／割り込み有効レジスタ（TIMx_DIER）

アドレスオフセット：0x0C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE**：トリガ DMA リクエストイネーブル
0：トリガ DMA リクエストは無効です。
1：トリガ DMA リクエストは有効です。

ビット 13 予約済み。常に 0 として読み出されます。

ビット 12 **CC4DE**：キャプチャ／比較 4 DMA リクエストイネーブル
0：CC4 DMA リクエストは無効です。
1：CC4 DMA リクエストは有効です。

ビット 11 **CC3DE**：キャプチャ／比較 3 DMA リクエストイネーブル
0：CC3 DMA リクエストは無効です。
1：CC3 DMA リクエストは有効です。

ビット 10 **CC2DE**：キャプチャ／比較 2 DMA リクエストイネーブル
0：CC2 DMA リクエストは無効です。
1：CC2 DMA リクエストは有効です。

ビット 9 **CC1DE**：キャプチャ／比較 1 DMA リクエストイネーブル
0：CC1 DMA リクエストは無効です。
1：CC1 DMA リクエストは有効です。

ビット 8 **UDE**：更新 DMA リクエストイネーブル
0：更新 DMA リクエストは無効です。
1：更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE**：トリガ割り込みイネーブル
0：トリガ割り込みは無効です。
1：トリガ割り込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IE**：キャプチャ／比較 4 割り込みイネーブル
0：CC4 割り込みは無効です。
1：CC4 割り込みは有効です。

ビット 3 **CC3IE**：キャプチャ／比較 3 割り込みイネーブル
0：CC3 割り込みは無効です。
1：CC3 割り込みは有効です。

- ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル
0 : CC2 割り込みは無効です。
1 : CC2 割り込みは有効です。
- ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル
0 : CC1 割り込みは無効です。
1 : CC1 割り込みは有効です。
- ビット 0 **UIE** : 更新割り込みイネーブル
0 : 更新割り込みは無効です。
1 : 更新割り込みは有効です。

17.4.5 TIMx ステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

- ビット 15:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **CC4OF** : キャプチャ／比較 4 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 11 **CC3OF** : キャプチャ／比較 3 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。ソフトウェアで“0”を書き込むことによってクリアされます。
0 : オーバキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。
- ビット 8:7 予約済みであり、リセット値に保持する必要があります。
- ビット 6 **TIF** : トリガ割り込みフラグ
このフラグは、トリガイイベント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。
0 : トリガイイベントは発生していません。
1 : トリガ割り込みが保留中です。
- ビット 5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **CC4IF** : キャプチャ／比較 4 割り込みフラグ
CC1IF の説明を参照してください。
- ビット 3 **CC3IF** : キャプチャ／比較 3 割り込みフラグ
CC1IF の説明を参照してください。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます (センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください)。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

- このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローまたはアンダーフロー時 (TIM2 から TIM5)、および TIMx_CR1 レジスタの UDIS=0 である場合。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき (同期制御レジスタの説明を参照)。

17.4.6 TIMx イベント生成レジスタ（TIMx_EGR）

アドレスオフセット：0x14

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									W		W	W	W	W	W

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG**：トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4G**：キャプチャ／比較 4 生成

CC1G の説明を参照してください。

ビット 3 **CC3G**：キャプチャ／比較 3 生成

CC1G の説明を参照してください。

ビット 2 **CC2G**：キャプチャ／比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G**：キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合：

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。

CC1 チャンネルが入力として設定されている場合：

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG**：更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。センターアラインモードが選択されている場合、または、DIR=0（カウントアップ）の場合、カウンタはクリアされます。そうでない場合、DIR=1（カウントダウン）であれば、自動再ロード値（TIMx_ARR）をとりまします。

17.4.7 TIMx キャプチャ／比較モードレジスタ 1（TIMx_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード

ビット 15 **OC2CE**：出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]**：出力比較 2 モード

ビット 11 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 10 **OC2FE**：出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]**：キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC2 チャンネルは出力として設定されます。

01：CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10：CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11：CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC2S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC2E=0）のときにのみ書き込み可能です。

ビット 7 **OC1CE**：出力比較 1 クリアイネーブル

OC1CE：出力比較 1 クリアイネーブル

0：OC1Ref は ETRF 入力の影響を受けません。

1：OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合は非アクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 : PWM モード 1 または 2 では、比較結果が変化したとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。**

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E=0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。**

17.4.8 TIMx キャプチャ／比較モードレジスタ 2（TIMx_CCMR2）

アドレスオフセット：0x1C

リセット値：0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード

ビット 15 **OC4CE**：出力比較 4 クリアイネーブル

ビット 14:12 **OC4M**：出力比較 4 モード

ビット 11 **OC4PE**：出力比較 4 プリロードイネーブル

ビット 10 **OC4FE**：出力比較 4 高速イネーブル

ビット 9:8 **CC4S**：キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC4 チャンネルは出力として設定されます。

01：CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10：CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11：CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。

ビット 7 **OC3CE**：出力比較 3 クリアイネーブル

ビット 6:4 **OC3M**：出力比較 3 モード

ビット 3 **OC3PE**：出力比較 3 プリロードイネーブル

ビット 2 **OC3FE**：出力比較 3 高速イネーブル

ビット 1:0 **CC3S**：キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC3 チャンネルは出力として設定されます。

01：CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10：CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11：CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。

入力キャプチャモード

- ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ
- ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケアラ
- ビット 9:8 **CC4S** : キャプチャ／比較 4 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC4 チャンネルは出力として設定されます。
- 01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
- 10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
- 11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。
- ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ
- ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケアラ
- ビット 1:0 **CC3S** : キャプチャ／比較 3 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC3 チャンネルは出力として設定されます。
- 01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。
- 10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。
- 11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。

17.4.9 TIMx キャプチャ／比較有効レジスタ（TIMx_CCER）

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rW		rW	rW	rW		rW	rW	rW		rW	rW	rW		rW	rW

- ビット 15 **CC4NP** : キャプチャ／比較 4 出力極性
- CC1NP の説明を参照してください。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **CC4P** : キャプチャ／比較 4 出力極性
- CC1P の説明を参照してください。
- ビット 12 **CC4E** : キャプチャ／比較 4 出力イネーブル。
- CC1E の説明を参照してください。
- ビット 11 **CC3NP** : キャプチャ／比較 3 出力極性
- CC1NP の説明を参照してください。
- ビット 10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **CC3P** : キャプチャ／比較 3 出力極性
- CC1P の説明を参照してください。
- ビット 8 **CC3E** : キャプチャ／比較 3 出力イネーブル。
- CC1E の説明を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

- ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性
CC1NP の説明を参照してください。
- ビット 6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **CC2P** : キャプチャ／比較 2 出力極性
CC1P の説明を参照してください。
- ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル。
CC1E の説明を参照してください。
- ビット 3 **CC1NP** : キャプチャ／比較 1 出力極性
CC1 チャンネルが出力として設定されている場合 :
この場合、CC1NP はクリアされたままでなければなりません。
CC1 チャンネルが入力として設定されている場合 :
このビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **CC1P** : キャプチャ／比較 1 出力極性
CC1 チャンネルが出力として設定されている場合 :
0 : OC1 はアクティブハイです。
1 : OC1 はアクティブローです。
CC1 チャンネルが入力として設定されている場合 :
CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。
00 : 非反転／立ち上がりエッジ
回路は TIXFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIXFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。
01 : 反転／立ち下がりエッジ
回路は TIXFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIXFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。
10 : 予約済み。この設定は使用しないでください。
11 : 非反転／両エッジ
回路は TIXFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIXFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。
- ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。
CC1 チャンネルが出力として設定されている場合 :
0 : オフ - OC1 はアクティブではありません。
1 : オン - OC1 信号は、対応する出力ピンに出力されます。
CC1 チャンネルが入力として設定されている場合 :
このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。
0 : キャプチャは無効です。
1 : キャプチャは有効です。

表 96. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx=0、OCx_EN=0)
1	OCx=OCxREF + 極性、OCx_EN=1



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

注：標準 OCx チャンネルに接続されている外部 IO ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

17.4.10 TIMx カウンタ（TIMx_CNT）

アドレスオフセット：0x24
リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

17.4.11 TIMx プリスケアラ（TIMx_PSC）

アドレスオフセット：0x28
リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値
カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。
PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

17.4.12 TIMx 自動再ロードレジスタ（TIMx_ARR）

アドレスオフセット：0x2C
リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値
ARR は、実際の自動再ロードレジスタにロードされる値です。
ARP の更新と動作の詳細については、[セクション 17.3.1：タイムベースユニット（468 ページ）](#)を参照してください。
自動再ロード値が null のときには、カウンタはブロックされます。

17.4.13 TIMx キャプチャ／比較モードレジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR1[31:16]**：キャプチャ／比較 1 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値下位ビット

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

17.4.14 TIMx キャプチャ／比較モードレジスタ 2（TIMx_CCR2）

アドレスオフセット：0x38

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR2[31:16]**：キャプチャ／比較 2 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR2[15:0]**：キャプチャ／比較 2 値下位ビット

CC2 チャンネルが出力として設定されている場合：

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合：

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。



17.4.15 TIMx キャプチャ／比較モードレジスタ 3（TIMx_CCR3）

アドレスオフセット：0x3C

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR3[31:16]**：キャプチャ／比較 3 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR3[15:0]**：キャプチャ／比較値下位ビット

CC3 チャンネルが出力として設定されている場合：

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC3 出力に送信される値を含みます。

CC3 チャンネルが入力として設定されている場合：

CCR3 は、最後の入力キャプチャ 3 イベント（IC3）によって転送されたカウンタ値です。

17.4.16 TIMx キャプチャ／比較モードレジスタ 4（TIMx_CCR4）

アドレスオフセット：0x40

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR4[31:16]**：キャプチャ／比較 4 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR4[15:0]**：キャプチャ／比較値下位ビット

1. **CC4 チャンネルが出力として設定されている場合（CC4S ビット）：**

CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC4 出力に送信される値を含みます。

2. **CC4 チャンネルが入力として設定されている場合（TIMx_CCMR4 レジスタの CC4S ビット）：**

CCR4 は、最後の入力キャプチャ 4 イベント（IC4）によって転送されたカウンタ値です。

17.4.17 TIMx DMA 制御レジスタ（TIMx_DCR）

アドレスオフセット：0x48

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送回数（タイマは、TIMx_DMAR アドレスに対して読み出したまたは書き込みアクセスが行われるときにバースト転送を認識します）を指定します。

00000 : 1 回転送

00001 : 2 回転送、

00010 : 3 回転送、

...

10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します（TIMx_DMAR アドレスを通じて読み出し／書き込みアクセスが行われるとき）。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx_CR1、

00001 : TIMx_CR2、

00010 : TIMx_SMCR

...

例：次の転送を考えます：DBL = 7 回転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

17.4.18 完全転送の TIMx DMA アドレス（TIMx_DMAR）

アドレスオフセット：0x4C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります：

$$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL は TIMx_DCR 内で設定）。

DMA パースト機能の使用例

この例では、タイマ DMA パースト機能を使って CCRx レジスタ値を更新します（x = 2、3、4）。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
4. TIMx を有効化
5. DMA チャンネルを有効化注：

注： この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

17.4.19 TIM2 オプションレジスタ（TIM2_OR）

アドレスオフセット：0x50

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ITR1_RMP		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw										

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 ITR1_RMP：内部トリガ 1 再配置

ソフトウェアでセット／クリアされます。

00：TIM8_TRGOUT

01：予約済み

10：OTG FS SOF は TIM2_ITR1 入力に接続されます。

11：OTG HS SOF は TIM2_ITR1 入力に接続されます。

ビット 9:0 予約済みであり、リセット値に保持する必要があります。

17.4.20 TIM5 オプションレジスタ（TIM5_OR）

アドレスオフセット：0x50

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP		Res.	Res.	Res.	Res.	Res.	Res.
								rw	rw						

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **TI4_RMP**：タイマ 入力 4 再配置

ソフトウェアでセット／クリアされます。

00：TIM5 チャンネル 4 を GPIO に接続：データシートの代替機能配置表を参照してください。

01：LSI の内部クロックは較正のために TIM5_CH4 入力へ接続

10：LSE の内部クロックは較正のために TIM5_CH4 入力へ接続

11：RTC ウェイクアップ割り込みは較正のために TIM5_CH4 入力へ接続ウェイクアップ割り込みを有効化する必要があります。

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

17.4.21 TIMx レジスタマップ

TIMx レジスタは、次の表のようにマップされます。

表 97. TIM2 から TIM5 のレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CKD [1:0]	ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CFN		
	リセット値																							0	0	0	0	0	0	0	0	0	
0x04	TIMx_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	T1S	MMS[2:0]			CCDS	Res		
	リセット値																									0	0	0	0	0			
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ETP	ECE	ETPS [1:0]	ETF[3:0]			MSM			TS[2:0]			Res	SMS[2:0]		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	リセット値																		0	0	0	0	0	0	0		0		0	0	0	0	
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	Res	Res	TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	リセット値																				0	0	0	0			0		0	0	0	0	0
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	CC4G	CC3G	CC2G	CC1G	UG
	リセット値																										0		0	0	0	0	0
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC2CE	OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]	OC1CE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIMx_CCMR1 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]			IC2 PSC [1:0]	CC2S [1:0]	IC1F[3:0]			IC1 PSC [1:0]	CC1S [1:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	TIMx_CCMR2 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC24CE	OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]	OC3CE	OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIMx_CCMR2 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]			IC4 PSC [1:0]	CC4S [1:0]	IC3F[3:0]			IC3 PSC [1:0]	CC3S [1:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC4NP	CC4P	CC4E	CC3NP	Res	CC3P	CC3E	CC2NP	Res	CC2P	CC2E	CC1NP	Res	CC1P	CC1E
	リセット値																		0	0	0	0		0	0	0		0	0		0	0	
0x24	TIMx_CNT	CNT[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																CNT[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 97. TIM2 から TIM5 のレジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	TIMx_ARR	ARR[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																ARR[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x34	TIMx_CCR1	CCR1[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																CCR1[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38	TIMx_CCR2	CCR2[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																CCR2[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x3C	TIMx_CCR3	CCR3[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																CCR3[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x40	TIMx_CCR4	CCR4[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約 済み。)																CCR4[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x48	TIMx_DCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBL[4:0]				Res	Res	Res	DBA[4:0]						
	リセット値																				0	0	0	0	0				0	0	0	0	0	
0x4C	TIMx_DMAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMAB[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50	TIM2_OR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ITR1_ RMP		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																					0	0											
0x50	TIM5_OR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			IT4_ RMP		Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																								0	0								

レジスタ境界アドレスについては、[セクション 2.2.2：メモリマップとレジスタ境界アドレス](#) を参照してください。

18 汎用タイマ (TIM9 から TIM14)

18.1 TIM9 から TIM14 の概要

TIM9 から TIM14 汎用タイマは、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

これらのカウンタは、入力信号パルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM）など様々な目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

タイマ TIM9～TIM14 は完全に独立していて、いかなるリソースも共用しません。これらのタイマは、[セクション 18.3.12](#)に示すように、相互に同期させることができます。

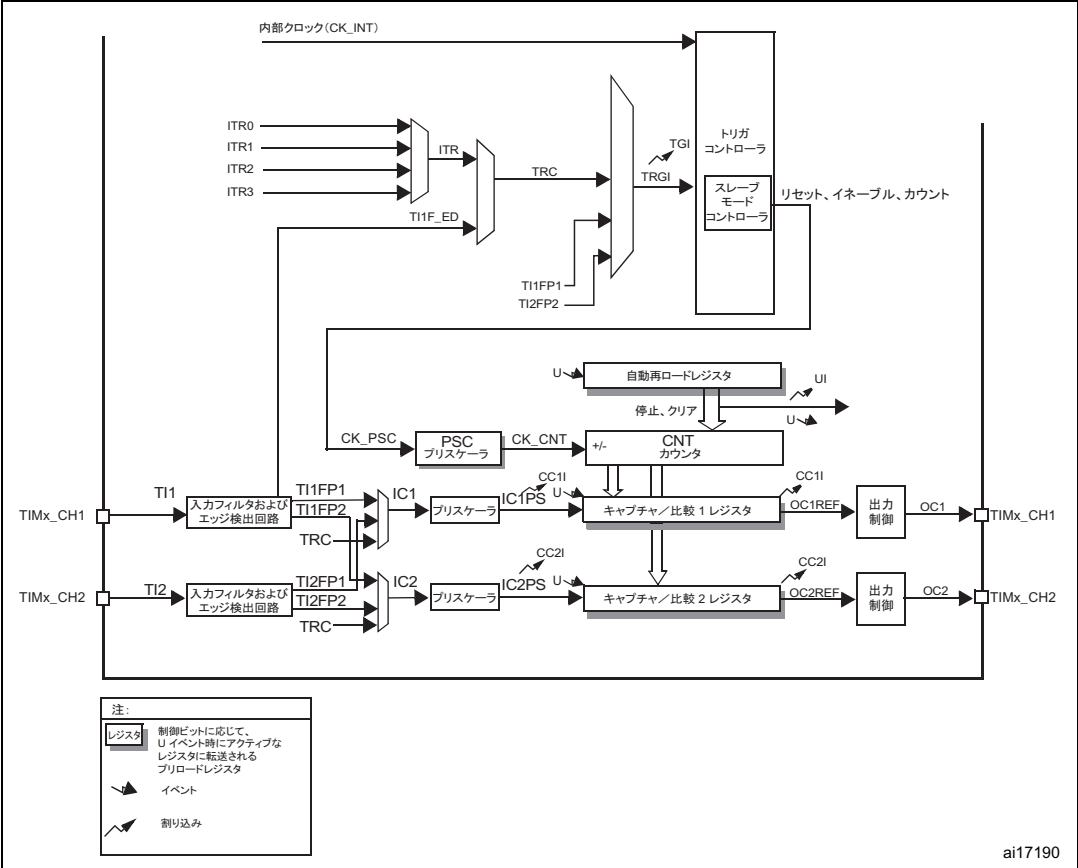
18.2 TIM9 から TIM14 の主な機能

18.2.1 TIM9/TIM12 の主な機能

TIM9 から TIM14 の汎用タイマの主な機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケラ（動作中に変更可能）。
- 次の機能を持つ、最大 2 つの独立チャネル。
 - － 入力キャプチャ
 - － 出力比較
 - － PWM 生成（エッジアラインモード）
 - － ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み生成。
 - － 更新：カウンタオーバーフロー、カウンタ初期化（ソフトウェアまたは内部トリガによる）
 - － トリガイベント（内部トリガによるカウンタの開始、停止、初期化、またはカウント）
 - － 入力キャプチャ
 - － 出力比較

図 170. 汎用 タイマのブロック図 (TIM9 とTIM12)

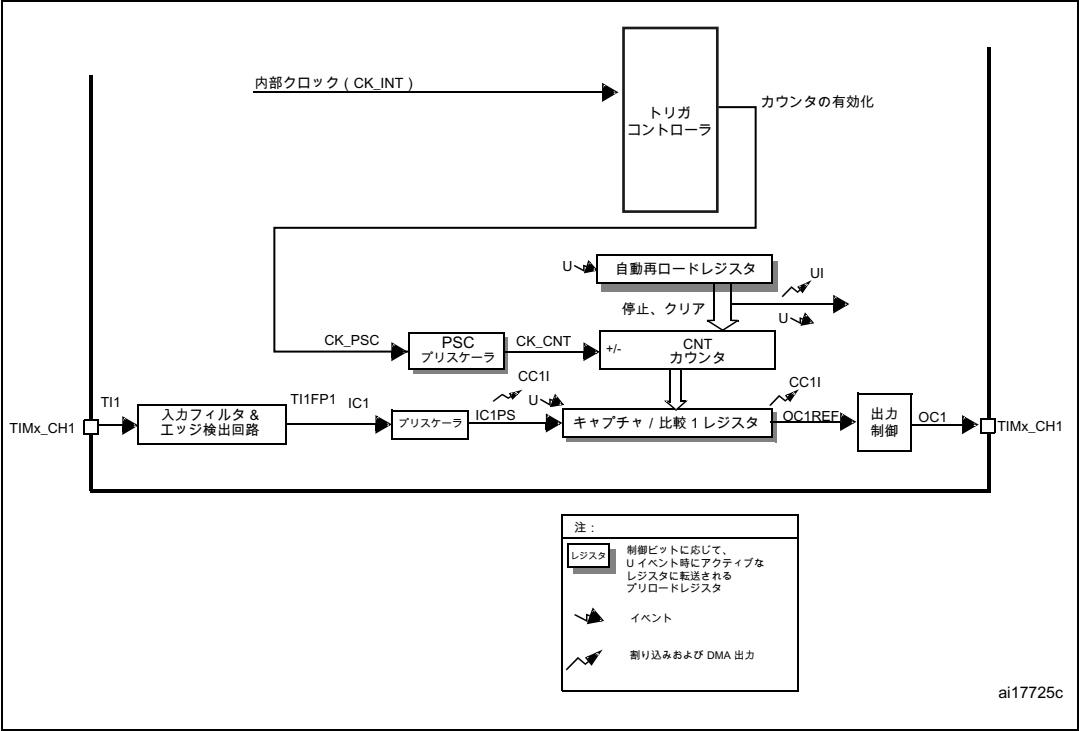


18.2.2 TIM10/TIM11 および TIM13/TIM14 の主な機能

汎用タイマ TIM10/TIM11 および TIM13/TIM14 には次の特長があります：

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ（動作中に変更可能）。
- 次の機能を持つ独立チャネル：
 - － 入力キャプチャ
 - － 出力比較
 - － PWM 生成（エッジアラインモード）
 - － ワンパルスモード出力
- 以下のイベント時の割り込み生成。
 - － 更新：カウンタオーバーフロー、カウンタの初期化（ソフトウェアによる）
 - － 入力キャプチャ
 - － 出力比較

図 171. 汎用タイマのブロック図（TIM10/11/13/14）



18.3 TIM9 から TIM14 の機能詳細

18.3.1 タイムベースユニット

タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。このカウンタはカウントアップします。

カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定ごとに詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。



 172 と  173 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 172. プリスケール分周比が 1 から 2 に変化したときのカウンタのタイミング図

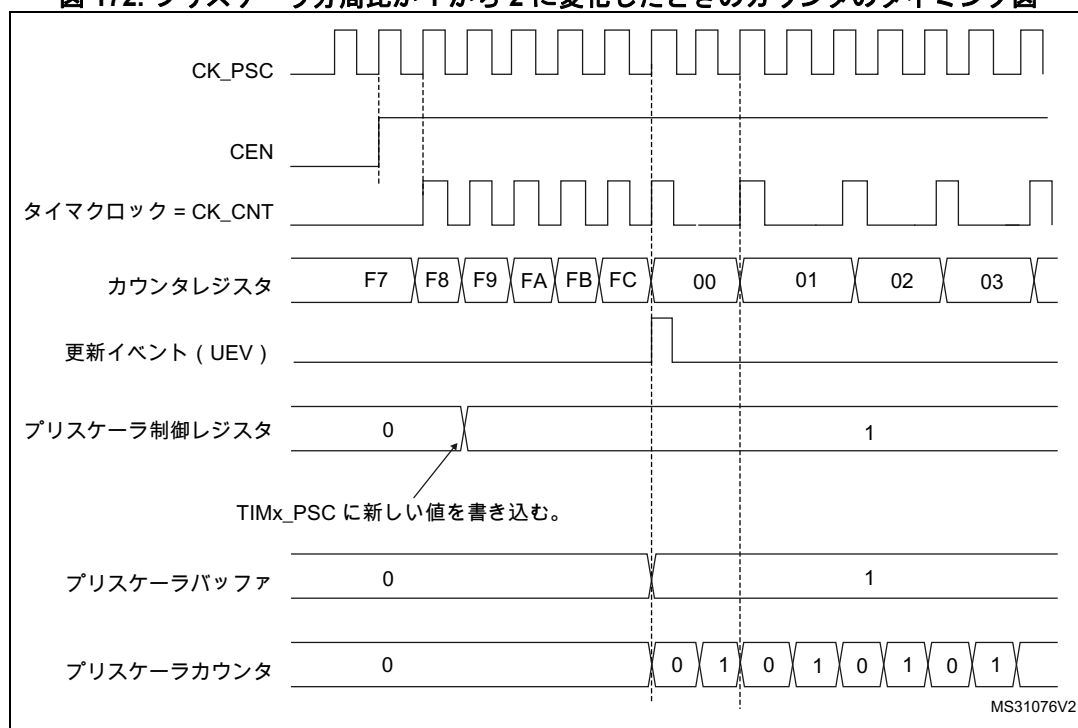
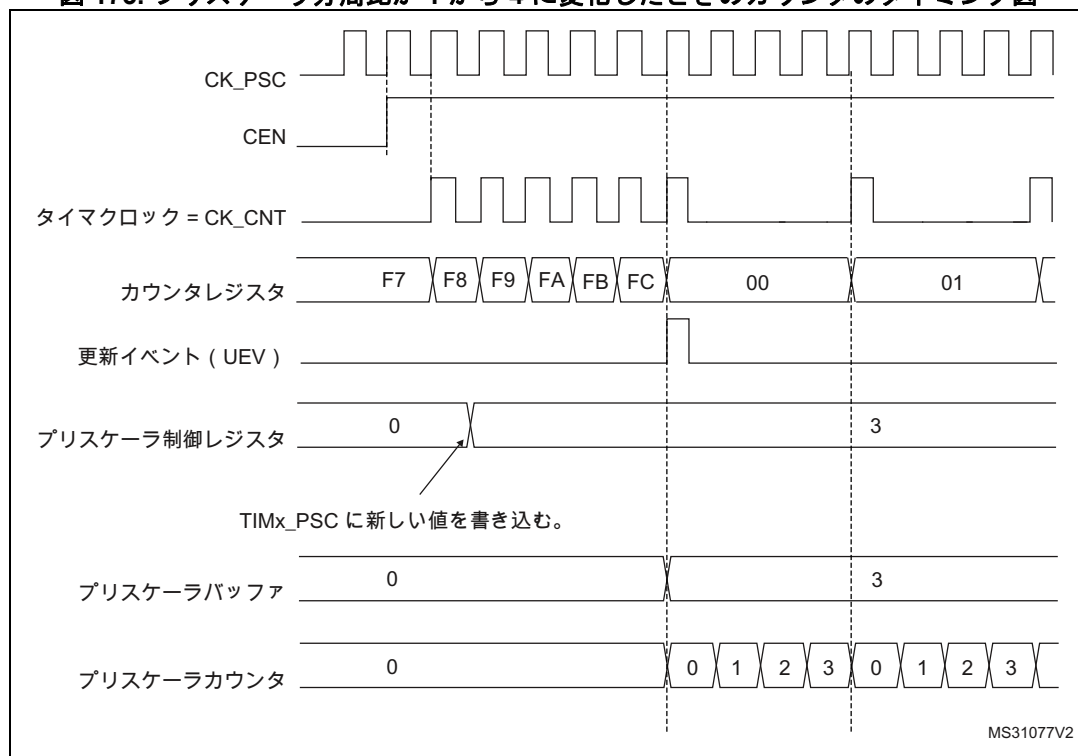


図 173. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



18.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

(ソフトウェアによって、または TIM9 および TIM12 のスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 174. 内部クロック分周比が 1 の場合のカウンタのタイミング図

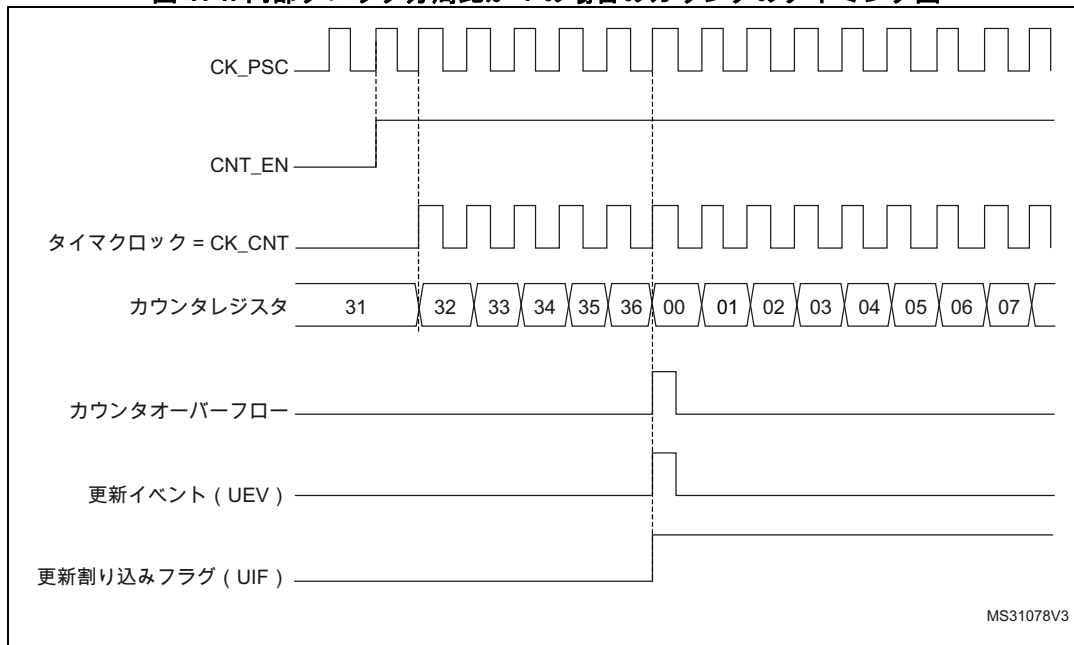


図 175. 内部クロック分周比が 2 の場合のカウンタのタイミング図

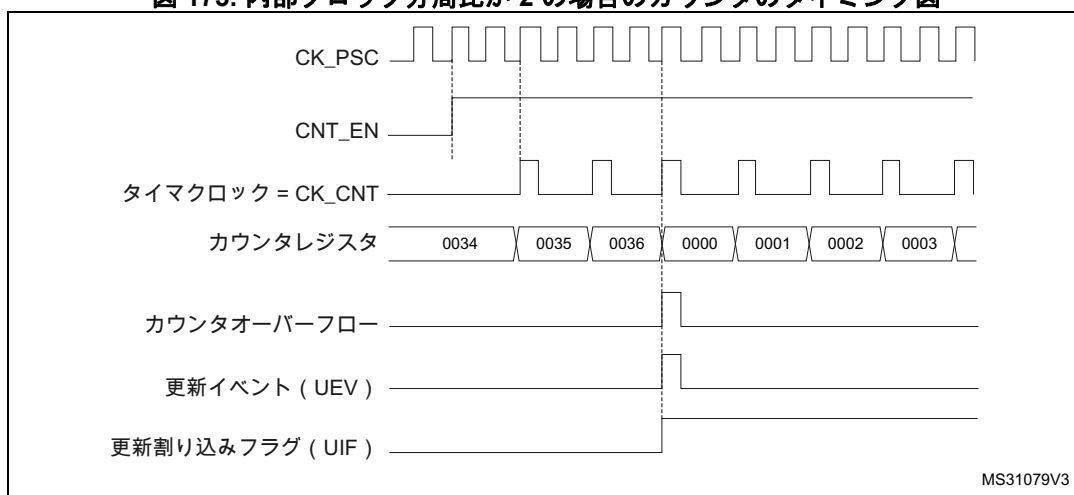


図 176. 内部クロック分周比が 4 の場合のカウンタのタイミング図

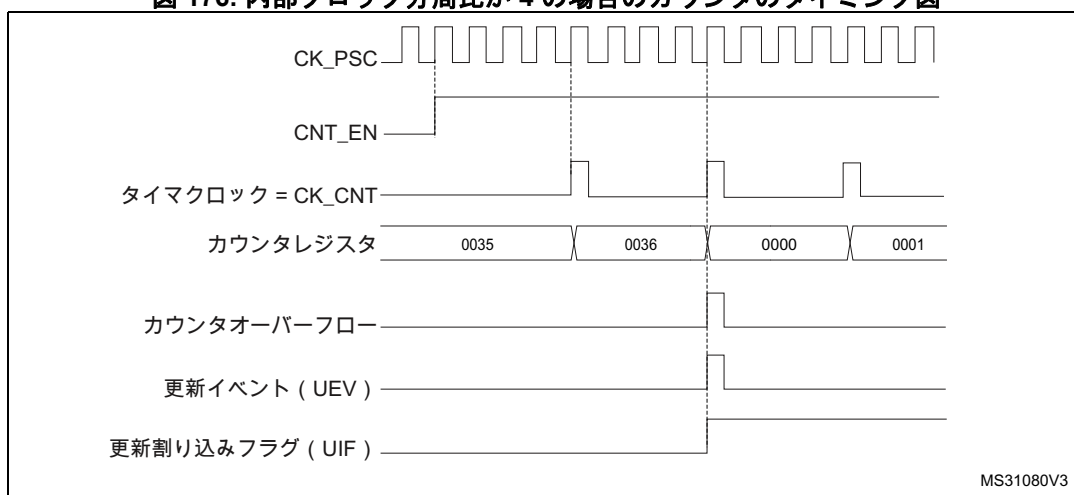


図 177. 内部クロック分周比が N の場合のカウンタのタイミング図

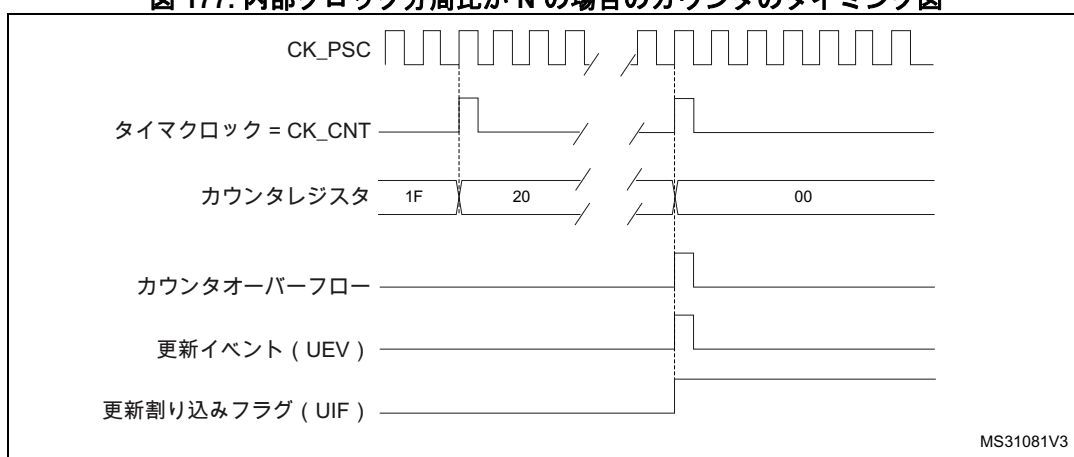


図 178. ARPE=0 の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR はプリロードされない)

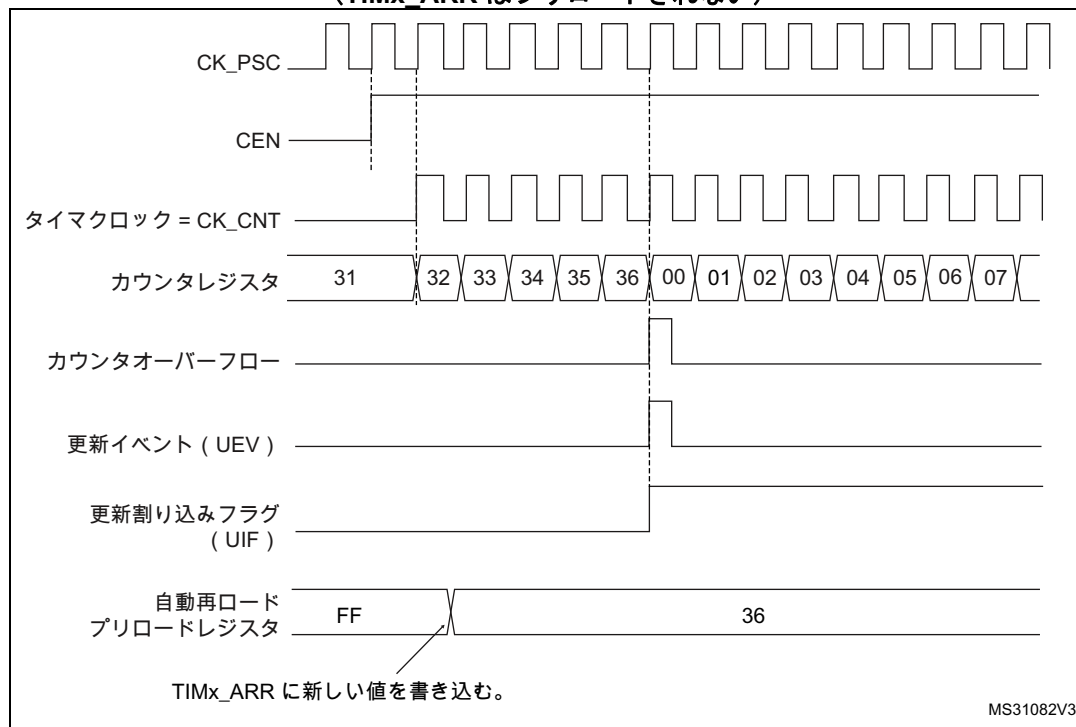
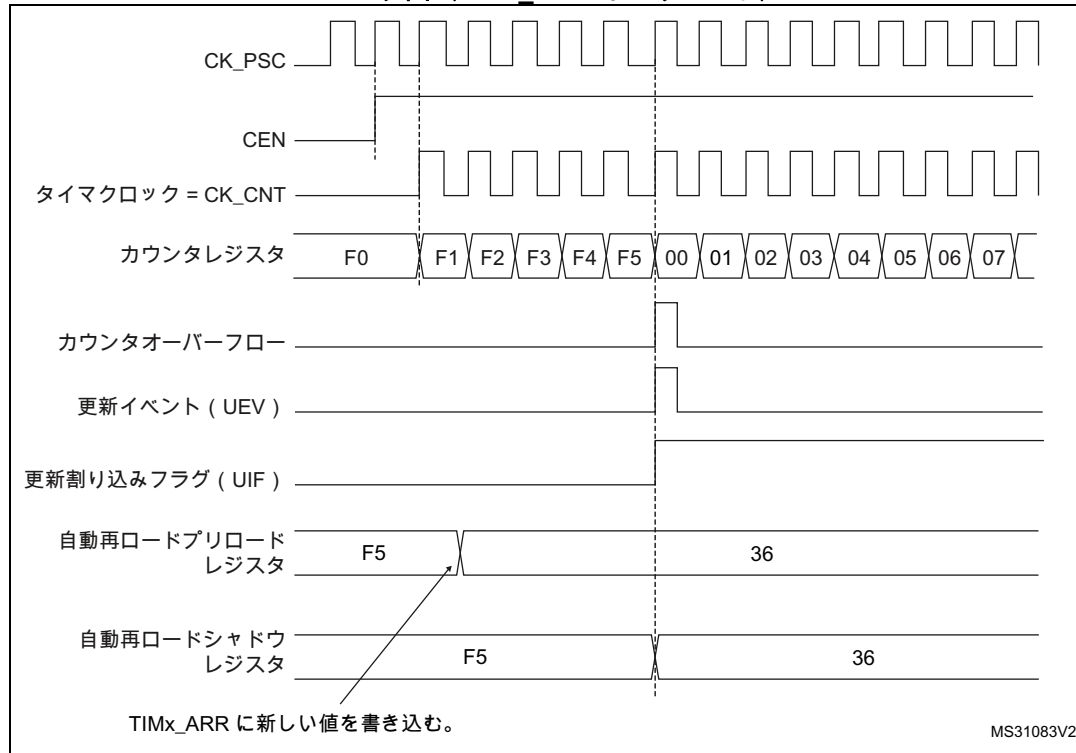


図 179. ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)



18.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

- 内部クロック (CK_INT)
- 外部クロックモード 1 (TIM9 および TIM12) : 外部入力ピン (Tlx)
- 内部トリガ入力 (ITRx) (TIM9 および TIM12) : 他のタイマからトリガ出力を接続。詳細については、[タイマを別のタイマのプリスケアラとして使用する](#) を参照してください。

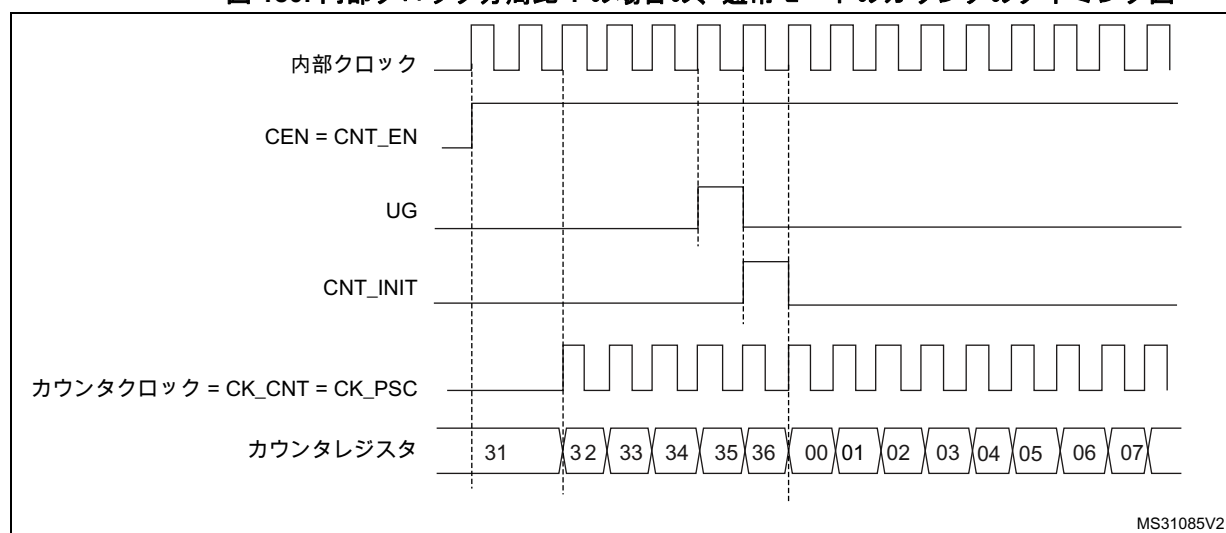
内部クロックソース (CK_INT)

内部 クロックソースは、TIM10/TIM11 および TIM13/TIM14 のデフォルトクロックソースです。

TIM9 および TIM12 に対しては、スレーブモードコントローラが無効の場合 (SMS="000")、内部クロックソースが選択されます。その際、TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットが制御ビットとして使用され、ソフトウェアからのみ変更できます (クリアされたままの UG ビットは除く)。CEN ビットに 1 が設定されると直ちに、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 180](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

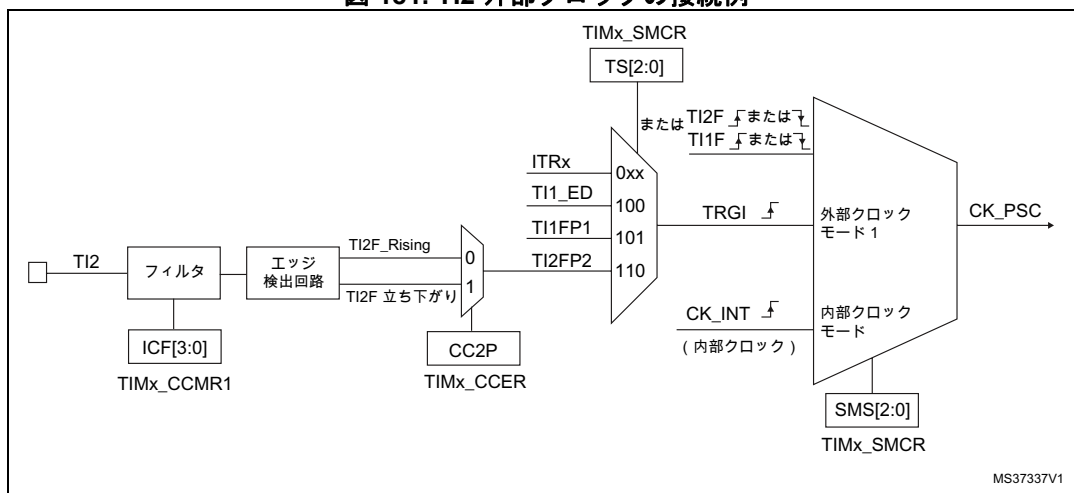
図 180. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1 (TIM9 および TIM12)

このモードは TIMx_SMCR レジスタの SMS = 111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 181. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

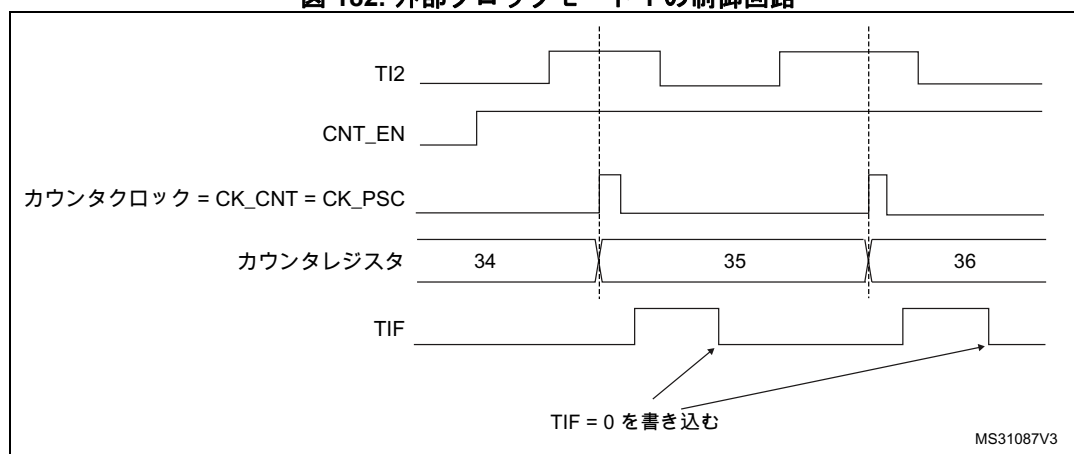
1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F = 0000 にしておきます）。
3. TIMx_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込むことによって、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタの SMS ビットに“111”を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタの TS ビットに“110”を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタの CEN ビットに“1”を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケラはトリガには使用されないため、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 182. 外部クロックモード 1 の制御回路



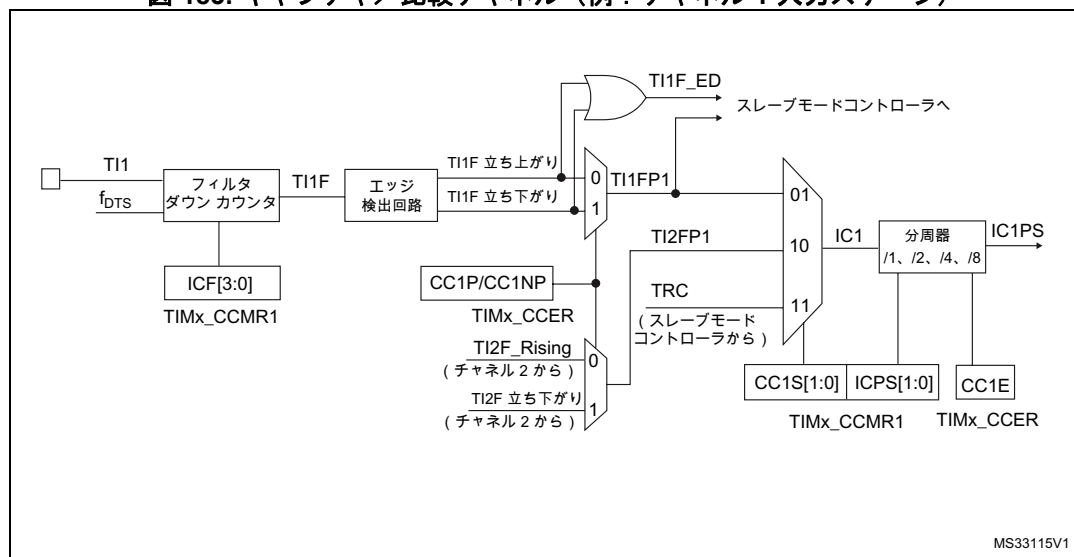
18.3.4 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケラ）、および出力ステージ（比較回路と出力制御）から構成されています。

図 183 から 図 185 に、1 つのキャプチャ／比較チャネルの概要を示します。

入力ステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレープモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 183. キャプチャ／比較チャネル（例：チャネル 1 入力ステージ）



出力ステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 184. キャプチャ／比較チャンネル 1 メイン回路

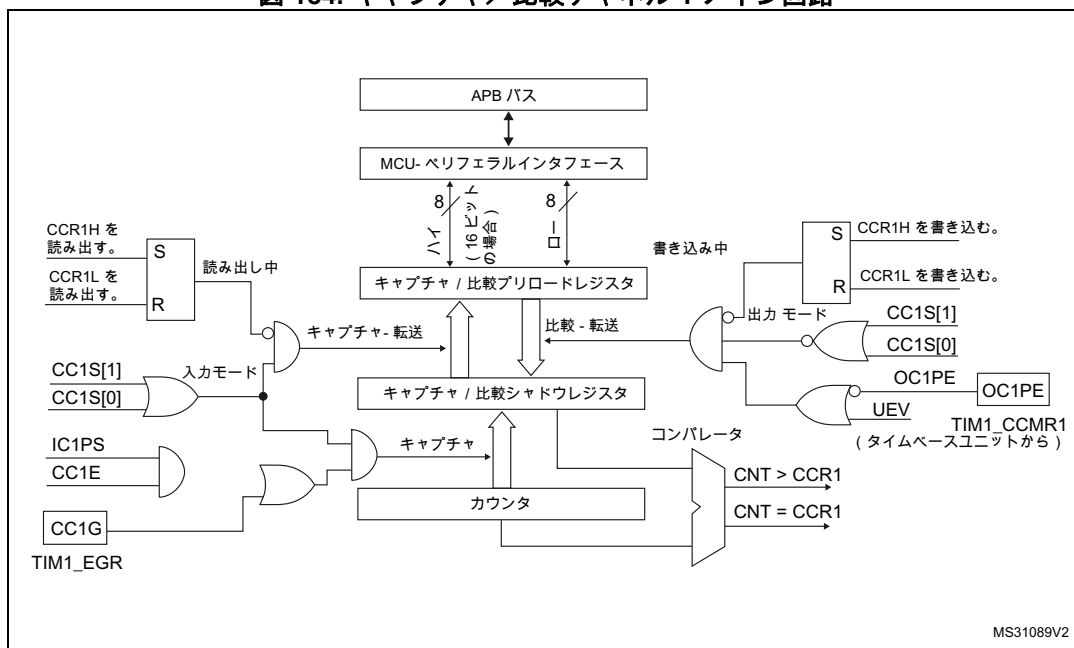
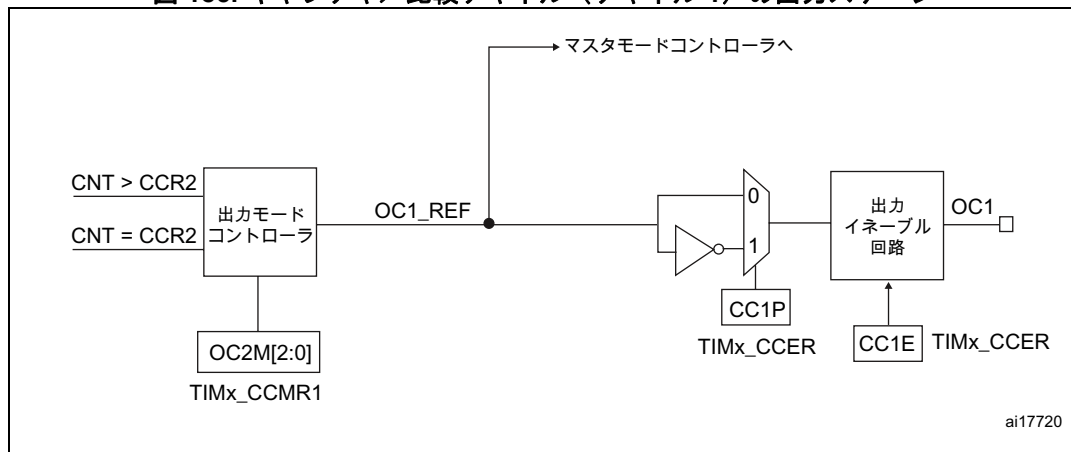


図 185. キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

18.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ／比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェア

アで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx_CCR1 は TI1 入力にリンクされていなければならないので、TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化するとすぐに、チャンネルは入力モードに設定され、TIMx_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に対して必要とする入力フィルタ時間を設定します（入力が Tix 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います）。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます（周波数 f_{DTS} でサンプリング）。次に、TIMx_CCMR1 レジスタの IC1F ビットに“0011”を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を設定します（この場合、立ち上がりエッジの選択）。
4. 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします（TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む）。
5. TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要に応じて、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます（割り込みフラグ）。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注： TIMx_EGR レジスタの対応する CCxG ビットをセットすることで、IC 割り込みリクエストをソフトウェアで発生させることができます。

18.3.6 PWM 入力モード (TIM9/12 の場合のみ)

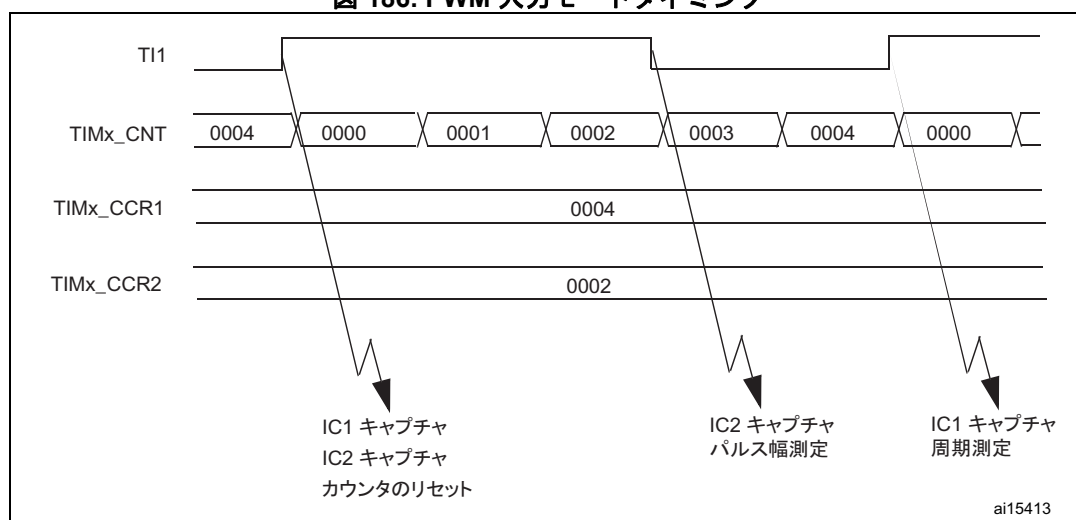
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tix 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TixFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます（手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります）。

1. TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
2. CC1P ビットと CC1NP ビットに“00”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用)。
3. TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
4. CC2P ビットと CC2NP ビットに“11”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
5. TIMx_SMCR レジスタの TS ビットに“101”を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
6. TIMx_SMCR レジスタの SMS ビットに“100”を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
7. TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 186. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx_CH1/TIMx_CH2 信号でのみ使用できます。

18.3.7 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF、次に OCx) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx_OCMRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP = 0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

18.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

1. 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM = 000)、アクティブにセットされるか (OCxM = 001)、インアクティブにセットされるか (OCxM = 010)、または反転されます (OCxM = 011)。
2. 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
3. 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

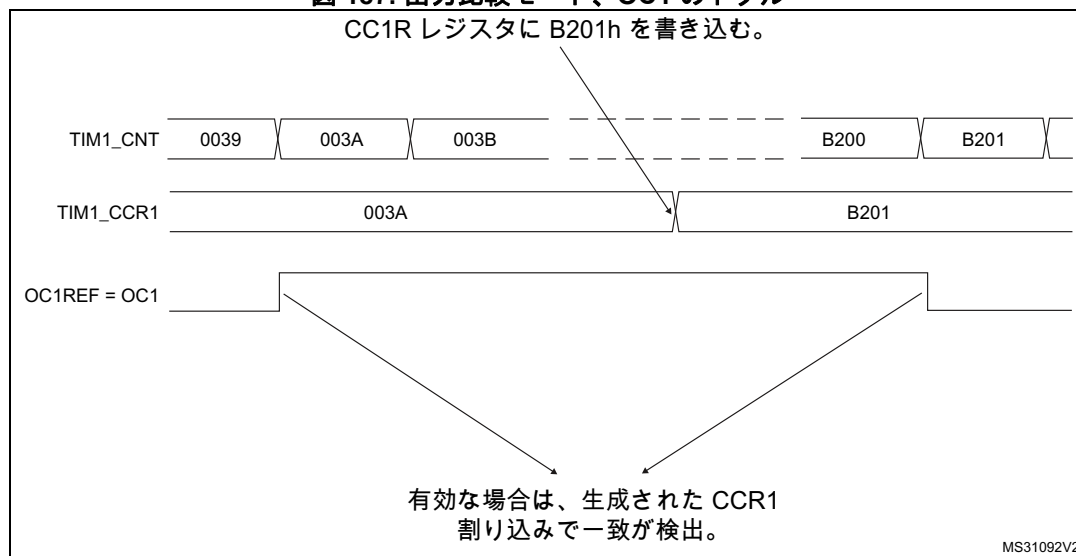
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順：

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
 - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに“011”を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに“0”を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに“0”を書き込みます。
 - 出力を有効にするには、CCxE ビットに“1”を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 187](#) に示します。

図 187. 出力比較モード、OC1 のトグル



18.3.9 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx の極性は、TIMx_CCER レジスタの CCxP ビットを使用して、ソフトウェアでプログラムできます。アクティブハイまたはアクティブラーとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効化されます。詳細については、TIMx_CCERx レジスタの説明を参照してください。

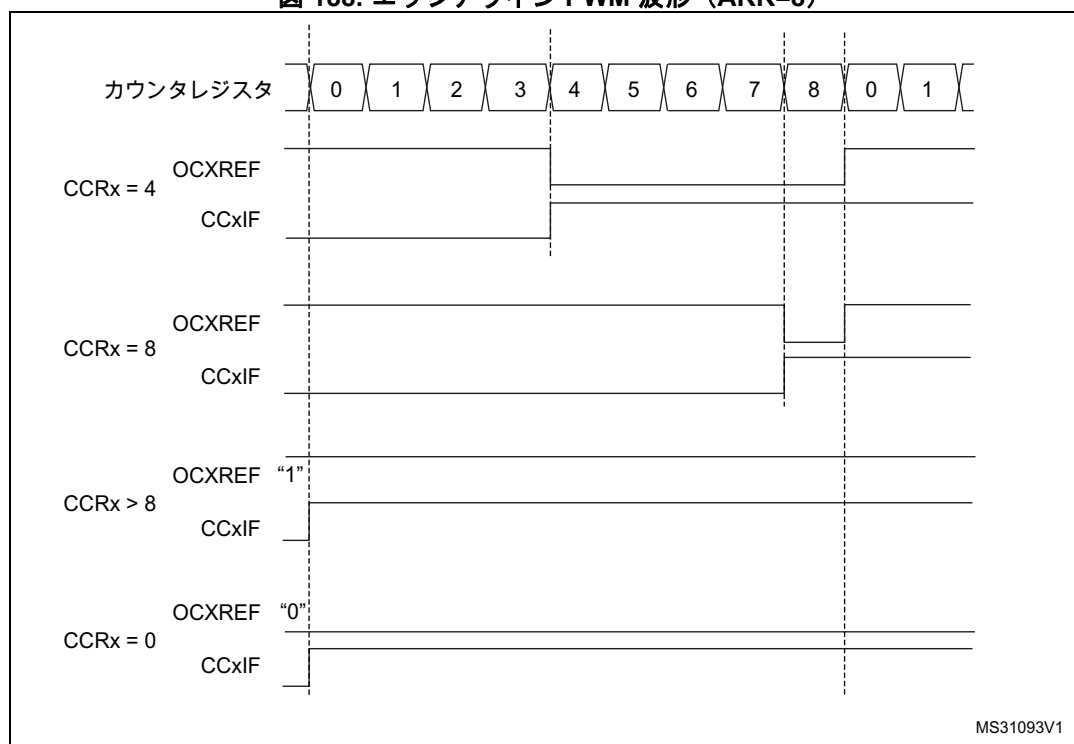
PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CNT ≤ TIMx_CCRx かどうか判断されます。

カウンタはカウントアップしているので、タイマはエッジアラインモードでのみ PWM を生成できます。

PWM エッジアラインモード

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 188 に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 188. エッジアライン PWM 波形（ARR=8）



18.3.10 ワンパルスモード

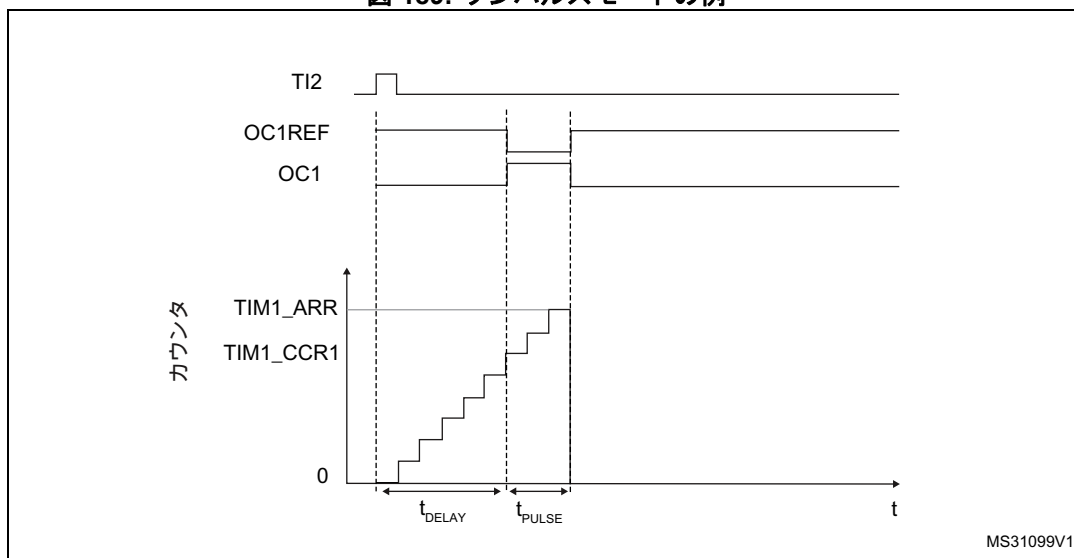
ワンパルスモード（OPM : One Pulse Mode）は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に（タイマがトリガを待っているときに）、設定が次のようであればなりません。

$$CNT < CCRx \leq ARR \text{ (特に、} 0 < CCRx \text{)}$$

図 189. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
2. TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込みます。
3. TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに“110”を書き込みます。
4. TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M ビットに“111”を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース : OCx 高速イネーブル

ワンパルスモードでは、Tlx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

18.3.11 TIM9/12 外部トリガ同期

TIM9/12 タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード : リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

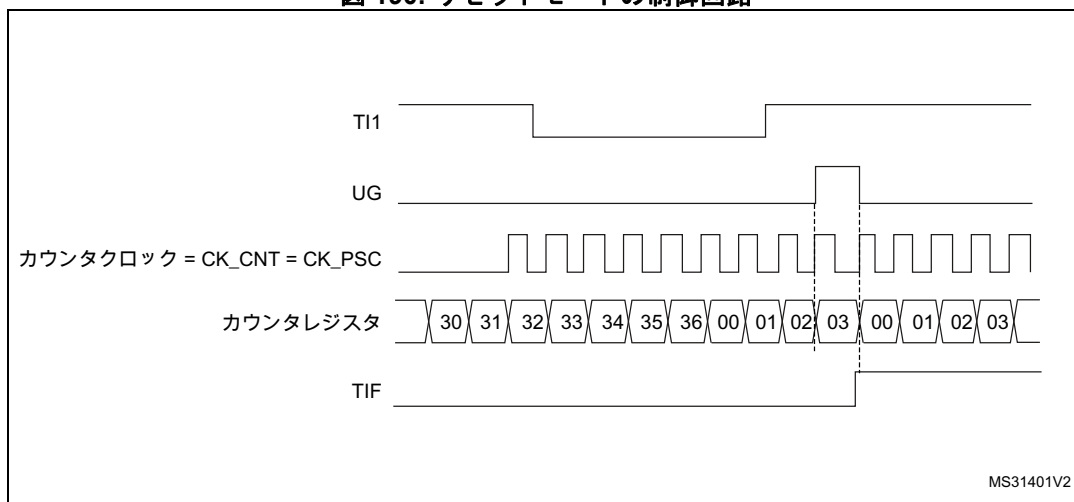
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S = 01)。TIMx_CCER レジスタの CC1P と CC1NP に "00" を書き込んで、極性を有効にします (その後、立ち上がりエッジのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに "100" を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタの TS ビットに "101" を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに "1" を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、(TIMx_DIER レジスタの TIE ビットが) 有効な場合は割り込みリクエストを送信できます。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 190. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

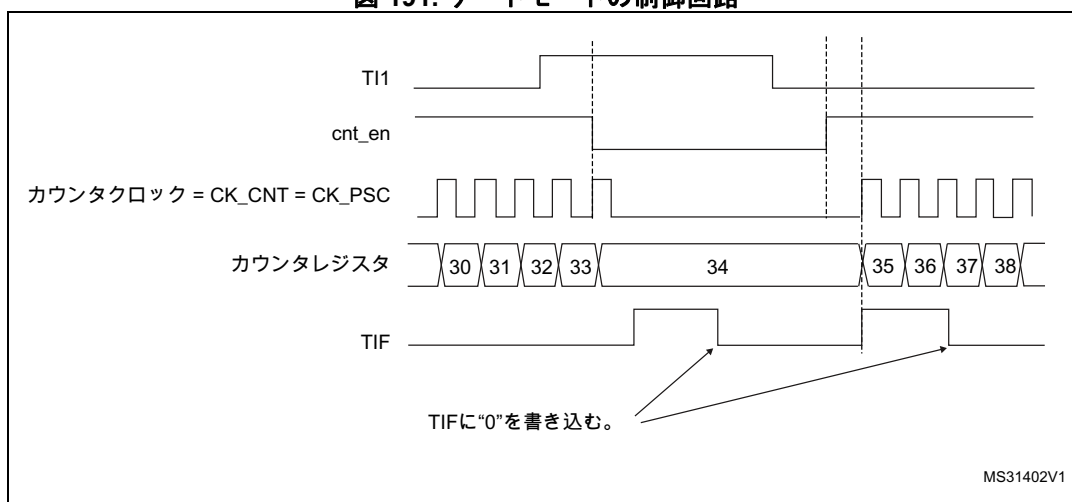
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

1. TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F = 0000 のままにしておく）。キャプチャプリスケラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S = 01）。TIMx_CCER レジスタの CC1P ビットに“1”、CC1NP ビットに“0”を書き込んで、極性を有効にします（その後、立ち上がりエッジのみを検出）。
2. TIMx_SMCR レジスタの SMS ビットに“101”を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタの TS ビットに“101”を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに“1”を書き込んで、カウンタを有効にします（ゲートモードでは、CEN = 0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 191. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

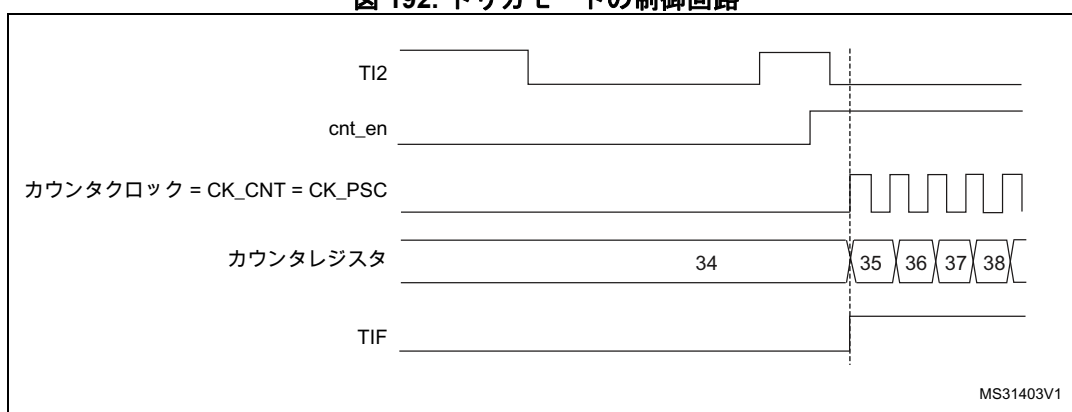
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S = 01)。TIMx_CCER レジスタの CC2P ビットに“1”、CC2NP ビットに“0”を書き込んで、極性を有効にします (その後、ローレベルのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに“110”を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタの TS ビットに“110”を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 192. トリガモードの制御回路



18.3.12 タイマ同期 (TIM9/12)

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 17.3.15 : タイマの同期](#)を参照してください。

18.3.13 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 30.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

18.4 TIM9 および TIM12 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#)を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

18.4.1 TIM9/12 制御レジスタ 1 (TIMx_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
						rw	rw	rw				rw	rw	rw	rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デジタルフィルタ (TlX) によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。
1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。
1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 URS : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。

0 : 次のイベントのいずれかが更新割り込みを生成します（有効な場合）。

- カウンタオーバーフロー

- UG ビットのセット

1 : カウンタオーバーフローのみが更新割り込みを生成します（有効な場合）。

ビット 1 UDIS : 更新ディセーブル

このビットは、更新イベント（UEV）生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー

- UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。UEV は生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 CEN : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

18.4.2 TIM9/12 のスレーブモード制御レジスタ（TIMx_SMCR）

アドレスオフセット：0x08

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]				Res.	SMS[2:0]		
								rW	rW	rW	rW		rW	rW	rW	

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM**：マスタ／スレーブモード

- 0：影響なし。
- 1：トリガ入力（TRGI）に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期（TRGO を通じて）を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期する場合に役立ちます。

ビット 6:4 **TS**：トリガ選択

- このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。
- 000：内部トリガ 0（ITR0）
 - 001：内部トリガ 1（ITR1）
 - 010：内部トリガ 2（ITR2）
 - 011：内部トリガ 3（ITR3）
 - 100：TI1 エッジ検出回路（TI1F_ED）
 - 101：フィルタタイマ入力 1（TI1FP1）
 - 110：フィルタタイマ入力 2（TI2FP2）
 - 111：予約済み。
- 各タイマにとっての ITRx の意味の詳細については、表 98 を参照してください。

注： 遷移時の誤ったエッジ検出を避けるために、これらのビットは必ず使用されていないとき（SMS = 000 のときなど）に変更してください。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS**：スレーブモード選択

- 外部信号が選択されると、トリガ信号（TRGI）のアクティブエッジが外部入力で選択された極性にリンクされます（入力制御レジスタおよび制御レジスタの説明を参照してください）。
- 000：スレーブモードは無効です。CEN = 1 の場合、プリスケアラは内部クロックによって直接クロック供給されます。
 - 001：予約済み
 - 010：予約済み
 - 011：予約済み
 - 100：リセットモード - 選択されたトリガ入力（TRGI）の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
 - 101：ゲートモード - カウンタクロックは、トリガ入力（TRGI）がハイのときに有効になります。トリガがローになると、カウンタは停止します（リセットはされません）。カウンタの開始と停止の両方が制御されます。
 - 110：トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します（リセットはされません）。カウンタの開始のみが制御されます。
 - 111：外部クロックモード 1 - 選択されたトリガ（TRGI）の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合（TS = 100）、ゲートモードを使用することはできません。TI1F_ED は TI1F の遷移ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。



表 98. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = '000')	ITR1 (TS = '001')	ITR2 (TS = '010')	ITR3 (TS = '011')
TIM9	TIM2	TIM3	TIM10_OC	TIM11_OC
TIM12	TIM4	TIM5	TIM13_OC	TIM14_OC

18.4.3 TIM9/12 割り込み有効レジスタ（TIMx_DIER）

アドレスオフセット：0x0C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
									rW				rW	rW	rW

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE**：トリガ割り込みイネーブル

0：トリガ割り込みは無効です。

1：トリガ割り込みは有効です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IE**：キャプチャ／比較 2 割り込みイネーブル

0：CC2 割り込みは無効です。

1：CC2 割り込みは有効です。

ビット 1 **CC1IE**：キャプチャ／比較 1 割り込みイネーブル

0：CC1 割り込みは無効です。

1：CC1 割り込みは有効です。

ビット 0 **UIE**：更新割り込みイネーブル

0：更新割り込みは無効です。

1：更新割り込みは有効です。

18.4.4 TIM9/12 のステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
0 : オーバーキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIF** : トリガ割り込みフラグ
このフラグは、トリガイベント時（スレープモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。
0 : トリガイベントは発生していません。
1 : トリガ割り込みが保留中です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ
CC1IF の説明を参照してください。

ビット 1 CC1IF：キャプチャ／比較 1 割り込みフラグ**CC1 チャンネルが出力として設定されている場合：**

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0：一致していません。

1：カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

CC1 チャンネルが入力として設定されている場合：

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0：入力キャプチャは発生していません。

1：カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（選択された極性に一致するエッジが IC1 で検出されました）。

ビット 0 UIF：更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0：更新は発生していません。

1：更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、トリガイベントによって CNT が再初期化されたとき（同期制御レジスタの説明を参照）。

18.4.5 TIM9/12 のイベント生成レジスタ (TIMx_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
									w				w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みが発生します。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2G** : キャプチャ/比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

現在のカウンタ値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラのカウンタもクリアされます (分周比は変化しません)。カウンタはクリアされます。

18.4.6 TIM9/12 のキャプチャ／比較モードレジスタ 1（TIMx_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のすべてのビットは、入力モードと出力モードで機能が異なります。特定のビットについて、OCxx は、チャンネルが出力モードに設定されているときの機能を記述し、ICxx は、チャンネルが入力モードに設定されているときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **OC2M[2:0]**：出力比較 2 モード

ビット 11 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 10 **OC2FE**：出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]**：キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
00：CC2 チャンネルは出力として設定されます。
01：CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。
10：CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。
11：CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC2S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC2E = 0）のときにのみ書き込み可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは、それぞれ CC1P および CC1NP ビットに依存します。

000: 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001: 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、TIMx_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にハイになります。

010: 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、TIMx_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にローになります。

011: 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100: 強制インアクティブレベル - OC1REF は強制的にローになります。

101: 強制アクティブレベル - OC1REF は強制的にハイになります。

110: PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF = 0) に、そうでない場合はアクティブ (OC1REF = 1) になります。

111: PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

注: PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0: TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1: TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注: PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0: CC1 は、トリガがオンのときでも、カウンタと CCR1 の値に依存して、通常どおりに動作します。トリガ入力にエッジが発生してから CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1: トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00: CC1 チャンネルは出力として設定されます。

01: CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10: CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11: CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注: CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E = 0) のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。**

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは $f_{DTS}/1000$: $f_{SAMPLING}=f_{DTS}/8$, N = 6 で行われます。

0001 : $f_{SAMPLING}=f_{CK_INT}$, N = 21001 : $f_{SAMPLING}=f_{DTS}/8$, N = 8

0010 : $f_{SAMPLING}=f_{CK_INT}$, N = 41010 : $f_{SAMPLING}=f_{DTS}/16$, N = 5

0011 : $f_{SAMPLING}=f_{CK_INT}$, N = 81011 : $f_{SAMPLING}=f_{DTS}/16$, N = 6

0100 : $f_{SAMPLING}=f_{DTS}/2$, N = 61100 : $f_{SAMPLING}=f_{DTS}/16$, N = 8

0101 : $f_{SAMPLING}=f_{DTS}/2$, N = 81101 : $f_{SAMPLING}=f_{DTS}/32$, N = 5

0110 : $f_{SAMPLING}=f_{DTS}/4$, N = 61110 : $f_{SAMPLING}=f_{DTS}/32$, N = 6

0111 : $f_{SAMPLING}=f_{DTS}/4$, N = 81111 : $f_{SAMPLING}=f_{DTS}/32$, N = 8

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E = 0) のときにのみ書き込み可能です。**

18.4.7 TIM9/12 のキャプチャ／比較有効レジスタ (TIMx_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								rW		rW	rW	rW		rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性
CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 相補出力極性
CC1 チャンネルが出力として設定されている場合 : CC1NP はクリア状態に維持する必要があります。
CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性
CC1 チャンネルが出力として設定されている場合 :
0 : OC1 はアクティブハイです。
1 : OC1 はアクティブローです。
CC1 チャンネルが入力として設定されている場合 :
CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。
00 : 非反転／立ち上がりエッジ
回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。
01 : 反転／立ち下がりエッジ
回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。
10 : 予約済み。この設定は使用しないでください。

注 : 11 : 非反転／両エッジ
回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル
CC1 チャンネルが出力として設定されている場合 :
0 : オフ - OC1 はアクティブではありません。
1 : オン - OC1 信号は、対応する出力ピンに出力されます。
CC1 チャンネルが入力として設定されている場合 :
このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。
0 : キャプチャは無効です。
1 : キャプチャは有効です。

表 99. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx = 0、OCx_EN = 0)
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

18.4.8 TIM9/12 のカウンタ（TIMx_CNT）

アドレスオフセット：0x24

リセット値：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

18.4.9 TIM9/12 のプリスケアラ（TIMx_PSC）

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブなプリスケアラレジスタにロードされる値を含みます。

18.4.10 TIM9/12 の自動再ロードレジスタ（TIMx_ARR）

アドレスオフセット：0x2C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

APR の更新と動作の詳細については、[セクション 18.3.1：タイムベースユニット](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

18.4.11 TIM9/12 のキャプチャ／比較レジスタ 1 (TIMx_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブなキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

18.4.12 TIM9/12 のキャプチャ／比較レジスタ 2 (TIMx_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR2[15:0]** : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブなキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

18.4.13 TIM9/12 のレジスタマップ

TIM9/12 のレジスタは、次のように 16 ビットのアドレス可能レジスタとしてマップされます：

表 100. TIM9/12 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIMx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CKD [1:0]	ARPE	Res	Res	Res	Res	OPM	URS	UDIS	CEN	
	リセット値																							0	0	0				0	0	0	0	
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSM	TS[2:0]		Res	SMS[2:0]		Res		
	リセット値																									0					0	0	0	
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIE	Res	Res	Res	CC2IE	CC1IE	UIE	
	リセット値																										0				0	0	0	
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIF	Res	Res	Res	CC2IF	CC1IF	UIF
	リセット値																							0	0						0	0	0	
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	Res	Res	CC2G	CC1G	UG
	リセット値																											0				0	0	0
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC2M [2:0]	OC2PE OC2FE		CC2S [1:0]	Res	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]		Res			
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0		
	TIMx_CCMR1 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]		IC2 PSC [1:0]	CC2S [1:0]	IC1F[3:0]		IC1 PSC [1:0]	CC1S [1:0]		Res					
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																										0		0	0	0	0	0	
0x24	TIMx_CNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	TIMx_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	

表 100. TIM9/12 レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIMx_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38	TIMx_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x3C～ 0x4C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

18.5 TIM10/11/13/14 レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読み出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

18.5.1 TIM10/11/13/14 制御レジスタ 1（TIMx_CR1）

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN
						r/w	r/w	r/w					r/w	r/w	r/w

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD**：クロック分周

このビットフィールドは、タイマクロック（CK_INT）周波数と、デジタルフィルタ（TIX）によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。
1: TIMx_ARR レジスタはバッファされます。

ビット 6:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **URS**：更新リクエストソース

このビットは、更新割り込み（UEV）ソースを選択するために、ソフトウェアによってセット／クリアされます。

0: 次のイベントのいずれかが UEV を発生します（有効な場合）。

- カウンタオーバーフロー
- UG ビットのセット

1: カウンタオーバーフローのみが UEV を発生します（有効な場合）。

ビット 1 **UDIS**：更新ディセーブル

このビットは、更新割り込み（UEV）イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0: UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー
- UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1: UEV は無効です。UEV は生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。カウンタとブリスケーラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN**：カウンタイネーブル

0: カウンタは無効です。
1: カウンタは有効です。

18.5.2 TIM10/11/13/14 割り込み有効レジスタ (TIMx_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

18.5.3 TIM10/11/13/14 ステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:2 予約済みであり、リセット値に保持する必要があります。

- ビット 1 **CC1IF** : キャプチャ／比較 1 割り込みフラグ
- CC1 チャンネルが出力として設定されている場合 :
このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。
0 : 一致していません。
1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。
CC1 チャンネルが入力として設定されている場合 :
このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。
0 : 入力キャプチャは発生していません。
1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（選択された極性に一致するエッジが IC1 で検出されました）。
- ビット 0 **UIF** : 更新割り込みフラグ
- このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。
0 : 更新は発生していません。
1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。
 - オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
 - TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

18.5.4 TIM10/11/13/14 のイベント生成レジスタ（TIMx_EGR）

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

- ビット 15:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **CC1G** : キャプチャ／比較 1 生成
- このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : チャンネル 1 でキャプチャ／比較イベントが生成されます。
CC1 チャンネルが出力として設定されている場合 :
CC1IF フラグがセットされ、対応する割り込みが送信されます（有効な場合）。
CC1 チャンネルが入力として設定されている場合 :
カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。
- ビット 0 **UG** : 更新生成
- このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。カウンタはクリアされます。

18.5.5 TIM10/11/13/14 のキャプチャ／比較モードレジスタ 1（TIMx_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]			IC1PSC[1:0]				
								rW	rW	rW	rW	rW	rW	rW	rW

出力比較モード

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 の元となる出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 のアクティブレベルは CC1P ビットに依存します。

000 : 停止。出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

111 : PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

注： PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときに、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注： PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 :

11 :

注： CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは $f_{DTS}/1000$: $f_{SAMPLING}=f_{DTS}/8$, $N = 6$ で行われます。

0001 : $f_{SAMPLING}=f_{CK_INT}$, $N = 2$ 1001 : $f_{SAMPLING}=f_{DTS}/8$, $N = 8$

0010 : $f_{SAMPLING}=f_{CK_INT}$, $N = 4$ 1010 : $f_{SAMPLING}=f_{DTS}/16$, $N = 5$

0011 : $f_{SAMPLING}=f_{CK_INT}$, $N = 8$ 1011 : $f_{SAMPLING}=f_{DTS}/16$, $N = 6$

0100 : $f_{SAMPLING}=f_{DTS}/2$, $N = 6$ 1100 : $f_{SAMPLING}=f_{DTS}/16$, $N = 8$

0101 : $f_{SAMPLING}=f_{DTS}/2$, $N = 8$ 1101 : $f_{SAMPLING}=f_{DTS}/32$, $N = 5$

0110 : $f_{SAMPLING}=f_{DTS}/4$, $N = 6$ 1110 : $f_{SAMPLING}=f_{DTS}/32$, $N = 6$

0111 : $f_{SAMPLING}=f_{DTS}/4$, $N = 8$ 1111 : $f_{SAMPLING}=f_{DTS}/32$, $N = 8$

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : 予約済み

11 : 予約済み

注: **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

18.5.6 TIM10/11/13/14 のキャプチャ／比較有効レジスタ（TIMx_CCER）

アドレスオフセット：0x20

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP**：キャプチャ／比較 1 相補出力極性
CC1 チャンネルが出力として設定されている場合：CC1NP はクリアされたままにする必要があります。
CC1 チャンネルが入力として設定されている場合：CC1NP ビットは、TI1FP1 の極性を定義するために CC1P と組み合わせて使用されます（CC1P の説明を参照してください）。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P**：キャプチャ／比較 1 出力極性
CC1 チャンネルが出力として設定されている場合：
0：OC1 はアクティブハイです。
1：OC1 はアクティブローです。
CC1 チャンネルが入力として設定されている場合：
CC1P ビットは、トリガ動作またはキャプチャ動作での TI1FP1 および TI2FP1 の極性を選択します。
00：非反転／立ち上がりエッジ
回路は TI1FP1 の立ち上がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されません。
01：反転／立ち下がりエッジ
回路は TI1FP1 の立ち下がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されます。
10：予約済み。この設定は使用しないでください。
11：非反転／両エッジ
回路は TI1FP1 の立ち上がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されません。

ビット 0 **CC1E**：キャプチャ／比較 1 出力エナブル。
CC1 チャンネルが出力として設定されている場合：
0：オフ - OC1 はアクティブではありません。
1：オン - OC1 信号は、対応する出力ピンに出力されます。
CC1 チャンネルが入力として設定されている場合：
このビットによって、カウンタ値のキャプチャ／比較レジスタ 1（TIMx_CCR1）へのキャプチャが実際に行われるかどうかが決まります。
0：キャプチャは無効です。
1：キャプチャは有効です。

表 101. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効（OCx=0、OCx_EN=0）
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

18.5.7 TIM10/11/13/14 のカウンタ (TIMx_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CNT[15:0]** : カウンタ値

18.5.8 TIM10/11/13/14 プリスケアラ (TIMx_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

18.5.9 TIM10/11/13/14 自動再ロードレジスタ (TIMx_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **ARR[15:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

APR の更新と動作の詳細については、[セクション 18.3.1 : タイムベースユニット](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

18.5.10 TIM10/11/13/14 キャプチャ／比較レジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

18.5.11 TIM11 オプションレジスタ 1（TIM11_OR）

アドレスオフセット：0x50

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]
															rW

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TI1_RMP[1:0]**：TIM11 入力 1 再配置機能

ソフトウェアでセット／クリアされます。

00、01、11：TIM11 のチャンネル 1 は、GPIO に接続されます（データシートの代替機能配置表を参照してください）。

10：HSE_RTC クロック（HSE は、プログラム可能なプリスケアラで分周されます）は、測定目的のため、TIM11_CH1 入力に接続されます。

18.5.12 TIM10/11/13/14 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 102. TIM10/11/13/14 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	TIMx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CKD [1:0]	ARPE	Res	Res	Res	Res	Res	URS	UDIS	CEN						
	リセット値																							0	0	0				0	0	0							
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res						
	リセット値																							Res	Res	Res	Res	Res	Res										
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1IE	UIE						
	リセット値																							Res	Res	Res	Res	Res	Res			0	0						
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1OF	Res	Res	Res	Res	Res	Res	Res	CC1IF	UIF						
	リセット値																							0		Res	Res	Res	Res	Res		0	0						
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1G	UG						
	リセット値																							Res	Res	Res	Res	Res	Res			0	0						
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OC1M [2:0]		OC1PE OC1FE		CC1S [1:0]									
	リセット値																										0	0	0	0	0	0							
	TIMx_CCMR1 入力キャプチャ モード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC1F[3:0]			IC1 PSC [1:0]		CC1S [1:0]									
	リセット値																									0	0	0	0	0	0	0	0						
0x1C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res						
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1NP	CC1P	CC1E							
	リセット値																												0		0	0							
0x24	TIMx_CNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x2C	TIMx_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x30	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					

表 102. TIM10/11/13/14 レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38～ 0x4C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x50	TIMx_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM_RMP	
	リセット値																														0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

19 基本タイマ (TIM6/7)

19.1 概要

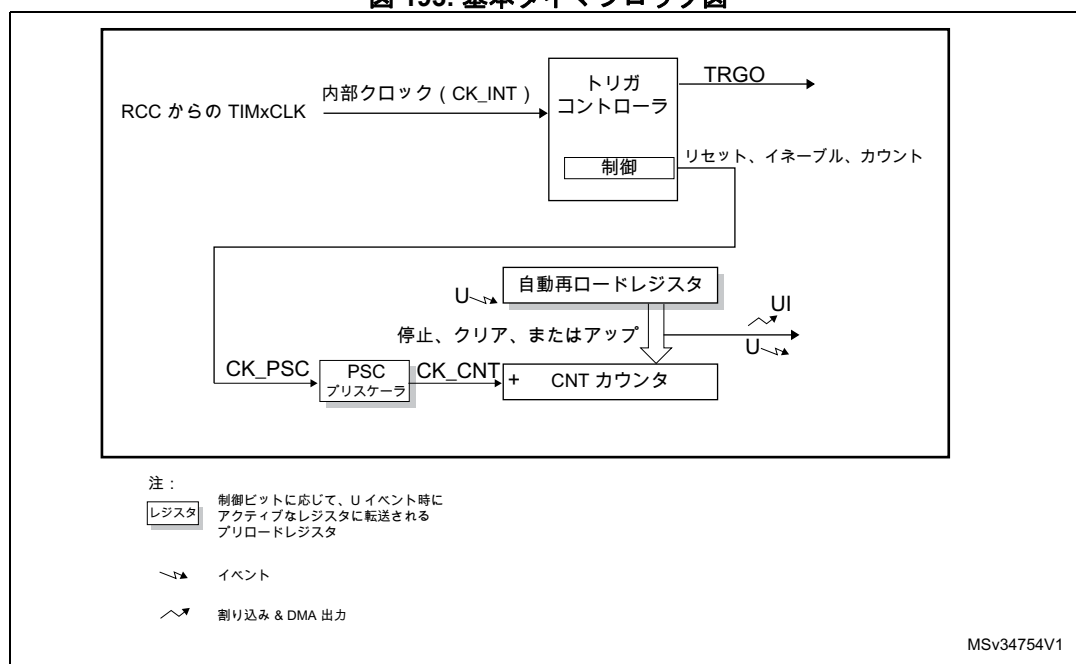
基本タイマ TIM6、TIM7 は、プログラム可能なプリスケアラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

19.2 TIM6/7 の主な機能

基本タイマ (TIM6/TIM7) の機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ (動作中も変更可能) で、カウンタクロック周波数を 1 から 65536 の間の値で分周可能。
- 更新イベント時の割り込み/DMA 生成: カウンタオーバーフロー

図 193. 基本タイマブロック図



19.3 TIM6/7 の機能説明

19.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きはプリロードレジスタへのアクセスとなります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント UEV ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、カウンタがオーバーフロー値に達すると、更新イベントが送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウントイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 CK_CNT から供給されます。

実際のカウンタイネーブル信号 CNT_EN は、CEN の 1 クロックサイクル後にセットされます。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。TIMx_PSC 制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 194](#) と [図 195](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 194. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

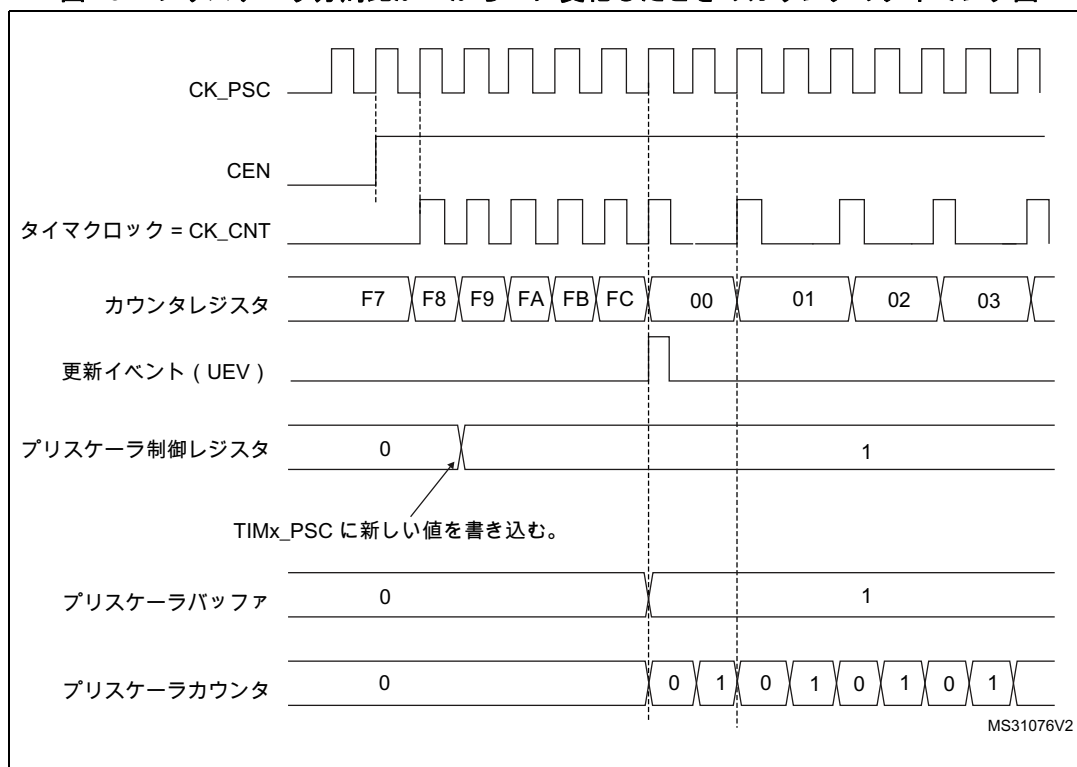
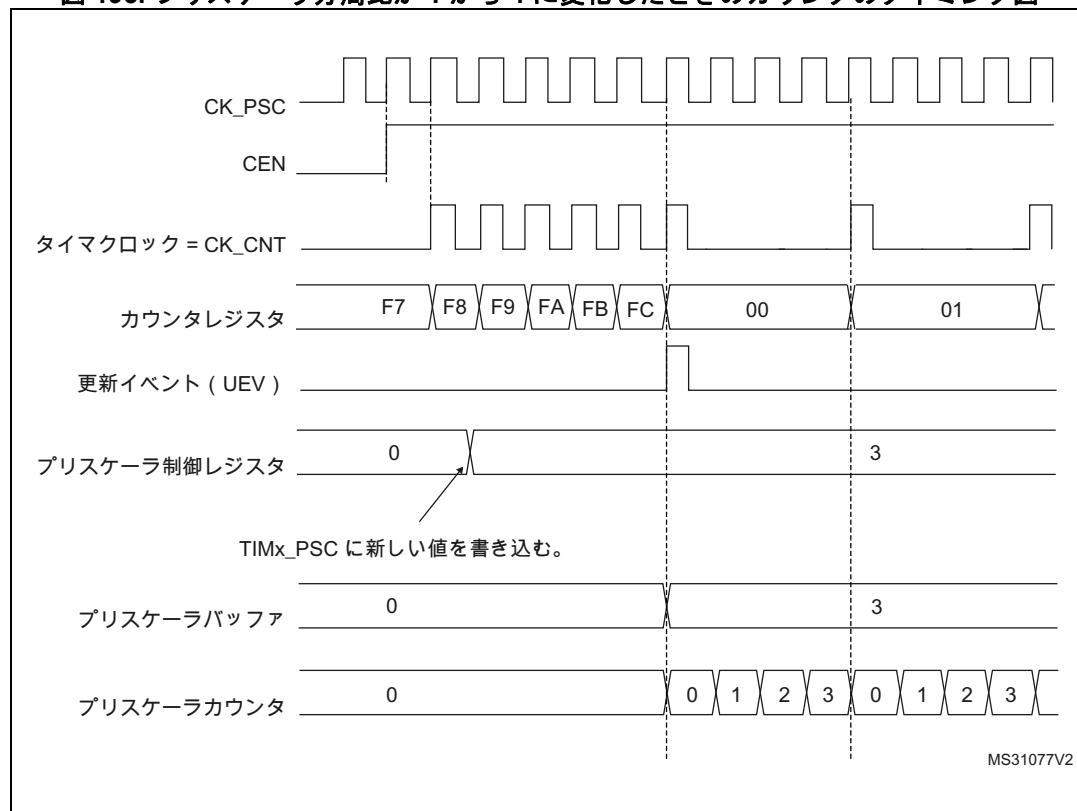


図 195. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



19.3.2 カウントモード

カウンタは、0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントした後、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、TIMx_EGR レジスタの UG ビットをセットすることによって (ソフトウェアで、または、スレーブモードコントローラを使用して) 生成できます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。このようにすると、UDIS ビットに 0 が書き込まれるまで更新イベントは発生しませんが、カウンタとプリスケアラカウンタは両方とも 0 からリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS (更新リクエスト選択) ビットがセットされている場合、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (割り込みや DMA リクエストは送信されません)。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットの設定に応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 196. 内部クロック分周比が 1 の場合のカウンタのタイミング図

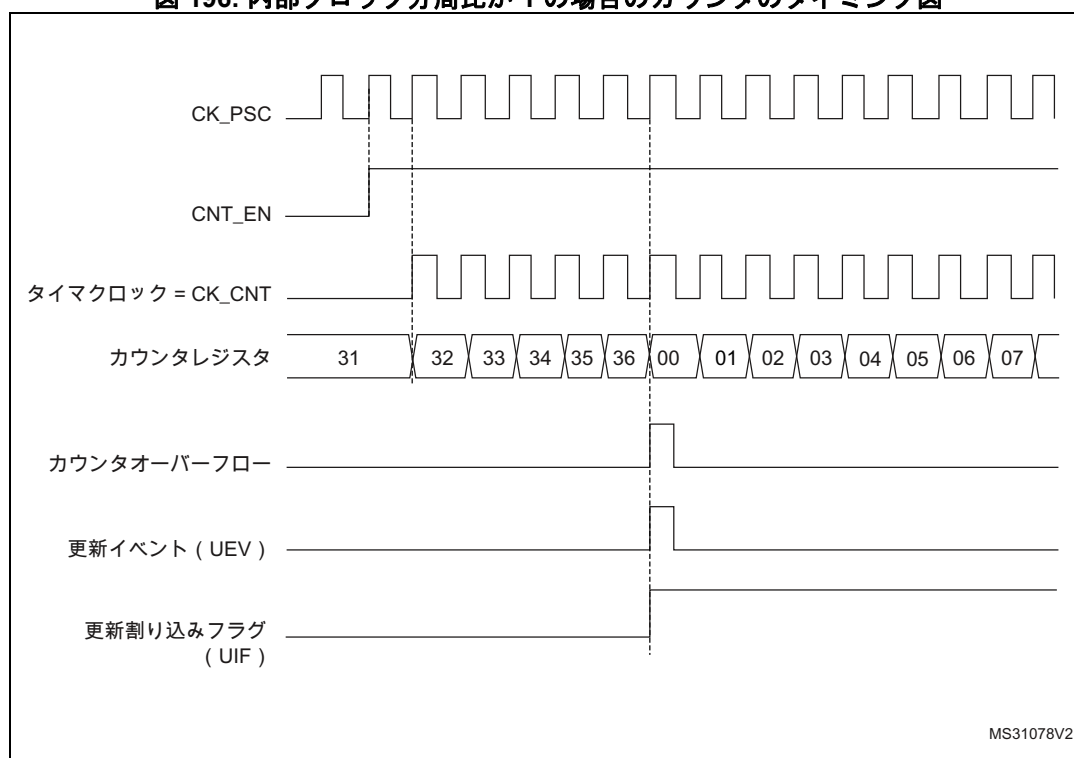


図 197. 内部クロック分周比が 2 の場合のカウンタのタイミング図

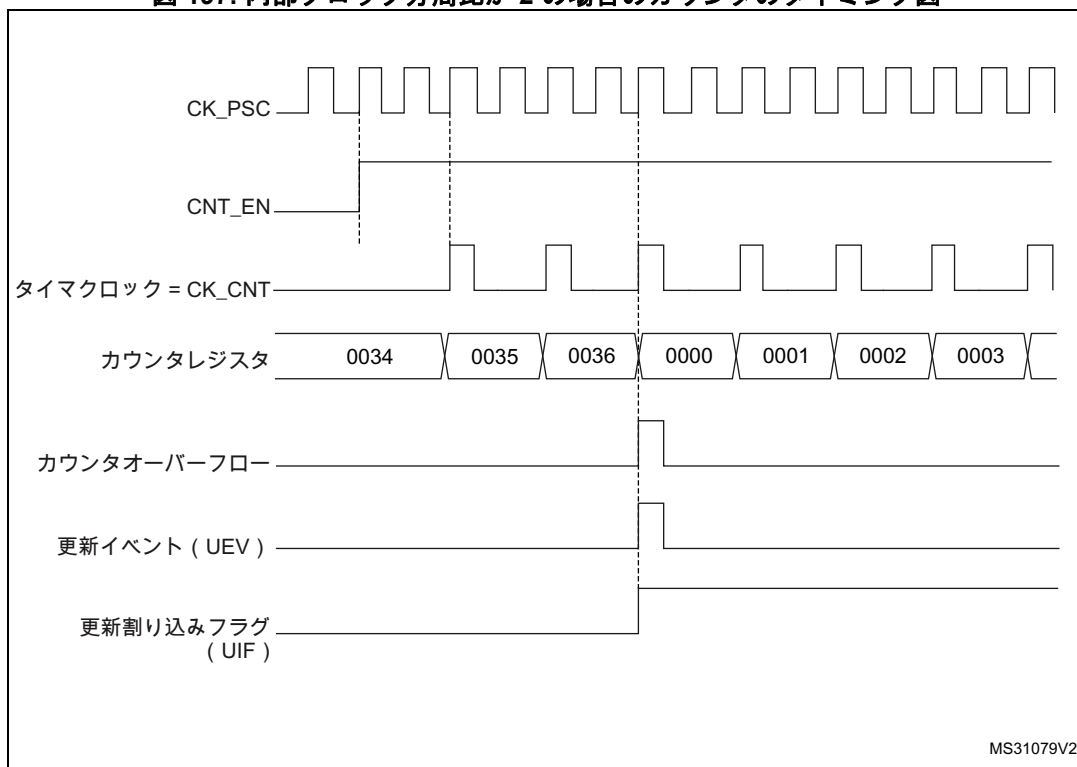


図 198. 内部クロック分周比が 4 の場合のカウンタのタイミング図

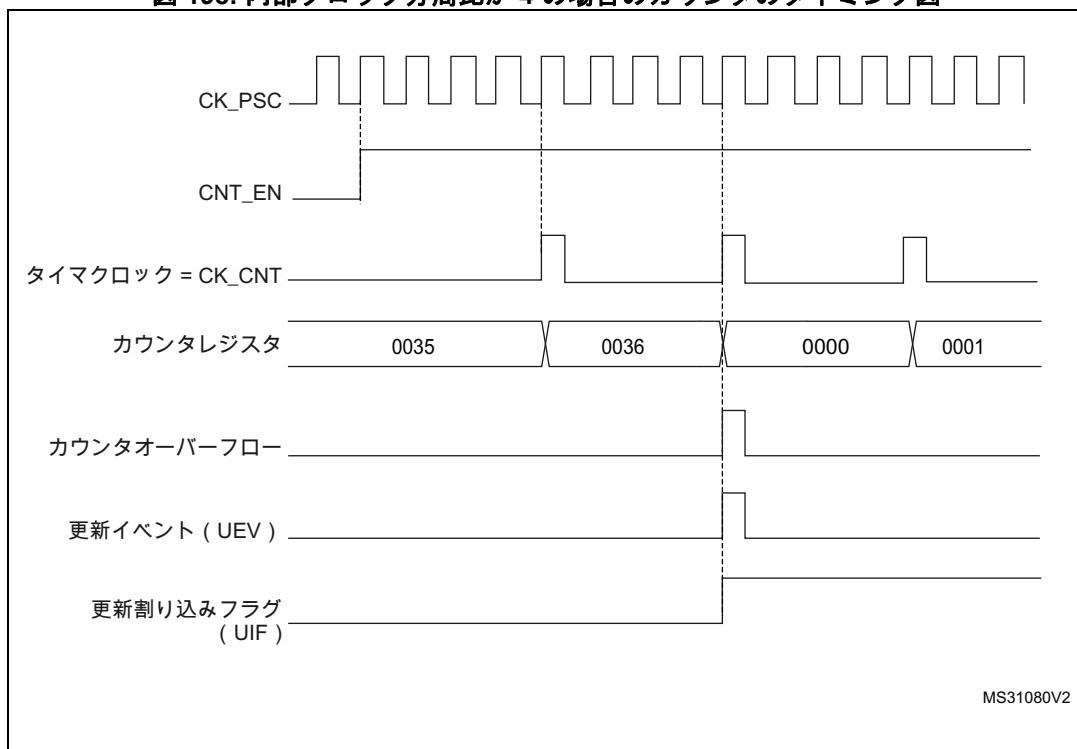


図 199. 内部クロック分周比が N の場合のカウンタのタイミング図

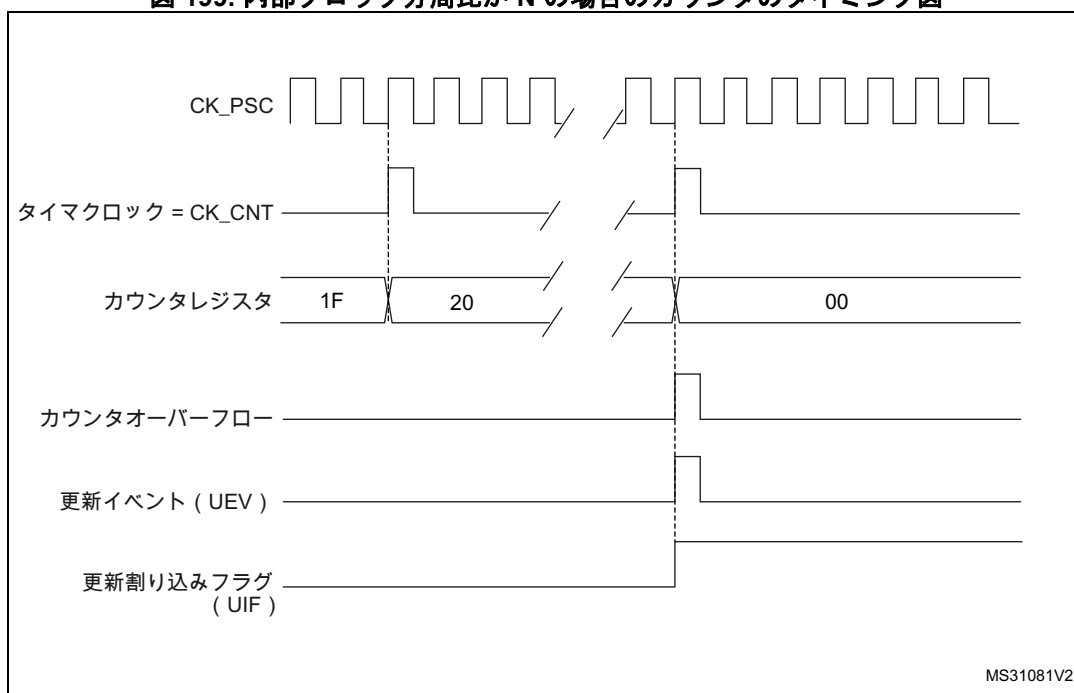


図 200. ARPE=0 の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR はプリロードされない)

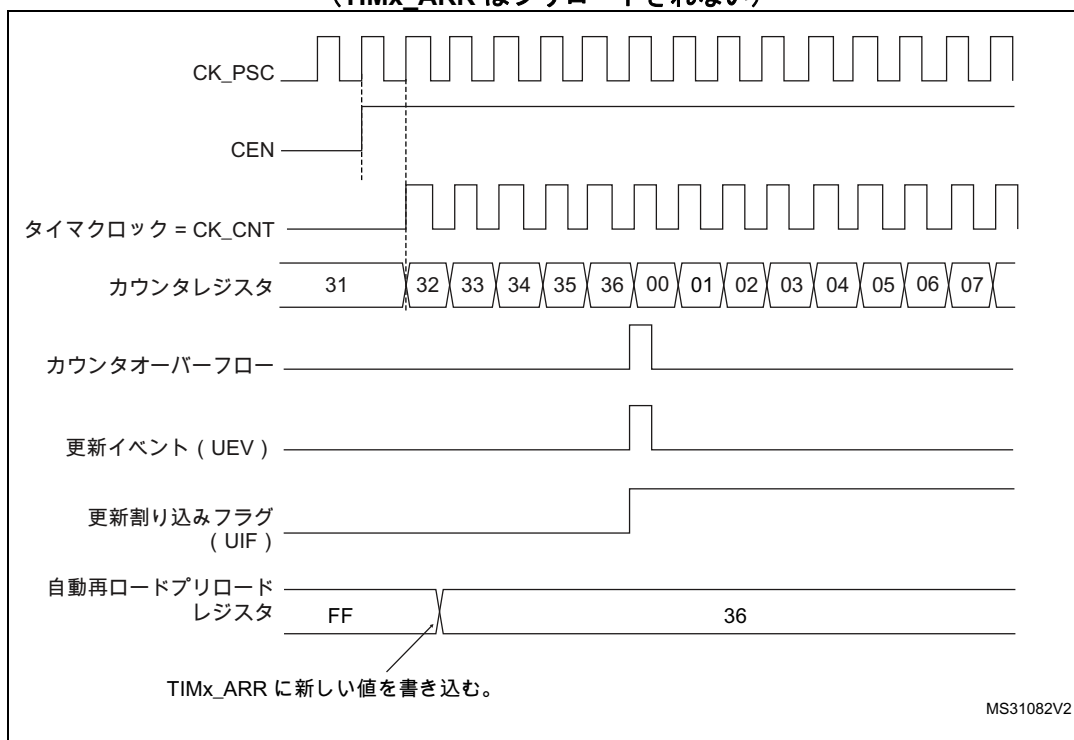
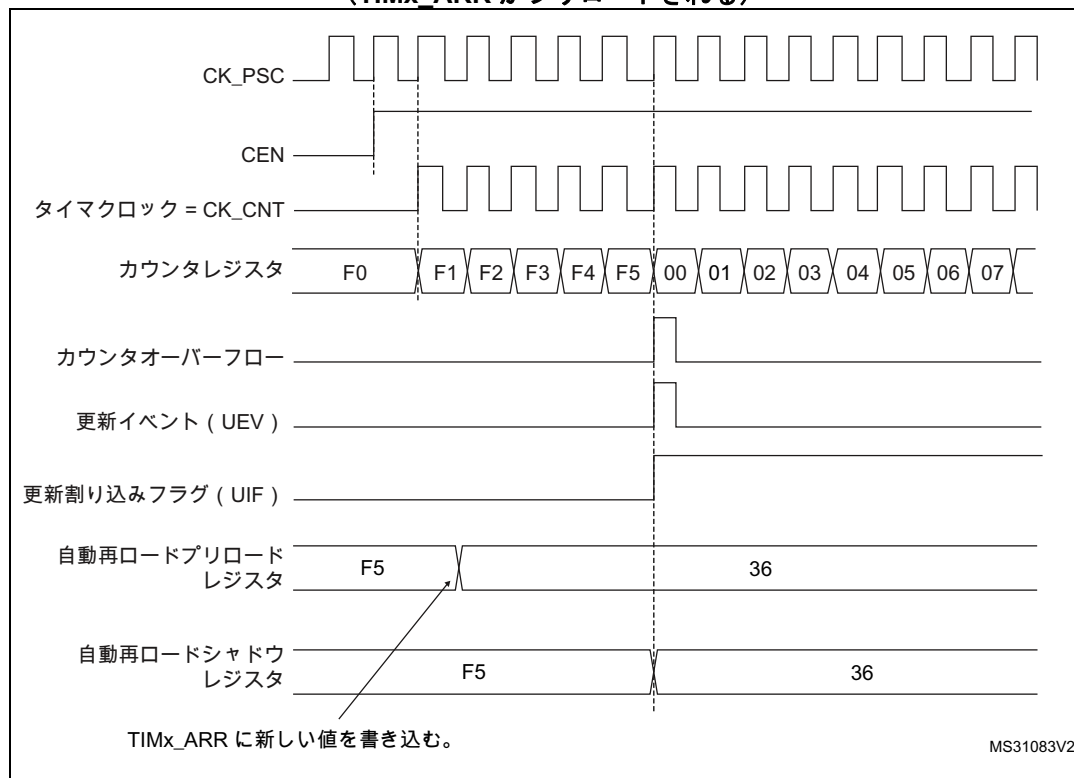


図 201. ARPE=1 の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR がプリロードされる)



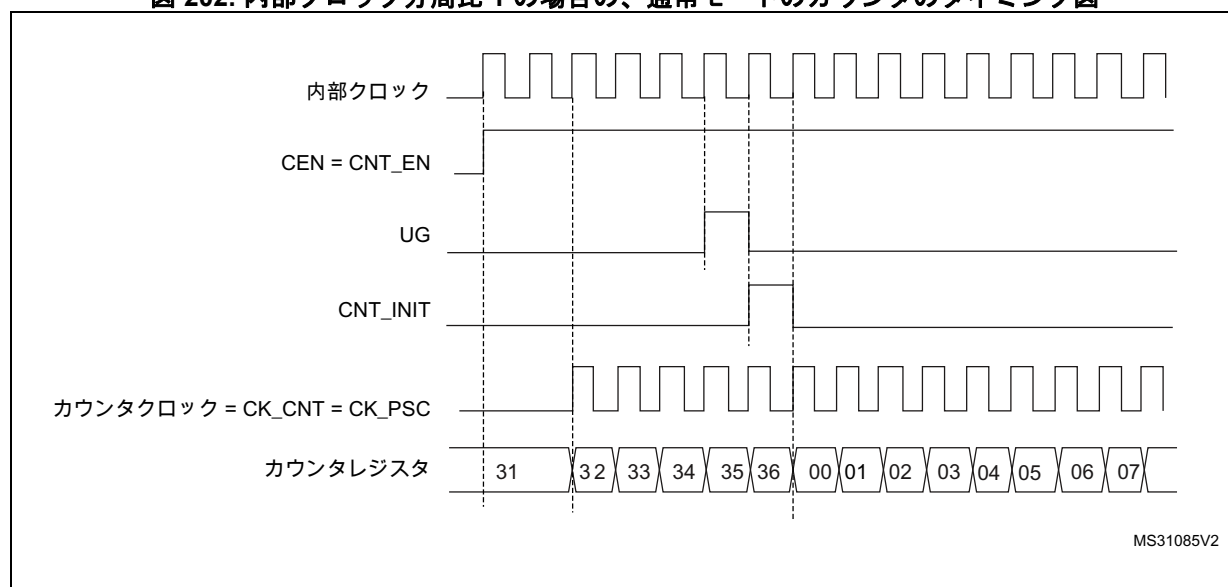
19.3.3 クロックソース

カウンタクロックは、内部クロック (CK_INT) ソースから供給されます。

TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットは実際の制御ビットであり、ソフトウェアによってのみ変更できます (ただし、自動的にクリア状態が保持される UG ビットを除く)。CEN ビットに 1 が書き込まれると、プリスケラにはクロックとして内部クロック CK_INT が供給されます。

図 202 に、プリスケラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

図 202. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



19.3.4 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 30.16.2: タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

19.4 TIM6/7 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 : レジスタに関する略記](#)を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読み出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

19.4.1 TIM6/7 制御レジスタ 1 (TIMx_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0 : TIMx_ARR レジスタはバッファされません。
- 1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC) はそれぞれの値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケールは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注 : ゲートモードは、**CEN** ビットが事前にソフトウェアでセットされている場合にのみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に **CEN** ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

19.4.2 TIM6/7 制御レジスタ 2 (TIMx_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **MMS** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (TRGO) を選択するために使用します。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが発生したとき (スレーブモードコントローラがリセットモードに設定されているとき) TRGO 信号は実際のリセットから遅れて発生します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除いて、TRGO に遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

19.4.3 TIM6/7 DMA/割り込み有効レジスタ (TIMx_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

19.4.4 TIM6/7 のステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- TIMx_CR1 レジスタの UDIS = 0 であり、繰り返しカウンタ値でオーバーフローまたはアンダーフローが発生したとき。
- TIMx_CR1 レジスタの URS = 0 かつ UDIS = 0 の場合に、TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで CNT が再初期化されたとき。

19.4.5 TIM6/7 のイベント生成レジスタ (TIMx_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : タイマカウンタを再初期化して、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

19.4.6 TIM6/7 のカウンタ (TIMx_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CNT[15:0]** : カウンタ値

19.4.7 TIM6/7 のプリスケアラ (TIMx_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **PSC[15:0]** : プリスケアラ値
カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。
PSC は、更新イベントごとにアクティブなプリスケアラレジスタにロードされる値を含みます。

19.4.8 TIM6/7 の自動再ロードレジスタ (TIMx_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **ARR[15:0]** : 自動再ロード値
ARR は、実際の自動再ロードレジスタにロードされる値です。
APR の更新と動作の詳細については、[セクション 19.3.1: タイムベースユニット \(576 ページ\)](#) を参照してください。
自動再ロード値が null のときには、カウンタはブロックされます。

19.4.9 TIM6/7 のレジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 103. TIM6/7 レジスタマップとリセット値

オフセット	レジスタ	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	リセット値									0				0	0	0	0
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
	リセット値										0	0	0				
0x08	Res.																
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
	リセット値								0								0
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
	リセット値																0
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
	リセット値																0
0x18	Res.																
0x1C	Res.																
0x20	Res.																
0x24	TIMx_CNT	CNT[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIMx_PSC	PSC[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	ARR[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

20 独立型ウォッチドッグ (IWDG)

20.1 IWDG の概要

デバイスは、2つの内蔵ウォッチドッグペリフェラルを搭載しており、使用上、高い安全レベル、タイミングの正確さ、および柔軟性を兼ね備えています。2つのウォッチドッグペリフェラル（独立型とウィンドウ型）は、ソフトウェア障害による誤動作を検出および解決し、カウンタが与えられたタイムアウト値に達すると、システムリセットまたは割込み（ウィンドウ型ウォッチドッグのみ）をトリガします。


独立型ウォッチドッグ (IWDG) は、独自の低速クロック (LSI) によってクロック供給されるので、メインクロックに障害があってもアクティブなままです。ウィンドウ型ウォッチドッグ (WWDG) のクロックは、APB1 クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、タイミング精度が低いという制約があります。WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。ウィンドウ型ウォッチドッグの詳細については、[セクション 21：ウィンドウ型ウォッチドッグ \(WWDG\)](#) を参照してください。

20.2 IWDG の主な機能

- フリーランニングダウンカウンタ
- 独立した RC オシレータからのクロック供給 (STANDBY および STOP モードで動作可能)
- ダウンカウンタ値が 0x000 に達したときにリセット (ウォッチドッグが有効な場合)

20.3 IWDG の機能説明

 **図 203** に、独立型ウォッチドッグモジュールの機能ブロックを示します。

キーレジスタ (IWDG_KR) に値 0xCCCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFFFF からカウントダウンを開始します。カウント値の終わり (0x000) に達すると、リセット信号が生成されます (IWDG_reset)。

IWDG_KR レジスタにキー値 0xAAAA が書き込まれると、IWDG_RLR の値がカウンタに再ロードされ、ウォッチドッグのリセットが防止されます。

20.3.1 ハードウェアウォッチドッグ

デバイスのオプションビットを通じて「ハードウェアウォッチドッグ」機能が有効な場合、ウォッチドッグは電源投入時に自動的に有効になり、カウンタがカウントの終わりに達する前にソフトウェアによってキーレジスタへ書き込まれない限り、リセットを生成します。

20.3.2 レジスタのアクセス保護

IWDG_PR および IWDG_RLR レジスタへの書き込みアクセスは保護されます。これらを変更するには、まず、IWDG_KR レジスタにコード 0x5555 を書き込む必要があります。このレジスタに別の値を書き込むと、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0xAAAA の書き込み) であることを意味します。



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などに付きましては如何なる責任にも負いません。

20.4 IWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

20.4.1 キーレジスタ (IWDG_KR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされる)

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **KEY[15:0]** : キー値（書き込み専用、読み出しでは 0000h）

これらのビットには、ソフトウェアによって一定間隔でキー値 AAAAh が書き込まれなければなりません。そうしないと、カウンタが 0 に達した時点でウォッチドッグがリセットを生成します。

キー値 5555h を書き込むことによって、IWDG_PR および IWDG_RLR レジスタへのアクセスが可能になります（[セクション 20.3.2](#) を参照）。

キー値 CCCCh を書き込むと、ウォッチドッグが開始します（ハードウェアウォッチドッグオプションが選択されている場合を除く）。

20.4.2 プリスケーラレジスタ (IWDG_PR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **PR[2:0]** : プリスケーラ分周回路

これらのビットは、書き込みアクセス保護されています ([セクション 20.3.2](#) を参照)。カウンタクロックを供給するプリスケーラ分周回路を選択するようにソフトウェアで書き込まれます。プリスケーラ分周比を変更できるようにするには、IWDG_SR レジスタの PVU ビットをリセットする必要があります。

- 000 : 4 分周
- 001 : 8 分周
- 010 : 16 分周
- 011 : 32 分周
- 100 : 64 分周
- 101 : 128 分周
- 110 : 256 分周
- 111 : 256 分周

注 : このレジスタを読み出すと、VDD 電圧ドメインからプリスケーラ値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG_SR レジスタの PVU ビットがリセットされているときのみとなります。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

20.4.3 再ロードレジスタ (IWDG_RLR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **RL[11:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書き込みアクセス保護されています ([セクション 20.3.2](#) を参照)。IWDG_KR レジスタに値 AAAAh が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とクロックプリスケアラによって決まります。表 104 を参照してください。

再ロード値を変更できるようにするには、IWDG_SR レジスタの RVU ビットをリセットする必要があります。

注： このレジスタを読み出すと、VDD 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG_SR レジスタの RVU ビットがリセットされているときのみとなります。

20.4.4 ステータスレジスタ (IWDG_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされません)

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **RVU** : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。VDD 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

再ロード値は、RVU ビットがリセットされているときのみ更新できます。

ビット 0 **PVU** : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。VDD 電圧ドメインでプリスケアラ更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。



注： 複数の再ロード値またはプリスケアラ値がアプリケーションで使用される場合は、再ロード値を変更する前に RVU ビットがリセットされるまで待つ必要があります、またプリスケアラ値を変更する前に PVU ビットがリセットされるまで待つ必要があります。ただし、プリスケアラ値や再ロード値を更新した後は、RVU または PVU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードに入った場合でも、書き込み操作は実行され、完了します)。

20.4.5 IWDG レジスタマップ

次の表に、IWDG レジスタマップとリセット値を示します。

表 105. IWDG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IWDG_KR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	IWDG_PR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR[2:0]		
	リセット値																														0	0	0
0x08	IWDG_RLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RL[11:0]											
	リセット値																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	IWDG_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RVU	PVU
	リセット値																															0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などに付きましては如何なる責任にも負いません。

アプリケーションプログラムは、通常動作時には定期的に WWDG_CR レジスタへの書き込みを行って、MCU リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値より小さいときに限られます。WWDG_CR レジスタに格納される値は、0xFF から 0xC0 の間でなければなりません。

ウォッチドッグの有効化

ウォッチドッグはリセット後は常に無効です。これを有効にするには、WWDG_CR レジスタの WDGA ビットをセットします。この後は、リセット以外の方法でウォッチドッグを無効にすることはできません。

ダウンカウンタの制御

このダウンカウンタはフリーランニングであり、ウォッチドッグが無効状態であってもカウントダウンを続けます。ウォッチドッグを有効にするときには、T6 ビットをセットして、ただちにリセットが生成されるのを防ぐ必要があります。

T[5:0] ビットは、ウォッチドッグがリセットを生成するまでの時間遅延を表すインクリメント数を含みます。このタイミングは、WWDG_CR レジスタへの書き込み時のプリスケアラの状態が不明なので最小値から最大値の間で変化します（[図 205](#) 参照）。設定レジスタ（WWDG_CFR）は、ウィンドウの上限値を含みます。リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x3F よりも大きい間にダウンカウンタを再ロードする必要があります。[図 205](#) に、ウィンドウ型ウォッチドッグのプロセスを示します。

注： T6 ビットを使用して、ソフトウェアリセットを生成することができます（WDGA ビットはセット、T6 ビットはクリアされる）。

高度なウォッチドッグ割り込み機能

実際にリセットが生成される前に特定の安全処理やデータロギングを実施する必要がある場合は、早期ウェイクアップ割り込み（EWI）が使用できます。EWI 割り込みは、WWDG_CFR レジスタの EWI ビットをセットすることによって有効になります。ダウンカウンタ値が 0x40 に到達すると、EWI 割り込みが生成され、対応する割り込みサービスルーチン（ISR）を使用してデバイスをリセットする前に特定の処理（通信やデータロギングなど）をトリガすることができます。

アプリケーションによっては、EWI 割り込みを使用して、WWDG リセットを生成せずにソフトウェアのシステムチェックやシステム復旧／グレースフルデグラデーションを管理することができます。この場合、対応する割り込みサービスルーチン（ISR）で WWDG カウンタを再ロードし、WWDG リセットを回避してから必要な操作をトリガしてください。

EWI 割り込みは、WWDG_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされます。

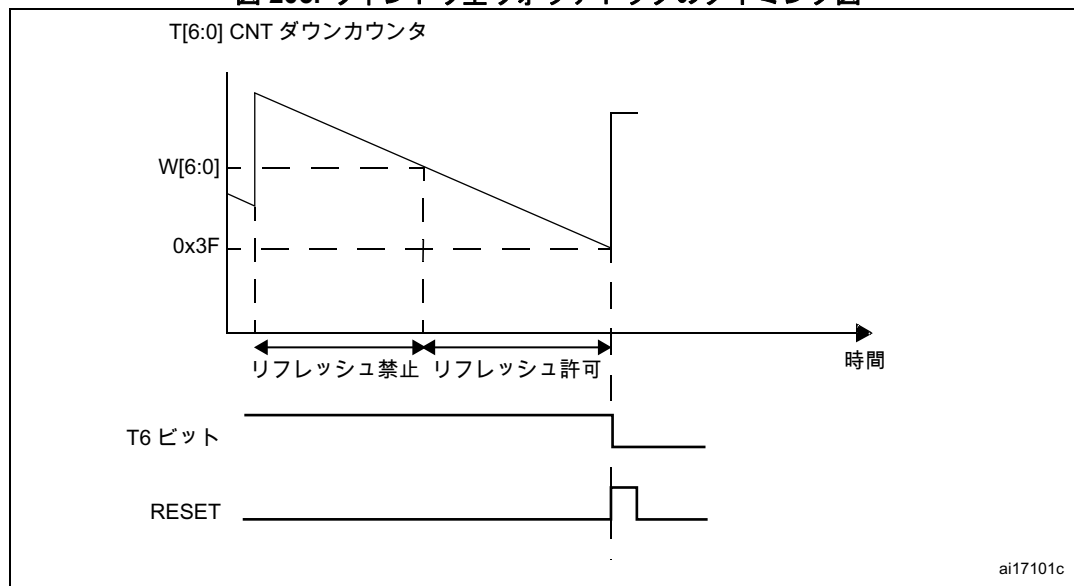
注： 優先順位の高いタスクにおけるシステムロックなどによって EWI 割り込みが使用できない場合、最終的には WWDG リセットが生成されます。

21.4 ウォッチドッグタイムアウトをプログラムする方法

WWDG のタイムアウトを計算するには、[図 205](#) の式を使用する必要があります。

警告： WWDG_CR レジスタに書き込むときには、ただちにリセットされるのを防ぐために、常に T6 ビットに 1 を書き込んでください。

図 205. ウィンドウ型ウォッチドッグのタイミング図



タイムアウト値は次の式で算出されます。

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (\text{T}[5:0] + 1) \quad (\text{ms})$$

ここで、

t_{WWDG} : WWDG タイムアウト

t_{PCLK1} : APB1 クロック周期 (ms)

4096 : 内部分周器に対応する値

たとえば、APB1 周波数が 24 MHz と等しい場合、WDGTB[1:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = 1/24000 \times 4096 \times 2^3 \times (63 + 1) = 21.85 \text{ ms}$$

t_{WWDG} の最小値と最大値については、データシートを参照してください。

21.5 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、WWDG カウンタは、DBG モジュールの DBG_WWDG_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 30.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

21.6 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

21.6.1 制御レジスタ (WWDG_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 007F

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA** : 有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。
WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。
0 : ウォッチドッグは無効です。
1 : ウォッチドッグは有効です。

ビット 6:0 **T[6:0]** : 7 ビットカウンタ (MSB から LSB まで)

これらのビットは、ウォッチドッグカウンタの値を含みます。(4096 x 2^{WDGTB[1:0]}) PCLK1 サイクルごとにデクリメントされます。0x40 に達して 0x3F に戻ると (T6 がクリアされると)、リセットが生成されます。

21.6.2 設定レジスタ (WWDG_CFR)

アドレスオフセット : 0x04

リセット値 : 0x0000 007F

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

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割り込み

このビットがセットされているときには、カウンタの値が 0x40 に達したときに割り込みが発生します。
この割り込みは、リセット後にハードウェアによってのみクリアされます。

ビット 8:7 **WDGTB[1:0]** : タイマーベース

プリスケアラのタイムベースは、次のように変更できます。

- 00 : CK カウンタクロック (PCLK1/4096) /1
- 01 : CK カウンタクロック (PCLK1/4096) /2
- 10 : CK カウンタクロック (PCLK1/4096) /4
- 11 : CK カウンタクロック (PCLK1/4096) /8

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

21.6.3 ステータスレジスタ (WWDG_SR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EWIF** : 早期ウェイクアップ割り込みフラグ

このビットは、カウンタの値が 0x40 に達したときにハードウェアによってセットされます。“0”を書き込んでソフトウェアでクリアする必要があります。“1”を書き込んでも、ビットの値は変化しません。このビットは、割り込みが有効でない場合にもセットされます。

21.6.4 WWDG レジスタマップ

次の表に、WWDG のレジスタマップとリセット値を示します。

表 106. WWDG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]										
	リセット値																								0	1	1	1	1	1	1	1				
0x04	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB1	WDGTB0	W[6:0]										
	リセット値																						0	0	0	1	1	1	1	1	1	1				
0x08	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF				
	リセット値																															0				

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

22 リアルタイムクロック (RTC)

22.1 概要

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。この RTC には、時刻クロック/カレンダー、プログラム可能なアラーム割り込みを 2 本、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグを搭載しています。RTC には、低電力モードを管理する自動ウェイクアップユニットも搭載されています。

2 つの 32 ビットレジスタには、2 進化 10 進数形式 (BCD) で表現した秒、分、時 (12 時間または 24 時間形式)、曜日、日、月、年が含まれています。サブセカンドの値もバイナリ形式で利用できます。

28 日、29 日 (うるう年)、30 日、31 日の補正は、自動的に行われます。サマータイム補正も行われます。

サブセカンド、秒、分、時、曜日、日付のプログラム可能なアラームを備えた 32 ビットレジスタが追加されています。

クリスタルオシレータ精度の偏差を補正するために、デジタル較正機能が利用可能です。

バックアップドメインリセット後、すべての RTC レジスタは、起こりうる不要な書き込みアクセスから保護されます。

供給電圧が動作範囲内にある間は、デバイスのステータス (実行モード、低電力モード、またはリセット中) に関係なく、RTC が停止することはありません。

22.2 RTC の主な機能

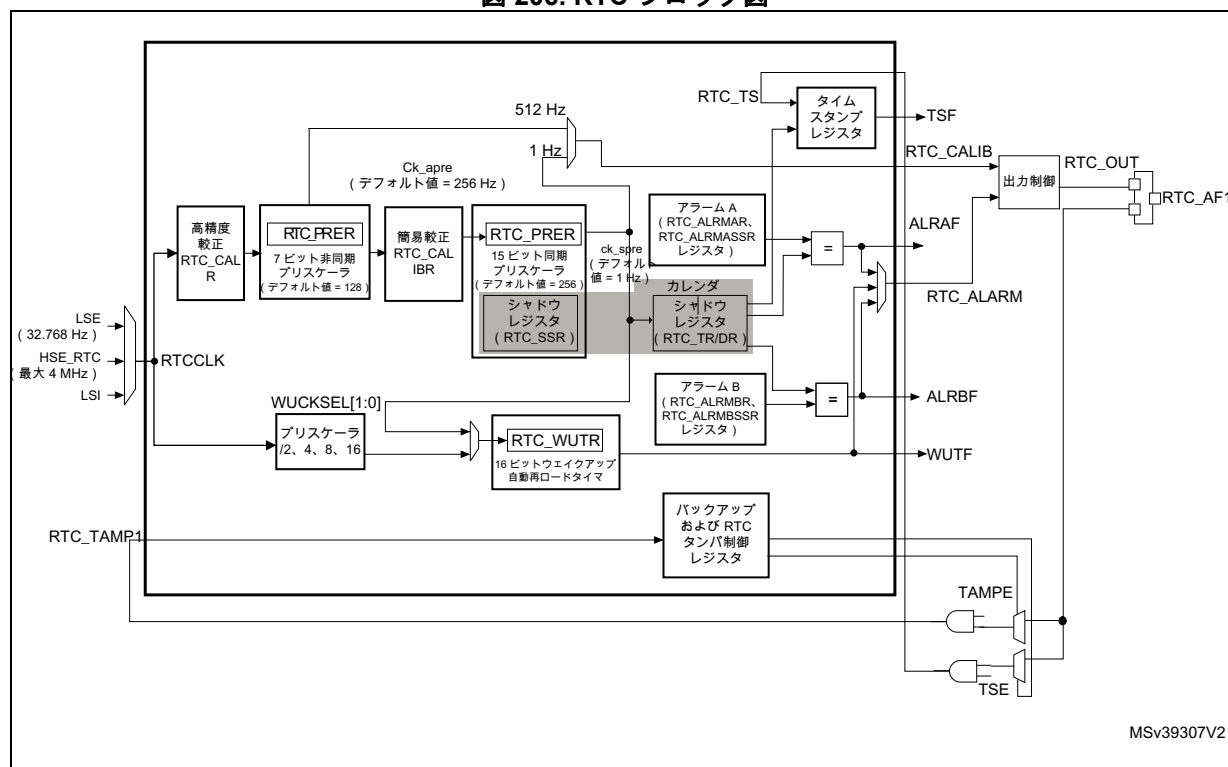
RTC ユニットの主な機能を以下に示します (図 206 を参照)。

- サブセカンド、秒、分、時 (12 または 24 時間形式)、曜日、日、月、年に対応するカレンダー。
- ソフトウェアでプログラム可能なサマータイム補正。
- 割り込み機能を備えた 2 本のプログラム可能なアラーム。アラームは、カレンダー項目のどの組み合わせでもトリガできます。
- 自動ウェイクアップ割り込みをトリガする周期的なフラグを生成する自動ウェイクアップユニット。
- リファレンスクロック検出: より正確な秒のクロックソース (50 または 60 Hz) の使用で、カレンダーの精度を向上。
- サブセカンドシフト機能を使用する外部クロックとの正確な同期。
- マスク可能な割り込み/イベント:
 - アラーム A
 - アラーム B
 - ウェイクアップ割り込み
 - タイムスタンプ
 - タンパ検出
- デジタル較正回路 (周期カウンタ修正)
 - 5 ppm の精度
 - 数秒の較正時間範囲で得られる 0.95 ppm の精度。
- イベントを保存するタイムスタンプ機能 (1 イベント)
- タンパ検出:
 - 設定可能なフィルタおよび内部プルアップのある 2 つのタンパイベント。

- 20 個のバックアップレジスタ (80 バイト)。このバックアップレジスタは、タンパ検出イベントが発生するとリセットされます。
- 次の 2 つの出力のうち 1 つを選択するオルタネート機能出力 RTC-OUT :
 - RTC_CALIB : 512 Hz または 1 Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。この出力は、RTC_CR レジスタの COE ビットをセットして有効にします。この出力はデバイスの RTC_AF1 機能に送られます。
 - RTC_ALARM (アラーム A、アラーム B またはウェイクアップ)。この出力は、RTC_CR レジスタの OSEL[1:0] ビットを設定して選択します。この出力はデバイスの RTC_AF1 機能に送られます。
- RTC オルタネート機能入力 :
 - RTC_TS : タイムスタンプイベント検出。この出力はデバイスの RTC_AF1 に送られます。
 - RTC_TAMP1 : TAMPER1 イベント検出。この出力はデバイスの RTC_AF1 に送られます。
 - RTC_REFIN : リファレンスクロック入力 (通常は 50 または 60 Hz の商用電源)。

セクション 7.3.15 : RTC 追加機能の選択を参照してください。

図 206. RTC ブロック図



MSv39307V2

22.3 RTC の機能説明

22.3.1 クロックとプリスケアラ

RTC クロックソース (RTCCLK) は、LSE クロック、LSI オシレータクロック、HSE クロックのうちから、クロックコントローラを介して選択されます。RTC クロックソースの設定に関する詳細は、[セクション 6 : STM32F412xx のリセットおよびクロック制御 \(RCC\)](#) を参照してください。

プログラム可能なプリスケアラステージで、カレンダーの更新に使用する 1 Hz のクロックを生成します。消費電力を最少に抑えるため、プリスケアラは以下に示す 2 つのプログラム可能なプリスケアラに分割されます ([図 206 : RTC ブロック図](#)を参照)。

- RTC_PRER レジスタの PREDIV_A ビットで設定される 7 ビットの非同期プリスケアラ
- RTC_PRER レジスタの PREDIV_S ビットで設定される 15 ビットの同期プリスケアラ

注： **両方のプリスケアラを使用する場合は、非同期プリスケアラを高い値に設定して消費を最低限に抑えることをお勧めします。**

LSE 周波数 32.768 kHz で 1 Hz (ck_spre) の内部クロック周波数を得るため、非同期プリスケアラ分周比は 128、同期プリスケアラの分周比は 256 に設定されます。

最低分周比は 1、最大分周比は 2^{22} です。

これは、約 4 MHz の最大入力周波数に相当します。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1}$$

ck_apre クロックは、サブセカンドダウンカウンタであるバイナリ RTC_SSR にクロックを供給するために使用されます。値がゼロになると、RTC_SSR は、PREDIV_S の内容で再ロードされます。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)}$$

ck_spre クロックは、カレンダーの更新に、または 16 ビットウェイクアップ自動再ロードタイマのタイムベースとして使用できます。短いタイムアウト期間を得るため、16 ビットウェイクアップ自動再ロードタイマを、プログラム可能な 4 ビット非同期プリスケアラで分周した RTCCLK で動作させることもできます (詳細は [セクション 22.3.4](#) を参照)。

22.3.2 リアルタイムクロックとカレンダー

RTC カレンダーの時間および日付レジスタには、PCLK1 (APB1 クロック) と同期するシャドウレジスタを通じてアクセスします。同期持続の待ち時間を避けるため、これらのレジスタに直接アクセスすることもできます。

- サブセカンド用 RTC_SSR
- 時刻用 RTC_TR
- 日付用 RTC_DR

RTCCLK 2 サイクルごとに現在のカレンダー値がシャドウレジスタにコピーされ、RTC_ISR レジスタの RSF ビットがセットされます ([セクション 22.6.4](#)を参照)。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 2 サイクル以内にシャドウレジスタが更新されます。

アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC_SSR、RTC_TR または RTC_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 (f_{APB}) は、RTC クロック (f_{RTCCLK}) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

22.3.3 プログラム可能なアラーム

本 RTC ユニットには、2 本のプログラム可能なアラームである、アラーム A およびアラーム B を搭載しています。

このプログラム可能なアラーム機能は、RTC_CR レジスタの ALRAIE ビットおよび ALRBIE ビットを通じて有効にします。ALRAF および ALRBF フラグは、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC_ALRMASR/RTC_ALRMAR および RTC_ALRMBSSR/RTC_ALRMBR にプログラムされている値と一致すると 1 にセットされます。各カレンダー項目は、RTC_ALRMAR および RTC_ALRMBR レジスタの MSKx ビット、および RTC_ALRMASR および RTC_ALRMBSSR レジスタの MASKSSx ビットで個別に選択できます。アラーム割り込みは、RTC_CR レジスタの ALRAIE ビットおよび ALRBIE ビットを通じて有効にします。

アラーム A および アラーム B (RTC_CR レジスタの OSEL[1:0] ビットで有効になっている場合) は、RTC_ALARM 出力に送ることができます。RTC_ALARM の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

注意： *秒の項目が選択されている場合 (RTC_ALRMAR または RTC_ALRMBR で MSK0 ビットがリセットされている)、正しい動作を保証するため、RTC_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上である必要があります。*

22.3.4 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)
RTCCLK が LSE (32.768 kHz) である場合、最小分解能 61 μ s で、ウェイクアップ割り込み周期を 122 μ s から 32 s の範囲で設定できます。
- ck_spre (通常は 1 Hz の内部クロック)
ck_spre 周波数が 1 Hz の場合、1 秒の分解能でウェイクアップ時間を 1 秒 からおよそ 36 時間までの範囲で設定できます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。
 - WUCKSEL[2:1] = 10 の場合は 1 秒から 18 時間、
 - WUCKSEL[2:1] = 11 の場合は約 18 時間から 36 時間です。後者の場合、16 ビットカウンタの現在値に 216 が加算されます。初期化シーケンスが完了すると ([ウェイクアップタイマのプログラミング](#)を参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効な場合、低電力モードでもカウントダウンはアクティブのままとなります。さらに、

カウンタがゼロに到達すると、RTC_ISR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC_WUTR レジスタ値) で自動的に再ロードされます。

その後、WUTF フラグはソフトウェアでクリアする必要があります。

RTC_CR2 レジスタの WUTIE ビットをセットして定期ウェイクアップ割り込みを有効にすると、デバイスは低電力モードを終了できます。

周期的なウェイクアップフラグは、RTC_CR レジスタの OSEL[0:1] ビットを通じて有効になっている場合に限り、RTC_ALARM 出力に送ることができます。RTC_ALARM の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

低電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイムには影響しません。

22.3.5 RTC の初期化と設定

RTC レジスタアクセス

RTC レジスタは、32 ビットのレジスタです。APB インターフェイスは、RTC レジスタアクセスに 2 ウェイトステートを挿入します。ただし、BYPHAD = 0 のときのカレンダーシャドウレジスタへの読み出しアクセスは除きます。

RTC レジスタ書き込み保護

システムリセット後、RTC レジスタは PWR 電源制御レジスタ (PWR_CR) の DBP ビットへの不要な書き込みアクセスから保護されます。RTC レジスタ書き込みアクセスを可能にするには、DBP ビットをセットする必要があります。

バックアップドメインリセット後、すべての RTC レジスタは書き込み保護されます。RTC レジスタへの書き込みは、書き込み保護レジスタ RTC_WPR にキーを書き込むことにより有効になります。

RTC_ISR[13:8]、RTC_TAFCR、RTC_BKPxR を除くすべての RTC レジスタの書き込み保護を解除するには、次のステップが必要です。

1. RTC_WPR レジスタに“0xCA”を書き込みます。
2. RTC_WPR レジスタに“0x53”を書き込みます。

誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC_ISR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC_ISR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。これには RTCCLK クロック 1 ~ 2 周期を必要とします (クロック同期のため)。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、まず RTC_PRER レジスタで同期プリスケアラ分周比をプログラムし、その後非同期プリスケアラ分周比をプログラムします。2 つの項目のうち 1 つだけを変更する必要がある場合でも、RTC_PRER レジスタに対し書き込みアクセスを 2 回に分けて行う必要があります。
4. シャドウレジスタ (RTC_TR および RTC_DR) に時刻と日付の初期値をロードし、RTC_CR レジスタの FMT ビットを介して時間形式 (12 時間または 24 時間) を設定します。

- INIT ビットをクリアして初期化モードを終了します。その後、カレンダーカウンタの実際の値が自動的にロードされ、4 RTCCLK クロックサイクル後にカウントが再開します。

初期化シーケンスが完了すると、カレンダーがカウントを開始します。

注： システムリセット後、アプリケーションは RTC_ISR レジスタの INITS フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 である場合、カレンダーの年の項目がバックアップドメインリセットデフォルト値 (0x00) にセットされているため初期化されていません。

初期化後にカレンダーを読み出すには、まずソフトウェアで RTC_ISR レジスタの RSF フラグがセットされていることを確認する必要があります。

サマータイム

サマータイム管理は、RTC_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

アラームのプログラミング

プログラム可能なアラーム (アラーム A またはアラーム B) をプログラムまたは更新するには、同様な手順を踏む必要があります。

- RTC_CR の ALRAE または ALRBIE をクリアしてアラーム A またはアラーム B を無効にします。
- RTC_ISR の ALRAWF または ALRBWF がセットされてアラームレジスタへのアクセスが許可されていることが確認できるまで、これらのビットをポーリングします。これには RTCCLK クロック約 2 周期を必要とします (クロック同期のため)。
- アラーム A レジスタまたはアラーム B レジスタ (RTC_ALRMASR/RTC_ALRMAR または RTC_ALRMBSSR/RTC_ALRMBR) をプログラムします。
- RTC_CR レジスタで ALRAE または ALRBIE をセットしてアラーム A またはアラーム B を再び有効にします。

注： RTC_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 周期後に有効になります。

ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値 (RTC_WUTR の WUT[15:0]) の設定または変更には、次の手順が必要です。

- RTC_CR の WUTE をクリアしてウェイクアップタイマを無効にします。
- RTC_ISR の WUTWF がセットされ、ウェイクアップ自動再ロードカウンタおよび WUCKSEL[2:0] ビットへのアクセスが許可されていることが確認されるまで WUTWF をポーリングします。これには RTCCLK クロック 1 ~ 2 周期を必要とします (クロック同期のため)。
- ウェイクアップ自動再ロード値 WUT[15:0] およびウェイクアップクロック選択 (RTC_CR の WUCKSEL[2:0] ビット) をプログラムします。RTC_CR で WUTE をセットしてタイマを再び有効にします。ウェイクアップタイマがカウントダウンを再開します。WUTWF ビットは、クロックの同期化により、WUTE クリア後、2 RTCCLK クロックサイクルまでにクリアされます。

22.3.6 カレンダの読み出し

RTC_CR レジスタの BYPSHAD 制御ビットがクリアされている場合

RTC カレンダレジスタ (RTC_SSR、RTC_TR および RTC_DR) を正しく読み出すには、APB1 クロック周波数 (f_{PCLK1}) が f_{RTCCLK} RTC クロック周波数の 7 倍以上である必要があります。これにより、同期メカニズムの安全な動作が保証されます。

APB1 クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR の 2 回目の読み出しが 1 回目の読み出しと同じ結果であれば、データが正しいことが保証されます。同じでない場合は、3 回目の読み出しアクセスを行う必要があります。どの場合も、APB1 クロック周波数は必ず RTC クロック周波数以上である必要があります。

RTC_ISR レジスタの RSF ビットは、カレンダーレジスタが RTC_SSR、RTC_TR、および RTC_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、2 RTCCLK サイクルごとに行われます。3 つの値における一貫性を保証するため、RTC_SSR または RTC_TR のどちらかを読み出すと、高次カレンダーシャドウレジスタの値は RTC_DR が読み出されるまでロックされます。ソフトウェアが 2 RTCCLK サイクル未満の間隔でカレンダーの読み出しアクセスを行う場合、最初のカレンダー読み出し後に RSF をソフトウェアでクリアする必要があり、その後ソフトウェアは、RSF ビットがセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再度読み出す必要があります。

低電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いまだ RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再度読み出す必要があります。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低電力モードに入る前には、その必要はありません。

注： システムリセット後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。実際、システムリセットがかかると、シャドウレジスタはデフォルト値にリセットされます。

初期化 (カレンダーの初期化と設定を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

同期 (セクション 22.3.8 を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

RTC_CR レジスタ (バイパスシャドウレジスタ) の BYPSHAD 制御ビットがセットされている場合

カレンダーレジスタを読み出すと、カレンダーカウンタの値が直接与えられるため、RSF ビットがセットされるのを待つ必要はありません。シャドウレジスタは低電力モード (STOP または STANDBY) では更新されないため、これは低電力モード終了後に特に有用です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読み出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読み出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

注： **BYPSHAD = 1 の間、カレンダーレジスタの読み出し命令が完了するには 1 APB サイクルだけ余計に必要となります。**

22.3.7 RTC のリセット

カレンダーシャドウレジスタ (RTC_SSR、RTC_TR、RTC_DR) および RTC ステータスレジスタ (RTC_ISR) の一部のビットは、利用可能なすべてのシステムリセットリソースによってデフォルト値にリセットされます。

逆に、次のレジスタは バックアップドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響は受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC_CR)、プリスケアラレジスタ (RTC_PRER)、RTC 較正レジスタ (RTC_CALIBR または RTC_CALR)、RTC シフトレジスタ (RTC_SHIFTR)、RTC タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、および RTC_TSDR)、RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)、RTC バックアップレジスタ (RTC_BKPxR)、ウェイクアップタイマレジスタ (RTC_WUTR)、およびアラーム A およびアラーム B レジスタ (RTC_ALRMASR/RTC_ALRMAR および RTC_ALRMBSSR/RTC_ALRMBR)。

さらに、LSE によってクロック供給されるとき、RTC はリセットソースがバックアップドメインリセットと異なる場合は、システムリセット中も動作し続けます (システムリセットによる影響を受けない RTC クロックソースのリストの詳細については RCC を参照)。バックアップドメインリセットが発生すると、RTC は停止し、すべての RTC レジスタがリセット値にセットされます。

22.3.8 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC_SSR または RTC_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC_SSR には、同期プリスケアラのカウンタの値が入っています。これにより、RTC によって維持されている正確な時刻を $1/(\text{PREDIV}_S + 1)$ 秒の分解能まで計算することができます。その結果、同期プリスケアラ値 ($\text{PREDIV}_S[14:0]$) を増加させることにより分解能を改善できます。許可されている最大分解能 (32768 Hz クロックで 30.52 μs) は、 PREDIV_S を 0x7FFF にセットすることにより得られます。

ただし、 PREDIV_S を増加させるということは、同期プリスケアラの出力を 1 Hz に維持するため PREDIV_A を減らす必要があることを意味します。このように、非同期プリスケアラの出力周波数が増加すると、RTC の動的消費電力が増加する場合があります。

RTC は、RTC シフト制御レジスタ (RTC_SHIFTR) を使って微調整できます。RTC_SHIFTR に書き込むことにより、 $1/(\text{PREDIV}_S + 1)$ 秒の分解能で、クロックを最大 1 秒だけシフト (遅れ/進み) させることができます。このシフト操作の本質は、同期プリスケアラのカウンタ $\text{SS}[15:0]$ に $\text{SUBFS}[14:0]$ 値を加算することであり、この操作はクロックを遅らせることになります。同時に ADD1S ビットがセットされた場合、1 秒追加すると同時に秒の小数部を差し引くことになるため、クロックを進めることになります。

注意： シフト操作を始める前に、ユーザは $\text{SS}[15] = 0$ であることを確認し、オーバーフローが発生しないようにする必要があります。

RTC_SHIFTR レジスタへの書き込みによってシフト操作が始まるとすぐに、シフト操作が保留中であることを示す SHPF フラグがハードウェアによってセットされます。このビットは、シフト操作が完了するとすぐに、ハードウェアによってクリアされます。

注意： この同期機能はリファレンスクロック検出機能とは両立できません。具体的には、 $\text{REFCKON} = 1$ のときにファームウェアから RTC_SHIFTR への書き込みはできません。



22.3.9 RTC リファレンスクロック検出

RTC カレンダの更新は、リファレンスクロックである RTC_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC_REFIN リファレンスクロックには、32.768 kHz LSE クロックよりも高い精度が必要です。RTC_REFIN 検出が有効になると (RTC_CR の REFCKON ビットが 1 にセット)、カレンダは引き続き LSE クロックによって駆動されますが、RTC_REFIN を使用してカレンダ更新周波数 (1 Hz) の不正確さを補正します。

各 1 Hz クロックエッジは、一番近いリファレンスクロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、その後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck_apre) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダ更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck_apre 7 周期に等しくなります。その後のカレンダ更新では、ck_apre 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、ck_apre クロックを出力する非同期プリスケラは強制的に再ロードされます。プリスケラは同時に再ロードされるので、リファレンスクロックおよび 1 Hz のクロックが整列するタイミングには影響しません。クロックが整列していない場合、後の 1 Hz クロックエッジは、リファレンスクロックと整列するように再ロードによって少しシフトされます。

リファレンスクロックが停止した (ck_apre 3 周期の枠内でリファレンスクロックエッジが発生しない) 場合、カレンダは LSE クロックのみを基準にして更新が継続されます。その後 RTC は ck_spre エッジを中心として ck_apre 7 周期という広い検出時間枠でリファレンスクロックを待ちます。

基準クロック検出を有効にした場合、PREDIV_A および PREDIV_S をデフォルト値にセットする必要があります。

- PREDIV_A = 0x007F
- PREDIV_S = 0x00FF

注： 基準クロック検出は、STANDBY モードでは利用できません。

注意： 基準クロック検出機能は、簡易デジタル校正と併用することはできません。REFCKON = 1 の場合、RTC_CALIBR は 0x0000 0000 に保持する必要があります。

22.3.10 RTC の簡易デジタル校正

デジタル校正手法には、簡易校正と高精度校正の 2 種類があります。簡易校正を実行するには、[セクション 22.6.7 : RTC 校正レジスタ \(RTC_CALIBR\)](#) を参照してください。

これら 2 種類の校正方法は同時に使用することは想定されていないので、アプリケーションは 2 種類の校正方法のうち、いずれかを選択する必要があります。簡易校正は、互換性を確保するために搭載されています。高精度校正を実行するには、[セクション 22.3.11 : RTC の高精度デジタル校正](#) および [セクション 22.6.16 : RTC 校正レジスタ \(RTC_CALR\)](#) を参照してください。

簡易デジタル校正を使用すると、非同期プリスケラ (ck_apre) の出力でクロックサイクルを追加 (正の校正) またはマスキング (負の校正) して、クリスタルの精度を補償することができます。

正および負の校正は、RTC_CALIBR レジスタの DCS ビットをそれぞれ“0”および“1”にセットして選択します。

正の較正が有効 (DCS = "0") の場合、2 x DC 分の間、毎分 (ck_apre 約 15360 サイクル) ck_apre 2 サイクルが追加されます。これによりカレンダーの更新更新速度が早くなり、それによって有効な RTC 周波数は少し高くなります。

負の較正が有効 (DCS = "1") の場合、2 x DC 分の間、毎分 (ck_apre 約 15360 サイクル) ck_apre 1 サイクルが取り除かれます。これによりカレンダーの更新速度が遅くなり、それによって有効な RTC 周波数は少し低くなります。

DC は RTC_CALIBR レジスタの DC[4:0] ビットで設定されます。この数値は 0 から 31 の範囲で設定可能であり、0 から 62 の範囲の時間間隔 (2 x DC) に相当します。

簡易デジタル較正は、初期化モードの期間でのみ設定でき、INIT ビットがクリアされると設定した動作を開始します。較正サイクルは全体で 64 分間続きます。64 分間のサイクルの最初の 2 x DC 分が、上記の設定のように変更されます。

負の較正は、約 2 ppm の分解能で実行できますが、正の較正は約 4 ppm の分解能になります。最大較正範囲は、-63 ppm から 126 ppm です。

較正は LSE または HSE クロックのどちらに対しても実行可能です。

注意： **PREDIV_A < 6 の場合、デジタル較正が正しく動作しない場合があります。**

RTCCLK=32.768 kHz かつ PREDIV_A+1=128 の場合

次の説明は、ck_apre 周波数が、LSE クロックの公称周波数 32.768 kHz をもとに、PREDIV_A を 127 (デフォルト値) にセットして得られた 256 Hz であることを前提としています。

ck_spre クロック周波数は、64 分間のサイクルの最初の 2 x DC 分間でのみ変更されます。たとえば、DC が 1 である場合、最初の 2 分のみが変更されます。つまり、各 ck_apre サイクルが 128 RTCCLK サイクル (PREDIV_A+1=128) に相当するならば、各 64 分サイクルの最初の 2 x DC 分に対して、毎分 1 回だけ、1 秒が RTCCLK 256 サイクル分短くなる、あるいは、RTCCLK 128 サイクル分長くなります。

したがって、各較正ステップには、125829120 RTCCLK サイクル (64 分 x 60 秒/分 x 32768 サイクル/秒) ごとに 512 オシレータサイクルを追加または 256 オシレータサイクルを差し引くという効果があります。これは、較正ステップにつき +4.069 ppm または -2.035 ppm と等価です。その結果、較正分解能は毎月 +10.5 秒または -5.27 秒となり、総較正範囲は毎月 +5.45 ~ -2.72 分となります。

クロック偏差を測定するため、較正用に 512 Hz クロックが出力されます。[セクション 22.3.14: 較正クロック出力](#)を参照してください。

22.3.11 RTC の高精度デジタル較正

RTC 周波数の精度は、-487.1 ppm から +488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に較正できます。周波数の修正は、一連の微調整 (個々の RTCCLK パルスの追加や削除) によって行われます。このような調整は、短い期間で観測された場合でも RTC が十分に較正されるように、かなり広範囲に分散して行われます。

この高精度デジタル較正は、入力周波数が 32768 Hz の場合、RTCCLK 約 2²⁰ パルスのサイクルまたは 32 秒の間で行われます。このサイクルは、RTCCLK によって駆動される 20 ビットカウンタ、cal_cnt[19:0] によって維持されます。

高精度校正レジスタ (RTC_CALR) によって、32 秒サイクル中にマスクされる RTCCLK クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると、32 秒サイクルの中でちょうど 1 パルスがマスクされます。
- CALM[1] ビットを 1 にセットすると、さらに 2 サイクルがマスクされます。
- CALM[2] ビットを 1 にセットすると、さらに 4 サイクルがマスクされます。
- CALM[8] ビットを 1 にセットするまで続けると、256 クロックがマスクされます。

注： CALM[8:0] (RTC_CALRx) によって、32 秒サイクル中にマスクされる RTCCLK パルス数を指定します。CALM[0] ビットを 1 にセットすると、cal_cnt[19:0] = 0x80000 になった時点で、32 秒サイクル中でちょうど 1 パルスがマスクされます。CALM[1] = 1 では、さらに 2 サイクルがマスクされ (cal_cnt = 0x40000 および 0xC0000)、CALM[2] = 1 では、さらに 4 サイクルがマスクされ (cal_cnt = 0x20000/0x60000/0xA0000/0xE0000)、CALM[8] = 1 まで続けると、256 クロックがマスクされず (cal_cnt = 0xXX800)。

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、RTCCLK 2^{11} サイクルごとに、RTCCLK パルスが 1 パルス追加で挿入されます。すなわち、32 秒サイクルごとに 512 クロックが追加されることとなります。

CALM を CALP と合わせて使用すると、32 秒サイクルの間に RTCCLK -511 から +512 サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で校正範囲 -487.1 ~ +488.5 ppm に換算されます。

有効校正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (\text{CALP} \times 512 - \text{CALM}) / (2^{20} + \text{CALM} - \text{CALP} \times 512)]$$

PREDIV_A < 3 の場合の校正

非同期プリスケアラ値 (RTC_PRER レジスタの PREDIV_A ビット) が 3 未満の場合、CALP ビットを 1 にセットすることはできません。CALP がすでに 1 にセットされていて、PREDIV_A ビットが 3 未満の値にセットされた場合、CALP の設定値は無視され、CALP が 0 に設定された場合と同じように校正されます。

PREDIV_A が 3 未満の状態では校正を実施するには、各秒のカウントが 8 RTCCLK クロックサイクル分早められるように同期プリスケアラ値 (PREDIV_S) を小さくする必要があります。これは 32 秒毎に 256 クロックサイクル追加することに相当します。結果として、CALM ビットのみを使用して 32 秒周期の間に 256 クロックパルスから 256 クロックパルス (243.3 から 244.1 ppm の校正範囲に相当)を追加することができます。

公称 RTCCLK 周波数が 32768 Hz で、PREDIV_A が 1 (分周比 2) の場合、PREDIV_S を 16383 ではなく 16379 (4 少ない) にセットする必要があります。また、PREDIV_A が 0 の場合、PREDIV_S を 32767 ではなく 32759 (8 少ない) にセットする必要がありますので注意して下さい。

PREDIV_S をこのように減少させた場合、校正された入力クロックの

有効周波数の式は次のようになります。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (256 - \text{CALM}) / (2^{20} + \text{CALM} - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

RTC 較正值の確認

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより向上することができます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。

ある時間間隔で RTC の周波数を精密に測定すると、デジタル較正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が較正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル較正の分解能による誤差となります。

- デフォルトでは、較正サイクル周期は 32 秒です。
このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (較正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。
- RTC_CALR レジスタの CALW16 ビットを 1 にセットして、較正サイクル周期を強制的に 16 秒にすることができます。
この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、較正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。
- RTC_CALR レジスタの CALW8 ビットを 1 にセットして、較正サイクル周期を強制的に 8 秒にすることができます。
この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

動作中の再較正

次の処理を実施することにより、RTC_ISR/INITF = 0 の間でも、較正レジスタ (RTC_CALR) を動作中に更新することができます。

1. RTC_ISR/RECALPF (再較正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC_CALR への書き込み動作後 ck_apre 3 サイクル以内に、新しい較正設定が有効になります。

22.3.12 タイムスタンプ機能

タイムスタンプは、RTC_CR レジスタの TSE ビットを 1 にセットすることにより有効になります。

TIMESTAMP にオルタネート機能が設定されているピンでタイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) にカレンダーが保存されます。また、タイムスタンプイベントが発生すると、RTC_ISR レジスタのタイムスタンプフラグビット (TSF) がセットされます。

RTC_CR レジスタの TSIE ビットをセットすることにより、タイムスタンプイベントが発生したときに割り込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC_TSTR および RTC_TSDR) は、その前のイベントの結果を維持します。

注： *同期処理のため、TSF はタイムスタンプイベント発生から ck_apre 2 サイクル後にセットされます。一方、TSOVF のセットに遅延はありません。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。*



注意： TSF ビットのクリア処理を行った直後にタイムスタンプイベントが発生した場合、TSF ビットおよび TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、アプリケーションは TSF がすでに“1”と読み出されていない限り“0”クリアの処理を行ってはなりません。

オプション機能として、タンパイイベントによってタイムスタンプイベントを記録することもできます。TAMPTS 制御ビットの詳細については、[セクション 22.6.17 : RTC タンパおよびオルタネート機能設定レジスタ \(RTC_TAFCR\)](#) を参照してください。タイムスタンプイベントがフィルタモードで設定されているタンパイイベントと同じピン上にある場合 (TAMPFLT がゼロ以外の値にセットされている場合)、RTC_TAFCR レジスタの TAMPTS を“1”にセットすることにより、タンパ検出イベントモードにおけるタイムスタンプを選択する必要があります。

TIMESTAMP オルタネート機能

TIMESTAMP 追加機能は RTC_AF1 に配置されます。

22.3.13 タンパ検出

2 つのタンパ検出入力を利用できます。これらの入力は、フィルタを通してエッジ検出またはレベル検出のどちらにも設定できます。

RTC バックアップレジスタ

バックアップレジスタ (RTC_BKPxR) は、20 個の 32 ビットレジスタであり、80 バイトのユーザーアプリケーションデータを保存できます。これらのレジスタは、V_{DD} 電源が遮断された場合に V_{BAT} によって電源が供給される、バックアップドメインに搭載されています。これらは、システムリセットや STANDBY モードからのウェイクアップではリセットされません。バックアップドメインリセットによりリセットされます。

このバックアップレジスタは、タンパ検出イベントが発生するとリセットされます ([セクション 22.6.20 : RTC バックアップレジスタ \(RTC_BKPxR\)](#) および [タンパ検出の初期化 \(614 ページ\)](#) を参照)。

タンパ検出の初期化

各タンパ検出入力、RTC_ISR2 レジスタのフラグ TAMP1F/TAMP2F に関連付けられています。各入力は、RTC_TAFCR レジスタにおいて該当する TAMP1E/TAMP2E ビットを 1 にセットすることにより有効にできます。

タンパ検出イベントが発生すると、すべてのバックアップレジスタ (RTC_BKPxR) がリセットされます。

RTC_TAFCR レジスタの TAMPIE ビットをセットすることにより、タンパ検出イベント発生時に割り込みが生成されます。

タンパイイベント時のタイムスタンプ

TAMPTS を“1”にセットすると、すべてのタンパイイベントがタイムスタンプイベントを発生させるようになります。この場合、通常のタイムスタンプイベント発生時と同様に TSF ビットまたは TSOVF ビットが RTC_ISR でセットされます。TSF または TSOVF がセットされるのと同時に、影響を受けるタンパフラグレジスタ (TAMP1F、TAMP2F) がセットされます。

タンパ入力でのエッジ検出

TAMPFLT ビットが“00”の場合、該当する TAMPxTRG ビットに応じて、立ち上りエッジまたは立ち下りエッジが観測されると TAMPER ピンがタンパ検出イベント (RTC_TAMP[2:1]) を生成します。エッジ検出を選択すると、TAMPER 入力ピンの内部プルアップ抵抗が無効になります。

注意： タンパ検出イベントを確実に検出するため、TAMPERx ピンが有効になる前にタンパ検出イベントが発生した場合でも検出できるよう、エッジ検出に使用される信号は TAMPxE と AND がとられます。

- TAMPxTRG = 0 の場合：タンパ検出が有効になる（TAMPxE ビットが 1 にセットされる）前に TAMPERx オルタネート機能がすでにハイであれば、TAMPxE がセットされた後に TAMPERx に立ち上がりエッジ入力がなくとも TAMPERx が有効になった直後にタンパイベントが検出されます。
- TAMPxTRG = 1 の場合：タンパ検出が有効になる前に TAMPERx オルタネート機能がすでにローであれば、TAMPxE がセットされた後に TAMPERx に立ち下がりエッジ入力がなくとも TAMPERx が有効になった直後にタンパイベントが検出されます。

タンパイベントが検出されクリアされた後に、バックアップレジスタ（RTC_BKPxR）を再プログラムする場合には、事前に、TAMPERx オルタネート機能を無効にしてから再度有効に（TAMPxE を 1 にセット）する処理を行う必要があります。これによって、TAMPERx の値がタンパ検出を示している期間中に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、TAMPERx オルタネート機能でのレベル検出と同じことです。

注： タンパ検出は、V_{DD} 電源がオフのときでも有効です。バックアップレジスタの不必要なリセットを避けるには、TAMPER オルタネート機能が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。

タンパ入力でのフィルタを使ったレベル検出

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、2、4 または 8（TAMPFLT による）回のいずれかの連続したサンプルが TAMPxTRG ビット（TAMP1TRG/TAMP2TRG）で指定するレベルで観測されると発生されます。

TAMPER 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、タンパ入力におけるより大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

注： プルアップ抵抗の電気的特性については、データシートを参照してください。

TAMPER オルタネート機能検出

TAMPER1 追加機能は RTC_AF1 ピンに配置されます。

22.3.14 較正クロック出力

RTC_CR レジスタで COE ビットが 1 にセットされると、RTC_CALIB デバイス出力にリファレンスクロックが供給されます。RTC_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV_A = 0x7F である場合、RTC_CALIB 周波数は $f_{\text{RTCCLK}}/64$ です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。

RTC_CALIB 出力は、RTC_CALIBR レジスタでプログラムされる較正值に影響されません。立ち下がリエッジには軽いジッタがあるため、RTC_CALIB のデューティサイクルは不規則になります。したがって、立ち上がりエッジの使用が推奨されます。

COSEL がセットされ、かつ “PREDIV_S+1” がゼロ以外の 256 の倍数である場合（すなわち、PREDIV_S[7:0] = 0xFF）、RTC_CALIB 周波数は $f_{\text{RTCCLK}}/(256 * (\text{PREDIV_A}+1))$ となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケラデフォルト値（PREDIV_A = 0x7F、PREDIV_S = 0xFF）に対する 1 Hz の較正出力に相当します。

較正クロック出力時の RTC_AF1 オルタネート機能

RTC_CR レジスタの COE ビットが 1 にセットされると、RTC_AF1 のオルタネート機能は、較正クロックの出力（RTC_CALIB）が有効になります。

注： RTC_CALIB または RTC_ALARM が選択されると、RTC_AF1 はオルタネート機能で自動的に出力に設定されます。

22.3.15 アラーム出力

アラーム出力については、3 つの機能が選択できます。ALRAF、ALRBF、WUTF です。これらの機能は、RTC_ISR レジスタの該当するフラグの内容を反映します。

RTC_CR レジスタの OSEL[1:0] 制御ビットを使用して RTC_AF1 でアラームオルタネート機能出力（RTC_ALARM）を有効にし、RTC_ALARM の出力となる機能を選択します。

RTC_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、出力の極性を決定します。

アラームオルタネート機能出力

RTC_ALARM は、RTC_TAFCR レジスタの ALARMOUTTYPE 制御ビットを使用して、出力オープンドレインまたは出力プッシュプルを設定できます。

注： RTC_ALARM が有効になると、この設定は、RTC_CALIB（COE ビットは RTC_AF1 では無視）の設定よりも優先されます。

RTC_CALIB または RTC_ALARM が選択されると、RTC_AF1 はオルタネート機能で自動的に出力に設定されます。

22.4 RTC と低電力モード

表 107. 低電力モードが RTC に与える影響

モード	説明
SLEEP	影響なし。 RTC 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンバイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスはストップモードから復帰します。
STANDBY	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンバイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスはスタンバイモードから復帰します。



22.5 RTC 割り込み

すべての RTC 割り込みは、EXTI コントローラに接続されています。

RTC アラーム割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで EXTI ライン 17 を設定して有効にし、立ち上りエッジ検出を選択します。
2. NVIC で RTC_Alarm IRQ チャンネルを設定し、有効にします。
3. RTC が RTC アラーム (アラーム A またはアラーム B) を発生させるように設定します。

RTC ウェイクアップ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで EXTI ライン 22 を設定して有効にし、立ち上りエッジ検出を選択します。
2. NVIC で RTC_WKUP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC ウェイクアップタイミントを発生させるように設定します。

RTC タンパ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで EXTI ライン 21 を設定して有効にし、立ち上りエッジ検出を選択します。
2. NVIC で TAMP_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC タンパイベントを検出するように設定します。

RTC タイムスタンプ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで EXTI ライン 21 を設定して有効にし、立ち上りエッジ検出を選択します。
2. NVIC で TAMP_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC タイムスタンプイベントを検出するように設定します。

表 108. 割り込み制御ビット

割り込みイベント	イベント フラグ	有効化 制御ビット	SLEEP モードの 終了	STOP モードの 終了	STANDBY モードの 終了
アラーム A	ALRAF	ALRAIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
アラーム B	ALRBF	ALRBIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
ウェイクアップ	WUTF	WUTIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
タイムスタンプ	TSF	TSIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
タンパ 1 検出	TAMP1F	TAMPIE	あり	あり ⁽¹⁾	あり ⁽¹⁾

1. STOP モードおよび STANDBY モードからのウェイクアップは、RTC クロックソースが LSE または LSI のときのみ可能です。

22.6 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

22.6.1 RTC 時刻レジスタ (RTC_TR)

RTC_TR は、カレンダー時刻シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。 [カレンダーの初期化と設定](#) および [カレンダーの読み出し](#) を参照してください。

アドレスオフセット : 0x00

バックアップドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31-24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(606 ページ\)](#) を参照してください。

22.6.2 RTC 日付レジスタ (RTC_DR)

RTC_DR は、カレンダー日付シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。 [カレンダーの初期化と設定](#) および [カレンダーの読み出し](#) を参照してください。

アドレスオフセット : 0x04

バックアップドメインリセット値 : 0x0000_2101

システムリセット : BYPSHAD = 0 の場合、0x0000 2101 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31-24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

000 : 禁止
001 : 月曜日
...
111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護](#)を参照してください。

22.6.3 RTC 制御レジスタ (RTC_CR)

アドレスオフセット : 0x08
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rW	rW	rW	rW	rW	rW	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBIE	ALRAIE	DCE	FMT	BYP SHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23 **COE** : 較正出力イネーブル
このビットは、RTC_CALIB 出力を有効にします。
0 : 較正出力は無効です。
1 : 較正出力は有効です。
- ビット 22:21 **OSEL[1:0]** : 出力選択
これらのビットは、RTC_ALARM 出力に送られるフラグの選択に使用します。
00 : 出力は無効です。
01 : アラーム A 出力は有効です。
10 : アラーム B 出力は有効です。
11 : ウェイクアップ出力は有効です。
- ビット 20 **POL** : 出力極性
このビットは、RTC_ALARM 出力の極性の設定に使用します。
0 : ALRAF/ALRBF/WUTF がアサートされると、このピンがハイになります (OSEL[1:0] に応じて)。
1 : ALRAF/ALRBF/WUTF がアサートされると、このピンがローになります (OSEL[1:0] に応じて)。
- ビット 19 **COSEL** : 較正出力選択
COE = 1 のとき、このビットによって RTC_CALIB に出力される信号を選択します。
0 : 較正出力は 512 Hz です。
1 : 較正出力は 1 Hz です。
これらの周波数は、RTCCLK が 32.768 kHz で、プリスケアラがデフォルト値 (PREDIV_A = 127 および PREDIV_S = 255) の場合に有効です。参照先 : [セクション 22.3.14 : 較正クロック出力](#)
- ビット 18 **BKP** : バックアップ
このビットは、サマータイムの変更を実施したか否かを記憶しておくため、ユーザが書き込むことができます。
- ビット 17 **SUB1H** : 1 時間差し引き (冬時間変更)
このビットを初期化モード以外の人にセットすると、現在時刻が 0 でない場合にカレンダー時刻から 1 時間を差し引きします。このビットは常に 0 として読み出されます。
現在時間が 0 のときにこのビットをセットしても、影響はありません。
0 : 影響なし。
1 : 現在時刻から 1 時間差し引きします。これは、冬時間変更で使用できます。
- ビット 16 **ADD1H** : 1 時間加算 (サマータイム変更)
このビットを初期化モード以外の人にセットすると、カレンダー時刻に 1 時間加算します。このビットは常に 0 として読み出されます。
0 : 影響なし。
1 : 現在時刻に 1 時間加算します。これは、サマータイム変更で使用できます。

ビット 15 **TSIE** : タイムスタンプ割り込みイネーブル

- 0 : タイムスタンプ割り込みは無効です。
- 1 : タイムスタンプ割り込みは有効です。

ビット 14 **WUTIE** : ウェイクアップタイマ割り込みイネーブル

- 0 : ウェイクアップタイマ割り込みは無効です。
- 1 : ウェイクアップタイマ割り込みは有効です。

ビット 13 **ALRBIE** : アラーム B 割り込みイネーブル

- 0 : アラーム B 割り込みは無効です。
- 1 : アラーム B 割り込みは有効です。

ビット 12 **ALRAIE** : アラーム A 割り込みイネーブル

- 0 : アラーム A 割り込みは無効です。
- 1 : アラーム A 割り込みは有効です。

ビット 11 **TSE** : タイムスタンプイネーブル

- 0 : タイムスタンプは無効です。
- 1 : タイムスタンプイネーブル

ビット 10 **WUTE** : ウェイクアップタイマイネーブル

- 0 : ウェイクアップタイマは無効です。
- 1 : ウェイクアップタイマは有効です。

ビット 9 **ALRBE** : アラーム B イネーブル

- 0 : アラーム B は無効です。
- 1 : アラーム B は有効です。

ビット 8 **ALRAE** : アラーム A イネーブル

- 0 : アラーム A は無効です。
- 1 : アラーム A は有効です。

ビット 7 **DCE** : 簡易デジタル較正イネーブル

- 0 : デジタル較正は無効です。
 - 1 : デジタル較正は有効です。
- PREDIV_A は 6 以上である必要があります。

ビット 6 **FMT** : 時間形式

- 0 : 24 時間 / 日形式
- 1 : AM / PM 時間形式

ビット 5 **BYPSHAD** : シャドウレジスタをバイパスします。

- 0 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。
- 1 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、カレンダカウンタから直接取得されます。

注 : *APB1 クロックの周波数が RTCCLK の 7 倍未満である場合、BYPSHAD は“1”にセットする必要があります。*

ビット 4 **REFCKON** : リファレンスクロック検出イネーブル（50 または 60 Hz）

- 0 : 基準クロック検出は無効です。
- 1 : 基準クロック検出は有効です。

注 : *PREDIV_S は 0x00FF である必要があります。*

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

- 0 : TIMESTAMP の立ち上りエッジによってタイムスタンプイベントを生成します。
 - 1 : TIMESTAMP の立ち下りエッジによってタイムスタンプイベントを生成します。
- 不要な TSF 設定を回避するため、TSEDGE が変化した場合には TSE をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ウェイクアップクロック選択

- 000 : RTC/16 クロックが選択されます。
- 001 : RTC/8 クロックが選択されます。
- 010 : RTC/4 クロックが選択されます。
- 011 : RTC/2 クロックが選択されます。
- 10x : ck_spre（通常は 1 Hz）クロックが選択されます。
- 11x : ck_spre（通常は 1 Hz）クロックが選択され、 2^{16} が WUT カウンタ値に加算されます（下記注を参照）。

注 : *WUT = ウェイクアップユニットカウンタ値 $WUT = (0x0000 \sim 0xFFFF) + 0x10000$ (**WUCKSEL[2:1]** = 11 の場合追加されます。)*

初期化モード (RTC_ISR/INITF = 1) の場合のみ、このレジスタのビット 7、6、4 が書き込みます。このレジスタのビット 2~0 は、RTC_CR WUTE ビット = 0 かつ RTC_ISR WUTWF ビット = 1 の場合にのみ書き込みます。

カレンダーの時間項目のインクリメント中は時間を変更しないことが推奨されます。カレンダーの時間項目のインクリメントがマスクされる可能性があるためです。

ADD1H および SUB1H の変更は、次の秒から有効になります。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(606 ページ\)](#) を参照してください。

22.6.4 RTC 初期化とステータスレジスタ（RTC_ISR）

アドレスオフセット : 0x0C

バックアップドメインリセット値 : 0x0000 0007

システムリセット値 : 0 にクリアされる INIT、INITF、RSF 以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF	ALRB WF	ALRA WF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

ビット 31:17 予約済みであり、リセット値のままにしておかなければなりません。

ビット 16 **RECALPF** : 再較正保留フラグ

ソフトウェアによって RTC_CALR レジスタに書き込みが行われると、RECALPF ステータスフラグが自動的に“1”にセットされ、RTC_CALR レジスタがブロックされたことを示します。新たな較正設定が認識されると、このビットは“0”に戻ります。*動作中の再較正*を参照してください。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。



ビット 13 TAMP1F : タンパ検出フラグ

このフラグは、タンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 12 TSOVF : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 11 TSF : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 10 WUTF : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。
このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 9 ALRBF : アラーム B フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム B レジスタ (RTC_ALRMBR) と一致したときにハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 8 ALRAF : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム A レジスタ (RTC_ALRMAR) と一致したときにハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 7 INIT : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC_TR と RTC_DR)、およびプリスケアラレジスタ (RTC_PRER) のプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

ビット 6 INITF : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

ビット 5 RSF : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC_SSRx、RTC_TRx および RTC_DRx) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1) に初期化モードで、またはバイパスシャドウレジスタモード (BYP SHAD = 1) で、ハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

ビット 4 INITS : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (バックアップドメインリセット値状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

- ビット 3 **SHPF** : シフト操作保留
- 0 : 保留中のシフト操作はありません。
1 : 保留中のシフト操作があります。
- このフラグは、RTC_SHIFTR への書き込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF への書き込みは無効です。
- ビット 2 **WUTWF** : ウェイクアップタイマ書き込みフラグ
- このビットは、RTC_CR で WUTE ビットが 0 にセットされた後、最大 2 RTCCLK サイクルでハードウェアによってセットされます。このビットは、WUTE ビットが 1 にセットされた後、最大 2 RTCCLK サイクルでクリアされます。ウェイクアップタイマ値は、WUTE ビットがクリアされ、WUTWF がセットされたときに変更されます。
- 0 : ウェイクアップタイマ設定は更新できません。
1 : ウェイクアップタイマ設定は更新できます。
- ビット 1 **ALRBWF** : アラーム B 書き込みフラグ
- このビットは、RTC_CR で ALRBIE ビットが 0 にセットされた後、アラーム B 値が変更可能な時にハードウェアによってセットされます。
- 初期化モードでハードウェアによってクリアされます。
- 0 : アラーム B は更新できません。
1 : アラーム B は更新できます。
- ビット 0 **ALRAWF** : アラーム A 書き込みフラグ
- このビットは、RTC_CR で ALRAIE ビットが 0 にセットされた後、アラーム A 値が変更可能な時にハードウェアによってセットされます。
- 初期化モードでハードウェアによってクリアされます。
- 0 : アラーム A は更新できません。
1 : アラーム A は更新できます。

注 : **ALRAF, ALRBF, WUTF および TSF ビットは、0 にプログラムされてから 2 APB クロック サイクル後にクリアされます。**

このレジスタは、書き込み保護されています (RTC_ISR[13:8] ビットを除く)。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(606 ページ\)](#) を参照してください。

22.6.5 RTC プリスケアラレジスタ (RTC_PRER)

アドレスオフセット : 0x10

バックアップドメインリセット値 : 0x007F 00FF

システムリセット : 影響なし

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



ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22:16 **PREDIV_A[6:0]** : 非同期プリスケアラ分周比
非同期分周比です。
 $ck_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV_A + 1)$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV_S[14:0]** : 同期プリスケアラ分周比
同期分周比です。
 $ck_spre \text{ 周波数} = ck_apre \text{ 周波数} / (PREDIV_S + 1)$

注 : このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。カレンダーの初期化と設定 (606 ページ) を参照してください。
このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護 (606 ページ) を参照してください。

22.6.6 RTC ウェイクアップタイマレジスタ (RTC_WUTR)

アドレスオフセット : 0x14
バックアップドメインリセット値 : 0x0000 FFFF
システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **WUT[15:0]** : ウェイクアップ自動再ロード値ビット
ウェイクアップタイマが有効 (WUTE が 1 にセット) なとき、ck_wut の (WUT[15:0] + 1) サイクル毎に WUTF フラグがセットされます。ck_wut の周期は、RTC_CR レジスタの WUCKSEL[2:0] ビットで選択します。
WUCKSEL[2] = 1 のとき、ウェイクアップタイマは 17 ビットとなり、WUCKSEL[1] が事実上タイマに再ロードされる最上位ビットである WUT[16] となります。
注 : WUTF の最初のアサートは、WUTE がセットされてからck_wut の (WUT+1) サイクル後に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。

注 : このレジスタは、RTC_ISR の WUTWF が 1 にセットされているときにのみ書き込みます。
このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護 (606 ページ) を参照してください。

22.6.7 RTC 校正レジスタ (RTC_CALIBR)

アドレスオフセット : 0x18
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

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

ビット 31:8 予約済みであり、リセット値のままにしておかなければなりません。

ビット 7 **DCS** : デジタル校正符号

0 : 正の校正 : カレンダ更新頻度が増加します。
1 : 負の校正 : カレンダ更新頻度が減少します。

ビット 6:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DC[4:0]** : デジタル校正

DCS = 0 (正校正)

00000 : + 0 ppm
00001 : + 4 ppm (概数値)
00010 : + 8 ppm (概数値)
..
11111 : + 126 ppm (概数値)

DCS = 1 (負校正)

00000 : -0 ppm
00001 : -2 ppm (概数値)
00010 : -4 ppm (概数値)
..
11111 : -63 ppm (概数値)

正確なステップ値については、[RTCCLK=32.768 kHz かつ PREDIV_A+1=128 の場合](#) を参照してください。

注 : このレジスタは、初期化モード (RTC_ISR/INITF = “1”) の場合のみ書き込めます。

 このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(606 ページ\)](#) を参照してください。

22.6.8 RTC アラーム A レジスタ (RTC_ALRMAR)

アドレスオフセット : 0x1C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]		MNU[3:0]				MSK1	ST[2:0]		SU[3:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31 **MSK4** : アラーム A 日付マスク
0 : 日付／曜日が一致すると、アラーム A がセットされます。
1 : アラーム A の比較では日付／曜日を無視します。
- ビット 30 **WDSEL** : 曜日選択
0 : DU[3:0] は日付の一の位を表します。
1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。
- ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位
- ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日
- ビット 23 **MSK3** : アラーム A 時マスク
0 : 時が一致すると、アラーム A がセットされます。
1 : アラーム A の比較では時を無視します。
- ビット 22 **PM** : AM/PM 表記
0 : AM または 24 時間形式
1 : PM
- ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位
- ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位
- ビット 15 **MSK2** : アラーム A 分マスク
0 : 分が一致すると、アラーム A がセットされます。
1 : アラーム A の比較では分を無視します。
- ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位
- ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位
- ビット 7 **MSK1** : アラーム A 秒マスク
0 : 秒が一致すると、アラーム A がセットされます。
1 : アラーム A の比較では秒を無視します。
- ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位
- ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタは、RTC_ISR の ALRAWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護 (606 ページ) を参照してください。

22.6.9 RTC アラーム B レジスタ (RTC_ALRMBR)

アドレスオフセット : 0x20
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31 **MSK4** : アラーム B 日付マスク
0 : 日付／曜日が一致すると、アラーム B がセットされます。
1 : アラーム B の比較では日付／曜日を無視します。
- ビット 30 **WDSEL** : 曜日選択
0 : DU[3:0] は日付の一の位を表します。
1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。
- ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位
- ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日
- ビット 23 **MSK3** : アラーム B 時マスク
0 : 時が一致すると、アラーム B がセットされます。
1 : アラーム B の比較では時を無視します。
- ビット 22 **PM** : AM/PM 表記
0 : AM または 24 時間形式
1 : PM
- ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位
- ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位
- ビット 15 **MSK2** : アラーム B 分マスク
0 : 分が一致すると、アラーム B がセットされます。
1 : アラーム B の比較では分を無視します。
- ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位
- ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位
- ビット 7 **MSK1** : アラーム 秒マスク
0 : 秒が一致すると、アラーム B がセットされます。
1 : アラーム B の比較では秒を無視します。
- ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位
- ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタは、RTC_ISR の ALRBWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護 (606 ページ) を参照してください。

22.6.10 RTC 書き込み保護レジスタ (RTC_WPR)

アドレスオフセット : 0x24

バックアップドメインリセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **KEY** : 書き込み保護キー

このバイトはソフトウェアで書き込まれます。

このバイトを読み出すと常に 0x00 が返されます。

RTC レジスタの書き込み保護解除方法については、[RTC レジスタ書き込み保護](#)を参照してください。

22.6.11 RTC サブセカンドレジスタ (RTC_SSR)

アドレスオフセット : 0x28

バックアップドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0]は、同期プリスケアラのカウンタ内の値です。秒の小数部は、下の式によって与えられます。

秒の小数部 = (PREDIV_S - SS) / (PREDIV_S + 1)

注 : **SS** は、シフト操作後に限り、PREDIV_S より大きな値となる場合があります。この場合、正確な時刻/日付は、RTC_TR/RTC_DR で示される値よりも 1 秒少ない値となります。

22.6.12 RTC シフト制御レジスタ (RTC_SHIFTR)

アドレスオフセット : 0x2C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **ADD1S** : 1 秒加算
0 : 影響なし。
1 : 時計／カレンダーに 1 秒加算します。
このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。
この機能は、SUBFS (下記説明を参照) と共に使用されることを想定しており、不可分操作で、効果的に時計に秒の小数部を加算することを目的としています。

ビット 30:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SUBFS** : 秒の小数部差し引き
このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。
SUBFS に書き込まれた値は、同期プリスケアラのカウンタに加算されます。このカウンタはカウントダウンしていくので、この操作によって、次の式で求める値が効果的にクロックから差し引き(遅延)されます。
$$\text{遅れ (秒)} = \text{SUBFS} / (\text{PREDIV_S} + 1)$$

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。
$$\text{進み (秒)} = (1 - (\text{SUBFS} / (\text{PREDIV_S} + 1)))$$

注 : SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実になります。
[セクション 22.3.8 : RTC の同期](#)を参照してください。

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(606 ページ\)](#) を参照してください。

22.6.13 RTC タイムスタンプ時刻レジスタ (RTC_TSTR)

アドレスオフセット : 0x30
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記
0 : AM または 24 時間形式
1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタの内容は、RTC_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

22.6.14 RTC タイムスタンプ日付レジスタ (RTC_TSDR)

アドレスオフセット : 0x34
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **WDU[1:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

注 : このレジスタの内容は、**RTC_ISR** で **TSF** が 1 にセットされている場合にのみ有効です。また、**TSF** ビットがリセットされるとクリアされます。

22.6.15 RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)

アドレスオフセット : 0x38

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:16 予約済み

ビット 15:0 **SS** : サブセカンド値

SS[15:0]は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。

注 : このレジスタの内容は、**RTC_ISR/TSF** がセットされている場合にのみ有効です。また、**RTC_ISR/TSF** ビットがリセットされるとクリアされます。

22.6.16 RTC 較正レジスタ (RTC_CALR)

アドレスオフセット : 0x3C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
rW	rW	rW	r	r	r	r	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15 CALP : RTC 周波数を 488.5 ppm 増加
0 : RTCCLK パルスは加えられません。
1 : RTCCLK の 2^{11} パルスごとに 1 パルス効果的に挿入されます (周波数が 488.5 ppm 増加)。
この機能は、CALM と共に使用されることを想定しており、カレンダーの周波数を高分解能で下げることができます。入力周波数が 32768 Hz の場合、32 秒枠の間に追加される RTCCLK パルス数は次のように算出されます : $(512 * CALP) - CALM$
[セクション 22.3.11 : RTC の高精度デジタル較正](#)を参照してください。

ビット 14 CALW8 : 8 秒較正サイクル周期の使用
CALW8 が“1”にセットされると、8 秒較正サイクル周期が選択されます。
CALW8 = 1 の場合、CALM[1:0] は“00”に固定されます。
[セクション 22.3.11 : RTC の高精度デジタル較正](#)を参照してください。

ビット 13 CALW16 : 16 秒較正サイクル周期の使用
CALW16 が“1”にセットされると、16 秒較正サイクル周期が選択されます。CALW8=1 の場合、このビットを“1”にセットすることはできません。
注 : CALW16 = 1 の場合、CALM[0] は“0”に固定されます。
[セクション 22.3.11 : RTC の高精度デジタル較正](#)を参照してください。

ビット 12:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8:0 CALM[8:0] : 較正マイナス
RTCCLK 2^{20} パルス (入力周波数が 32768 Hz の場合 32 秒) 内の CALM をマスクすることによって、カレンダーの周波数が下げられます。この方法により、カレンダーの周波数を 0.9537 ppm の分解能で下げることができます。
カレンダーの周波数を上げるには、この機能を CALP と共に使用する必要があります。
[セクション 22.3.11 : RTC の高精度デジタル較正 \(611 ページ\)](#)を参照してください。

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護を参照してください](#)。

22.6.17 RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)

アドレスオフセット : 0x40
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALARMOUT TYPE	TSIN SEL	TAMP11 NSEL
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP PUDIS	TAMP PRCH[1:0]	TAMP FLT[1:0]	TAMP FREQ[2:0]			TAMPTS	Res.	Res.	Res.	Res.	TAMPIE	TAMP1TRG	TAMP1E		
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	

ビット 31:19 予約済みであり、リセット値に保持する必要があります。常に 0 が読み出されます。

ビット 18 ALARMOUTTYPE : RTC_ALARM 出力形式
0 : RTC_ALARM はオープンドレイン出力になります。
1 : RTC_ALARM はプッシュプル出力になります。



ビット 17 **TSINSEL** : TIMESTAMP マッピング

0 : RTC_AF1 が TIMESTAMP として使用されます。

1 : 予約済み

ビット 16 **TAMP1INSEL** : TAMPER1 マッピング

0 : RTC_AF1 が TAMPER1 として使用されます。

1 : 予約済み

注 : **TAMP1E** ビットは、**TAMP1INSEL** が変更されたらクリアし、**TAMP1F** の不要なセットを回避する必要があります。

ビット 15 **TAMPPUDIS** : TAMPER ブルアップ無効

このビットにより、毎回のサンプリング前に各タンパピンをプリチャージするかどうか決定します。

0 : サンプリング前にタンパピンをプリチャージします (内部ブルアップを有効化)。

1 : タンパピンのプリチャージを無効化します。

注 :

ビット 14:13 **TAMPPRCH[1:0]** : タンパプリチャージ持続時間

これらのビットにより、各サンプリングの前にブルアップを有効化している時間を決定します。
TAMPPRCH は、各タンパ入力に対して有効です。

0x0 : 1 RTCCLK サイクル

0x1 : 2 RTCCLK サイクル

0x2 : 4 RTCCLK サイクル

0x3 : 8 RTCCLK サイクル

ビット 12:11 **TAMPFLT[1:0]** : タンパフィルタカウント

これらのビットにより、タンパイベントをアクティブにするのに必要な指定のレベル (TAMP*TRG) での連続サンプリングの数を決定します。TAMPFLT は、各タンパ入力に対して有効です。

0x0 : タンパ入力 that アクティブレベル (タンパ入力における内部ブルアップ無し) に変化するときのエッジで、タンパがアクティブになります。

0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパがアクティブになります。

0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパがアクティブになります。

0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパがアクティブになります。

ビット 10:8 **TAMPFREQ[2:0]** : タンパサンプリング周波数

これらのビットにより、各タンパ入力 that サンプリングされる周波数を決定します。

0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)

0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)

0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)

0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)

0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)

0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)

0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)

0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

ビット 7 **TAMPTS** : タンパ検出イベント時のタイムスタンプの有効化

0 : タンパ検出イベントがあっても、タイムスタンプは保存されません。

1 : タンパ検出イベント時、タイムスタンプが保存されます。

RTC_CR レジスタで TSE = 0 であっても TAMPTS は有効です。

ビット 6:3 予約済み。常に 0 が読み出されます。

ビット 2 **TAMPIE** : タンパ割り込みイネーブル

0 : タンパ割り込みは無効です。

1 : タンパ割り込みは有効です。

ビット 1 **TAMP1TRG** : タンパ 1 のアクティブレベル

TAMPFLT != 00の場合 :

0 : TAMPER1 がローのままのとき、タンパ検出イベントがトリガされます。

1 : TAMPER1 がハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00の場合 :

0 : TAMPER1 の立ち上りエッジでタンパ検出イベントがトリガされます。

1 : TAMPER1 の立ち下りエッジでタンパ検出イベントがトリガされます。

注意 : **TAMPFLT = 0 の場合、TAMP1E ビットは、TAMP1TRG が変更されたらリセットし、TAMP1F の不要なセットを回避する必要があります。**

ビット 0 **TAMP1E** : タンパ 1 検出イネーブル

0 : タンパ 1 検出は無効です。

1 : タンパ 1 検出は有効です。

22.6.18 **RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)**

アドレスオフセット : 0x44

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:24 **MASKSS[3:0]** : このビットから始まる最上位ビットのマスク

0 : アラーム A に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

1 : アラーム A の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

2 : アラーム A の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

3 : アラーム A の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

12 : アラーム A の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

13 : アラーム A の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

14 : アラーム A の比較では SS[14] を無視します。SS[13:0] が比較されます。

15 : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SS[14:0]** : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

注： このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護 (606 ページ) を参照してください。

22.6.19 RTC アラーム B サブセカンドレジスタ（RTC_ALRMBSSR）

アドレスオフセット：0x48

バックアップドメインリセット値：0x0000 0000

システムリセット：影響なし

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:24 **MASKSS[3:0]**：このビットから始まる最上位ビットのマスク

0x0：アラーム B に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます（他の項目が一致していることを前提として）。

0x1：アラーム B の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

0x2：アラーム B の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

0x3：アラーム B の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

0xC：アラーム B の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

0xD：アラーム B の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

0xE：アラーム B の比較では SS[14] を無視します。SS[13:0] が比較されます。

0xF：アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット（ビット 15）が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SS[14:0]**：サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

注： このレジスタは、RTC_CR レジスタの ALRBIE がリセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、次を参照してください。[セクション：RTC レジスタ書き込み保護](#)

22.6.20 RTC バックアップレジスタ (RTC_BKPxR)

アドレスオフセット : 0x50 から 0x9C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

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

ビット 31:0 BKP[31:0]

アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。
これらのレジスタは、V_{DD} がオフになった場合、V_{BAT} によって電源が供給されるため、システムリセットによりリセットされず、デバイスが低電力モードで動作する場合、レジスタの内容は有効なまま保持されます。
このレジスタは、TAMPx_F=1 である限り、タンパ検出イベントでリセットされます。

22.6.21 RTC レジスタマップ

表 109. RTC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RTC_TR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]	HU[3:0]			Res.	MNT[2:0]			MNU[3:0]			Res.	ST[2:0]		SU[3:0]								
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0	
0x04	RTC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]			YU[3:0]			WDU[2:0]			MT	MU[3:0]			Res.	Res.	DT [1:0]		DU[3:0]							
	リセット値																	0	0	1	0	0	0	0	1			0	0	0	0	0	1	
0x08	RTC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DCE	FMT	BYPHAD	REFCKON	TSEDGE	WCKSEL [2:0]			
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	RTC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INTS	SHPF	WUTWF	ALBWF	ALRAWF	
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	1	0	1	
0x10	RTC_PRER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						Res.	PREDIV_S[14:0]																
	リセット値										1	1	1	1	1	1	1		0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
0x14	RTC_WUTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUT[15:0]																
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x18	RTC_CALIBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCS	Res.	Res.	DC[4:0]					
	リセット値																									0			0	0	0	0	0	
0x1C	RTC_ALRMAR	MSK4	WDSEL	DT [1:0]	DU[3:0]				MSK3	PM	HT [1:0]	HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]			MSK1	ST[2:0]		SU[3:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	RTC_ALRMBR	MSK4	WDSEL	DT [1:0]	DU[3:0]				MSK3	PM	HT [1:0]	HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]			MSK2	ST[2:0]		SU[3:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x24	RTC_WPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[7:0]								
	リセット値																									0	0	0	0	0	0	0	0	
0x28	RTC_SSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	RTC_SHIFTR	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]															
	リセット値	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 109. RTC レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
0x30	RTC_TSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]			Res.		ST[2:0]			SU[3:0]											
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0								
0x38	RTC_TSSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]																							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x3C	RTC_CALR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]																
	リセット値																	0	0	0					0	0	0	0	0	0	0	0	0	0							
0x40	RTC_TAFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALARMOUTTYPE	TSINSEL	TAMP1INSEL	TAMPPUDIS	TAMPPRCH[1:0]			TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS		Res.	Res.	Res.	Res.	TAMPIE	TAMP1ETR	TAMPIE			
	リセット値															0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0					
0x44	RTC_ALRMASSR	Res.	Res.	Res.	Res.	MASKSS[3:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]																					
	リセット値					0	0	0	0											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x48	RTC_ALRMBSSR	Res.	Res.	Res.	Res.	MASKSS[3:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]																					
	リセット値					0	0	0	0											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x50 0x9C へ	RTC_BKP0R	BKP[31:0]																																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	RTC_BKP19R へ	BKP[31:0]																																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

注意： [表 109](#) では、リセット値とはバックアップドメインリセット後の値です。レジスタの大部分は、システムリセットによる影響を受けません。詳細については、[セクション 22.3.7 : RTC のリセット](#) を参照してください。

23 FMPI2C（高速モードプラス Inter-Integrated Circuit）インタフェース

23.1 概要

I²C（Inter-integrated circuit）バスインタフェースは、マイクロコントローラとシリアル I²C バス間の通信を処理します。マルチマスタ機能を備え、すべての I²C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準モード（Sm）、高速モード（Fm）、および高速モードプラス（Fm+）をサポートします。

また、SMBus（System Management Bus）および PMBus（Power Management Bus）と互換性があります。

DMA を使用して、CPU の負荷を軽減できます。

23.2 FMPI2C の主な機能

- I²C バス仕様 rev03 との互換性：
 - － スレーブおよびマスタモード
 - － マルチマスタ機能
 - － 標準モード（最大 100 kHz）
 - － 高速モード（最大 400 kHz）
 - － 高速モードプラス（最大 1 MHz）
 - － 7ビットおよび 10 ビットアドレッシングモード
 - － 複数の 7 ビットスレーブアドレス（2 つのアドレス、1 つは設定可能なマスク付き）
 - － すべての 7 ビットアドレス確認応答モード
 - － 同報（General call）コール
 - － プログラム可能なセットアップおよびホールド時間
 - － 使いやすいイベント管理
 - － クロックストレッチオプション
 - － ソフトウェアリセット

- DMA 機能付きの 1 バイトバッファ
- プログラム可能なアナログおよびデジタルノイズフィルタ

製品の実装によっては、次の追加機能も使用できます（[セクション 23.3 : FMPI2C の実装](#) を参照）：

- SMBus 仕様 rev 2.0 との互換性：
 - － ハードウェア PEC（Packet Error Checking）の生成と ACK 制御による確認
 - － コマンドおよびデータ確認応答制御
 - － アドレス解決プロトコル（ARP）サポート
 - － ホストおよびデバイスのサポート
 - － SMBus アラート
 - － タイムアウトおよびアイドル条件の検出
- PMBus rev 1.1 標準との互換性
- 独立したクロック：独立したクロックソースの選択により、FMPI2C の通信速度は PCLK の再プログラミングから独立

23.3 FMPI2C の実装

このマニュアルでは、FMPI2C1 に実装されているすべての機能について説明しています。

表 110. STM32F412xx FMPI2C の実装

I2C の機能 ⁽¹⁾	I2CFMP1
独立クロック	X
SMBus	X
STOP モードからのウェイクアップ	-

1. X：サポートされています。

23.4 FMPI2C 機能の詳細

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I2C バスに接続されます。標準（最大 100 kHz）、高速モード（最大 400 kHz）、または高速モードプラス（最大 1 MHz）の I²C バスで接続できます。

このインタフェースは、データピン（SDA）とクロックピン（SCL）によって SMBus に接続することもできます。

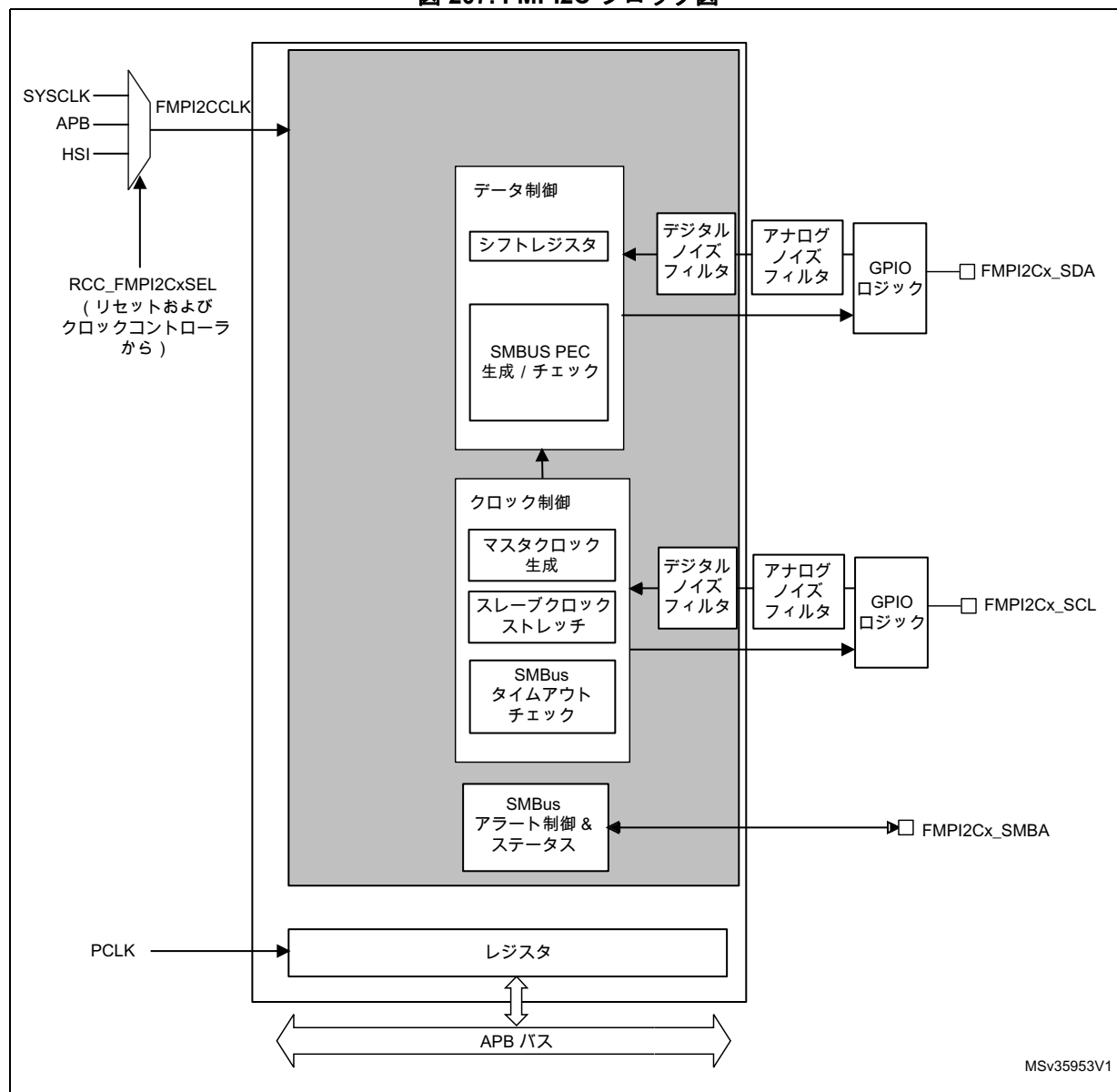
SMBus 機能がサポートされる場合、追加の SMBus アラートピン（SMBA）オプションも使用できます。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

23.4.1 FMPI2C ブロック図

FMP I2C インタフェースのブロック図を [図 207](#) に示します。

図 207. FMPI2C ブロック図



FMPI2C は、独立したクロックソースによってクロック供給されるため、FMPI2C は PCLK 周波数から独立して動作できます。

この独立したクロックソースは、次のクロックソースから選択できます：

- PCLK1：APB1 クロック（デフォルト値）
- HSI：高速内部オシレータ
- SYSCLK：システムクロック

詳細については、[セクション 6：STM32F412xx のリセットおよびクロック制御（RCC）](#)を参照してください。

23.4.2 FMPI2C クロックの要件

FMPI2C カーネルは FMPI2CCLK によってクロック供給されます。

FMPI2CCLK の周期 t_{I2CCLK} は、次の条件を満たす必要があります。

$$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4 \text{ and } t_{I2CCLK} < t_{HIGH}$$

ここで：

t_{LOW} : SCL ロー時間、および t_{HIGH} : SCL ハイ時間

$t_{filters}$: 有効なときには、アナログフィルタとデジタルフィルタによる遅延の合計。

アナログフィルタの遅延は、最大 260 ns です。デジタルフィルタの遅延は、 $DNF \times t_{I2CCLK}$ です。

PCLK の周期 t_{PCLK} は、次の条件を満たす必要があります。

$$t_{PCLK} < 4/3 t_{SCL}$$

t_{SCL} : SCL 周期

注意： *FMPI2C カーネルが PCLK によってクロック供給されるとき、PCLK は t_{I2CCLK} の条件を満たす必要があります。*

23.4.3 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成したときにはスレーブからマスタへ、アービトレーションの喪失または STOP 生成が発生したときにはマスタからスレーブへ自動的に切り替わるため、マルチマスタ機能を使用できます。

通信の流れ

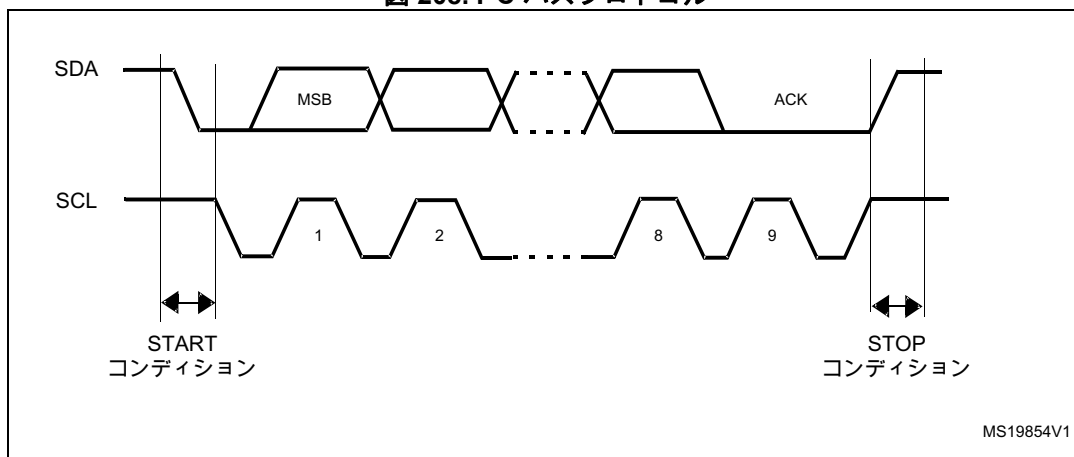
マスタモードでは、FMPI2C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス（7 または 10 ビット）と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効または無効にできます。予約済みの SMBus アドレスもソフトウェアによって有効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト（7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト）にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。次の図を参照してください。

図 208. I²C バスプロトコル



確認応答（Acknowledge）は、ソフトウェアによって有効または無効にできます。FMPI2C インタフェースのアドレスは、ソフトウェアによって選択できます。

23.4.4 FMPI2C の初期化

ペリフェラルの有効化と無効化

FMPI2C ペリフェラルクロックは、クロックコントローラで設定し、有効にする必要があります（[セクション 6：STM32F412xx のリセットおよびクロック制御 \(RCC\)](#)を参照）。

そして、FMPI2C_CR1 レジスタの PE ビットをセットすることによって、FMPI2C を有効にできます。

FMPI2C が無効なときには (PE=0)、I²C はソフトウェアリセットを実行します。詳細については、[セクション 23.4.5：ソフトウェアリセット](#)を参照してください。

ノイズフィルタ

FMPI2C_CR1 レジスタの PE ビットをセットすることによって FMPI2C ペリフェラルを有効にする前に、必要な場合は、ノイズフィルタを設定する必要があります。デフォルトでは、SDA および SCL 入力にアナログノイズフィルタがあります。このアナログフィルタは I²C 仕様に準拠しており、高速モードおよび高速モードプラスで最大 50 ns のパルス幅を持つスパイクを抑制します。ANFOFF ビットをセットすることによって、このアナログフィルタを無効にし、FMPI2C_CR1 レジスタの DNF[3:0] ビットを設定することによってデジタルフィルタを選択することができます。

デジタルフィルタが有効なときには、SCL または SDA ラインのレベルは、DNF x FMPI2CCCLK 周期より長く安定していた場合のみ、内部で変更されます。これにより、プログラム可能な 1 ~ 15 FMPI2CCCLK 周期の長さを持つスパイクを抑制できます。

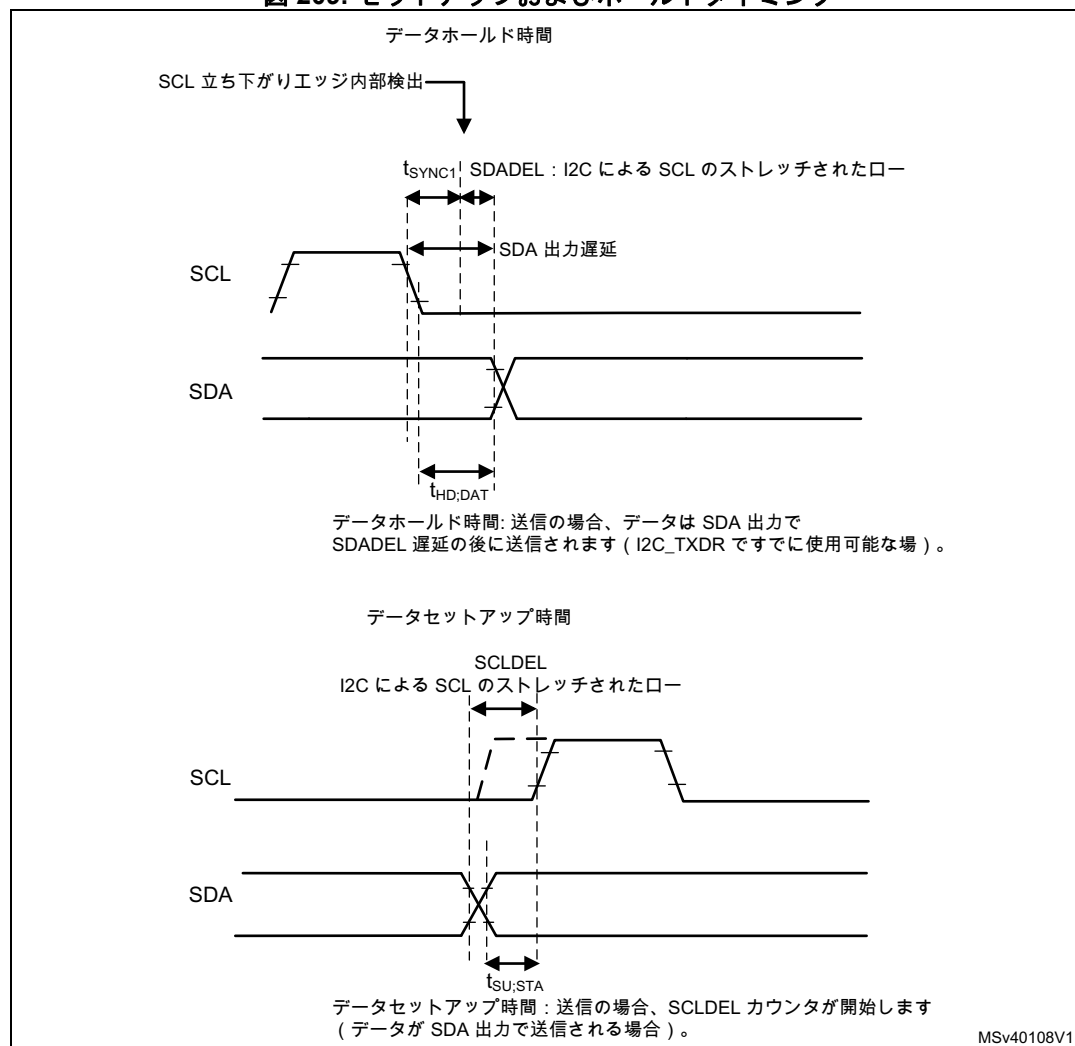
注意： FMPI2C が有効なときには、フィルタ設定の変更はできません。

FMPI2C のタイミング

マスタおよびスレーブモードで正しいデータホールドおよびセットアップ時間が使用されるのを保証するためには、タイミングを設定する必要があります。これを行うには、FMPI2C_TIMINGR レジスタの PRESC[3:0]、SCLDEL[3:0]、および SDADEL[3:0] ビットをプログラムします。

STM32CubeMX ツールは、I2C 設定ウィンドウの I2C_TIMINGR の内容を計算して示します。

図 209. セットアップおよびホールドタイミング



- SCL 立ち下がりエッジが内部で検出されると、SDA 出力を送信する前に遅延が挿入されます。この遅延は、 $t_{SDA\Delta EL} = SDA\Delta EL \times t_{PRESC} + t_{I2CCLK}$ であり $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。 $t_{SDA\Delta EL}$ はホールド時間 $t_{HD;DAT}$ に影響を与えます。

SDA 出力遅延の合計は、次のとおりです：

$$t_{SYNC1} + \{[SDA\Delta EL \times (PRESC+1) + 1] \times t_{I2CCLK}\}$$

t_{SYNC1} の長さは、次のパラメータに依存します。

- SCL 立ち下がり傾斜
- アナログフィルタが有効なときの入力遅延： $t_{AF(min)} < t_{AF} < t_{AF(max)}$ ns
- デジタルフィルタが有効なときの入力遅延： $t_{DNF} = DNF \times t_{I2CCLK}$
- SCL と FMPI2CCLK クロックの同期による遅延（2 ～ 3 FMPI2CCLK 周期）

SCL 立ち下がりエッジの未定義の領域をブリッジするためには、SDA ΔEL を次のようにプログラムする必要があります：

$$\{t_r(max) + t_{HD;DAT(min)} - t_{AF(min)} - [(DNF+3) \times t_{I2CCLK}]\} / \{(PRESC+1) \times t_{I2CCLK}\} \leq SDA\Delta EL$$

$$SDA\Delta EL \leq \{t_{HD;DAT(max)} - t_{AF(max)} - [(DNF+4) \times t_{I2CCLK}]\} / \{(PRESC+1) \times t_{I2CCLK}\}$$

注： $t_{AF(min)} / t_{AF(max)}$ は、アナログフィルタが有効なときのみ、等式に含まれます。 t_{AF} の値については、デバイスのデータシートを参照してください。

最大 $t_{HD;DAT}$ は、標準モード、高速モード、および高速モードプラスで 3.45 μ s、0.9 μ s、および 0.45 μ s ですが、遷移時間による $t_{VD;DAT}$ の最大値より短い必要があります。この最大値を満たす必要があるのは、デバイスが SCL 信号の LOW 周期（ t_{LOW} ）をストレッチしない場合だけです。クロックが SCL をストレッチする場合、クロックをリリースする前に、データがセットアップ時間まで有効である必要があります。

SDA 立ち上がりエッジは、通常、最悪ケースであり、この場合、前の等式は次のようになります：

$$SDA\Delta EL \leq \{t_{VD;DAT(max)} - t_r(max) - 260\text{ ns} - [(DNF+4) \times t_{I2CCLK}]\} / \{(PRESC+1) \times t_{I2CCLK}\}$$

注： **NOSTRETCH=0 のときには、SCLDEL の値に従って、デバイスはセットアップ時間を保証するために SCL ローをストレッチするので、この条件に違反することがあります。**

t_r 、 t_r 、 $t_{HD;DAT}$ 、および $t_{VD;DAT}$ の標準値については、表 111：I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

- $t_{SDA\Delta EL}$ 遅延の後や、データが I2C_TXDR レジスタにまだ書き込まれていないためスレーブがクロックをストレッチしなければならない場合の SDA 出力送信後、SCL ラインはセットアップ時間中、ローレベルで保持されます。このセットアップ時間は、 $t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$ であり $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。

t_{SCLDEL} は、セットアップ時間 $t_{SU;DAT}$ に影響を与えます。

SDA 遷移（立ち上がりエッジは通常、最悪のケース）の未定義の領域をブリッジするためには、SCLDEL を次のようにプログラムする必要があります：

$$\{[t_r(max) + t_{SU;DAT(min)}] / [(PRESC+1) \times t_{I2CCLK}]\} - 1 \leq SCLDEL$$

t_r および $t_{SU;DAT}$ の標準値については、表 111：I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

使用される SDA および SCL 遷移時間の値は、アプリケーションの値です。標準から最大値を使用すると、SDA ΔEL と SCLDEL の計算の制約が増えますが、アプリケーションにかかわらず、この機能を使用できます。

注：各クロックパルスで、SCL 立ち下がりエッジの検出後、I2C マスタまたはスレーブは、最低 $\lceil (SDADEL+SCLDEL+1) \times (PRESC+1) + 1 \rceil \times t_{I2CCLK}$ の間、送信および受信の両モードで SCL ローをストレッチします。送信モードで、SDADEL カウンタ終了時にデータがまだ I2C_TXDR に書き込まれていない場合、I2C は次のデータが書き込まれるまで SCL ローをストレッチし続けます。その時、新しいデータ MSB が SDA 出力で送信され、SCLDEL カウンタが開始し、SCL ローのストレッチを継続して、データセットアップ時間を保証します。

スレーブモードで NOSTRETCH=1 の場合、SCL はストレッチされません。そのため、十分なセットアップ時間を保証するためにも、SDADEL はこのようにプログラムされる必要があります。

表 111. I²C-SMBUS 仕様のデータのセットアップおよびホールド時間

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
t _{HD;DAT}	データホールド時間	0	-	0	-	0	-	0.3	-	μs
t _{VD;DAT}	データ有効時間	-	3.45	-	0.9	-	0.45	-	-	
t _{SU;DAT}	データセットアップ時間	250	-	100	-	50	-	250	-	ns
t _r	SDA および SCL 信号の立ち上がり時間	-	1000	-	300	-	120	-	1000	
t _f	SDA および SCL 信号の立ち下がり時間	-	300	-	300	-	120	-	300	

また、マスタモードでは、FMPI2C_TIMINGR レジスタの PRESC[3:0]、SCLH[7:0]、および SCLL[7:0] ビットをプログラムすることによって、SCL クロックのハイおよびローレベルを設定する必要があります。

- SCL 立ち下がりエッジが内部で検出されると、SCL 出力をリリースする前に遅延が挿入されます。この遅延は、 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。
 t_{SCLL} は、SCL ロー時間 t_{LOW} に影響を与えます。
- SCL 立ち上がりエッジが内部で検出されると、SCL 出力を強制的にローレベルにする前に遅延が挿入されます。この遅延は、 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。
 t_{SCLH} は、SCLハイ時間 t_{HIGH} に影響を与えます。

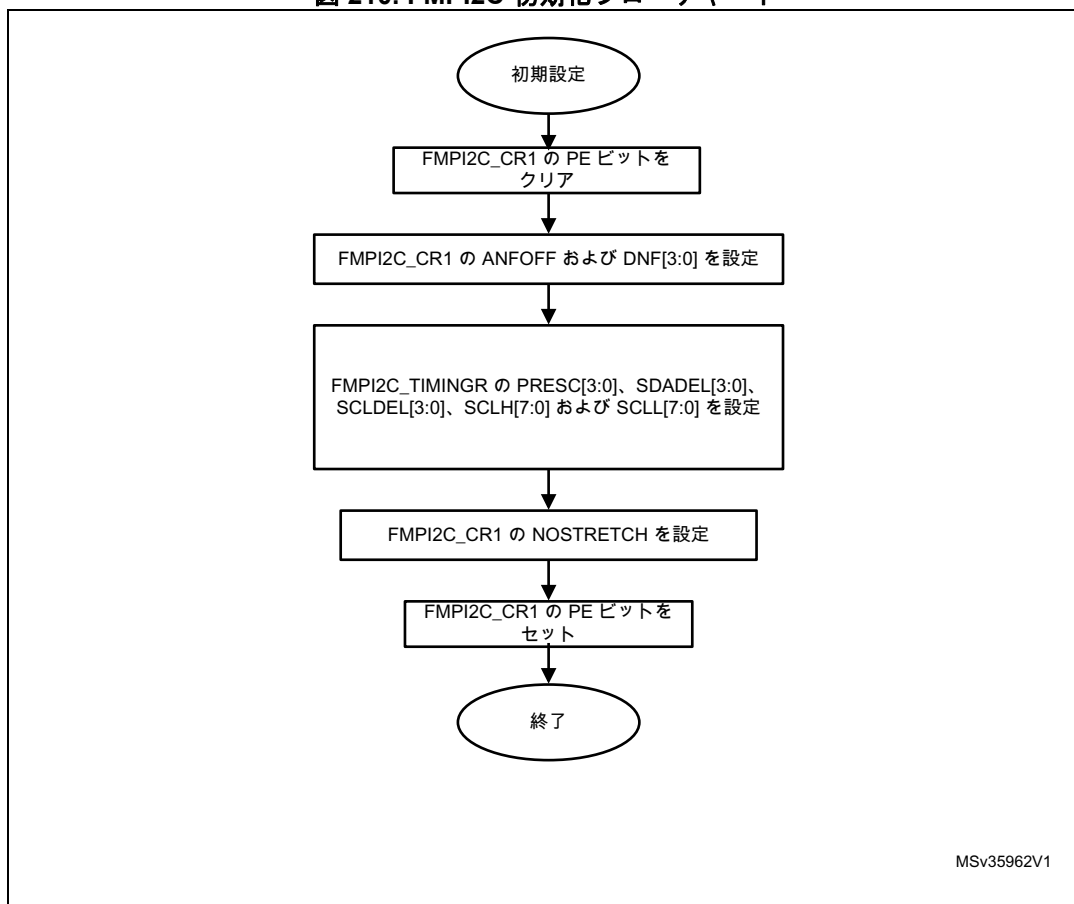
詳細については、[FMPI2C マスタ初期化](#) を参照してください。

注意：FMPI2C が有効なときには、タイミング設定の変更はできません。

ペリフェラルを有効にする前に、FMPI2C スレーブ NOSTRETCH モードも設定する必要があります。詳細については、[FMPI2C スレーブ初期化](#) を参照してください。

注意：FMPI2C が有効なときには、NOSTRETCH 設定の変更はできません。

図 210. FMPI2C 初期化フローチャート



23.4.5 ソフトウェアリセット

ソフトウェアリセットを行うには、FMPI2C_CR1 レジスタの PE ビットをクリアします。その場合、FMPI2C のライン SCL および SDA がリリースされます。内部状態マシンがリセットされ、通信制御ビットとステータスビットがリセット値に戻ります。構成レジスタは影響を受けません。

影響を受けるレジスタのビットは、以下のとおりです：

1. FMPI2C_CR2 レジスタ：START、STOP、NACK
2. FMPI2C_ISR レジスタ：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO、OVR

SMBus 機能がサポートされるときには、以下も影響を受けます：

1. FMPI2C_CR2 レジスタ：PECBYTE
2. FMPI2C_ISR レジスタ：PECERR、TIMEOUT、ALERT

ソフトウェアリセットを実行するためには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たなければなりません。このためには、次のソフトウェアシーケンスを書き込みます：- PE=0 を書き込む - PE=0 を確認する - PE=1 を書き込む

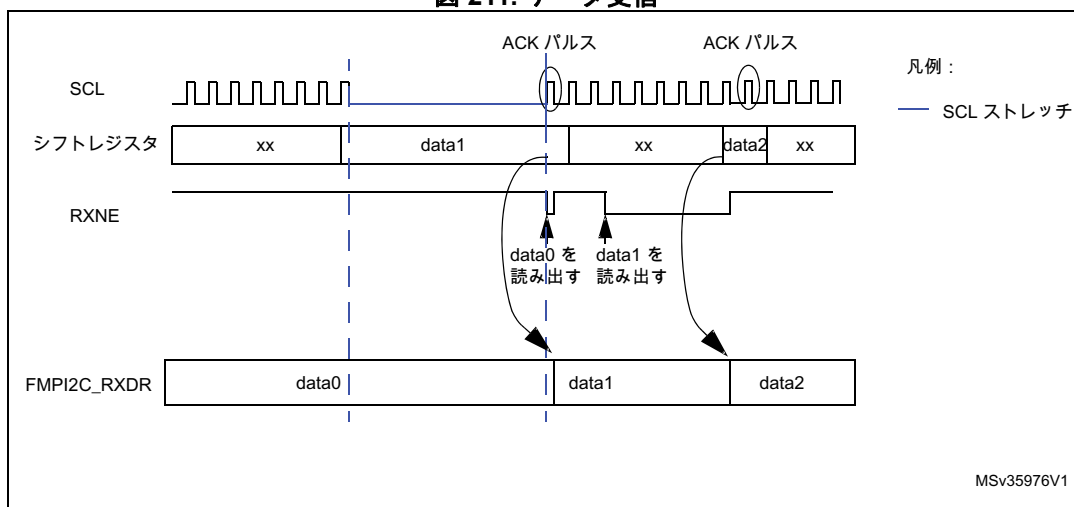
23.4.6 データ転送

データ転送は、送受信データレジスタとシフトレジスタを通じて管理されます。

受信

SDA 入力はシフトレジスタに送られます。8 番目の SCL パルスの後（完全なデータバイトの受信後）、シフトレジスタは、FMPI2C_RXDR レジスタが空の場合（RXNE=0）、このレジスタにコピーされます。RXNE=1 の場合、すなわち、前に受信されたデータバイトがまだ読み出されていなかった場合、SCL ラインは FMPI2C_RXDR が読み出されるまでストレッチされます。ストレッチは、8 番目と 9 番目の SCL パルスの間（確認応答パルスの前）に挿入されます。

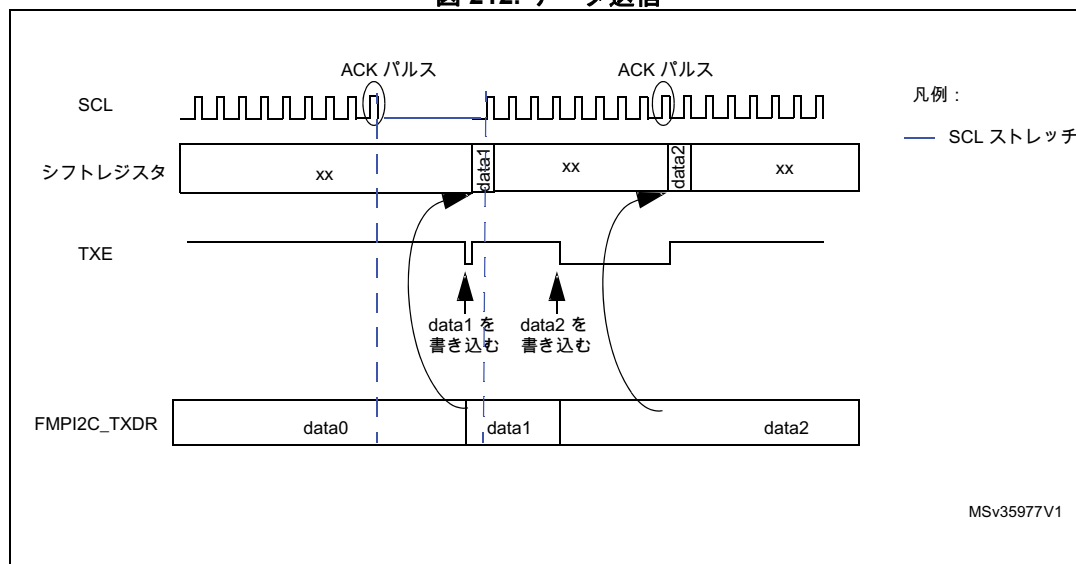
図 211. データ受信



送信

FMPI2C_TXDR レジスタが空（TXE=0）でない場合、その内容が 9 番目の SCL パルス（確認応答パルス）の後、シフトレジスタにコピーされます。次に、シフトレジスタの内容が SDA ラインにシフトアウトされます。TXE=1 の場合、すなわち、FMPI2C_TXDR にデータがまだ書き込まれていない場合、SCL ラインは FMPI2C_TXDR に書き込まれるまでストレッチされます。ストレッチは、9 番目の SCL パルスの後で行われます。

図 212. データ送信



ハードウェア転送管理

次のようにさまざまなモードでバイト転送を管理し、通信をクローズするために、FMPI2C にはハードウェアにバイトカウンタが組み込まれています：

- マスタモードでの NACK、STOP、および ReSTART 生成
- スレーブレシーバモードでの ACK 制御
- SMBus 機能がサポートされているときの PEC 生成／確認

バイトカウンタは、マスタモードでは常に使用されます。デフォルトでは、スレーブモードでは無効ですが、FMPI2C_CR2 レジスタの SBC（スレーブバイト制御）ビットをセットすることによって、ソフトウェアにより有効にできます。

転送されるバイト数は、FMPI2C_CR2 レジスタの NBYTES[7:0] ビットフィールドでプログラムされます。転送バイト数（NBYTES）が 255 より大きい場合、またはレシーバが受信データバイトの確認応答値を制御したい場合には、FMPI2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。このモードでは、NBYTES でプログラムされたバイト数が転送されると、TCR フラグがセットされ、TCIE がセットされている場合は割り込みが生成されます。SCL は、TCR フラグがセットされている間、ストレッチされます。TCR は、NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

NBYTES カウンタに最後のバイト数が再ロードされたときには、RELOAD ビットがクリアされる必要があります。

マスタモードで RELOAD=0 のときには、カウンタは 2 つのモードで使用できます：

- **自動終了モード** (FMPI2C_CR2 レジスタの AUTOEND = 1)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、マスタは STOP コンディションを自動的に送信します。
- **ソフトウェア終了モード** (FMPI2C_CR2 レジスタの AUTOEND = 0)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、ソフトウェアアクションが求められます。TC フラグがセットされ、TCIE ビットがセットされている場合は割り込みが生成されます。SCL 信号は、TC フラグがセットされている間、ストレッチされます。TC フラグは、FMPI2C_CR2 レジスタの START または STOP ビットがセットされたときに、ソフトウェアによってクリアされます。マスタが RESTART コンディションを送信したいときには、このモードを使用する必要があります。

注意： *AUTOEND ビットは、RELOAD ビットがセットされているときには効果がありません。*

表 112. FMPI2C 設定表

機能	SBC ビット	RELOAD ビット	AUTOEND ビット
マスタ Tx/Rx NBYTES + STOP	x	0	1
マスタ Tx/Rx + NBYTES + RESTART	x	0	0
スレーブ Tx/Rx すべての受信バイトに ACK	0	x	x
スレーブ Rx および ACK 制御	1	1	x

23.4.7 FMPI2C スレーブモード

FMPI2C スレーブ初期化

スレーブモードで動作するには、少なくとも 1 つのスレーブアドレスを有効にする必要があります。2 つのレジスタ FMPI2C_OAR1 と FMPI2C_OAR2 を使用して、スレーブ専用アドレス OA1 および OA2 をプログラムできます。

- OA1 は、FMPI2C_OAR1 レジスタの OA1MODE ビットをセットすることによって、7 ビットモード（デフォルト）または 10 ビットアドレッシングモードに設定できます。
OA1 を有効にするには、FMPI2C_OAR1 レジスタの OA1EN ビットをセットします。
- 追加のスレーブアドレスが必要な場合は、2 番目のスレーブアドレス OA2 を設定できます。FMPI2C_OAR2 レジスタの OA2MSK[2:0] ビットを設定することによって、最大 7 つの OA2 LSB をマスクできます。したがって、OA2MSK が 1 から 6 まで設定された場合、OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]、または OA2[7] のみが受信アドレスと比較されます。OA2MSK が 0 に等しくなるとすぐに、OA2 のアドレスコンパレータは、確認応答されない FMPI2C 予約済みアドレス（0000 XXX および 1111 XXX）を除外します。OA2MSK=7 の場合、受信されたすべてのアドレスが確認応答されます（予約済みアドレスを除く）。OA2 は常に 7 ビットアドレスです。
これらの予約済みアドレスは、特定のイネーブルビットによって有効化された場合、FMPI2C_OAR1 または FMPI2C_OAR2 レジスタが OA2MSK=0 でプログラムされた場合、確認応答できます。
OA2 を有効にするには、FMPI2C_OAR2 レジスタの OA2EN ビットをセットします。
- 同報アドレスは、FMPI2C_CR1 レジスタの GCEN ビットをセットすることで有効になります。

FMPI2C が有効アドレスの 1 つによって選択されると、ADDR 割り込みステータスフラグがセットされ、ADDRIE ビットがセットされている場合は割り込みが生成されます。

デフォルトでは、スレーブはクロックストレッチ機能を使用し、必要なときには、ソフトウェアアクションを実行するために、SCL 信号をローレベルでストレッチすることを意味します。マスタがクロックストレッチをサポートしない場合、FMPI2C_CR1 レジスタの NOSTRETCH=1 で FMPI2C を設定する必要があります。

ADDR 割り込みの受信後、いくつかのアドレスが有効な場合は、FMPI2C_ISR レジスタの ADDCODE[6:0] ビットを読み出して、一致するアドレスを確認する必要があります。転送方向を知るために、DIR フラグも確認する必要があります。

スレーブクロックストレッチ（NOSTRETCH = 0）

デフォルトモードでは、FMPI2C スレーブは次の状況で SCL クロックをストレッチします：

- ADDR フラグがセットされると：受信アドレスは有効なスレーブアドレスの 1 つと一致します。このストレッチは、ADDRCF ビットをセットすることによりソフトウェアによって ADDR フラグがクリアされたときにリリースされます。
- 送信時、前のデータ送信が完了し、新しいデータが FMPI2C_TXDR レジスタに書き込まれなかった場合、または ADDR フラグがクリアされたときに（TXE=1）、最初のデータバイトが書き込まれていなかった場合。このストレッチは、データが FMPI2C_TXDR レジスタに書き込まれたときにリリースされます。
- 受信時、FMPI2C_RXDR レジスタがまだ読み出されておらず、新しいデータ受信が完了したとき。このストレッチは、FMPI2C_RXDR が読み出されたときにリリースされます。
- スレーブバイト制御モードおよび再ロードモード（SBC=1 および RELOAD=1）で TCR = 1 のとき、すなわち、最後データバイトが転送されたとき。このストレッチは、NBYTES[7:0] フィールドにゼロ以外の値を書き込むことによって TCR がクリアされたときにリリースされます。
- SCL 立ち下がリエッジの検出後、FMPI2C は、 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{\text{CCCLK}}$ の間、SCL ローをストレッチします。

クロックストレッチなしのスレーブ（NOSTRETCH = 1）

FMPI2C_CR1 レジスタの NOSTRETCH = 1 のとき、FMPI2C スレーブは SCL 信号をストレッチしません。

- ADDR フラグがセットされている間、SCL クロックはストレッチされません。
- 送信時、転送に対応する最初の SCL パルスが発生する前に、FMPI2C_TXDR レジスタにデータが書き込まれる必要があります。そうでない場合、アンダーランが発生し、FMPI2C_ISR レジスタで OVR フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。OVR フラグは、最初のデータ送信が開始し、STOPF ビットがまだセットされている（クリアされていない）ときにもセットされます。したがって、次の転送で送信される最初のデータを書き込んだ後でのみ、前に転送の STOPF フラグをクリアすることによって、送信される最初のデータについても、OVR ステータスが提供されることを確実にできます。
- 受信時、次のデータバイトの 9 番目の SCL パルス（ACK パルス）が発生する前に、FMPI2C_RXDR レジスタからデータが読み出される必要があります。そうでない場合、オーバーランが発生し、FMPI2C_ISR レジスタで OVR フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

スレーブバイト制御モード

スレーブ受信モードでバイト ACK 制御を可能にするためには、FMPI2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。これは、SMBus 標準に準拠する必要があります。

スレーブ受信モードでバイト ACK 制御を可能にするためには、再ロードモードを選択する必要があります（RELOAD=1）。各バイトの制御を得るには、ADDR 割り込みサブルーチンで NBYTES を 0x1 に初期化し、各受信バイト後に 0x1 に再ロードする必要があります。バイトが受信されると、TCR ビットがセットされ、8 番目と 9 番目の SCL パルスの間で、SCL 信号ローをストレッチします。FMPI2C_RXDR レジスタからデータを読み出すことができ、その後、FMPI2C_CR2 レジスタの ACK ビットを設定することによって、確認応答するかどうかを決定できます。SCL ストレッチは、NBYTES をゼロ以外の値にプログラムすることによってリリースされ、確認応答または非確認応答が送信され、次のバイトを受信できます。

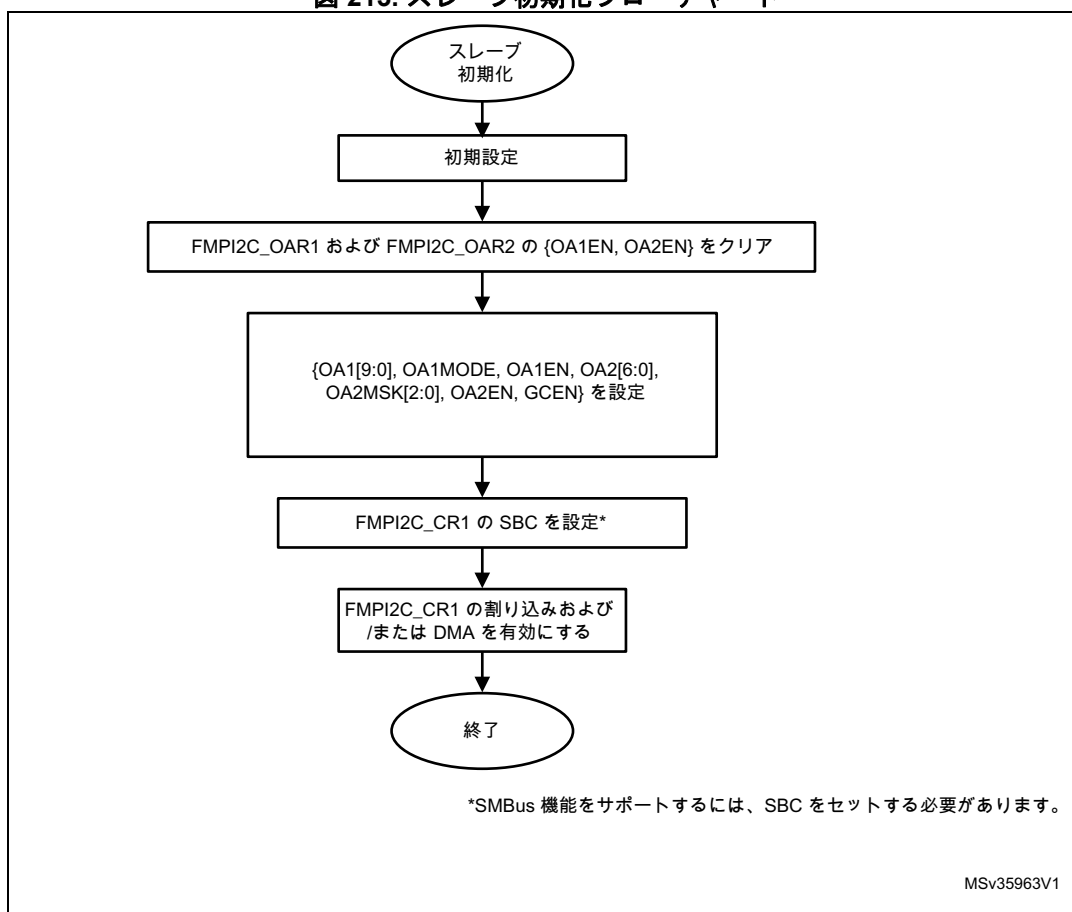
NBYTES に 0x1 より大きい値をロードでき、この場合、受信フローは NBYTES データ受信、継続します。

注： SBC ビットは、FMPI2C が無効なとき、またはスレーブがアドレス指定されていないとき、または ADDR=1 のときに設定する必要があります。

RELOAD ビットの値は、ADDR=1 のとき、または TCR=1 のときに変更できます。

注意： スレーブバイト制御モードは、NOSTRETCH モードと互換性がありません。NOSTRETCH=1 のときに SBC をセットすることはできません。

図 213. スレーブ初期化フローチャート



スレーブトランスミッタ

FMPI2C_TXDR レジスタが空になると、送信割り込みステータス (TXIS) が生成されます。FMPI2C_CR1 レジスタの TXIE ビットがセットされている場合は、割り込みが生成されます。

TXIS ビットは、FMPI2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

NACK が受信されると、FMPI2C_ISR レジスタの NACKF ビットがセットされ、FMPI2C_CR1 レジスタの NACKIE ビットがセットされていた場合は割り込みが生成されます。マスタが STOP または RESTART コンディションを実行できるように、スレーブは SCL および SDA ラインを自動的にリリースします。TXIS ビットは、NACK 受信時にはセットされません。

STOP が受信され、FMPI2C_CR1 レジスタの STOPIE ビットがセットされると、FMPI2C_ISR レジスタの STOPF フラグがセットされ、割り込みが生成されます。ほとんどのアプリケーションでは、SBC は通常、0 にプログラムされます。この場合、スレーブアドレスが受信されたときに (ADDR=1)、TXE = 0 であった場合、FMPI2C_TXDR レジスタの内容を最初のデータバイトとして送信するか、新しいデータバイトをプログラムするために TXE ビットをセットすることによって FMPI2C_TXDR レジスタをフラッシュするかを選択できます。

スレーブバイト制御モード (SBC=1) では、送信バイト数をアドレス一致割り込みサブルーチンの NBYTES でプログラムする必要があります (ADDR=1)。この場合、転送中の TXIS イベントの数は、NBYTES でプログラムされた値に対応します。

注意： ***NOSTRETCH=1 のとき、SCL クロックは ADDR フラグがセットされている間はストレッチされない***
ので、最初のデータバイトをプログラムするために ADDR サブルーチンで FMPI2C_TXDR レジスタの内容をフラッシュすることはできません。最初に送信されるデータバイトは、FMPI2C_TXDR レジスタで前もってプログラムされている必要があります。

- このデータは、前の送信メッセージの最後の TXIS イベントで書き込まれたデータでもかまいません。
- このデータバイトが送信データバイトでない場合、新しいデータバイトをプログラムするために TXE ビットをセットすることによって FMPI2C_TXDR レジスタをフラッシュできます。アドレスの確認応答に続いて、最初のデータ送信が開始する前にこれらが実行されることを保証するためには、STOPF ビットのクリアは、これらのアクションの後でのみ行う必要があります。

最初のデータ送信が開始したときに STOPF がまだセットされていた場合、アンダーランエラーが生成されます (OVR フラグがセットされます)。

TXIS イベントが必要な場合 (送信割り込みまたは送信 DMA リクエスト)、TXIS イベントを生成するためには、TXE ビットに加えて TXIS ビットもセットする必要があります。

図 214. FMPI2C スレーブトランスマッタの転送シーケンスフローチャート
(NOSTRETCH=0)

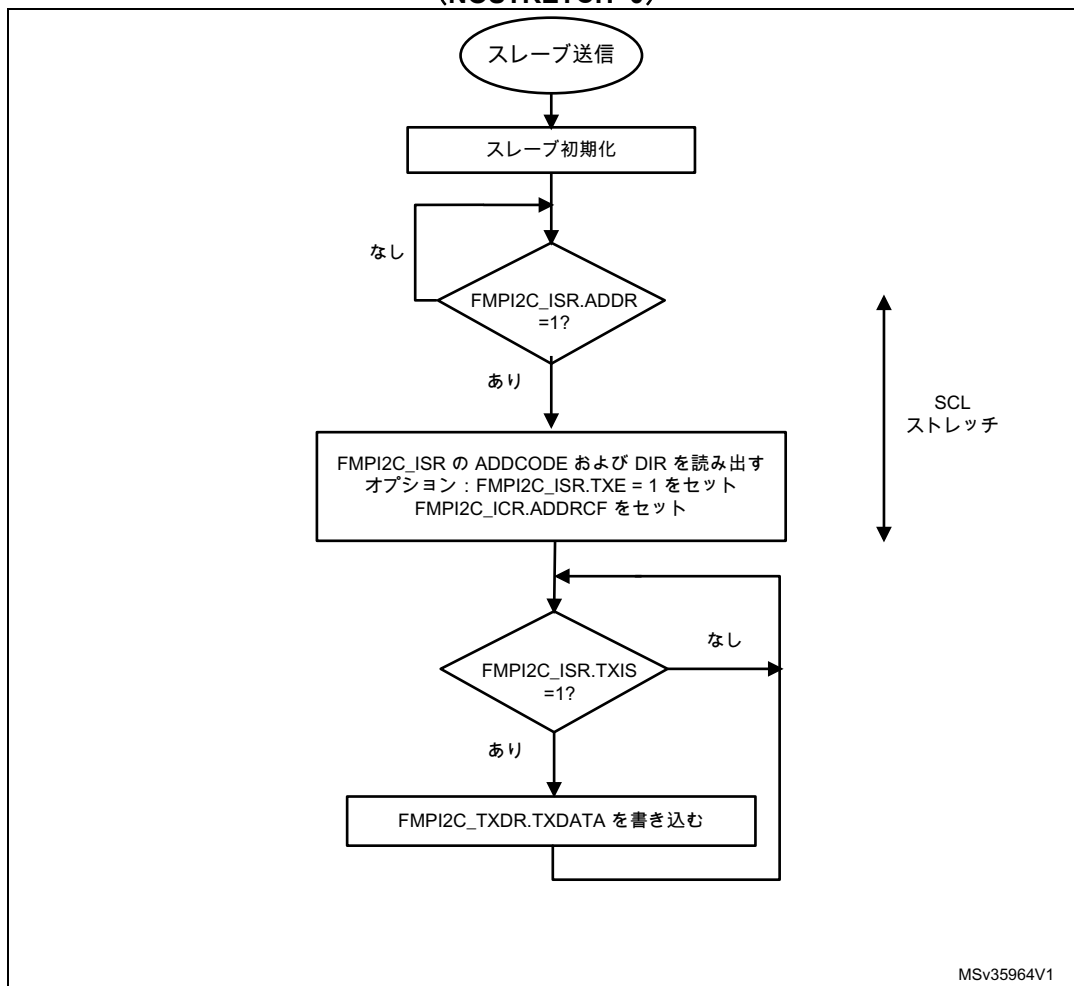


図 215. FMPI2C スレーブトランスマッタの転送シーケンスフローチャート
(NOSTRETCH=1)

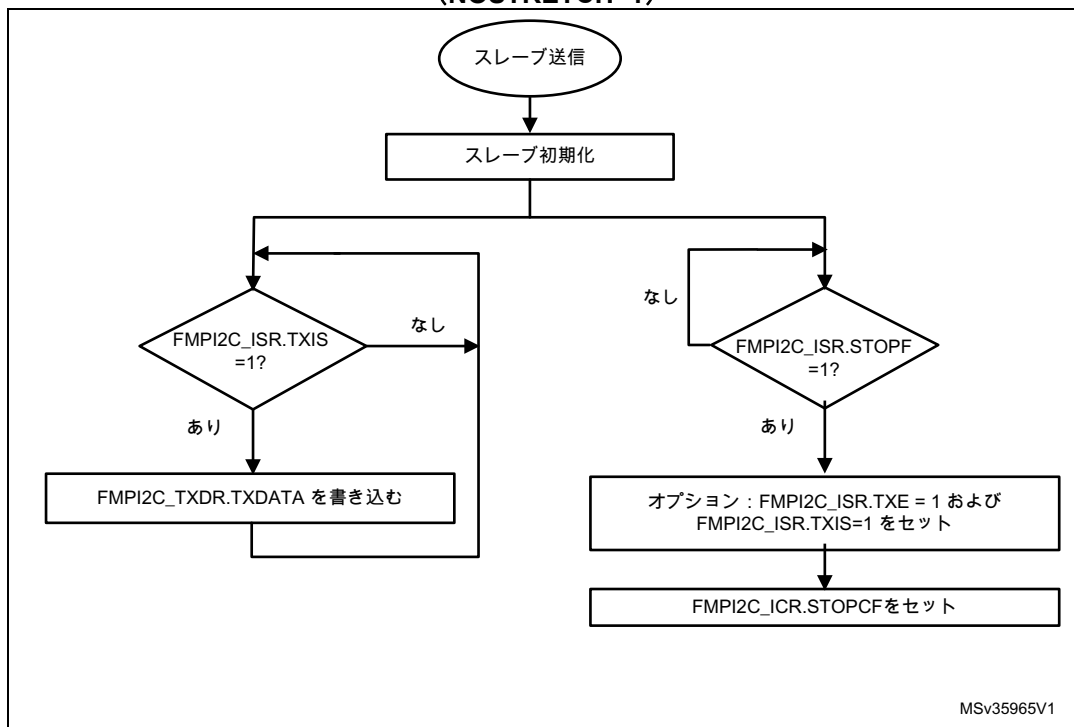
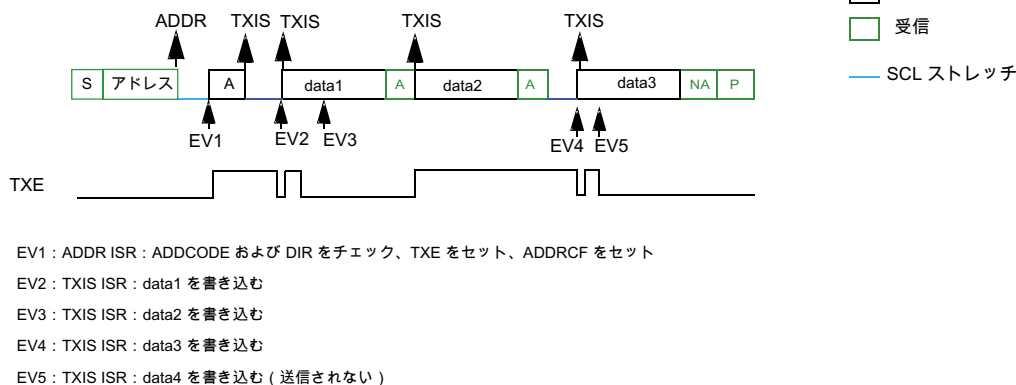
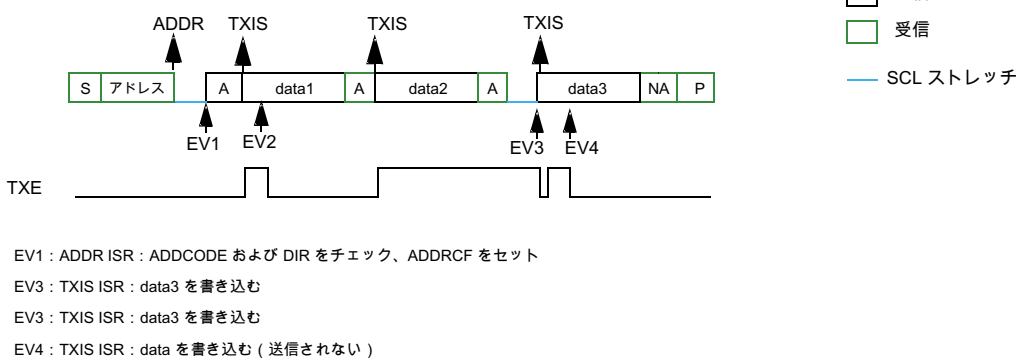


図 216. FMPI2C スレーブトランスミッタの転送バス図

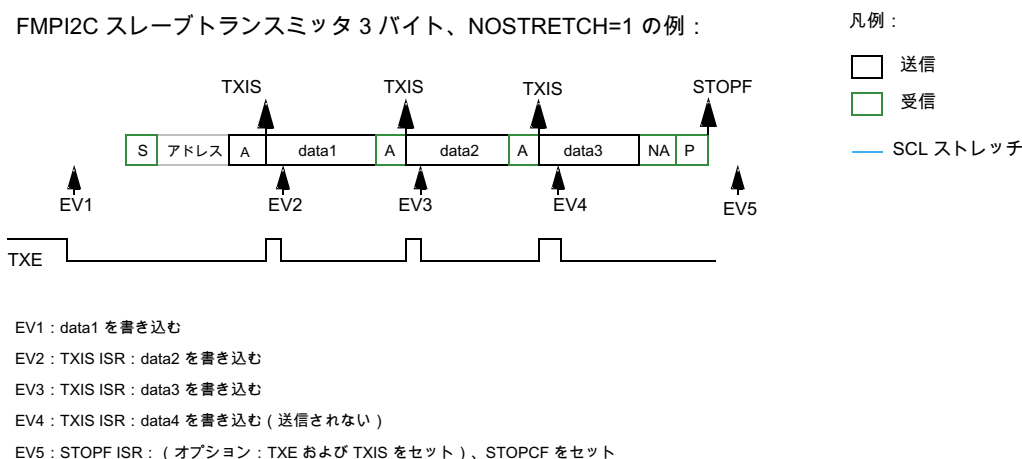
フラッシュされた最初のデータ付き FMPI2C スレーブトランスミッタ 3 バイト、NOSTRETCH=0 の例：



最初のデータをフラッシュしない FMPI2C スレーブトランスミッタ 3 バイト、NOSTRETCH=0 の例：



FMPI2C スレーブトランスミッタ 3 バイト、NOSTRETCH=1 の例：



MS35975V1

スレーブレシーバ

FMPI2C_RXDR がフルのときには、FMPI2C_ISR の RXNE がセットされ、FMPI2C_CR1 の RXIE がセットされている場合は割り込みが生成されます。RXNE は、FMPI2C_RXDR が読み出されたときにクリアされます。

STOP が受信され、FMPI2C_CR1 レジスタの STOPIE ビットがセットされると、FMPI2C_ISR の STOPF がセットされ、割り込みが生成されます。

図 217. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)

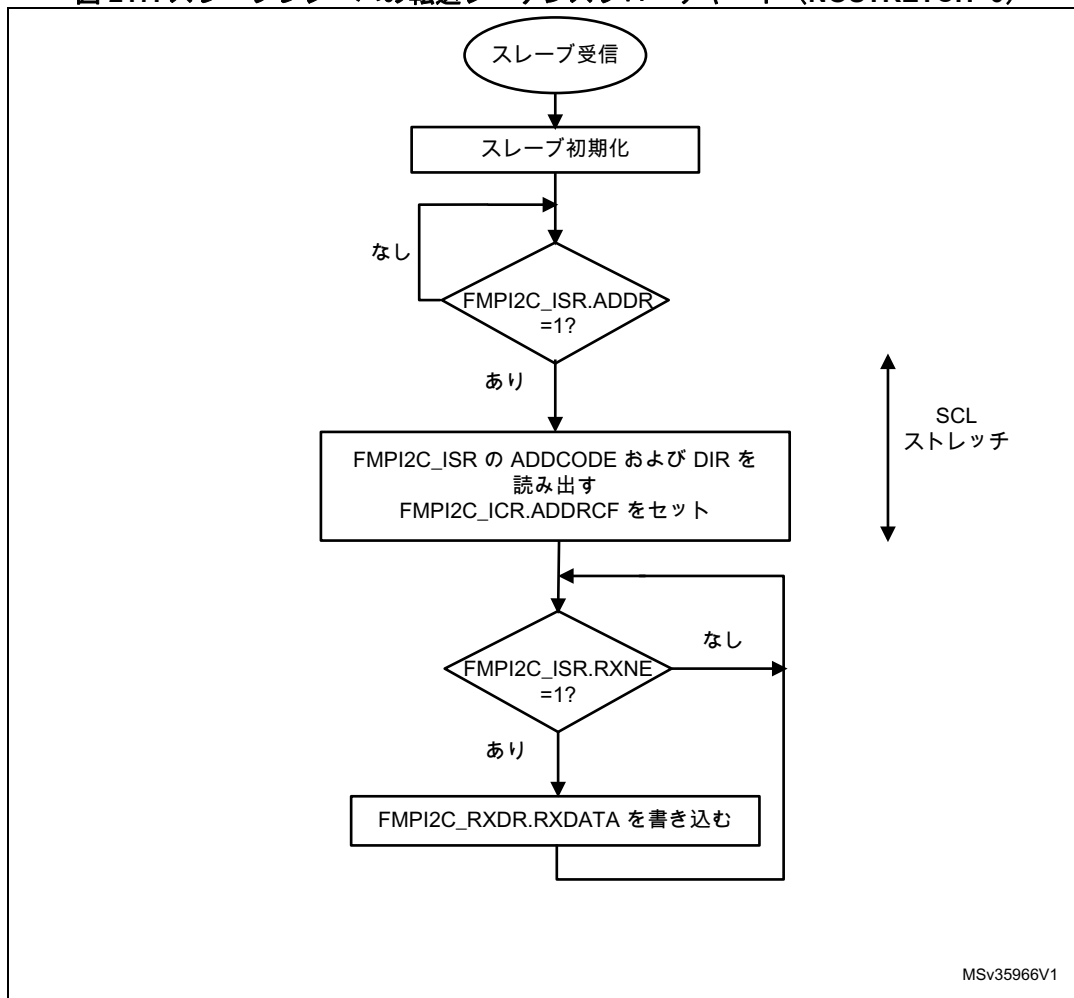


図 218. スレーブレシーバの転送シーケンスフローチャート（NOSTRETCH=1）

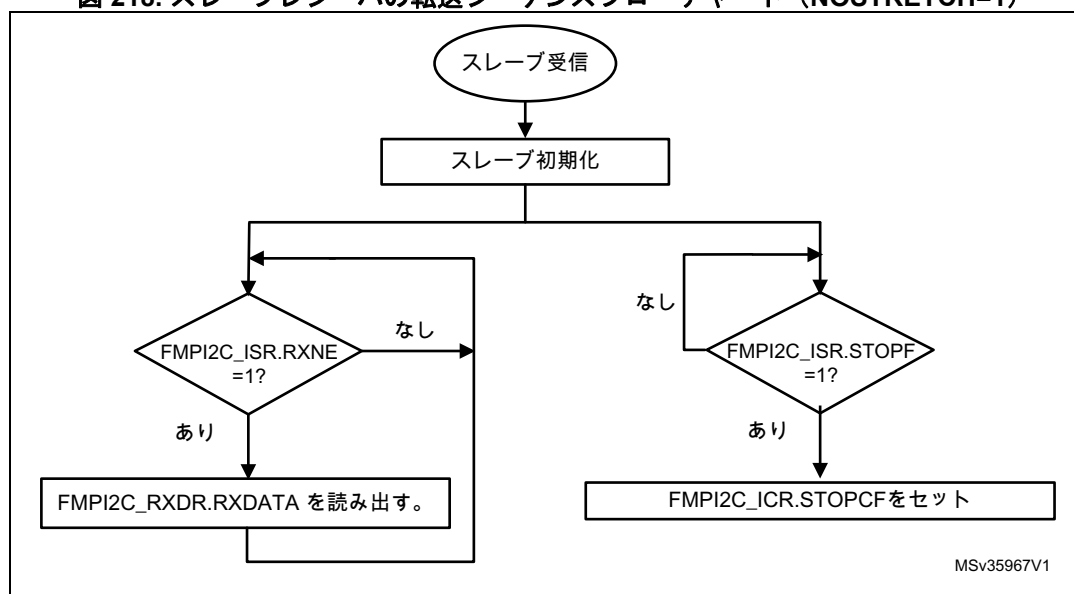
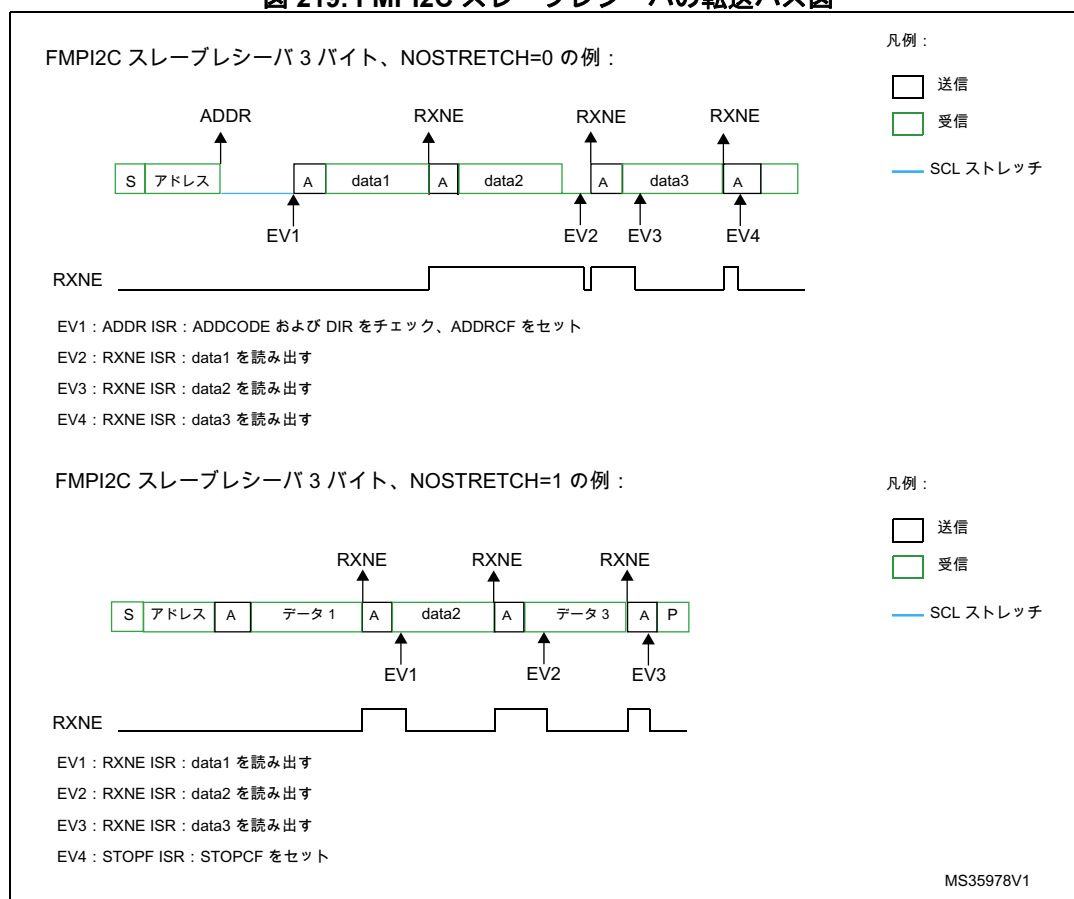


図 219. FMPI2C スレーブレシーバの転送バス図



23.4.8 FMPI2C マスタモード

FMPI2C マスタ初期化

ペリフェラルを有効にする前に、FMPI2C_TIMINGR レジスタの SCLH および SCLL ビットをセットすることによって、FMPI2C マスタクロックを設定する必要があります。

STM32CubeMX ツールは、I2C 設定ウィンドウの I2C_TIMINGR の内容を計算して示します。

マルチマスタ環境とスレーブクロックストレッチをサポートするために、クロック同期メカニズムが実装されています。

クロック同期を可能にするために：

- クロックのローレベルは SCLL カウンタを使用してカウントされ、SCL ローレベル内部検出から開始されます。
- クロックのハイレベルは SCLH カウンタを使用してカウントされ、SCL ハイレベル内部検出から開始されます。

FMPI2C は、SCL 立ち下がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC1} の後に SCL ローレベルを検出します。SCLL カウンタが FMPI2C_TIMINGR レジスタの SCLL[7:0] ビットでプログラムされた値に達すると、FMPI2C は SCL をハイレベルにリリースします。

I2C は、SCL 立ち上がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および FMPI2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC2} の後に SCL ハイレベルを検出します。SCLH カウンタが FMPI2C_TIMINGR レジスタの SCLH[7:0] ビットでプログラムされた値に達すると、FMPI2C は SCL をローレベルにします。

結果として、マスタクロック周期は次のとおりです：

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}+1) + (\text{SCLL}+1)] \times (\text{PRESC}+1) \times t_{\text{I2CCLK}}\}$$

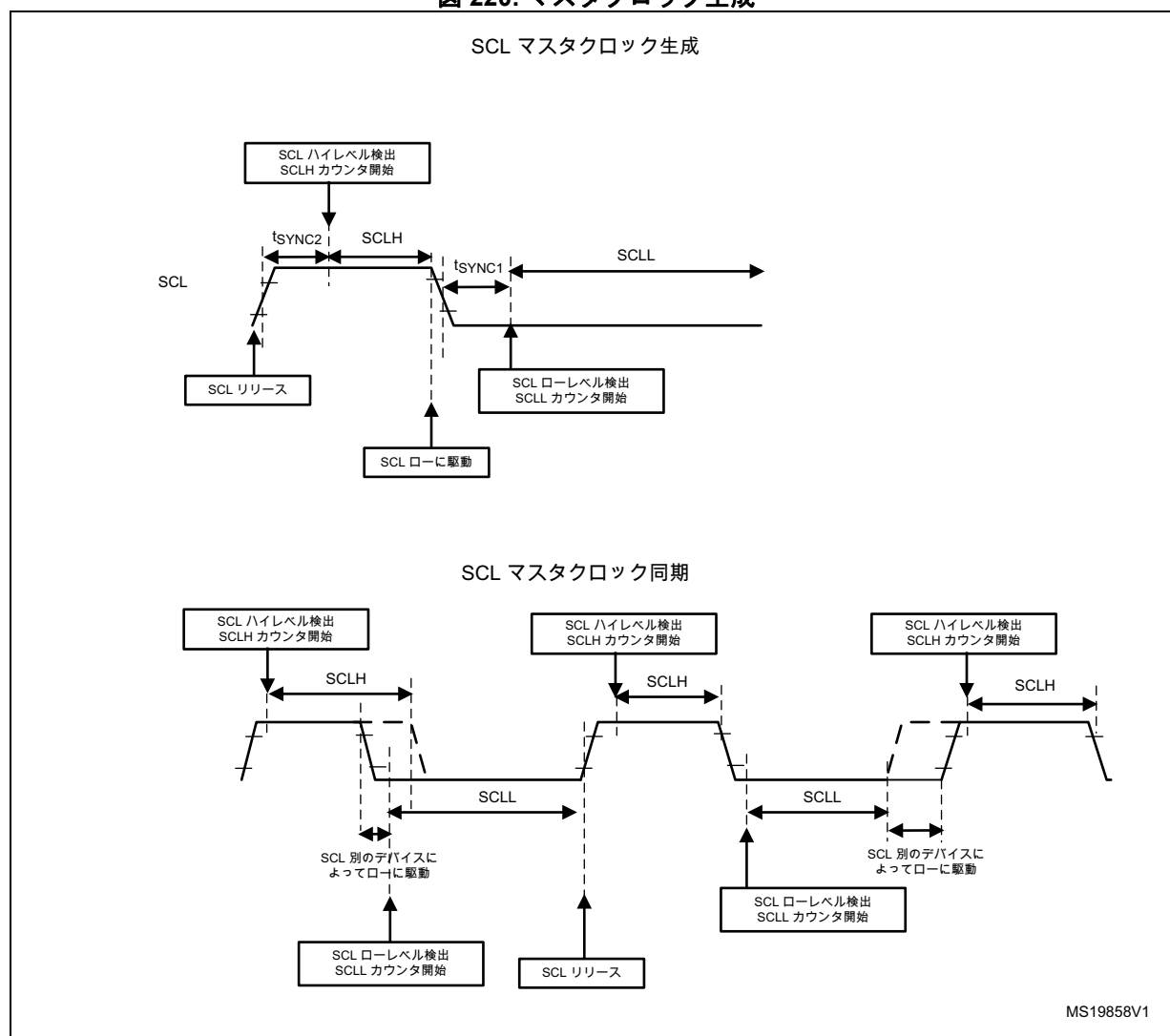
t_{SYNC1} の長さは、次のパラメータに依存します：

- SCL 立ち下がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF $\times t_{\text{I2CCLK}}$
- SCL と FMPI2CCLK クロックの同期による遅延（2 ～ 3 FMPI2CCLK 周期）

t_{SYNC2} の長さは、次のパラメータに依存します：

- SCL 立ち上がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF $\times t_{\text{I2CCLK}}$
- SCL と FMPI2CCLK クロックの同期による遅延（2 ～ 3 FMPI2CCLK 周期）

図 220. マスタクロック生成



注意： I^2C または SMBus 準拠のためには、マスタクロックは次のタイミングを満たす必要があります：

表 113. I²C-SMBUS 仕様のクロックタイミング

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
f _{SCL}	SCL クロック周波数		100		400		1000		100	kHz
t _{HD:STA}	(反復) START コンディションのホールド時間	4.0	-	0.6		0.26	-	4.0	-	μs
t _{SU:STA}	反復 START コンディションのセットアップ時間	4.7	-	0.6		0.26	-	4.7	-	μs
t _{SU:STO}	STOP コンディションのセットアップ時間	4.0	-	0.6		0.26	-	4.0	-	μs
t _{BUF}	STOP コンディションと START コンディションの間のバスフリー時間	4.7	-	1.3		0.5	-	4.7	-	μs
t _{LOW}	SCL クロックのロー周期	4.7	-	1.3		0.5	-	4.7	-	μs
t _{HIGH}	SCL クロックの周期	4.0	-	0.6		0.26	-	4.0	50	μs
t _r	SDA および SCL 信号の立ち上がり時間	-	1000	-	300		120	-	1000	ns
t _f	SDA および SCL 信号の立ち下がり時間	-	300	-	300		120	-	300	ns

注： SCLL は、t_{BUF} および t_{SU:STA} タイミングの生成にも使用されます。

SCLH は、t_{HD:STA} および t_{SU:STO} タイミングの生成にも使用されます。

FMPI2C_TIMINGR 設定と FMPI2CCLKI2CCLK 周波数の例については、[セクション 23.4.9 : FMPI2C_TIMINGR レジスタの設定例](#)を参照してください。

マスタ通信の初期化（アドレスフェーズ）

通信を初期化するためには、FMPI2C_CR2 レジスタでアドレス指定されたスレーブについて次のパラメータをプログラムする必要があります：

- アドレッシングモード（7 ビットまたは 10 ビット）：ADD10
- 送信されるスレーブアドレス：SADD[9:0]
- 転送方向：RD_WRN
- 10 ビットアドレスが読み出される場合：HEAD10R ビット。HEAD10R を設定して、完全なアドレスシーケンスが送信されなければならないか、ヘッダのみ（方向の変更の場合）かを示す必要があります。
- 転送されるバイト数：NBYTES[7:0]。バイト数が 255 バイト以上の場合は、NBYTES[7:0] に 0xFF を書き込む必要があります。

次に、FMPI2C_CR2 レジスタの START ビットをセットする必要があります。START ビットがセットされているとき、上記のすべてのビットを変更することはできません。

その場合、マスタは、バスがフリーである（BUSY = 0）ことを検出すると、t_{BUF} の遅延後に、自動的に START コンディションとスレーブアドレスを送信します。



アービトレーション喪失の場合、マスタはスレーブモードに自動的に切り替えて、スレーブとしてアドレス指定された場合は専用アドレスを確認応答できます。

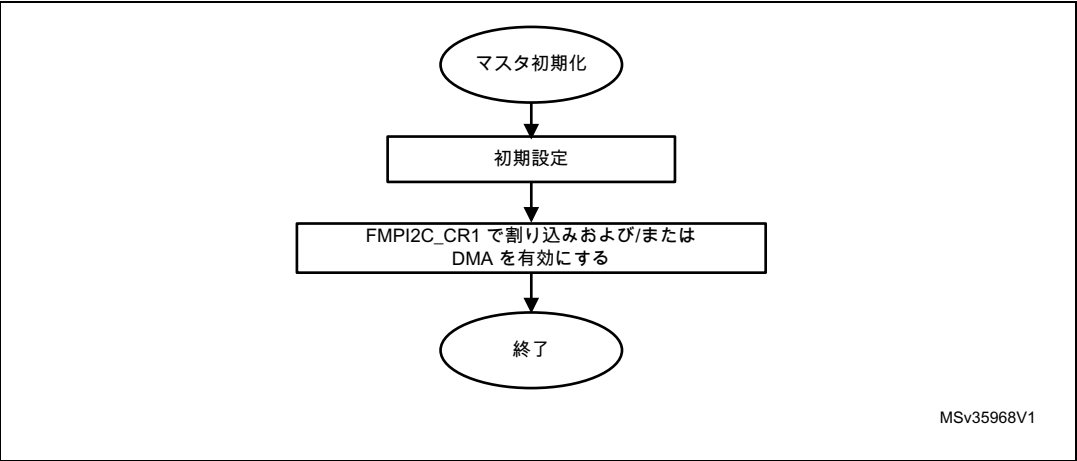
注： START ビットは、スレーブアドレスがバスに送信されたとき、受信した確認応答値にかかわらず、ハードウェアによってリセットされます。START ビットは、アービトレーション喪失が発生した場合にも、ハードウェアによってリセットされます。

10 ビットアドレッシングモードでは、スレーブアドレスの最初の 7 ビットがスレーブによって NACK されている場合、マスタは ACK が受信されるまで自動的にスレーブアドレスの送信を再開します。この場合、NACK をスレーブから受信するかどうか、ADDRCF をセットしてスレーブアドレスの送信を停止する必要があります。

START ビットがセットされているときに、FMPI2C がスレーブとしてアドレス指定された場合 (ADDR=1)、FMPI2C はスレーブモードに切り替わり、ADDRCF ビットがセットされたときに START ビットがクリアされます。

注： 反復スタートコンディションにも同じ手順が適用されます。この場合、BUSY=1 です。

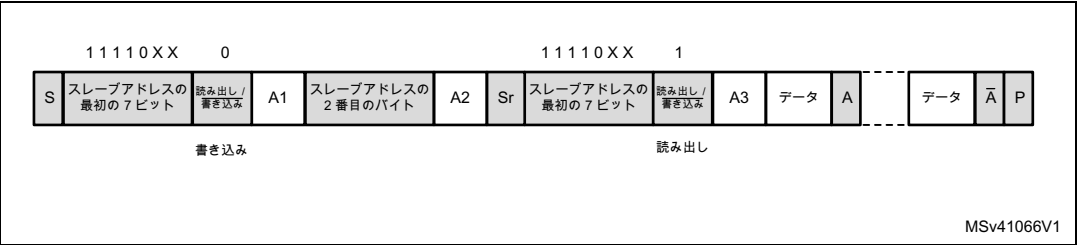
図 221. マスタ初期化フローチャート



10 ビットアドレススレーブをアドレス指定するマスタレシーバの初期化

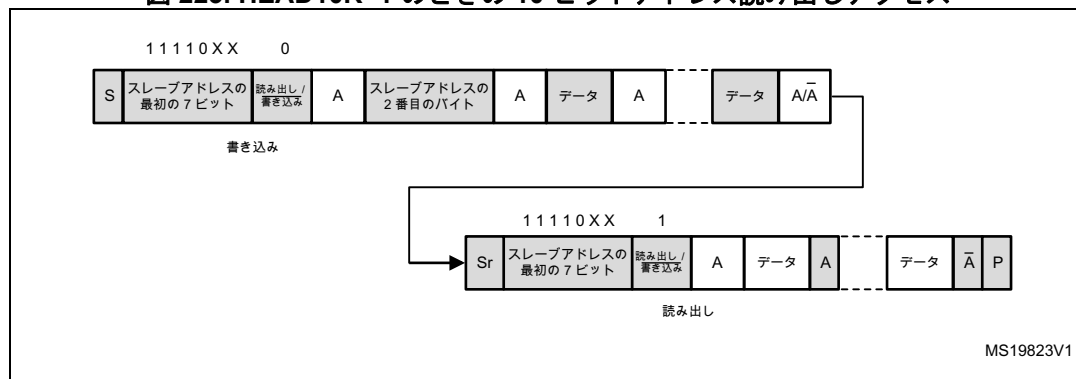
- スレーブアドレスが 10 ビット形式の場合、FMPI2C_CR2 レジスタの HEAD10R ビットをクリアすることによって、完全な読み出しシーケンスを送信することができます。この場合、マスタは、START ビットがセットされた後、次のような完全なシーケンスを自動的に送信します：(Re) START + スレーブアドレス 10 ビットヘッダ書き込み + スレーブアドレスの 2 番目のバイト + RStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 222. HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス



- マスタが 10 ビットアドレススレーブをアドレス指定して、このスレーブアドレスにデータを送信した後、同じスレーブからデータを読み出す場合には、まず、マスタ送信フローが行われる必要があります。その場合、反復開始が、HEAD10R=1 で設定された 10 ビットスレーブアドレスでセットされます。この場合、マスタは次のシーケンスを送信します：ReStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 223. HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス



マスタトランスマッタ

書き込み転送の場合、ACK が受信されたときの9 番目の SCL パルス後、各バイトの送信後にTXIS フラグがセットされます。

FMPI2C_CR1 レジスタの TXIE ビットがセットされている場合、TXIS イベント時に割り込みが生成されます。このフラグは、FMPI2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

転送中の TXIS イベントの数は、NBYTES[7:0] でプログラムされた値に対応します。送信されるデータバイト数の合計が 255 より大きい場合、FMPI2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

TXIS フラグは、NACK 受信時にはセットされません。

- RELOAD=0 でNBYTES データが転送されたとき：
 - － 自動終了モード（AUTOEND=1）では、STOP が自動的に送信されます。
 - － ソフトウェア終了モード（AUTOEND=0）では、TC フラグがセットされ、ソフトウェアアクションを実行するために SCL ラインがローでストレッチされます：

正しいスレーブアドレス設定と転送バイト数で FMPI2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションがバスに送信されます。

FMPI2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。
- NACK が受信された場合：TXIS フラグはセットされず、NACK 受信後、自動的に STOP コンディションが送信されます。FMPI2C_ISR レジスタの NACKF フラグがセットされ、NACKIE ビットがセットされていた場合は割り込みが生成されます。

図 224. N≤255 バイトの場合の FMPI2C マスタトランスミッタの転送シーケンスフローチャート

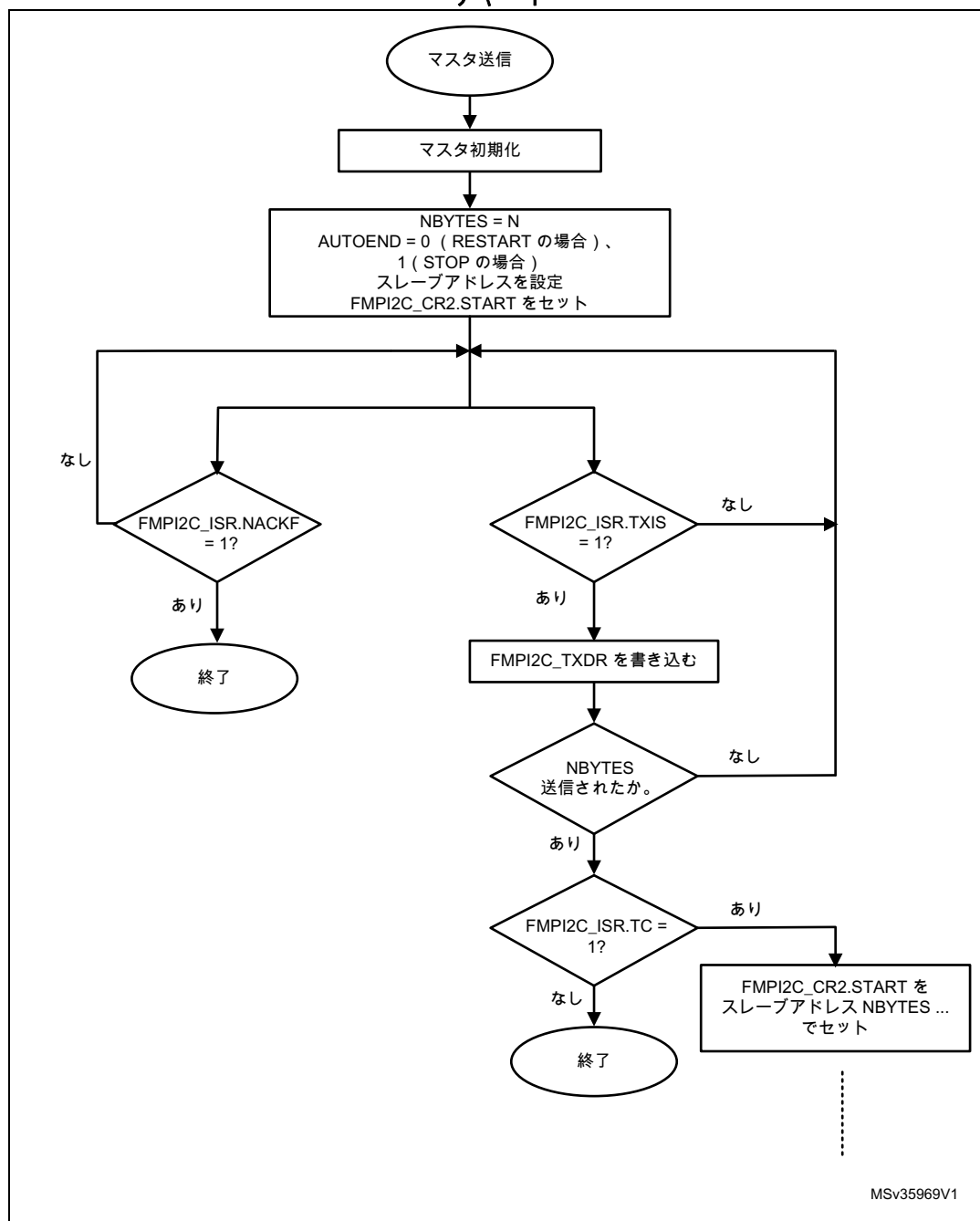


図 225. N>255 バイトの場合の FMPI2C マスタトランスミッタの転送シーケンスフローチャート

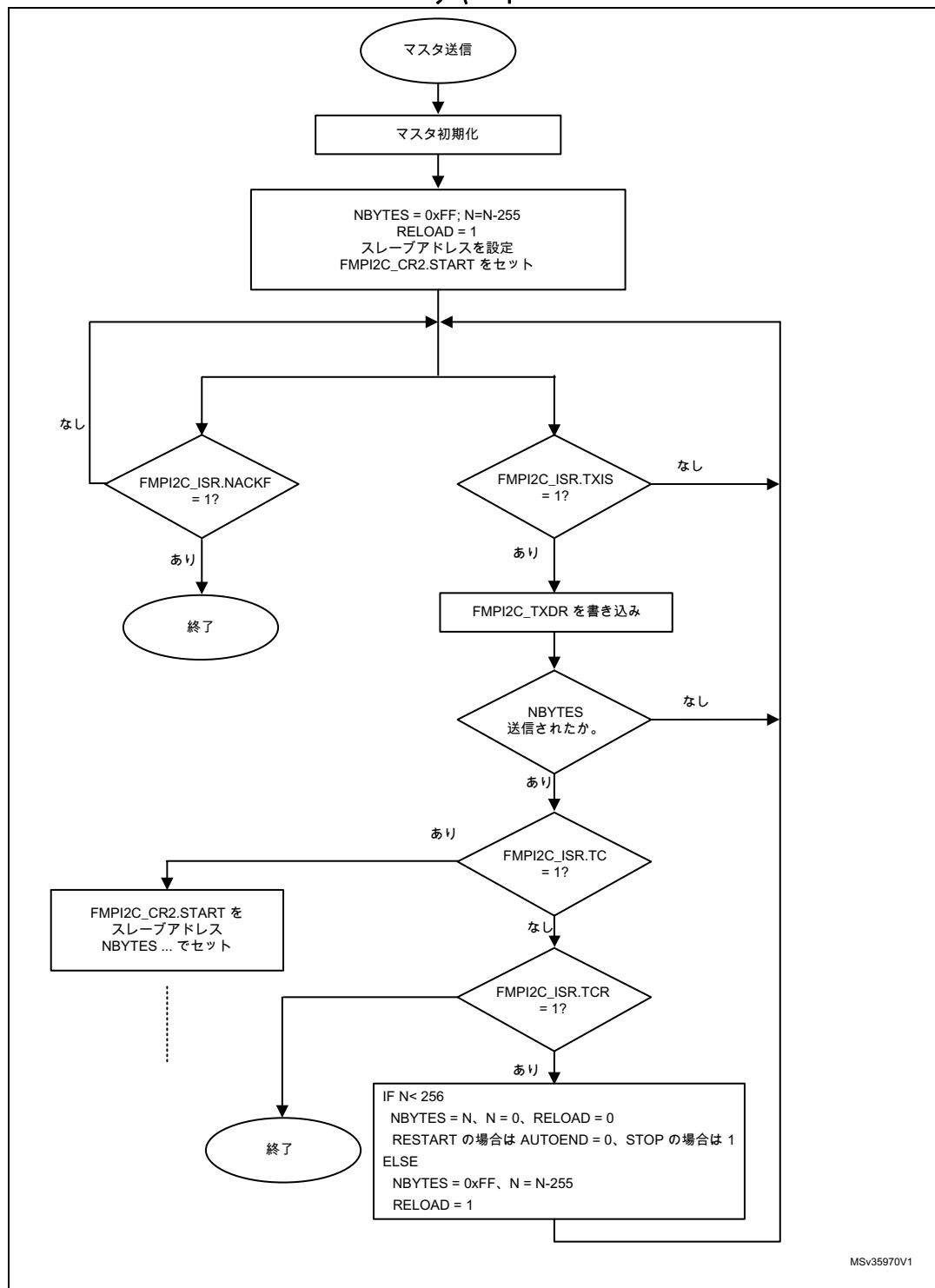
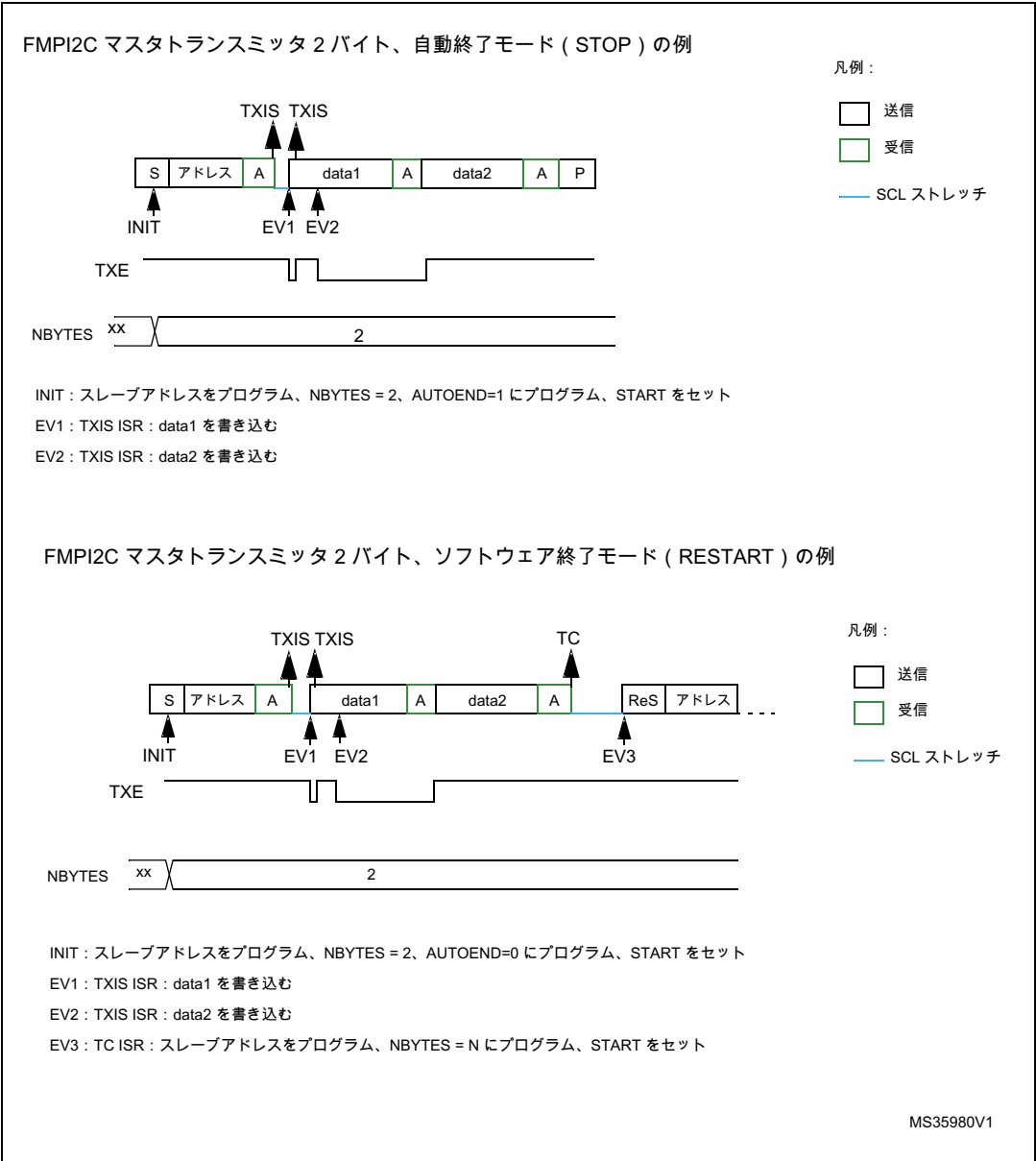


図 226. FMPI2C マスタトランスミッタの転送バス図



マスタレシーバ

読み出し転送の場合、各バイトの受信後や8 番目の SCL パルス後に RXNE フラグがセットされます。FMPI2C_CR1 レジスタの RXIE ビットがセットされている場合、RXNE イベント時に割り込みが生成されます。このフラグは、FMPI2C_RXDR が読み出されたときにクリアされます。

受信されるデータバイト数の合計が 255 より大きい場合、FMPI2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES[7:0] データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

- RELOAD=0 でNBYTES[7:0] データが転送されたとき：
 - － 自動終了モード（AUTOEND=1）では、最後の受信バイト後に NACK および STOP が自動的に送信されます。
 - － ソフトウェア終了モード（AUTOEND=0）では、最後の受信バイト後に NACK が自動的に送信され、TC フラグがセットされ、ソフトウェアアクションを実行できるように、SCL ラインがローでストレッチされます。

正しいスレーブアドレス設定と転送バイト数で FMPI2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションとスレーブアドレスがバスに送信されます。

FMPI2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。

図 227. N≤255 バイトの場合の FMPI2C マスタレシーバの転送シーケンスフローチャート

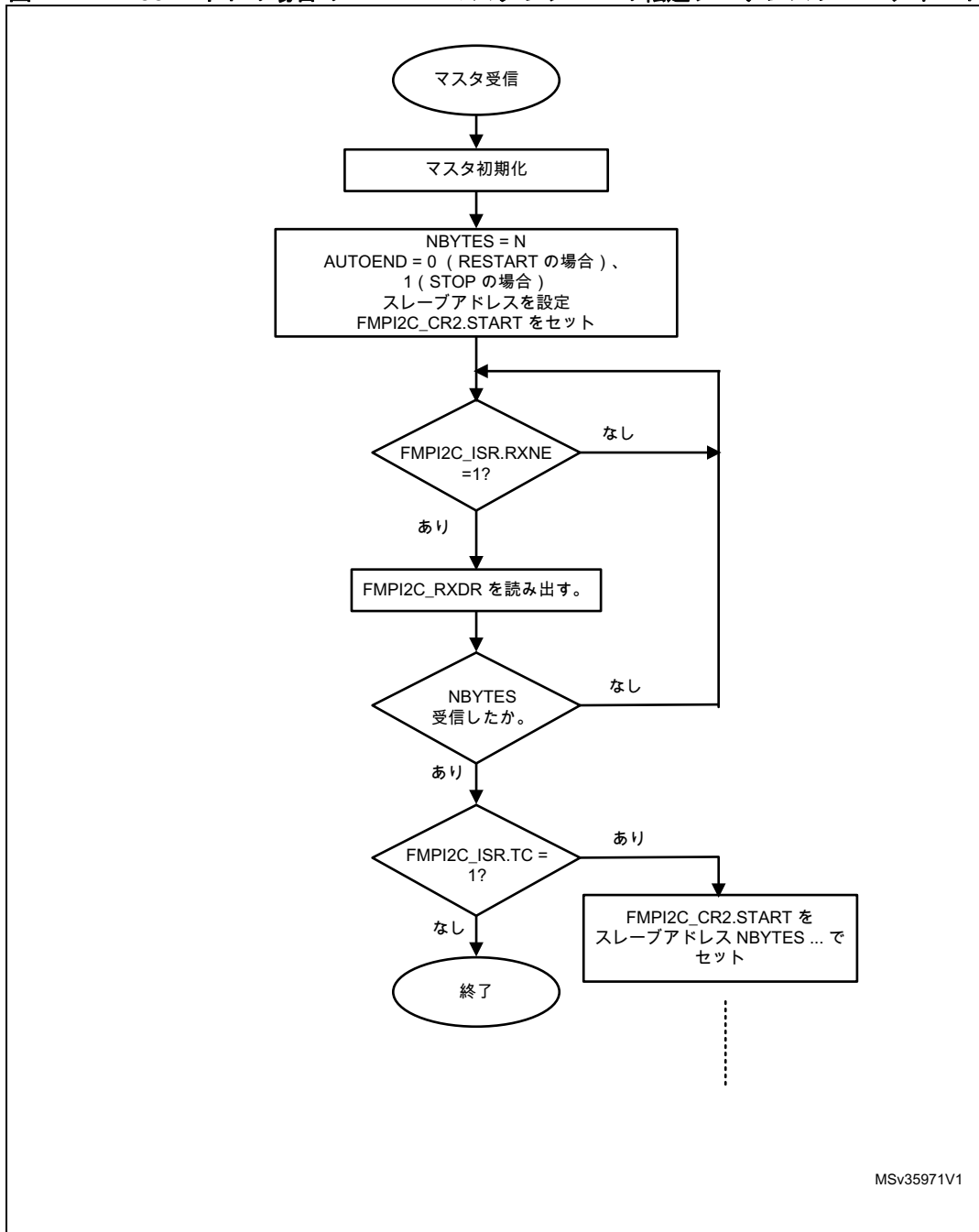


図 228. N>255 バイトの場合の FMPI2C マスタレーバの転送シーケンスフローチャート

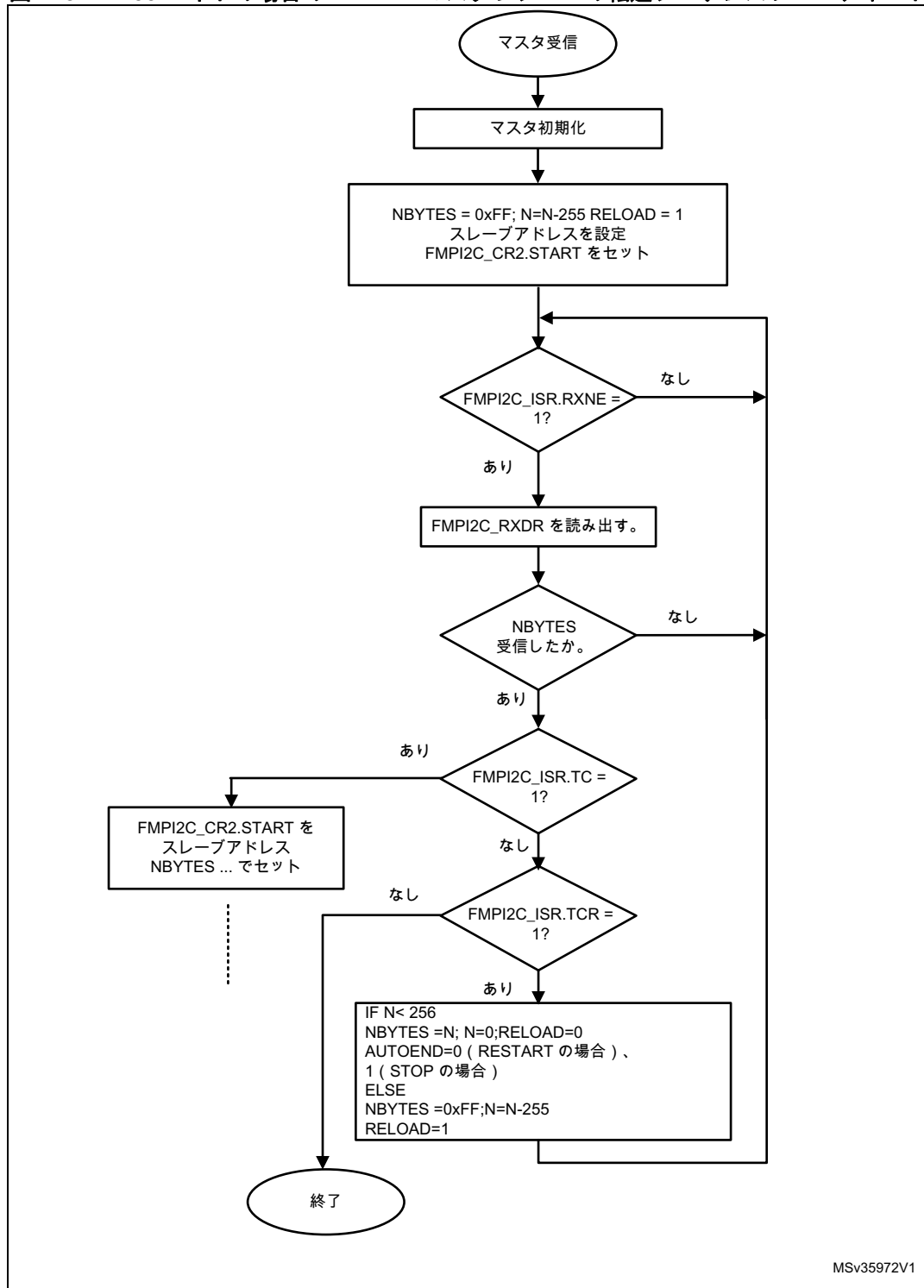
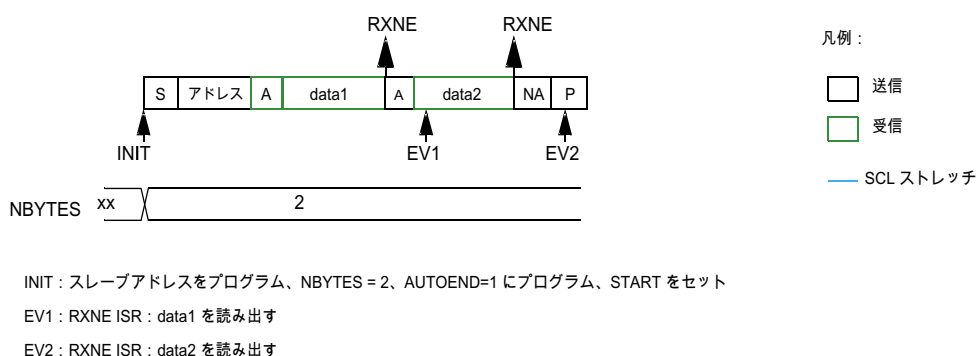
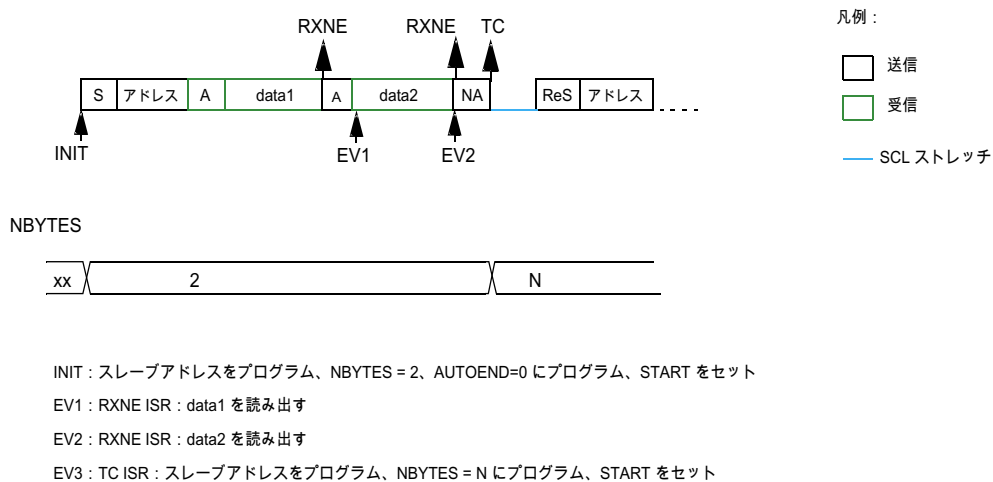


図 229. FMPI2C マスタレシーバの転送バス図

FMPI2C マスタレシーバ 2 バイト、自動終了モード (STOP) の例



FMPI2C マスタレシーバ 2 バイト、ソフトウェア終了モード (RESTART) の例



MS35979V1

23.4.9 FMPI2C_TIMINGR レジスタの設定例

下の表に、I²C 仕様に準拠したタイミングを得るための FMPI2C_TIMINGR をプログラムする方法の例を示します。より正確な設定値を得るには、アプリケーションノート I²C タイミング設定ツール (AN4235) および該当するソフトウェア STSW-STM32126 を参照してください。

23.4.10 SMBus 固有の機能

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 23.3 : FMPI2C の実装](#) を参照してください。

概要

システム管理バス（SMBus）は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I²C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

このペリフェラルは、SMBUS 仕様 rev 2.0 (<http://smbus.org>) と互換性があります。

システム管理バス仕様では、3 種類のデバイスを規定しています。

- スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。
- マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。
- ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

このペリフェラルは、マスタまたはスレーブデバイスとして、また、ホストとしても設定できます。

SMBUS は、I²C 仕様 rev 2.1 に基づきます。

バスプロトコル

特定のデバイスについて、11 の可能なコマンドプロトコルがあります。デバイスは、11 のプロトコルの一部または全部を使用して通信できます。プロトコルは、Quick Command、Send Byte、Receive Byte、Write Byte、Write Word、Read Byte、Read Word、Process Call、Block Read、Block Write、および Block Write-Block Read Process Call です。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

これらのプロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

アドレス解決プロトコル（ARP）

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス割り当てを目的とする各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子（UDID）を実装する必要があります。128 ビットの番号がソフトウェアによって実装されます。

このペリフェラルは、アドレス解決プロトコル（ARP）をサポートします。SMBus デバイスのデフォルトアドレス (0b1100 001) は、FMPI2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。ARP コマンドは、ユーザのソフトウェアによって実装してください。

ARP サポートのために、スレーブモードでアービトレーションも行われます。

SMBus アドレス解決プロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

受信コマンドおよびデータ確認応答制御

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、FMPI2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、[スレーブバイト制御モード \(654 ページ\)](#) を参照してください。

Host Notify プロトコル

このペリフェラルは、FMPI2C_CR1 レジスタの SMBHEN ビットをセットすることによって、Host Notify（ホスト通知）プロトコルをサポートします。この場合、ホストは SMBus ホストアドレス（0b0001 000）を確認応答します。

このプロトコルが使用されると、デバイスはマスタとして動作し、ホストはスレーブとして動作します。

SMBus アラート

SMBus ALERT オプション信号がサポートされます。スレーブ専用デバイスは、通信したいホストの SMBALERT# ピンを通じてホストに信号を送信します。ホストは、割り込みを処理し、アラート応答アドレス（0b0001 100）を通じて全 SMBALERT# デバイスに同時にアクセスします。SMBALERT# をローに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。

スレーブデバイスとして設定されたとき（SMBHEN=0）、FMPI2C_CR1 レジスタの ALERTEN ビットをセットすることによって、SMBA ピンはローに引き下げられます。同時に、アラート応答アドレスが有効になります。

ホストとして設定されたとき（SMBHEN=1）、SMBA ピンで立ち下がりエッジが検出され、ALERTEN=1 のとき、FMPI2C_ISR レジスタの ALERT フラグがセットされます。FMPI2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。ALERTEN=0 のときには、外部 SMBA ピンがローの場合でも、ALERT ラインはハイとみなされます。

SMBus ALERT ピンが不要な場合には、ALERTEN=0 の場合、SMBA ピンを標準 GPIO として使用できます。

パケットエラーチェック

信頼性と通信の堅牢性を向上させるために、SMBus 仕様にパケットエラーチェックメカニズムが導入されました。パケットエラーチェックは、各メッセージ転送の終わりにパケットエラーコード（PEC）を付加することによって実装されます。PEC は、すべてのメッセージバイト（アドレスと読み出し／書き込みビットを含む）に対して $C(x) = x_8 + x^2 + x + 1$ CRC-8 多項式を使用して計算されます。

ペリフェラルはハードウェア PEC 計算機が組み込まれ、受信バイトがハードウェアによって計算された PEC に一致しないときには自動的に非確認応答を送信できます。

タイムアウト

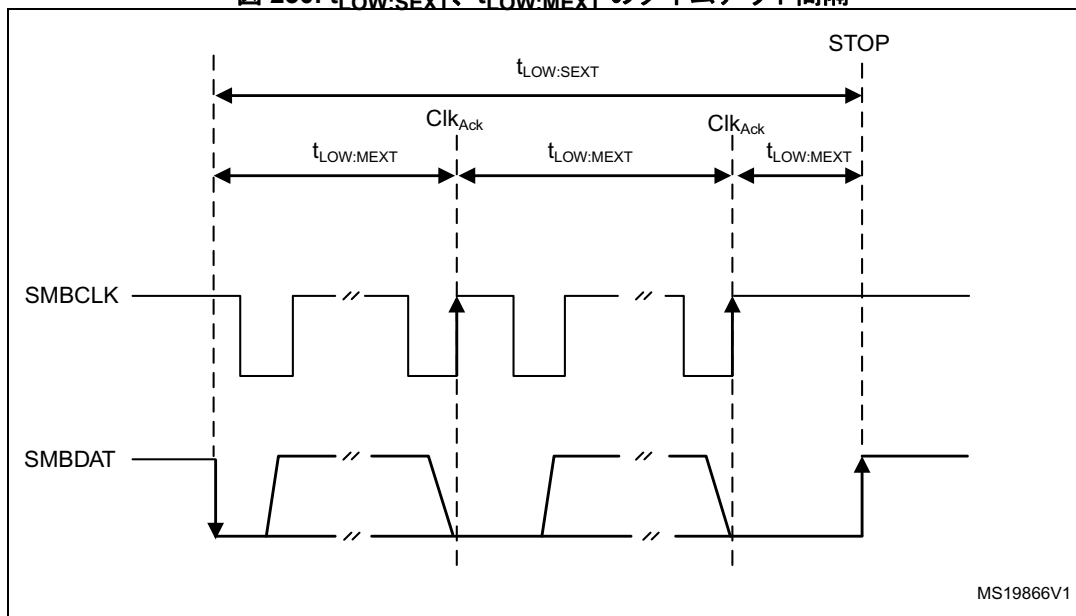
このペリフェラルは、SMBus 仕様バージョン 2.0 で定義された 3 つのタイムアウトに準拠するために、ハードウェアタイマが組み込まれています。

表 114. SMBus タイムアウト仕様

記号	パラメータ	リミット		単位
		最小値	最大値	
t_{TIMEOUT}	クロックロータイムアウト検出	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累積クロックロー延長時間（スレーブデバイス）	-	25	ms
$t_{\text{LOW:MEXT}}^{(2)}$	累積クロックロー延長時間（マスタデバイス）	-	10	ms

- $t_{\text{LOW:SEXT}}$ は、特定のスレーブデバイスが初めての START から STOP までの 1 つのメッセージのクロックサイクルを延長できる累積時間です。別のスレーブデバイスまたはマスタもクロックを延長して、合計のクロックロー延長時間が $t_{\text{LOW:SEXT}}$ より大きくなる場合があります。したがって、このパラメータは、スレーブデバイスをフルスピードのマスタの単独のターゲットとして測定されます。
- $t_{\text{LOW:MEXT}}$ は、マスタがメッセージの各バイト内のクロックサイクルを START-to-ACK、ACK-to-ACK、または ACK-to-STOP から定義に従って延長できる累積時間です。スレーブデバイスまたは別のマスタもクロックを延長して、合計のクロックロー時間が $t_{\text{LOW:MEXT}}$ より大きくなる場合があります。したがって、このパラメータは、フルスピードスレーブデバイスをマスタの単独のターゲットとして測定されます。

図 230. $t_{\text{LOW:SEXT}}$ 、 $t_{\text{LOW:MEXT}}$ のタイムアウト間隔



MS19866V1

バスアイドル検出

マスタは、クロックおよびデータ信号が $t_{\text{HIGH,MAX}}$ より大きい t_{IDLE} の間ハイであった場合、バスはフリーであるとみなすことができます（表 113：I2C-SMBUS 仕様のクロックタイミングを参照してください）

このタイミングパラメータは、マスタがバスに動的に追加し、SMBCLK または SMBDAT ラインで状態遷移を検出しなかった可能性のあるコンディションをカバーします。この場合、マスタは十分に長い時間待って、転送が進行中でないことを確認する必要があります。このペリフェラルは、ハードウェアバスアイドル検出をサポートします。

23.4.11 SMBus 初期化

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 23.3：FMPI2C の実装を参照してください。

SMBus 通信を行うためには、FMPI2C 初期化に加えて、他にも特定の初期化を行う必要があります：

受信コマンドおよびデータ確認応答制御（スレーブモード）

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、FMPI2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、スレーブバイト制御モード (654 ページ) を参照してください。

特定アドレス（スレーブモード）

必要な場合は、特定の SMBus アドレスを有効にしてください。詳細については、バスアイドル検出 (676 ページ) を参照してください。

- SMBus デバイスのデフォルトアドレス (0b1100 001) は、FMPI2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。
- SMBus ホストアドレス (0b0001 000) は、FMPI2C_CR1 レジスタの SMBHEN ビットをセットすることによって有効になります。
- アラート応答アドレス (0b0001100) は、FMPI2C_CR1 レジスタの ALERTEN ビットをセットすることで有効になります。

パケットエラーチェック

PEC 計算を有効にするには、FMPI2C_CR1 レジスタの PECEN ビットをセットします。その場合、PEC 転送はハードウェアバイトカウンタ (FMPI2Cx_CR2 レジスタの NBYTES[7:0]) を使用して管理されます。PECEN ビットは、FMPI2C を有効にする前に設定する必要があります。

PEC 転送はハードウェアバイトカウンタによって管理されるので、スレーブモードで SMBus とインタフェースするときには SBC ビットをセットする必要があります。PEC は、PECBYTE ビットがセットされ、RELOAD ビットがクリアされたとき、NBYTES-1 データの転送後に転送されます。RELOAD がセットされた場合、PECBYTE は効果がありません。

注意： FMPI2C が有効なときには、PECEN 設定の変更はできません。

表 115. SMBUS の PEC 設定

モード	SBC ビット	RELOAD ビット	AUTOEND ビット	PECBYTE ビット
マスタ Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
マスタ Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
スレーブ Tx/Rx と PEC	1	0	x	1

タイムアウト検出

タイムアウト検出は、FMPI2C_TIMEOUTR レジスタの TIMOUTEN および TEXTEN ビットをセットすることによって有効になります。SMBus 仕様バージョン 2.0 で指定された最大時間の前にタイムアウトを検出するようにタイマをプログラムする必要があります。

● t_{TIMEOUT} チェック

t_{TIMEOUT} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] ビットを t_{TIMEOUT} パラメータをチェックするためにタイマ再ロード値でプログラムする必要があります。SCL ローレベルタイムアウトを検出するためには、TIDLE ビットを 0 に設定する必要があります。

その場合、タイマは、FMPI2C_TIMEOUTR レジスタの TIMOUTEN をセットすることによって有効になります。

SCL が (TIMEOUTA+1) x 2048 x t_{I2CCLK} より長い時間、ローに設定された場合、FMPI2C_ISR レジスタの TIMEOUT フラグがセットされます。

表 116: さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例（最大値 t_{TIMEOUT} = 25 ms）を参照してください。

注意： **TIMEOUTEN ビットがセットされているときには、TIMEOUTA[11:0] ビットおよび TIDLE ビットの設定変更はできません。**

● t_{LOW:SEXT} および t_{LOW:MEXT} チェック

ペリフェラルがマスタとして設定されているか、スレーブとして設定されているかに応じて、12 ビットの TIMEOUTB タイマは、スレーブの場合は t_{LOW:SEXT} をチェックするために、マスタの場合は t_{LOW:MEXT} をチェックするために、設定する必要があります。標準では最大値のみが規定されているので、両方について同じ値を選ぶことができます。

その場合、タイマは、FMPI2C_TIMEOUTR レジスタの TEXTEN ビットをセットすることによって有効になります。

SMBus ペリフェラルが、(TIMEOUTB+1) x 2048 x t_{I2CCLK} より長い時間および [バスアイドル検出 \(676 ページ\)](#) セクションで述べられているタイムアウト間隔で、累積 SCL ストレッチを実行した場合、FMPI2C_ISR レジスタの TIMEOUT フラグがセットされます。

表 117: さまざまな FMPI2CCLK 周波数での TIMEOUTB の設定例を参照してください。

注意： **TEXTEN ビットがセットされているときには、TIMEOUTB 設定変更はできません。**

バスアイドル検出

t_{IDLE} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] フィールドを t_{IDLE} パラメータを得るためにタイマ再ロード値でプログラムする必要があります。SCL および SDA ハイレベルタイムアウトを検出するためには、TIDLE ビットを 1 に設定する必要があります。

その場合、タイマは、FMPI2C_TIMEOUTR レジスタの TIMOUTEN ビットをセットすることによって有効になります。

SCL および SDA の両方のラインが (TIMEOUTA+1) x 4 x t_{I2CCLK} より長い間ハイのままであった場合、FMPI2C_ISR レジスタの TIMEOUT フラグがセットされます。



表 118：さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例（最大値 $t_{IDLE} = 50 \mu s$ ）を参照してください。

注意： **TIMEOUTEN** がセットされているときに、**TIMEOUTA** および **TIDLE** 設定を変更することはできません。

23.4.12 SMBus：FMPI2C_TIMEOUTR レジスタの設定例

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。[セクション 23.3：FMPI2C の実装](#)を参照してください。

- $t_{TIMEOUT}$ の最大時間を 25 ms に設定：

表 116. さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例
(最大値 $t_{TIMEOUT} = 25 \text{ ms}$)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{TIMEOUT}$
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$

- $t_{LOW:SEXT}$ および $t_{LOW:MEXT}$ の最大時間を 8 ms に設定：

表 117. さまざまな FMPI2CCLK 周波数での TIMEOUTB の設定例

f_{I2CCLK}	TIMEOUTB[11:0] ビット	TEXTEN ビット	$t_{LOW:EXT}$
8 MHz	0x1F	1	$32 \times 2048 \times 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 \times 2048 \times 62.5 \text{ ns} = 8 \text{ ms}$

- t_{IDLE} の最大時間を 50 μs に設定

表 118. さまざまな FMPI2CCLK 周波数での TIMEOUTA の設定例
(最大値 $t_{IDLE} = 50 \mu s$)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	t_{TIDLE}
8 MHz	0x63	1	1	$100 \times 4 \times 125 \text{ ns} = 50 \mu s$
16 MHz	0xC7	1	1	$200 \times 4 \times 62.5 \text{ ns} = 50 \mu s$

23.4.13 SMBus スレーブモード

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 23.3 : FMPI2C の実装](#)を参照してください。

FMPI2C スレーブ転送管理（[セクション 23.4.7 : FMPI2C スレーブモード](#)を参照）に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

SMBus スレーブトランスミッタ

IP が SMBus で使用されるときには、SBCは、プログラムされたデータバイト数の終わりの PEC 送信を可能にするため、1 にプログラムする必要があります。PECBYTE ビットがセットされているときには、NBYTES[7:0] でプログラムされたバイト数には PEC 送信が含まれます。その場合、TXIS 割り込みの合計数は NBYTES-1 であり、NBYTES-1 データ転送後にマスタが追加のバイトをリクエストした場合、FMPI2C_PECR レジスタの内容が自動的に送信されます。

注意： **PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。**

図 231. N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート

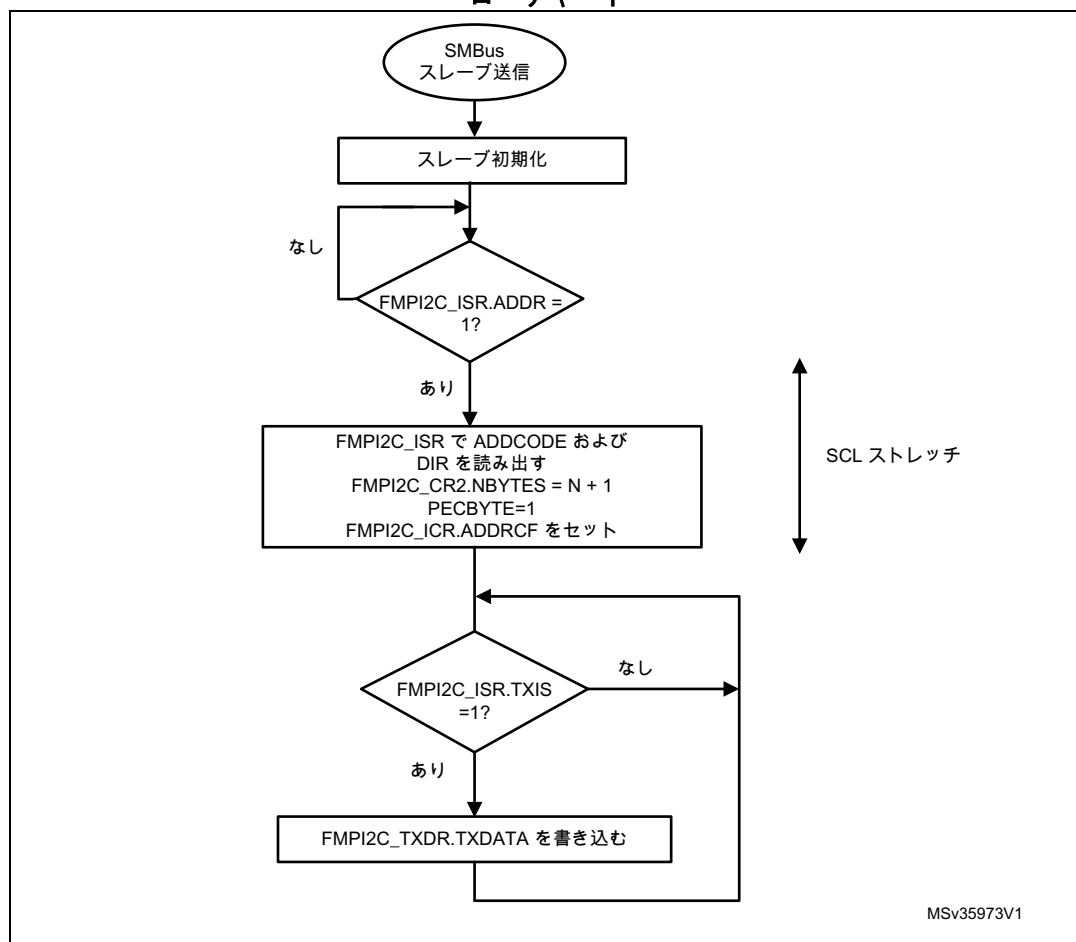
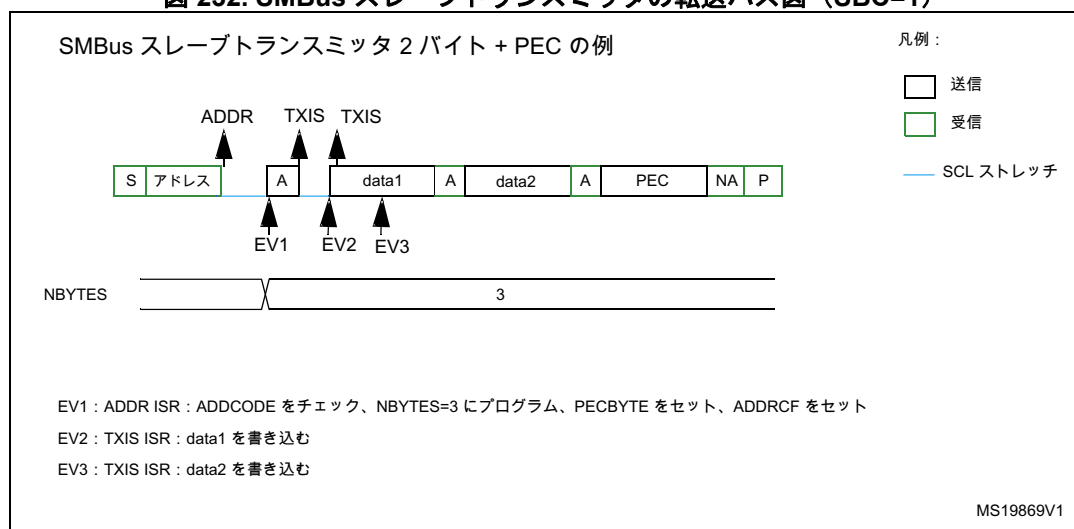


図 232. SMBus スレーブトランスミッタの転送バス図（SBC=1）



SMBus スレーブレシーバ

FMPI2C が SMBus モードで使用されるときには、SBC は、プログラムされたデータバイト数の終わりの PEC チェックを可能にするため、1 にプログラムする必要があります。各バイトの ACK 制御を可能にするためには、再ロードモードを選択する必要があります（RELOAD=1）。詳細については、[スレーバイト制御モード \(654 ページ\)](#) を参照してください。

PEC バイトをチェックするためには、RELOAD ビットをクリアして、PECBYTE ビットをセットする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが内部 FMPI2C_PECR レジスタの内容と比較されます。ACK ビットの値にかかわらず、比較が一致しなかった場合は NACK が自動的に生成され、比較が一致した場合は ACK が自動的に生成されます。PEC バイトが受信されると、他のデータと同様に FMPI2C_RXDR レジスタにコピーされ、RXNE フラグがセットされます。

PEC 不一致の場合、PECERR フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

ACK ソフトウェア制御が不要な場合は、PECBYTE=1 をプログラムし、同じ書き込み操作で NBYTES を連続フローで受信するバイト数にプログラムします。NBYTES-1 が受信された後、次の受信バイトが PEC であるかどうかチェックされます。

注意： *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 233. N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート

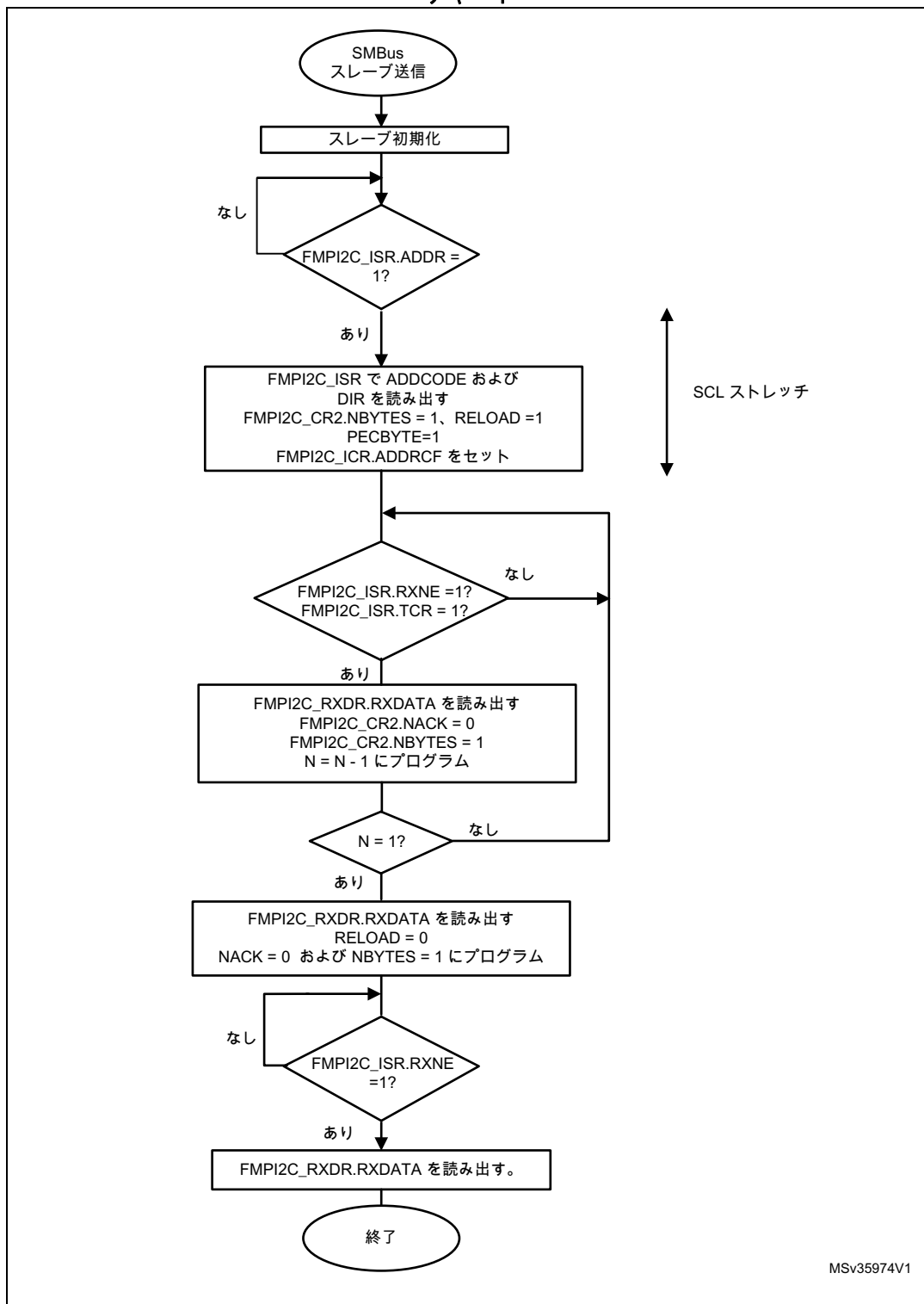
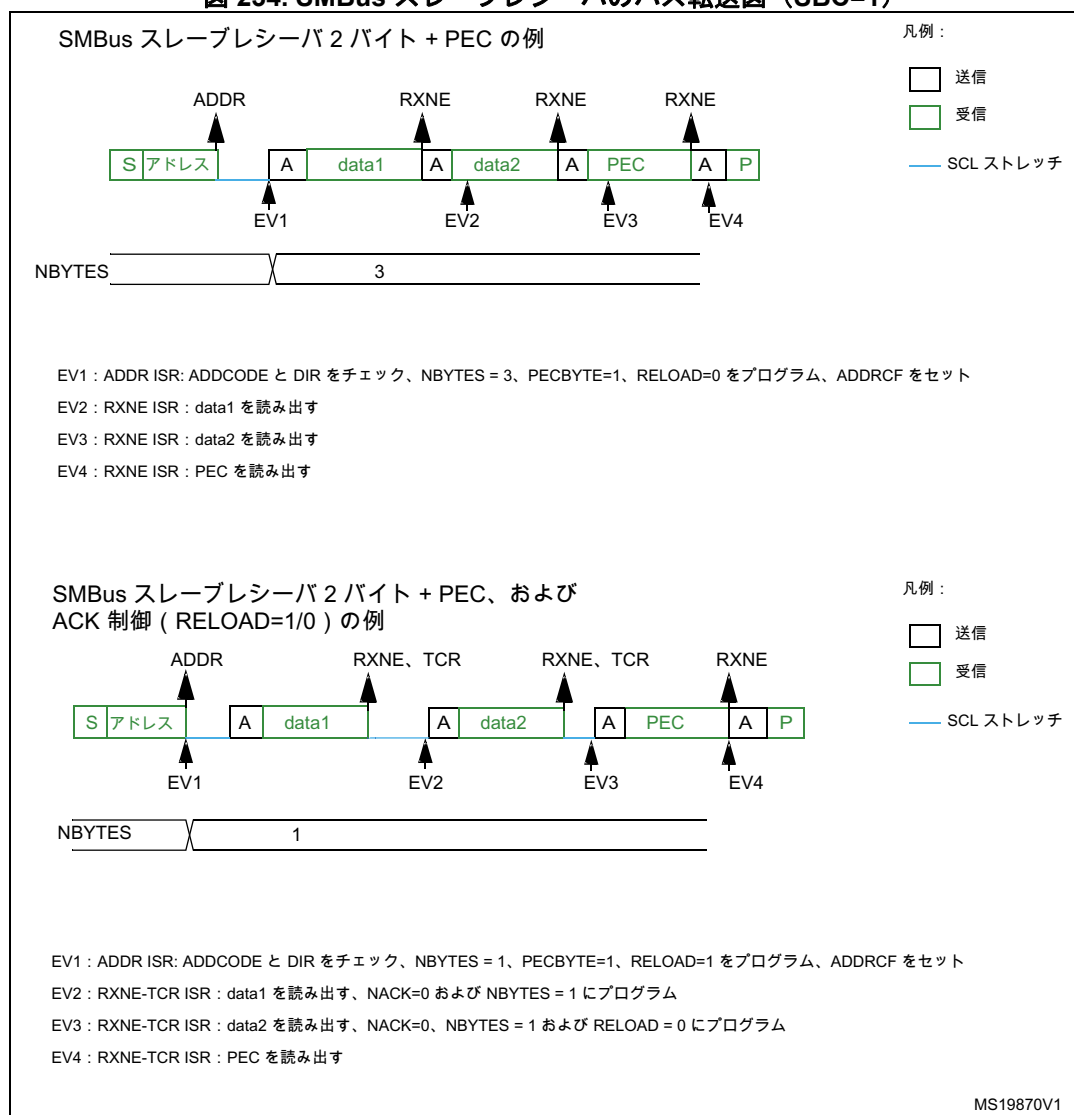


図 234. SMBus スレーブレシーバのバス転送図（SBC=1）



このセクションは、SMBus 機能がサポートされるときにのみ適用されます。[セクション 23.3 : FMPI2C の実装](#)を参照してください。

FMPI2C マスタ転送管理（[セクション 23.4.8 : FMPI2C マスタモード](#)を参照）に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

SMBus マスタトランスミッタ

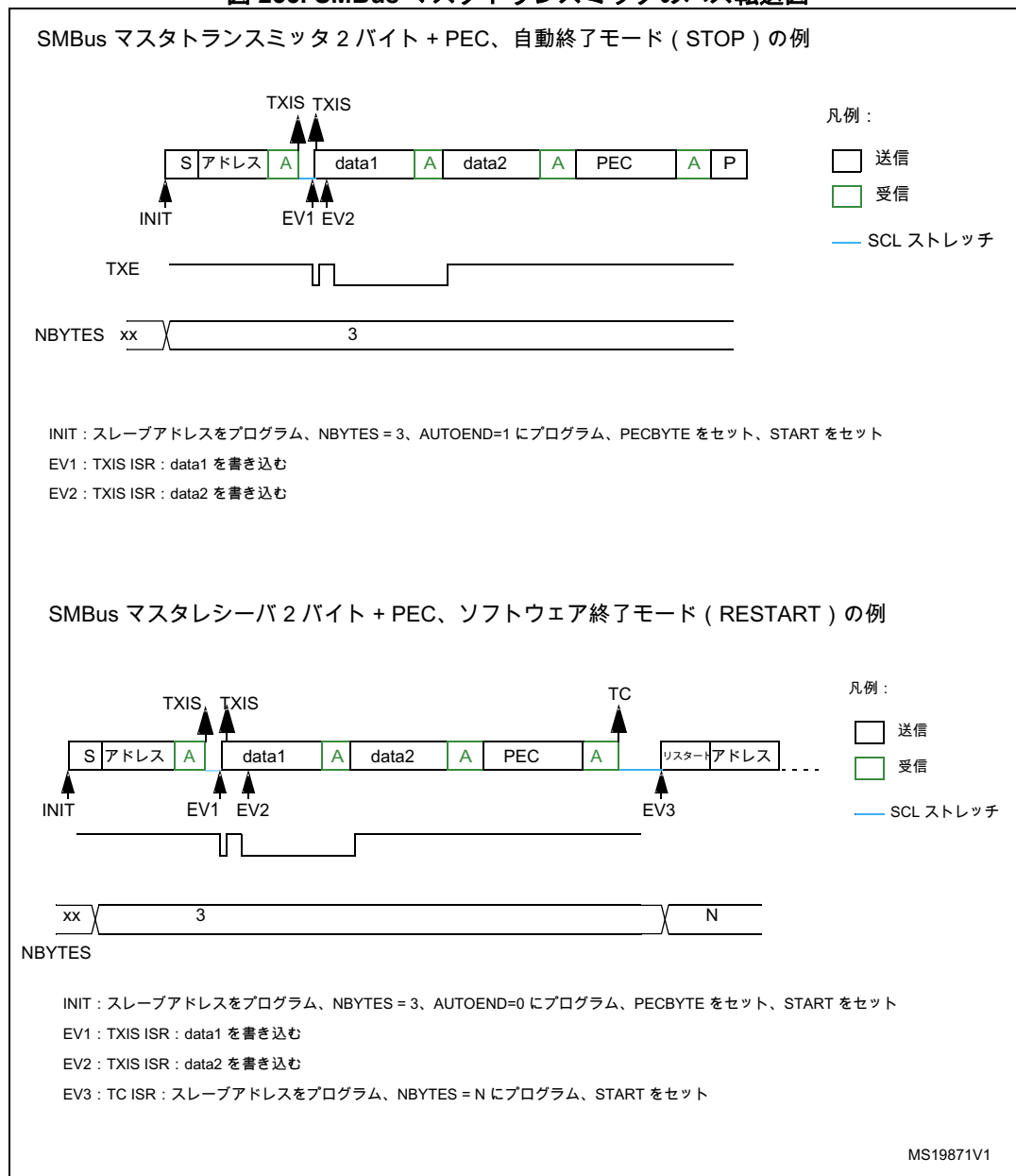
SMBus マスタが PEC を送信したいときには、START ビットをセットする前に、PECBYTE ビットをセットする必要があります。バイト数を NBYTES[7:0] フィールドでプログラムする必要があります。この場合、TXIS 割り込みの合計数は NBYTES-1 になります。したがって、NBYTES=0x1 のときに PECBYTE ビットがセットされた場合、FMPI2C_PECR レジスタの内容が自動的に送信されます。

SMBus マスタが PEC 後に STOP コンディションを送信したい場合は、自動終了モードを選択してください（AUTOEND=1）。この場合、PEC 送信に続いて、STOP コンディションが自動的に送信されます。

SMBus マスタが PEC 後に RESTART コンディションを送信したい場合は、ソフトウェアモードを選択してください（AUTOEND=0）。この場合、NBYTES-1 が送信されると、PEC 送信後に FMPI2C_PECR レジスタの内容が送信され、TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションを TC 割り込みサブルーチンでプログラムする必要があります。

注意： **PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。**

図 235. SMBus マスタトランスミッタのバス転送図



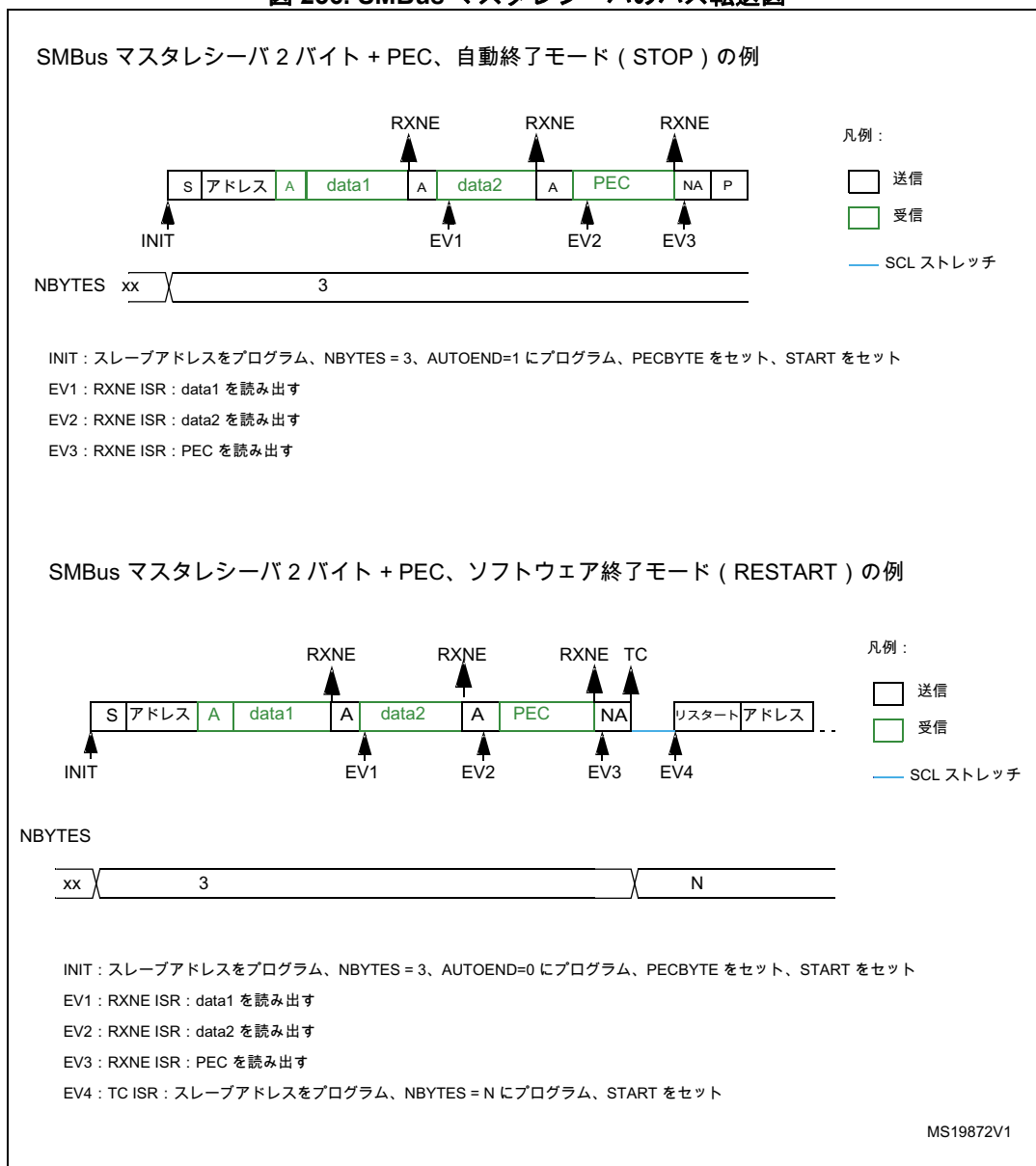
SMBus マスタレシーバ

SMBus マスタが転送終了時に PEC を受信してから STOP を受信したいときには、自動終了モードを選択できます（AUTOEND=1）。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが FMPI2C_PECR レジスタの内容と自動的に照合されます。PEC バイトに対して NACK 応答が与えられた後、STOP コンディションが送信されます。

SMBus マスタが転送終了時に PEC バイトを受信してから RESTART を受信したいときには、ソフトウェアモードを選択する必要があります（AUTOEND=0）。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが FMPI2C_PECR レジスタの内容と自動的に照合されます。PEC バイト受信後に TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションは、TC 割り込みサブルーチンでプログラムできます。

注意： *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 236. SMBus マスタレシーバのバス転送図



23.4.14 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

バスエラー (BERR)

バスエラーは、START または STOP コンディションが検出され、複数の 9 SCL クロックパルス後になかったときに検出されます。START または STOP コンディションは、SCL がハイとどき SDA エッジが発生した場合に検出されます。

バスエラーフラグは、FMPI2C がマスタまたはアドレス指定されたスレーブとして転送に関与する場合にのみ（すなわち、スレーブモードのアドレスフェーズでないとき）、セットされます。

スレーブモードで START または RESTART の誤配置が検出された場合、FMPI2C は、正しい START コンディションの場合と同様に、アドレス認識状態に入ります。

バスエラーが検出されると、FMPI2C_ISR レジスタの BERR フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

アービトレーション喪失（ARLO）

アービトレーション喪失は、SDA ラインでハイレベルが送信されたが、SCL 立ち上がりエッジでローレベルがサンプリングされたときに検出されます。

- マスタモードでは、アービトレーション喪失は、アドレスフェーズ、データフェーズ、およびデータ確認応答フェーズで検出されます。この場合、SDA および SCL ラインはリリースされ、START 制御ビットがハードウェアによってクリアされ、マスタは自動的にスレーブモードに切り替わります。
- スレーブモードでは、アービトレーション喪失は、データフェーズとデータ確認応答フェーズで検出されます。この場合、転送は中止され、SCL および SDA ラインがリリースされます。

アービトレーション喪失が検出されると、FMPI2C_ISR レジスタの ARLO フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

オーバーラン／アンダーランエラー（OVR）

オーバーランまたはアンダーラインエラーは、スレーブモードで NOSTRETCH=1 のとき、および次のときに検出されます：

- 受信時、新しいバイトが受信され、RXDR レジスタがまだ読み出されていないとき。新しい受信バイトは失われ、新しいバイトへの応答として NACK が自動的に送信されます。
- 送信時：
 - － STOPF=1 のときには、最初のデータバイトが送信されなければなりません。TXE=0、0xFF の場合、FMPI2C_TXDR レジスタの内容が送信され、そうでない場合は送信されません。
 - － 新しいバイトが送信されるべきときに、FMPI2Cx_TXDR レジスタがまだ書き込まれていなかった場合、0xFF が送信されます。

オーバーランまたはアンダーランエラーが検出されると、FMPI2Cx_ISR レジスタの OVR フラグがセットされ、FMPI2Cx_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

パケットエラーチェックエラー（PECERR）

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。[セクション 23.3 : FMPI2C の実装](#) を参照してください。

PEC エラーは、受信した PEC バイトが FMPI2C_PECR レジスタの内容と一致しなかったときに検出されます。正しくない PEC の受信後、NACK が自動的に送信されます。

PEC エラーが検出されると、FMPI2C_ISR レジスタの PECERR フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

タイムアウトエラー（TIMEOUT）

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。[セクション 23.3 : FMPI2C の実装](#) を参照してください。

タイムアウトエラーは、次のような条件で発生します：

- TIDLE=0 であり、SCL が TIMEOUTA[11:0] ビットで定義された時間だけローのままであった場合：これは SMBus タイムアウトの検出に使用されます。
- TIDLE=1 であり、SDA および SCL が TIMEOUTA[11:0] ビットで定義された時間だけハイのままであった場合：これはバスアイドル状態の検出に使用されます。
- マスタ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合（SMBus $t_{\text{LOW:MEXT}}$ パラメータ）。
- スレーブ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合（SMBus $t_{\text{LOW:SEXT}}$ パラメータ）。

マスタモードでタイムアウト違反が検出されると、STOP コンディションが自動的に送信されます。

スレーブモードでタイムアウト違反が検出されると、SDA および SCL ラインが自動的にリリースされます。

タイムアウトエラーが検出されると、FMPI2C_ISR レジスタの TIMEOUT フラグがセットされ、FMPI2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

アラート（ALERT）

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。[セクション 23.3 : FMPI2C の実装](#) を参照してください。

ALERT フラグは、FMPI2C インタフェースがホストとして設定され（SMBHEN=1）、アラートピン検出が有効であり（ALERTEN=1）、SMBA ピンで立ち下がりエッジが検出されたときにセットされます。FMPI2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。

23.4.15 DMA リクエスト

DMA を使用した送信

送信について DMA (Direct Memory Access) を有効にするには、FMPI2C_CR1 レジスタの TXDMAEN ビットをセットします。TXIS ビットがセットされるたびに、データは、DMA ペリフェラル（[セクション 9 : ダイレクトメモリアクセスコントローラ \(DMA\) \(189 ページ\)](#)）を使用して設定された SRAM 領域から FMPI2C_TXDR レジスタにロードされます。

データのみが DMA で転送されます。

- マスタモード：初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます（送信されたスレーブアドレスを DMA で転送することはできません）。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。[マスタトランスミッタ \(665 ページ\)](#) を参照してください。

- スレーブモードでは：
 - － NOSTRETCH=0 では、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
 - － NOSTRETCH=1 では、アドレス一致イベントの前に DMA を初期化する必要があります。
- SMBus をサポートする場合：PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブトランスミッタ \(679 ページ\)](#) および [SMBus マスタトランスミッタ \(682 ページ\)](#) を参照してください。

注： DMA が送信に使用される場合、TXIE ビットが有効である必要はありません。

DMA を使用した受信

受信について DMA (Direct Memory Access) を有効にするには、FMPI2C_CR1 レジスタの RXDMAEN ビットをセットします。RXNE ビットがセットされているときには、データは、FMPI2C_RXDR レジスタから DMA ペリフェラル（[セクション 9：ダイレクトメモリアクセスコントローラ \(DMA\)](#) を参照）を使用して設定された SRAM 領域にロードされます。データのみ（PEC を含む）が DMA で転送されます。

- マスタモード、初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。
- NOSTRETCH=0 のスレーブモードでは、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
- SMBus がサポートされる場合（[セクション 23.3：FMPI2C の実装](#) を参照）：PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブレシーバ \(680 ページ\)](#) および [SMBus マスタレシーバ \(684 ページ\)](#) を参照してください。

注： DMA が受信に使用される場合、RXIE ビットが有効である必要はありません。

23.4.16 デバッグモード

マイクロコントローラがデバッグモードに入ると（コア停止）、DBG モジュールの DBG_I2Cx_ 設定ビットに応じて、SMBus タイムアウトは、通常の動作を続行するか、あるいは停止します。

23.5 FMPI2C 低電力モード

表 119. 低電力モード

モード	説明
SLEEP	影響なし。 FMPI2C 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	FMPI2C レジスタの内容を保存します。
STANDBY	FMPI2C ペリフェラルはパワーダウンされ、STANDBY 終了後に再初期化する必要があります。

23.6 FMPI2C 割り込み

次の表に、FMPI2C 割り込みリクエストの一覧を示します。

表 120. FMPI2C 割り込みリクエスト

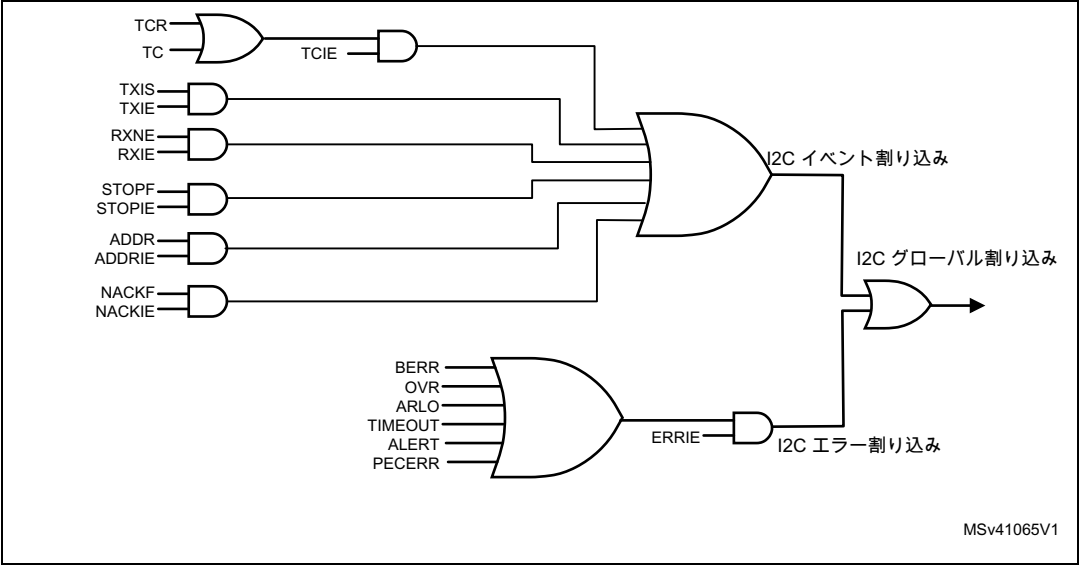
割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
受信バッファノットエンプティ	RXNE	FMPI2C_RXDR レジスタを 読み出す	RXIE
送信バッファ割り込みステータス	TXIS	FMPI2C_TXDR レジスタに 書き込む	TXIE
STOP 検出割り込みフラグ	STOPF	STOPCF=1 を書き込む	STOPIE
転送完了再ロード	TCR	FMPI2C_CR2 の NBYTES[7:0] ≠ 0 で書き 込む	TCIE
転送完了	TC	START=1 または STOP=1 を書き込む	
アドレス一致	ADDR	ADDRCF=1 を書き込む	ADDRIE
NACK 受信	NACKF	NACKCF=1 を書き込む	NACKIE
バスエラー	BERR	BERRCF=1 を書き込む	ERRIE
アービトレーション喪失	ARLO	ARLOCF=1 を書き込む	
オーバーラン/アンダーラン	OVR	OVRCF=1 を書き込む	
PEC エラー	PECERR	PECERRCF=1 を書き込む	
タイムアウト/t _{LOW} エラー	TIMEOUT	TIMEOUTCF=1 を書き込む	
SMBus アラート	ALERT	ALERTCF=1 を書き込む	

製品実装に応じて、これらの割り込みイベントのすべてで同じ割り込みベクトルを共有するか（FMPI2C グローバル割り込み）、2つの割り込みベクトル（FMPI2C イベント割り込みと FMPI2C エラー割り込み）にグループ化することができます。詳細については、を参照してください。

FMPI2C 割り込みを有効にするには、次のシーケンスが必要です：

1. NVIC で FMPI2C IRQ チャンネルを設定し、有効にします。
2. 割り込みを生成するように FMPI2C を設定します。

図 237. FMPI2C 割り込みマッピング図



MSv41065V1

23.7 FMPI2C レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタは、ワード（32 ビット）単位でアクセスされます。

23.7.1 制御レジスタ 1 (FMPI2C_CR1)

アドレスオフセット：0x00

リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{FMPI2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERT EN	SMBD EN	SMBH EN	GCEN	Res.	NOSTR ETCH	SBC
								rw	rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.	ANF OFF	DNF				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PECEN** : PEC イネーブル

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

注 : *SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。*

ビット 22 **ALERTEN** : SMBus アラート有効

デバイスモード (SMBHEN=0) :

0 : SMBA ピンをハイにリリースし、アラート応答アドレスヘッダを無効にします。0001100x の後に NACK が続きます。

1 : SMBA ピンをローに駆動し、アラート応答アドレスヘッダを有効にします : 0001100x の後に ACK が続きます。

ホストモード (SMBHEN=1) :

0 : SMBus アラートピン (SMBA) はサポートされません。

1 : SMBus アラートピン (SMBA) はサポートされます。

注 : *ALERTEN=0 のときには、SMBA ピンを標準 GPIO として使用できます。*

SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。

ビット 21 **SMBDEN** : SMBus デバイスデフォルトアドレス有効

0 : デバイスデフォルトアドレス無効。アドレス 0b1100001x は NACK されます。

1 : デバイスデフォルトアドレス有効。アドレス 0b1100001x は ACK されます。

注 : *SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。*

ビット 20 **SMBHEN** : SMBus ホストアドレス有効

0 : ホストアドレス無効。アドレス 0b0001000x は NACK されます。

1 : ホストアドレス有効。アドレス 0b0001000x は ACK されます。

注 : *SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。*

ビット 19 **GCEN** : 同報イネーブル

0 : 同報は無効です。アドレス 0b00000000 は NACK されます。

1 : 同報は有効です。アドレス 0b00000000 は ACK されます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **NOSTRETCH** : クロックストレッチ無効

このビットは、スレーブモードでのクロックストレッチを無効にするために使用されます。マスタモードではクリアされたままでなければなりません。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

注 : *このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。*

ビット 16 **SBC** : スレーブバイト制御

このビットは、スレーブモードでのハードウェアバイト制御を有効にするために使用されます。

0 : スレーブバイト制御無効

1 : スレーブバイト制御有効

ビット 15 **RXDMAEN** : DMA 受信リクエスト有効

0 : DMA モードは受信に無効

1 : DMA モードは受信に有効

ビット 14 **TXDMAEN** : DMA 送信リクエスト有効

0 : DMA モードは送信に無効

1 : DMA モードは送信に有効

ビット 13 予約済みであり、リセット値に保持する必要があります。



ビット 12 **ANFOFF** : アナログノイズフィルタ OFF

0 : アナログノイズフィルタ有効

1 : アナログノイズフィルタ無効

注 : このビットは、FMPI2C が無効 ($PE = 0$) のときのみプログラムできます。

ビット 11:8 **DNF[3:0]** : デジタルノイズフィルタ

これらのビットは、SDA および SCL 入力のデジタルノイズフィルタを設定するために使用されます。デジタルフィルタは、最大 $DNF[3:0] * t_{I2CCLK}$ の長さのスパイクを除去します。

0000 : デジタルフィルタ無効

0001 : デジタルフィルタは有効であり、最大 $1 t_{I2CCLK}$ の除去能力を持ちます。

...

1111 : デジタルフィルタは有効であり、最大 $15 t_{I2CCLK}$ の除去能力を持ちます。

注 : アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

このフィルタは、FMPI2C が無効 ($PE = 0$) のときのみプログラムできます。

ビット 7 **ERRIE** : エラー割り込み有効

0 : エラー検出割り込み無効

1 : エラー検出割り込み有効

注 : 次のようなエラーが発生すると、割り込みが生成されます :

アービトレーション喪失 (ARLO)

バスエラー検出 (BERR)

オーバーラン/アンダーラン (OVR)

タイムアウト検出 (TIMEOUT)

PEC エラー検出 (PECERR)

アラートピンイベント検出 (ALERT)

ビット 6 **TCIE** : 転送完了割り込み有効

0 : 転送完了割り込み無効

1 : 転送完了割り込み有効

注 : 次のようなイベントが発生すると、割り込みが生成されます :

転送完了 (TC)

転送完了再ロード (TCR)

ビット 5 **STOPIE** : STOP 検出割り込み有効

0 : STOP 検出 (STOPF) 割り込み無効

1 : STOP 検出 (STOPF) 割り込み有効

ビット 4 **NACKIE** : 非確認応答受信割り込み有効

0 : 非確認応答 (NACKF) 受信割り込み無効

1 : 非確認応答 (NACKF) 受信割り込み有効

ビット 3 **ADDRIE** : アドレス一致割り込み有効 (スレーブのみ)

0 : アドレス一致 (ADDR) 割り込み無効

1 : アドレス一致 (ADDR) 割り込み有効

ビット 2 **RXIE** : RX 割り込み有効

0 : 受信 (RXNE) 割り込み無効

1 : 受信 (RXNE) 割り込み有効

ビット 1 **TXIE** : TX 割り込み有効

0 : 送信 (TXIS) 割り込み無効

1 : 送信 (TXIS) 割り込み有効

ビット 0 PE：ペリフェラルは有効です。
0：ペリフェラルは無効です。
1：ペリフェラルは有効です。

注： PE=0 のとき、FMPI2C SCL および SDA ラインはリリースされます。内部ステートマシンおよびステータスビットはリセット値に戻されます。クリアされたときには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たれる必要があります。

23.7.2 制御レジスタ 2 (FMPI2C_CR2)

アドレスオフセット：0x04
リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 2 x PCLK1 + 6 x FMPI2CCLK です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_W RN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 PECBYTE：パケットエラーチェックバイト

このビットはソフトウェアによってセットされ、PEC が転送されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、また、PE=0 のとき、ハードウェアによってクリアされます。

0：PEC 転送なし。
1：PEC 送信／受信がリクエストされます。

注： このビットに 0 を書き込んでも、効果はありません。
このビットは、RELOAD がセットされているときには効果がありません。
このビットは、SBC=0 のとき、スレーブモードでは効果がありません。
SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3：FMPI2C の実装を参照してください。

ビット 25 AUTOEND：自動終了モード（マスタモード）

このビットは、ソフトウェアによってセット／クリアされます。

0：ソフトウェア終了モード：NBYTES データが転送されると TC フラグがセットされ、SCL ローをストレッチします。
1：自動終了モード：NBYTES データが転送されると、STOP コンディションが自動的に送信されます。

注： このビットは、スレーブモードまたは RELOAD ビットがセットされているときには効果がありません。

ビット 24 RELOAD : NBYTES 再ロードモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : 転送は、NBYTES データの転送後、完了します (STOP または RESTART が続きます)。

1 : 転送は、NBYTES データの転送後に完了しません (NBYTES が再ロードされます)。NBYTES データが転送されると TCR フラグがセットされ、SCLローをストレッチします。

ビット 23:16 NBYTES[7:0] : バイト数

送受信されるバイト数は、ここでプログラムされます。このフィールドは、SBC=0 のスレーブモードでは効果がありません。

注： *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 15 NACK : NACK 生成 (スレーブモード)

このビットはソフトウェアによってセットされ、NACK が送信されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、または PE=0 のとき、ハードウェアによってクリアされます。

0 : 現在の受信バイト後に ACK が送信されます。

1 : 現在の受信バイト後に NACK が送信されます。

注： *このビットに 0 を書き込んでも、効果はありません。*

このビットは、スレーブモードでのみ使用されます：マスタレシーバモードでは、NACK ビットの値にかかわらず、STOP または RESTART コンディション前の最後のバイト後に NACK が自動的に生成されます。

スレーブレシーバ NOSTRETCH モードでオーバーランが発生すると、NACK ビットの値にかかわらず、NACK が自動的に生成されます。

ハードウェア PEC チェックが有効なとき (PECBYTE=1)、PEC 確認応答値は NACK 値に依存しません。

ビット 14 STOP : STOP 生成 (マスタモード)

このビットはソフトウェアによってセットされ、STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

マスタモード：

0 : STOP生成なし。

1 : 現在のバイト転送後の STOP生成。

注： *このビットに 0 を書き込んでも、効果はありません。*

ビット 13 START : START 生成

このビットはソフトウェアによってセットされ、START とアドレスシーケンスが送信された後、アービトラクション喪失によって、タイムアウトエラー検出によって、または PE=0 のときに、ハードウェアによってクリアされます。FMPI2C_ICR レジスタの ADDRCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアすることもできます。

0 : START生成なし。

1 : RESTART/START生成：

– FMPI2C がすでにマスタモードであり、AUTOEND=0 の場合、このビットをセットすると、RELOAD=0 のとき、NBYTES 転送の終了後に反復 START コンディションが生成されます。

– そうでない場合、このビットをセットすると、バスがフリーになると、START コンディションが生成されます。

注： *このビットに 0 を書き込んでも、効果はありません。*

バスが BUSY の場合、または FMPI2C がスレーブモードの場合でも、START ビットをセットできます。

このビットは、RELOAD がセットされているときには効果がありません。10 ビットアドレッシングモードでは、アドレスの最初の部分で NACK が受信される場合、START ビットはハードウェアによってクリアされず、START ビットがソフトウェアによってクリアされない限り、マスタはアドレスシーケンスを再送します。

ビット 12 **HEAD10R** : 読み出し方向のみの10 ビットアドレスヘッダ（マスタレーバモード）

0 : マスタは完全な 10 ビットスレーブアドレス読み出しシーケンスを送信します : START + 2 バイトの書き込み方向の 10 ビットアドレス + RESTART + 読み出し方向の 10 ビットアドレスの最初の 7 ビット。

1 : マスタは 10 ビットアドレスの最初の 7 ビットのみを送信し、その後に読み出し方向を送信します。

注 : *START ビットがセットされているときに、このビットを変更することはできません。*

ビット 11 **ADD10** : 10 ビットアドレッシングモード（マスタモード）

0 : マスタは 7 ビットアドレッシングモードで動作します。

1 : マスタは 10 ビットアドレッシングモードで動作します。

注 : *START ビットがセットされているときに、このビットを変更することはできません。*

ビット 10 **RD_WRN** : 転送方向（マスタモード）

0 : マスタは書き込み転送をリクエストします。

1 : マスタは読み出し転送をリクエストします。

注 : *START ビットがセットされているときに、このビットを変更することはできません。*

ビット 9:8 **SADD[9:8]** : スレーブアドレスビット 9:8（マスタモード）

7 ビットアドレッシングモード（ADD10=0） :

これらのビットは無視されます。

10 ビットアドレッシングモード（ADD10=1） :

これらのビットには、送信されるスレーブアドレスのビット 9:8 を書き込みます。

注 : *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 7:1 **SADD[7:1]** : スレーブアドレスビット 7:1（マスタモード）

7 ビットアドレッシングモード（ADD10=0） :

これらのビットには、送信される 7 ビットのスレーブアドレスを書き込みます。

10 ビットアドレッシングモード（ADD10=1） :

これらのビットには、送信されるスレーブアドレスのビット 7:1 を書き込みます。

注 : *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 0 **SADD0** : スレーブアドレスビット 0（マスタモード）

7 ビットアドレッシングモード（ADD10=0） :

このビットは無視されます。

10 ビットアドレッシングモード（ADD10=1） :

このビットには、送信されるスレーブアドレスのビット 0 を書き込みます。

注 : *START ビットがセットされているときに、これらのビットを変更することはできません。*

23.7.3 Own Address 1 レジスタ（FMPI2C_OAR1）

アドレスオフセット：0x08

リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{FMPI2CCCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA1EN**：Own Address 1 有効

- 0：Own Address 1 無効。受信されたスレーブアドレス OA1 は NACK されます。
- 1：Own Address 1 有効。受信されたスレーブアドレス OA1 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OA1MODE** Own Address 1 10 ビットモード

- 0：Own Address 1は 7 ビットアドレスです。
- 1：Own Address 1は 10 ビットアドレスです。

注： このビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 9:8 **OA1[9:8]**：インタフェースアドレス

- 7 ビットアドレッシングモード：無視されます。
- 10 ビットアドレッシングモード：アドレスのビット 9:8。

注： これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 7:1 **OA1[7:1]**：インタフェースアドレス

- アドレスのビット 7:1
- 注：** これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 0 **OA1[0]**：インタフェースアドレス

- 7 ビットアドレッシングモード：無視されます。
- 10 ビットアドレッシングモード：アドレスのビット 0。

注： このビットは、OA1EN=0 のときのみ書き込むことができます。

23.7.4 Own Address 2 レジスタ（FMPI2C_OAR2）

アドレスオフセット：0x0C

リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{FMPI2CCCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]				OA2[7:1]						Res.
rw					rw				rw						

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA2EN** : Own Address 2 有効

- 0 : Own Address 2 無効。受信されたスレーブアドレス OA2 は NACK されます。
- 1 : Own Address 2 有効。受信されたスレーブアドレス OA2 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **OA2MSK[2:0]** : Own Address 2 マスク

- 000 : マスクなし。
- 001 : OA2[1] はマスクされ、無視されます。OA2[7:2] のみ比較されます。
- 010 : OA2[2:1] はマスクされ、無視されます。OA2[7:3] のみ比較されます。
- 011 : OA2[3:1] はマスクされ、無視されます。OA2[7:4] のみ比較されます。
- 100 : OA2[4:1] はマスクされ、無視されます。OA2[7:5] のみ比較されます。
- 101 : OA2[5:1] はマスクされ、無視されます。OA2[7:6] のみ比較されます。
- 110 : OA2[6:1] はマスクされ、無視されます。OA2[7] のみ比較されます。
- 111 : OA2[7:1] はマスクされ、無視されます。比較は行われず、すべての（予約済みを除く）7 ビット受信アドレスが確認応答されます。

注： これらのビットは、OA2EN=0 のときのみ書き込むことができます。
OA2MSK が 0 でなくなると、予約済み FMPI2C アドレス (0b0000xxx および 0b1111xxx) は、比較が一致した場合でも確認応答されません。

ビット 7:1 **OA2[7:1]** : インタフェースアドレス

アドレスのビット 7:1

注： これらのビットは、OA2EN=0 のときのみ書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。



23.7.5 タイミングレジスタ（FMPI2C_TIMINGR）

アドレスオフセット：0x10

リセット値：0x0000 0000

アクセス：ウェイト状態なし

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

ビット 31:28 **PRESC[3:0]**：タイミングプリスケアラ

このフィールドは、データのセットアップおよびホールドカウンタ（[FMPI2C のタイミング \(646 ページ\)](#)）を参照）と SCL ハイおよびローレベルカウンタ（[FMPI2C マスタ初期化 \(661 ページ\)](#)）を参照）に使用されるクロック周期 t_{PRESC} を生成して、FMPI2CCLK をプリスケールするために使用されます。

$$t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{2CCLK}}$$

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **SCLDEL[3:0]**：データセットアップ時間

このフィールドは、SDA エッジと SCL 立ち上がりエッジの間に遅延 t_{SCLDEL} を生成するために使用されます。NOSTRETCH = 0 でのマスタモードおよびスレーブモードでは、SCL ラインは、 t_{SCLDEL} の間ローにストレッチされます。

$$t_{\text{SCLDEL}} = (\text{SCLDEL} + 1) \times t_{\text{PRESC}}$$

注： t_{SCLDEL} は、 $t_{\text{SU:DAT}}$ タイミングを生成するために使用されます。

ビット 19:16 **SDADEL[3:0]**：データホールド時間

このフィールドは、SCL 立ち下がりエッジと SDA エッジの間に遅延 t_{SDADEL} を生成するために使用されます。NOSTRETCH = 0 でのマスタモードおよびスレーブモードでは、SCL ラインは、 t_{SDADEL} の間ローにストレッチされます。

$$t_{\text{SDADEL}} = \text{SDADEL} \times t_{\text{PRESC}}$$

注： t_{SDADEL} は、 $t_{\text{HD:DAT}}$ タイミングを生成するために使用されます。

ビット 15:8 **SCLH[7:0]**：SCL ハイ周期（マスタモード）

このフィールドは、マスタモードで SCL ハイ周期を生成するために使用されます。

$$t_{\text{SCLH}} = (\text{SCLH} + 1) \times t_{\text{PRESC}}$$

注： t_{SCLH} は、 $t_{\text{SU:STO}}$ および $t_{\text{HD:STA}}$ タイミングを生成するためにも使用されます。

ビット 7:0 **SCLL[7:0]**：SCL ロー周期（マスタモード）

このフィールドは、マスタモードで SCL ロー周期を生成するために使用されます。

$$t_{\text{SCLL}} = (\text{SCLL} + 1) \times t_{\text{PRESC}}$$

注： t_{SCLL} は、 t_{BUF} および $t_{\text{SU:STA}}$ タイミングを生成するためにも使用されます。

注： このレジスタは、FMPI2C が無効（PE=0）のときに設定する必要があります。

注： STM32CubeMX ツールは、I2C 設定ウィンドウの I2C_TIMINGR の内容を計算して示します。

23.7.6 タイムアウトレジスタ（FMPI2C_TIMEOUTR）

アドレスオフセット：0x14

リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{FMPI2CCCLK}$ です。

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

ビット 31 **TEXTEN**：拡張クロックタイムアウト有効

- 0：拡張クロックタイムアウト検出は無効です。
1：拡張クロックタイムアウト検出は有効です。 $t_{\text{LOW:EXT}}$ を超える累積 SCL ストレッチが FMPI2C インタフェースによって行われると、タイムアウトエラーが検出されます（TIMEOUT=1）。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **TIMEOUTB[11:0]**：バスタイムアウト B

このフィールドは、累積クロック拡張タイムアウトを設定するために使用されます：
マスタモードでは、マスタ累積クロックロー拡張時間（ $t_{\text{LOW:MEXT}}$ ）が検出されます。
スレーブモードでは、スレーブ累積クロックロー拡張時間（ $t_{\text{LOW:SEXT}}$ ）が検出されます。
 $t_{\text{LOW:EXT}} = (\text{TIMEOUTB} + 1) \times 2048 \times t_{\text{2CCLK}}$

注： これらのビットは、**TEXTEN=0** のときのみ書き込むことができます。

ビット 15 **TIMOUTEN**：クロックタイムアウト有効

- 0：SCL タイムアウト検出は無効です。
1：SCL タイムアウト検出は有効です。SCL が t_{TIMEOUT} （**TIDLE=0**）を超えてローであるか、 t_{IDLE} （**TIDLE=1**）を超えてハイであった場合、タイムアウトエラーが検出されます（TIMEOUT=1）。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TIDLE**：アイドルクロックタイムアウト検出

- 0：TIMEOUTA は、SCL ロータイムアウトの検出に使用されます。
1：TIMEOUTA は、SCL と SDA の両方のハイタイムアウト（バスアイドル条件）の検出に使用されます。

注： このビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

ビット 11:0 **TIMEOUTA[11:0]**：バスタイムアウト A

このフィールドは、以下を設定するために使用されます：
– **TIDLE=0** のときの SCL ロータイムアウト条件 t_{TIMEOUT}
 $t_{\text{TIMEOUT}} = (\text{TIMEOUTA} + 1) \times 2048 \times t_{\text{2CCLK}}$
– **TIDLE=1** のときのバスアイドル条件（SCL と SDA の両方のハイ）
 $t_{\text{IDLE}} = (\text{TIMEOUTA} + 1) \times 4 \times t_{\text{2CCLK}}$

注： これらのビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

注： **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 23.3：FMPI2C の実装を参照してください。**

23.7.7 割り込みおよびステータスレジスタ（FMPI2C_ISR）

アドレスオフセット：0x18
リセット値：0x0000 0001
アクセス：ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]						DIR	
								r						r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23:17 **ADDCODE[6:0]**：アドレス一致コード（スレーブモード）
これらのビットは、アドレス一致イベントが発生したときに（ADDR = 1）、受信したアドレスで更新されます。
10 ビットアドレスの場合、ADDCODE は 10 ビットのヘッダとその後のアドレスの 2 つの MSB を示します。
- ビット 16 **DIR**：転送方向（スレーブモード）
このフラグは、アドレス一致イベントが発生したときに（ADDR=1）、更新されます。
0：書き込み転送、スレーブはレシーバモードになります。
1：読み出し転送、スレーブはトランスミッタモードになります。
- ビット 15 **BUSY**：バスビジー
このフラグは、バスで通信が進行中であることを示します。START コンディションが検出されたときに、ハードウェアによってセットされます。STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **ALERT**：SMBus アラート
このフラグは、SMBHEN=1（SMBus ホスト設定）、ALERTEN=1、および SMBALERT イベント（立ち下がりエッジ）が SMBA ピンで検出されたときに、ハードウェアによってセットされます。ALERTCF ビットをセットすることによって、ソフトウェアによってクリアされます。
注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。
SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3：FMPI2C の実装を参照してください。
- ビット 12 **TIMEOUT**：タイムアウトまたは t_{LOW} 検出フラグ
このフラグは、タイムアウトまたは拡張クロックタイムアウトが発生したときに、ハードウェアによってセットされます。TIMEOUTCF ビットをセットすることによって、ソフトウェアによってクリアされます。
注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。
SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3：FMPI2C の実装を参照してください。



ビット 11 PECERR：受信時の PEC エラー

このフラグは、受信した PEC が PEC レジスタの内容に一致しないときに、ハードウェアによってセットされます。正しくない PEC の受信後、NACK が自動的に送信されます。PECCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。

ビット 10 OVR：オーバーラン／アンダーラン（スレーブモード）

このフラグは、NOSTRETCH=1 のスレーブモードで、オーバーラン／アンダーランエラーが発生したときに、ハードウェアによってセットされます。OVRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 9 ARLO：アービトレーション喪失

このフラグは、アービトレーション喪失の場合に、ハードウェアによってセットされます。ARLOCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 8 BERR：バスエラー

このフラグは、ペリフェラルが転送に関与しているので、START または STOP コンディションの誤配置が検出されたときに、ハードウェアによってセットされます。このフラグは、スレーブモードのアドレスフェーズではセットされません。BERRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 7 TCR：転送完了再ロード

このフラグは、RELOAD=1 および NBYTES データが転送されたときに、ハードウェアによってセットされます。NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

このフラグは、マスタモード、または SBC ビットがセットされているときのスレーブモードのみ使用されます。

ビット 6 TC：転送完了（マスタモード）

このフラグは、RELOAD=0、AUTOEND=0、および NBYTES データが転送されたときに、ハードウェアによってセットされます。START ビットまたは STOP ビットがセットされたときに、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 5 STOPF：STOP 検出フラグ

このフラグは、バス上で STOP コンディションが検出され、ペリフェラルがこの転送に関与しているときに、ハードウェアによってセットされます：

－ マスタとして。ただし、STOP コンディションがペリフェラルによって生成される場合。

－ または、スレーブとして。ただし、ペリフェラルがこの転送中にアドレス指定されていた場合。

STOPCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 4 NACKF：非確認応答受信フラグ

このフラグは、バイト送信後に NACK を受信したときに、ハードウェアによってセットされます。NACKCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 3 ADDR：アドレス一致（スレーブモード）

このビットは、受信したスレーブアドレスが有効なスレーブアドレスの 1 つに一致したときに、ハードウェアによってセットされます。ADDRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。



- ビット 2 **RXNE** : 受信データレジスタノットエンプティ（レシーバ）
このビットは、受信データが FMPI2C_RXDR レジスタにコピーされ、読み出す準備ができたときに、ハードウェアによってセットされます。RXNE は、FMPI2C_RXDR が読み出されたときにクリアされます。
注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。
- ビット 1 **TXIS** : 送信割り込みステータス（トランスミッタ）
このビットは、FMPI2C_TXDR レジスタが空であり、送信データを FMPI2C_TXDR レジスタに書き込む必要があるときに、ハードウェアによってセットされます。次の送信データが FMPI2C_TXDR レジスタに書き込まれたときにクリアされます。
このビットは、NOSTRETCH=1 のときのみ、ソフトウェアによって 1 を書き込んで、TXIS イベントを生成することができます（TXIE=1 の場合に割り込み、または TXDMAEN=1 の場合に DMA リクエスト）。
注： このビットは、PE=0 のとき、ハードウェアによってクリアされます。
- ビット 0 **TXE** : 送信データレジスタエンプティ（トランスミッタ）
このビットは、FMPI2C_TXDR レジスタが空のときに、ハードウェアによってセットされます。次の送信データが FMPI2C_TXDR レジスタに書き込まれたときにクリアされます。
このビットは、ソフトウェアによって 1 を書き込んで、送信データレジスタ FMPI2C_TXDR をフラッシュできます。
注： このビットは、PE=0 のとき、ハードウェアによってセットされます。

23.7.8 割り込みクリアレジスタ（FMPI2C_ICR）

アドレスオフセット : 0x1C
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERT CF	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

- ビット 31:14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **ALERTCF** : アラートフラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの ALERT フラグがクリアされます。
注： SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。
- ビット 12 **TIMOUTCF** : タイムアウト検出フラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの TIMEOUT フラグがクリアされます。
注： SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。
- ビット 11 **PECCF** : PEC エラーフラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの PECERR フラグがクリアされます。
注： SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。
- ビット 10 **OVRCF** : オーバーラン／アンダーランフラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの OVR フラグがクリアされます。

- ビット 9 **ARLOCF** : アービトレーション喪失フラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの ARLO フラグがクリアされます。
- ビット 8 **BERRCF** : バスエラーフラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの BERRF フラグがクリアされます。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **STOPCF** : STOP 検出フラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの STOPF フラグがクリアされます。
- ビット 4 **NACKCF** : 非確認応答フラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの ACKF フラグがクリアされます。
- ビット 3 **ADDRCF** : アドレス一致フラグクリア
このビットに 1 を書き込むと、FMPI2C_ISR レジスタの ADDR フラグがクリアされます。このビットに 1 を書き込むと、FMPI2C_CR2 レジスタの START ビットもクリアされます。
- ビット 2:0 予約済みであり、リセット値に保持する必要があります。

23.7.9 PEC レジスタ (FMPI2C_PECR)

アドレスオフセット : 0x20
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

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

- ビット 31:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7:0 **PEC[7:0]** パケットエラーチェックレジスタ
PECEN=1 のとき、このフィールドは内部 PEC を含みます。
PEC は、PE=0 のとき、ハードウェアによってクリアされます。

注 : **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 23.3 : FMPI2C の実装を参照してください。**

23.7.10 受信データレジスタ（FMPI2C_RXDR）

アドレスオフセット：0x24
リセット値：0x0000 0000
アクセス：ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。
ビット 7:0 **RXDATA[7:0]** 8 ビット受信データ
I²C バスから受信したデータバイト。

23.7.11 送信データレジスタ（FMPI2C_TXDR）

アドレスオフセット：0x28
リセット値：0x0000 0000
アクセス：ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。
ビット 7:0 **TXDATA[7:0]** 8 ビット送信データ
I²C バスに送信されるデータバイト。
注： これらのビットは、TXE=1 のときのみ書き込むことができます。

23.7.12 FMPI2C レジスタマップ

次の表に、FMPI2C のレジスタマップとリセット値を示します。

表 121. FMPI2C レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	FMPI2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	Res.	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	リセット値									0	0	0	0	0		0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x4	FMPI2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]										
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	FMPI2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]										
	リセット値																0					0	0	0	0	0	0	0	0	0	0	0	
0xC	FMPI2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]				Res.						
	リセット値																0					0	0	0	0	0	0	0	0	0	0		
0x10	FMPI2C_TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			SCLH[7:0]					SCLL[7:0]											
	リセット値	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	FMPI2C_TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]										TIMEOUTEN	Res.	TIDLE	TIMEOUTA[11:0]														
	リセット値	0				0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	FMPI2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]						DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDRF	RXNE	TXIS	TXE	
	リセット値									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	FMPI2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMEOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.	
	リセット値																		0	0	0	0	0	0			0	0	0				
0x20	FMPI2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]											
	リセット値																						0	0	0	0	0	0	0	0	0	0	
0x24	FMPI2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]												
	リセット値																						0	0	0	0	0	0	0	0	0	0	

表 121. FMPI2C レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x28	FMPI2C_TXDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXDATA[7:0]													
	リセット値																									0	0	0	0	0	0	0	0					

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

24 I²C (Inter-integrated circuit) インタフェース

24.1 I²C の概要

I²C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I²C バス間のインタフェースとして機能します。マルチマスタ機能を備え、すべての I²C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準（最大 100 kHz）モードと高速（最大 400 kHz）モードをサポートしています。

また、SMBus2.0 とも互換性を持っています。このインタフェースは、CRC の生成と検証、SMBus (System Management Bus)、PMBus (PowerManagement Bus) など、さまざまな目的に使用できます。

デバイスの実装によっては、CPU の負荷を軽減するために DMA 機能を使用できます。

24.2 I²C の主な特長

- パラレルバス/I²C プロトコル間のコンバータ
- マルチマスタ機能：同じインタフェースがマスタまたはスレーブとして機能
- I²C マスタの機能：
 - － クロック生成
 - － START 信号、STOP 信号の生成
- I²C スレーブの機能：
 - － プログラム可能な I²C アドレス検出
 - － 2 つのスレーブアドレスを確認応答するデュアルアドレス機能
 - － STOP ビット検出
- 7 ビット/10 ビットアドレッシングおよび同報 (General Call) の生成と検出
- 複数の通信速度をサポート：
 - － 標準モード (最大 100 kHz)
 - － 高速モード (最大 400 kHz)
- アナログノイズフィルタ
- プログラム可能なデジタルノイズフィルタ
- ステータスフラグ：
 - － トランスマッタ/レシーバモードフラグ
 - － 最終バイト送信フラグ
 - － I²C ビジーフラグ
- エラーフラグ：
 - － マスタモードのアービトレーションロスト条件
 - － アドレス/データ送信後の確認応答障害
 - － START/STOP コンディションの誤配置検出
 - － クロックストレッチが無効な場合のオーバーラン/アンダーラン
- 2 つの割り込みベクタ：
 - － 正常なアドレス/データ通信用の割り込み
 - － エラー条件用の割り込み
- クロックストレッチオプション
- DMA 機能付きの 1 バイトバッファ
- 設定可能な PEC (Packet Error Checking) の生成/検証：
 - － Tx モードでは PEC 値を最終バイトとして送信可能
 - － 最後の受信バイトの PEC エラーチェック
- SMBus 2.0 互換性：
 - － クロックロータイムアウト遅延：25 ms
 - － マスタのクロックロー累積延長時間：10 ms
 - － スレーブのクロックロー累積延長時間：25 ms
 - － ハードウェアによる PEC 生成/検証 (ACK 制御付き)
 - － アドレス解決プロトコル (ARP) サポート
- PMBus 互換性

注：製品によっては、上記の機能の一部を使用できないことがあります。搭載されている I²C インタフェースによってサポートされる特定の機能については、製品のデータシートを参照してください。

24.3 I²C の機能説明

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I²C バスに接続されます。標準（最大 100 kHz）または高速（最大 400 kHz）の I²C バスと接続できます。

24.3.1 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成した後はスレーブからマスタに、アービトレーションロスまたは STOP 生成が生じた場合はマスタからスレーブに、それぞれ自動的に切り替わることで、マルチマスタ機能を可能にします。

通信の流れ

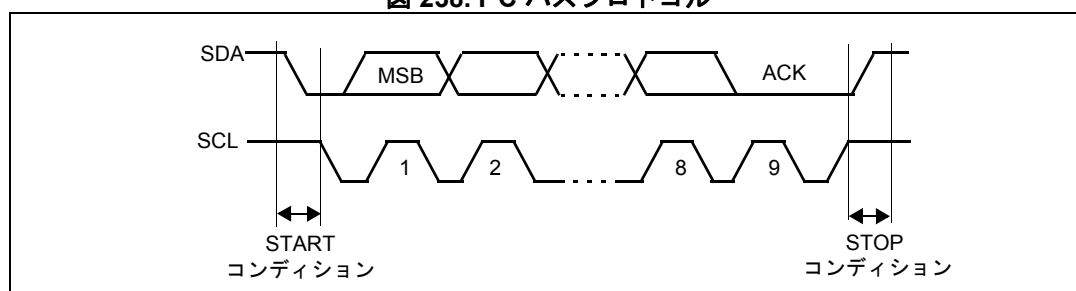
マスタモードでは、I²C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで始まり、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス（7 または 10 ビット）と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効／無効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト（7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト）にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。図 238 を参照してください。

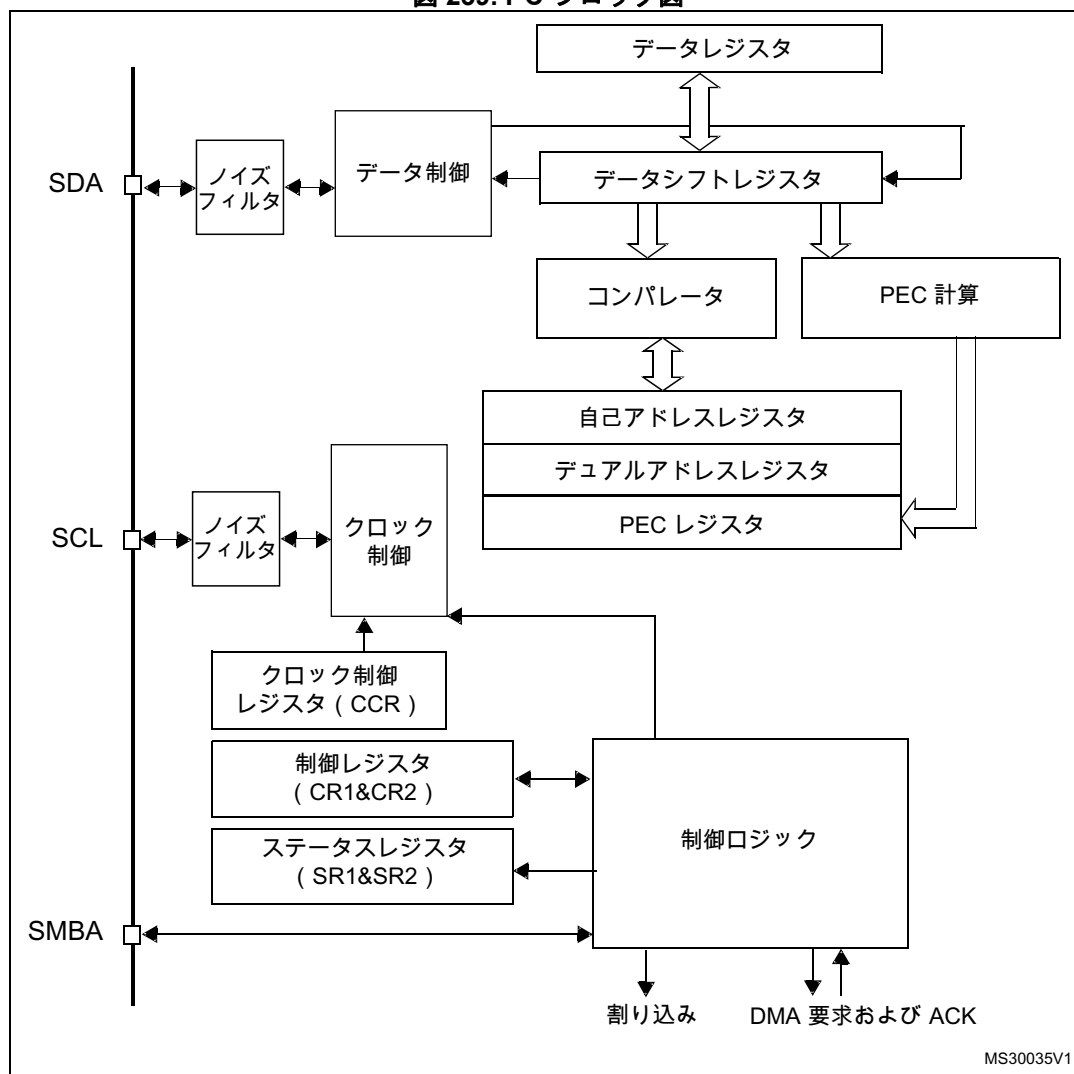
図 238. I²C バスプロトコル



確認応答は、ソフトウェアによって有効／無効にできます。I²C インタフェースアドレス（デュアルアドレッシングの 7 ビット／10 ビットや同報アドレス）は、ソフトウェアによって選択できます。

I²C インタフェースのブロック図を 図 239 に示します。

図 239. I²C ブロック図



1. SMBA は、SMBus モードでのオプション信号です。SMBus が無効な場合、この信号は適用されません。

24.3.2 I²C スレーブモード

デフォルトでは、I²C インタフェースはスレーブモードで動作します。デフォルトのスレーブモードからマスタモードに切り替えるには、START コンディションの生成が必要です。

正しいタイミングを生成するには、ペリフェラル入カクロックを I2C_CR2 レジスタにプログラミングする必要があります。ペリフェラル入カクロックは、少なくとも次の周波数以上でなければなりません。

- 標準モード : 2 MHz
- 高速モード : 4 MHz

START コンディションが検出されると、アドレスは、SDA ラインから受信されて、シフトレジスタに送信されます。その後、このアドレスはインタフェースのアドレス (OAR1) および OAR2 (ENDUAL=1 の場合) または同報アドレス (ENGCG = 1 の場合) と比較されます。

注： 10 ビットアドレスモードでは、この比較にはヘッダシーケンス (11110xx0) が含まれます。ここで、xx はアドレスの上位 2 ビットを示します。

ヘッダまたはアドレスの不一致：インタフェースはこれを見逃し、別の START コンディションを待ちます。

ヘッダ一致 (10 ビットモードのみ)：インタフェースは、ACK ビットがセットされている場合は確認応答パルスを生成し、8 ビットのスレーブアドレスを待ちます。

アドレス一致：インタフェースは以下のイベントを順に生成します。

- ACK ビットがセットされている場合、確認応答パルスを生成します。
- ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割り込みが生成されます。
- ENDUAL=1 の場合、ソフトウェアは、DUALF ビットを読み出して、どのスレーブアドレスが確認応答されたかをチェックする必要があります。

10 ビットモードでは、アドレスシーケンスの受信後、スレーブは常にレシーバモードになります。スレーブがトランスミッタモードに入るのは、反復 START コンディションの後、一致するアドレスビットを持ち、最下位ビットがセットされた (11110xx1) ヘッダシーケンスを受信したときです。

TRA ビットは、スレーブがレシーバモードかトランスミッタモードかを示します。

スレーブトランスミッタ

アドレス受信に続いて、ADDR をクリアした後、スレーブは、内部シフトレジスタを介して、DR レジスタから SDA ラインにバイトを送信します。

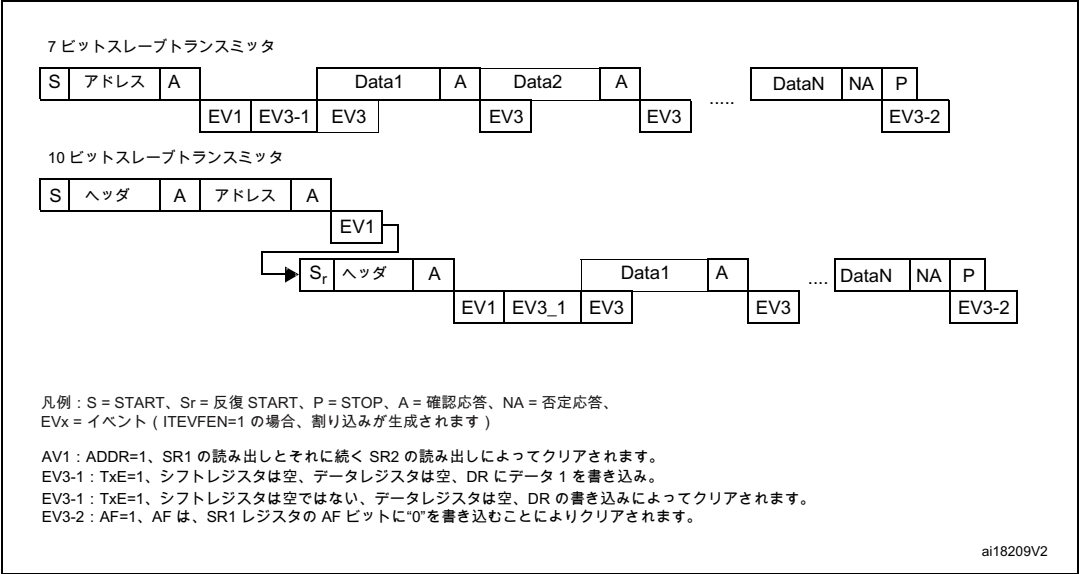
スレーブは、ADDR がクリアされ、DR に送信データが書き込まれるまで、SCL のローレベルを伸長します (図 240 転送シーケンスの EV1 EV3 を参照)。

確認応答パルスが受信されると：

- TxE ビットがハードウェアによってセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます。

TxE がセットされ、次のデータ送信の終わりまでに I2C_DR レジスタに一部のデータが書き込まれていない場合、BTF ビットがセットされ、インタフェースは、I2C_SR1 の読み出しと、それに続く I2C_DR レジスタへの書き込みによって BTF がクリアされるまで、SCL のローレベルを伸長して待ちます。

図 240. スレーブトランスミッタの転送シーケンス



1. EV1 イベントおよび EV3-1 イベントは、対応するソフトウェアシーケンスの終わりまで SCL のローレベルを伸長します。
2. 次のバイト送信の終わりにソフトウェアシーケンスが完了しない場合、EV3 イベントは SCL のローレベルを伸長します。

スレーブレシーバ

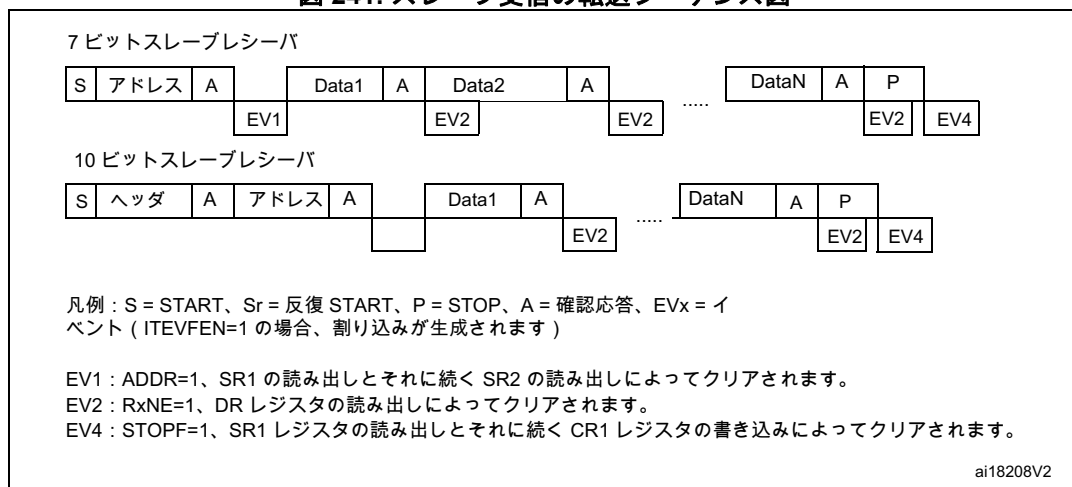
アドレス受信に続いて、ADDR がクリアされた後、スレーブは、SDA ラインからのバイトを内部シフトレジスタを介して DR レジスタに受信します。各バイトの後、インタフェースは以下のイベントを順に生成します。

- ACK ビットがセットされている場合、確認応答パルスを生成します。
- RxNE ビットがハードウェアによってセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は、割り込みが生成されます。

RxNE がセットされ、次のデータ受信の終わりに DR レジスタ内のデータが読み出されなかった場合、BTF ビットがセットされます。さらにインタフェースは、I2C_DRの読み出しによって BTF がクリアされるまで、SCL のローレベルを伸長して待ちます (図 241 転送シーケンスを参照)。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

図 241. スレーブ受信の転送シーケンス図



- EV1 イベントは、対応するソフトウェアシーケンスの終わりまで SCL のローレベルを伸長します。
- 次のバイト受信の終わりまでにソフトウェアシーケンスが完了しない場合、EV2 イベントは SCL のローレベルを延長します。
- SR1 レジスタの状態をチェックした後、ユーザは、セットされているフラグごとに、完全なクリアシーケンスを実行してください。
 すなわち、ADDR フラグと STOPF フラグに対し、I2C 割り込みルーチンの中に次のシーケンスを記述することが必要です。

```

READ SR1
if (ADDR == 1) {READ SR1; READ SR2}
if (STOPF == 1) {READ SR1; WRITE CR1}
        
```

 これは、ADDR フラグと STOPF フラグがセットされていることが分かった場合、両方とも確実にクリアするための処理です。

スレーブ通信の終了

最後のデータバイトが転送された後、マスタは STOP コンディションを生成します。インタフェースは、STOP コンディションを検出して、次の動作を行います。

- STOPF ビットをセットし、ITEVFEN ビットがセットされている場合は割り込みを生成します。

STOPF は、SR1 レジスタの読み出しと、それに続く CR1 レジスタへの書き込みによってクリアされます (図 241 : スレーブ受信の転送シーケンス図 EV4 を参照)。

24.3.3 I²C マスタモード

マスタモードでは、I²C インタフェースは、データ転送を開始し、クロック信号を生成します。データ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START ビットによってバス上に START コンディションが生成されると、マスタモードが選択されます。

次に、マスタモードに必要なシーケンスを示します。

- 正しいタイミングを生成するため、I2C_CR2 レジスタにペリフェラル入カクロックを設定します。
- クロック制御レジスタを設定します。
- 立ち上がり時間レジスタを設定します。
- I2C_CR1 レジスタを設定して、ペリフェラルを有効にします。
- I2C_CR1 レジスタの START ビットをセットして、START コンディションを生成します。

ペリフェラル入カクロックは、少なくとも次の周波数以上でなければなりません。

- 標準モード : 2 MHz
- 高速モード : 4 MHz

SCL マスタクロック生成

CCR ビットは、立ち上がりエッジおよび立ち下がりエッジの生成（それぞれ）を始め、ハイおよびローレベルの SCL クロックの生成に使用します。スレーブは SCL ラインをストレッチする場合がありますため、ペリフェラルは立ち上がりエッジ生成後、TRISE ビットにプログラムされた時間の最後にバスからの SCL 入力をチェックします。

- SCL ラインがローの場合、スレーブがバスをストレッチしており、ハイレベルカウンタが SCL ラインがハイで検出されるまで停止することを示します。これにより、SCL クロックパラメータの最小ハイ周期を保証できます。
- SCL ラインがハイの場合、ハイレベルカウンタはカウントを継続します。

スレーブがクロックをストレッチしていなくても、ペリフェラルによる SCL 立ち上がりエッジ生成からペリフェラルによる SCL 立ち上がりエッジ検出までのフィードバックループには時間がかかります。このループバック時間は、SCL 立ち上がり時間（SCL VIH 入力検出に影響）、SCL 入力パスにあるノイズフィルタによる遅延、および APB クロックとの内部 SCL 入力同期による遅延と関連します。フィードバックループで使用する最大時間は、TRISE ビットでプログラムされているため、SCL 周波数は SCL 立ち上がり時間にかかわらず安定した状態になります。

START コンディション

BUSY ビットがクリアされているときに START ビットをセットすると、インタフェースは、START コンディションを生成し、マスタモードに切り替わります（MSL ビットをセット）。

注： マスタモードで START ビットをセットすると、インタフェースは、現在のバイト転送の最後に RESTART コンディションを生成します。

START コンディションが送信されると、以下のイベントが発生します。

- SB ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割り込みが生成されます。

次に、マスタは、SR1 レジスタの読み出しと、それに続く DR レジスタへのスレーブアドレスの書き込みを待ちます（図 242 と図 243 の転送シーケンスの EV5 を参照）。

スレーブアドレス送信

次に、スレーブアドレスが内部シフトレジスタを介して SDA ラインに送信されます。

- 10 ビットアドレスモードでは、ヘッダシーケンスの送信によって、次のイベントが発生します。
 - ADD10 ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割り込みが生成されます。

次に、マスタは、SR1 レジスタの読み出しと、それに続く DR レジスタへの 2 番目のアドレスバイトの書き込みを待ちます（図 242 と図 243 の転送シーケンスを参照）。

- ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割り込みが生成されます。

次にマスタは SR1 レジスタの読み出しと、それに続く SR2 レジスタの読み出しを待ちます（図 242 と図 243 転送シーケンスを参照）。

- 7 ビットアドレスモードでは、1 個のアドレスバイトが送信されます。

アドレスバイトが送信されると、以下のイベントが発生します。

 - ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割り込みが生成されます。

次にマスタは SR1 レジスタの読み出しと、それに続く SR2 レジスタの読み出しを待ちます（図 242 と図 243 転送シーケンスを参照）。

マスタは、送信されたスレーブアドレスの LSB に応じて、トランスミッタモードに入るか、レシーバモードに入るかを決定できます。

- 7 ビットアドレスモードでは、
 - トランスミッタモードに入るには、マスタは、LSB をリセットしたスレーブアドレスを送信します。
 - レシーバモードに入るには、マスタは、LSB をセットしたスレーブアドレスを送信します。
- 10 ビットアドレスモードでは、
 - トランスミッタモードに入るには、マスタは、ヘッダ (11110xx0) を送信してから、スレーブアドレスを送信します (xx はアドレスの上位 2 ビットです)。
 - レシーバモードに入るには、マスタは、ヘッダ (11110xx0) を送信してから、スレーブアドレスを送信します。さらに、反復 START コンディションと、それに続いてヘッダ (11110xx1) を送信します (xx はアドレスの上位 2 ビットです)。

TRA ビットは、マスタがレシーバモードにあるか、トランスミッタモードにあるかを示します。

マスタトランスミッタ

アドレス送信に続いて、ADDR をクリアした後、マスタは、内部シフトレジスタを介して、DR レジスタから SDA ラインにバイトを送信します。

マスタは、最初のデータバイトが I2C_DR に書き込まれるまで待ちます (図 242 転送シーケンスの EV8_1 を参照)。

ACK (アクノリッジパルス) を受信すると、TxE ビットがハードウェアによってセットされ、ITEVFEN ビットおよび ITBUFEN ビットがセットされている場合は割り込みが生成されます。

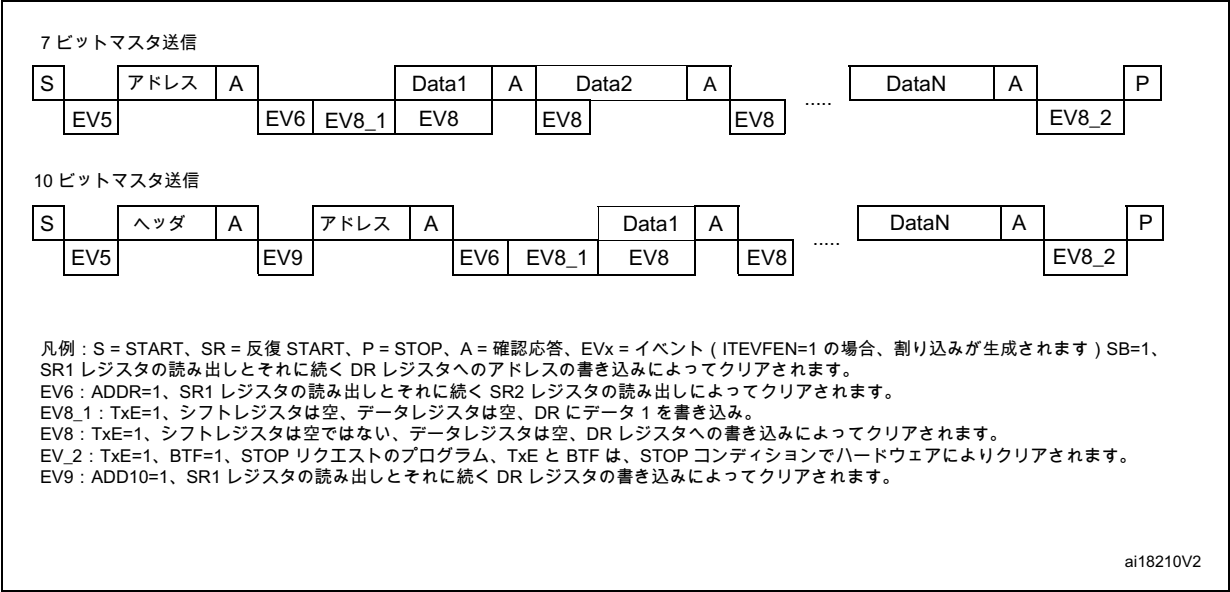
TxE がセットされていて、最後のデータ送信が終わるまでにデータバイトが DR レジスタに書き込まれなかった場合、BTF がセットされ、インタフェースは I2C_DR への書き込みによって BTF がクリアされるまで、SCL のローレベルを延長してウェイトします。

通信の終了

最後のバイトが DR レジスタに書き込まれた後、STOP ビットがソフトウェアによってセットされて STOP コンディションを生成します (図 242 の転送シーケンスの EV8_2 を参照)。インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。

注： **STOP コンディションは、TxE または BTF ビットがセットされたとき、EV8_2 イベント時にプログラミングしてください。**

図 242. マスタ送信の転送シーケンス図



1.

EV5、EV6、EV9、EV8_1、および EV8_2 の各イベントは、対応するソフトウェアシーケンスの終わりまで、SCL のローレベルを延長します。
2.

次のバイト送信の終わりまでにシーケンスが完了しない場合、EV8 イベントは SCL のローレベルを延長します。

マスタレシーバ

アドレス送信に続いて ADDR がクリアされると、I²C インタフェースはマスタレシーバモードに入ります。このモードでは、インタフェースは、SDA ラインから内部シフトレジスタを介して DR レジスタにバイトを受信します。各バイトの後、インタフェースは以下のイベントを順に生成します。

1. ACK ビットがセットされている場合、確認応答パルスを生成します。
2. RxNE ビットがセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます (図 243 転送シーケンスの EV7 を参照)。

RxNE ビットがセットされ、最後のデータ受信の終わりまでに DR レジスタ内のデータが読み出されなかった場合、BTF ビットがハードウェアによってセットされ、インタフェースは、DR レジスタの読み出しによって BTF がクリアされるまで、SCL のローレベルを延長して待ちます。

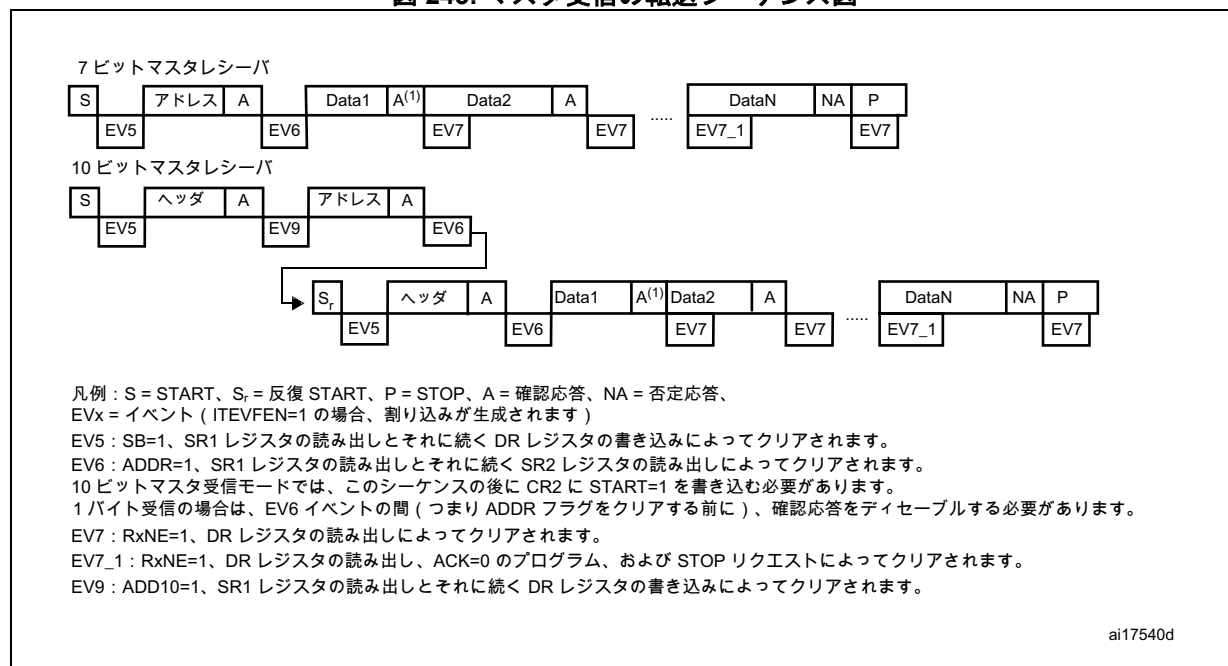
通信の終了

マスタは、スレーブから受信した最終バイトに対して NACK を送信します。この NACK を受信したスレーブは、SCL ラインと SDA ラインの制御を解除します。これによって、マスタは STOP/RESTART コンディションを送信できます。

1. 最後の受信データバイトの後で NACK パルスを生成するには、最後から 2 番目の RxNE イベントの後で最後から 2 番目のデータバイトの読み出し直後に、ACK ビットをクリアする必要があります。
2. STOP/RESTART コンディションを生成するには、ソフトウェアは、最後から 2 番目の RxNE イベントの後で最後から 2 番目のデータバイトの読み出し後に、STOP/START ビットをセットする必要があります。
3. 1 バイトを受信する必要がある場合は、EV6 の間 (ADDR フラグがクリアされる前) に ACK (アクノリッジ) が無効化され、EV6 の後に STOP コンディションが生成されます。

STOP コンディションの生成後、インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。

図 243. マスタ受信の転送シーケンス図



1. 1 バイトが受信された場合、無効となります。
2. EV5、EV6、および EV9 イベントは、対応するソフトウェアシーケンスの終わりまで、SCL のローレベルを延長します。
3. 次のバイト受信の終わりまでにソフトウェアシーケンスが完了しない場合、EV7 イベントは SCL のローレベルを延長します。
4. EV7 1 ソフトウェアシーケンスは、現在のバイト転送の ACK パルスより前に完了する必要があります。

現在のバイト転送の ACK パルスより前に EV7-1 ソフトウェアシーケンスが完了しない場合、以下の手順を推奨します。

以下の手順に従う必要があります。

- 最後のデータ受信が終わる前に、ACK ビットが遅れずにローにセットされる
- 最後のデータ受信の後、補助データの受信なしに STOP ビットがハイにセットされる。

2 バイト受信の場合:

- ADDR = 1 になるまで待ちます (ADDR フラグがクリアされるまで SCL のローレベルを伸長します)。
- ACK をローにセットし、POS をハイにセットします。
- ADDR フラグをクリアします。
- BTF = 1 になるまで待ちます (DR にデータ 1、シフトレジスタにデータ 2、データ 1 が読み出されるまで SCL のローレベルを伸長)。
- STOP をハイにセットします。
- データ 1 とデータ 2 を読み出します。

N > 2 バイト受信の場合 (N-2 データ受信から)

- BTF = 1 になるまで待ちます (DR にデータ N-2、シフトレジスタにデータ N-1、データ N-2 が読み出されるまで SCL のローレベルを伸長)。
- ACK をローにセットします。
- データ N-2 を読み出します。
- BTF = 1 になるまで待ちます (DR にデータ N-1、シフトレジスタにデータ N、データ N-1 が読み出されるまで SCL のローレベルを伸長)。
- STOP をハイにセットします。
- データ N-1 とデータ N を読み出します。

24.3.4 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

バスエラー (BERR)

このエラーは、I²C インタフェースがアドレスやデータの転送中に外部の STOP または START コンディションを検出したときに発生します。このとき、以下のイベントが発生します。

- BERR ビットがセットされ、ITERREN ビットがセットされている場合は割り込みが生成されます。
- スレーブモード：データは破棄され、バスはハードウェアによって解放されます。
 - START の誤配置である場合、スレーブはこれを RESTART とみなし、アドレスまたは STOP コンディションを待ちます。
 - STOP の誤配置である場合、スレーブは STOP コンディションの場合のように動作し、バスはハードウェアによって解放されます。
- マスタモード：バスは解放されず、現在の送信の状態は影響を受けません。現在の送信を中止するかどうかは、ソフトウェアに依存します。

確認応答障害 (AF)

このエラーは、インタフェースが NACK ビットを検出したときに発生します。このとき、

- AF ビットがセットされ、ITERREN ビットがセットされている場合は割り込みが生成されます。
- NACK を受信したトランスミッタは、通信をリセットする必要があります。
 - スレーブの場合：バスはハードウェアによって解放されます。
 - マスタの場合：ソフトウェアが STOP または反復 START コンディションを生成する必要があります。

アービトレーション喪失 (ARLO)

このエラーは、I²C インタフェースがアービトレーションロスト状態を検出したときに発生します。このとき、

- ARLO ビットがハードウェアによってセットされます (ITERREN ビットがセットされている場合、割り込みが生成されます)。
- I²C インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。I²C がアービトレーションを失うと、同じ転送ではそのスレーブアドレスを確認応答できませんが、通信を許可されたマスタからの反復 START 後に確認応答できます。
- バスはハードウェアによって解放されます。

オーバーラン／アンダーランエラー（OVR）

スレーブモードでオーバーランエラーが発生するのは、クロック伸長が無効であり、I²C インタフェースがデータを受信しているときです。インタフェースがすでにバイトを受信し（RxNE=1）、インタフェースが次のバイトを受信するまでに DR 内のデータがまだ読み出されていません。このとき、

- 最後に受信されたバイトは失われます。
- オーバーランエラーが発生した場合、ソフトウェアで RxNE ビットをクリアし、トランスミッターで最後の受信バイトを再送信してください。

スレーブモードでアンダーランエラーが発生するのは、クロック伸長が無効であり、I²C インタフェースがデータを送信しているときです。次のバイト送信用のクロックが現れるまでに、次に送信すべきデータがまだ DR レジスタに格納されていません（TxNE=1）。このとき、

- DR レジスタ内の同じバイトが再送信されます。
- ユーザは、アンダーランエラー中にレシーバ側で受信されたデータが捨てられ、I²C バス規格で指定されたクロックロー時間内に次のバイトが書き込まれるようにしてください。

送信される最初のバイトに関しては、ADDR がクリアされてから最初の SCL 立ち上がりエッジまでに DR を書き込む必要があります。さもなければ、レシーバは最初のデータを破棄しなければなりません。

24.3.5 プログラム可能なノイズフィルタ

高速 モードでは、I²C 標準により、SDA ラインおよび SCL ライン上のスパイク幅を 50 ns まで短くすることが求められています。

SDA および SCL I/O には、アナログノイズフィルタが内蔵されています。このフィルタはデフォルトで有効化されており、I2C_FLTR レジスタの ANOFF ビットをセットして無効化することができます。

DNF[3:0] ビットをゼロ以外の値に設定することにより、デジタルノイズフィルタを有効化することができます。このフィルタは、SDA 入力および SCL 入力のスパイク幅を DNF[3:0] * T_{PCLK1} まで抑えることができます。

デジタルノイズフィルタを有効化すると、SDA ホールド時間が（DNF[3:0] + 1）* T_{PCLK} だけ長くなります。

I²C-バス仕様バージョン 2.1（Thd:dat）の最大ホールド時間を満たすためには、アナログフィルタを無効化し、表 122 に示す制約に従って DNF ビットをプログラムする必要があります。

注： DNF[3:0] は、必ず I²C が無効化されているとき（PE = 0）に設定する必要があります。アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

表 122. Thd:dat（max）に準拠した最大 DNF[3:0] 値

PCLK1 周波数	最大 DNF 値	
	標準モード	高速 モード
2 <= F _{PCLK1} <= 5	2	0
5 < F _{PCLK1} <= 10	12	0
10 < F _{PCLK1} <= 20	15	1
20 < F _{PCLK1} <= 30	15	7
30 < F _{PCLK1} <= 40	15	13
40 < F _{PCLK1} <= 50	15	15

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

注： 各周波数範囲に対し、この制約は範囲の最小周波数である最悪ケースに基づいて決められています。システムが最大ホールド時間を超えた対応が可能な場合は、DNF 値を大きくすることができます。

24.3.6 SDA/SCL ライン制御

- クロック伸長が有効な場合：
 - － トランスマッターモード：TxNE=1 および BTF=1 の場合、インタフェースは、送信の前にクロックラインをローレベルに保ち、マイクロコントローラがバイトをデータレジスタに書き込むのを待ちます（パツファとシフトレジスタは、いずれも空）。
 - － レシーバモード：RxNE=1 および BTF=1 の場合、インタフェースは、受信の後にクロックラインをローレベルに保ち、マイクロコントローラがバイトをデータレジスタに読み出すのを待ちます（パツファとシフトレジスタは、いずれも満杯）。
- スレーブモードでクロック伸長が無効な場合：
 - － オーバーランエラー：RxNE=1 で、次のバイトが受信されるまでに DR が読み出されていない場合。最後に受信されたバイトは失われます。
 - － アンダーランエラー：TxNE=1 で、次のバイトを送信するまでに DR への書き込みが行われていない場合。同じバイトが再送信されます。
 - － 書き込み衝突は管理されません。

24.3.7 SMBus

概要

システム管理バス（SMBus）は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I²C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。システムが SMBus を使用することで、個々の制御ラインをトグルすることなく、デバイスとの間でメッセージをやり取りできます。

システム管理バス仕様では、3 種類のデバイスを規定しています。スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ／スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

SMBus と I²C の類似点

- 2 線バスプロトコル（1 クロック、1 データ）+ SMBus アラートライン（オプション）
- マスタ／スレーブ通信（マスタがクロックを供給します）
- マルチマスタ機能
- I²C 7 ビットアドレスフォーマットに類似した SMBus データフォーマット（[図 238](#)）

SMBus と I²C の相違点

次の表に、SMBus と I²C の相違点を示します。

表 123. SMBus と I²C

SMBus	I ² C
最大速度：100 kHz	最大速度：400 kHz
最小クロック速度：10 kHz	最小クロック速度の規定なし



表 123. SMBus と I²C (続き)

SMBus	I ² C
クロックロータイムアウト : 35 ms	タイムアウトなし
論理レベルは固定	ロジックレベルはV _{DD} に依存
種々のアドレスタイプ (予約、ダイナミック、など)	7 ビット、10 ビット、および同報のスレーブアドレスタイプ
種々のバスプロトコル (クイックコマンド、プロセスコール、など)	バスプロトコルなし

SMBus アプリケーションの用途

SMBus を使用すれば、デバイスは、製造者情報を提供し、モデル／部品番号をシステムに通知し、サスペンドイベントに備えてその状態を保存し、種々のエラーを報告し、制御パラメータを受け付け、そのステータスを返すことができます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

デバイス識別

SMBus 上にスレーブとして存在するデバイスには、スレーブアドレスと呼ばれるユニークなアドレスがあります。予約済みのスレーブアドレスの一覧については、SMBus 仕様 ver.2.0 (<http://smbus.org/specs/>) を参照してください。

バスプロトコル

SMBus 仕様では、最大 9 種のプロトコルをサポートします。これらのプロトコルと SMBus アドレスタイプの詳細については、SMBus 仕様 ver.2.0 を参照してください。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス解決プロトコル (ARP) には、以下の属性があります。

- アドレス割り当てには、標準の SMBus 物理層アービトレーション機構を使用します。
- 割り当てられたアドレスは、デバイスに電源が供給されている間は一定しています。デバイスの電源切断後もアドレスの保持が可能です。
- アドレス割り当ての後、新たな SMBus パケットのオーバーヘッドは生じません。(つまり、割り当てられたスレーブアドレスへのその後のアクセスでは、固定アドレスのデバイスへのアクセスと同じオーバーヘッドになります。)
- いずれの SMBus マスタも、バスの番号付けが可能です。

ユニークデバイス識別子 (UDID)

アドレス割り当てを目的として各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。

128 ビット UDID と ARP の詳細については、SMBus 仕様 ver. 2.0 を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

SMBus アラートモード

SMBus アラートは、マスタ機能をピンに置き換えたいデバイス用の割り込みラインを持つオプション信号です。SMBA は、SCL 信号や SDA 信号と同じく、ワイヤード AND 信号です。SMBA は、SMBus 同報アドレスとともに使用されます。SMBus で呼び出されるメッセージの長さは 2 バイトです。

スレーブ専用デバイスは、I2C_CR1 レジスタの ALERT ビットをセットすることによって、ホストに接続したいことを SMBA を通じて通知できます。ホストは、割り込みを処理し、アラート応答アドレス (ARA と呼ばれ、値は 0001 100X) を通じて全 SMBA デバイスに同時にアクセスします。SMBA をローレベルに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。このステータスは、I2C_SR1 レジスタの SMBALERT ステータスフラグを使用して識別されます。ホストは、モディファイド受信バイト動作を行います。スレーブ送信デバイスによって提供された 7 ビットのデバイスアドレスは、バイトの上位 7 ビットに置かれます。8 番目のビットは 0 または 1 にできます。

複数のデバイスが SMBA をローレベルに引き下げている場合、スレーブアドレス転送時の標準的なアービトラーションによって、最高優先順位 (最小アドレス) のデバイスが通信権を獲得します。スレーブアドレスの確認応答後、デバイスは、その SMBA のプルダウンを解放する必要があります。メッセージ転送の完了時に、SMBA が依然としてローレベルであることにホストが気づいた場合、再び ARA の読み出しが行われます。

SMBA 信号を実装していないホストは、周期的に ARA にアクセスすることができます。

SMBus アラートモードの詳細については、SMBus 仕様 ver. 2.0 (<http://smbus.org/specs/>) を参照してください。

タイムアウトエラー

I²C と SMBus では、タイミング仕様に違いがあります。

SMBus では、クロックロータイムアウトとして 35 ms の TIMEOUT が定義されています。さらに SMBus では、スレーブデバイスのクロックロー累積延長時間として TLOW: SEXT を規定しています。また、マスタデバイスのクロックロー累積延長時間として TLOW: MEXT も規定しています。これらのタイムアウトの詳細については、SMBus 仕様 ver. 2.0 を参照してください。

I2C_SR1 レジスタのステータスフラグ Timeout や Tlow Error は、この機能のステータスを示します。

SMBus モードでのインタフェースの使い方

I²C モードから SMBus モードに切り替えるには、次のシーケンスを実行してください。

- I2C_CR1 レジスタの SMBus ビットをセットします。
- アプリケーションの必要に応じて、I2C_CR1 レジスタの SMBTYPE ビットと ENARP ビットを設定します。

デバイスをマスタとして設定したい場合、[セクション 24.3.3 : I2C マスタモード](#)に記載されている START コンディションの生成手順に従ってください。あるいは、[セクション 24.3.2 : I2C スレーブモード](#)のシーケンスに従ってください。

アプリケーションは、ソフトウェアによってさまざまな SMBus プロトコルを制御する必要があります。

- ENARP=1 および SMBTYPE=0 の場合 : SMB デバイスのデフォルトアドレスが確認応答されます。
- ENARP=1 および SMBTYPE=1 の場合 : SMB ホストヘッダが確認応答されます。
- SMBALERT=1 の場合 : SMB アラート応答アドレスが確認応答されます。

24.3.8 DMA リクエスト

DMA リクエストは、データ転送用にのみ生成されます（有効な場合）。DMA リクエストが生成されるのは、送信中にデータレジスタが空になったとき、および受信中にデータレジスタが満杯になったときです。I²C データ 転送の前に DMA を初期化して有効化する必要があります。ADDR イベントの前に I2C_CR2 レジスタで DMAEN ビットをセットする必要があります。マスタモードまたはスレーブモードにおいてクロック伸長を有効化した場合、ADDR フラグをクリアする前に ADDR イベント中に DMAEN ビットをセットすることもできます。DMA リクエストは、現在のバイト転送の終わりにまでに処理される必要があります。対応する DMA ストリームにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT）信号を送信し、転送完了割り込みを生成します（有効な場合）。

- マスタトランスミッタ：EOT 割り込み後の割り込みルーチンでは、DMA リクエストを無効にし、BTF イベントの発生を待ってから、STOP コンディションをプログラミングします。
- マスタレシーバ
 - － 受信するバイト数が 2 以上であるとき、DMA コントローラは、最後から 2 番目のデータバイト（number_of_bytes - 1）に対応するハードウェア信号（EOT_1）を送信します。I2C_CR2 レジスタで LAST ビットがセットされた場合、I²C は、EOT_1 に続く次のバイトの後に、自動的に NACK を送信します。ユーザは、DMA 転送完了割り込みルーチンで STOP コンディションを生成できます（有効な場合）。
 - － 1 バイトを受信する必要がある場合：EV6 イベント中に NACK をプログラムする必要があります。すなわち、ADDR フラグをクリアする前に ADDR=1 のときに ACK=0 をプログラムします。すると、ADDR フラグのクリア後、または DMA 転送完了割り込みルーチン内で、STOP コンディションをプログラムすることができます。

DMA を使用した送信

DMA モードの送信を有効にするには、I2C_CR2 レジスタの DMAEN ビットをセットします。データは、TxE ビットがセットされるたびに、DMA ペリフェラルを使用して設定されたメモリ領域（DMA 仕様を参照）から I2C_DR レジスタにロードされます。I²C 送信に DMA ストリーム x（ここで x はストリーム番号）を割り当てるには、次のシーケンスを実行します：

1. I2C_DR レジスタのアドレスを DMA_SxPAR レジスタにセットします。データは、毎回の TxE イベント後に、メモリからこのアドレスに移動されます。
2. DMA_SxMA0R レジスタに（およびダブルバッファモードの場合は DMA_SxMA1R レジスタ）メモリアドレスを設定します。データは、毎回の TxE イベント後に、このメモリから I2C_DR にロードされます。
3. 転送されるバイトの総数を DMA_SxNDTR レジスタに設定します。毎回の TxE イベント後に、この値はデクリメントされます。
4. DMA_SxCR レジスタの PL[0:1] ビットを使用して、DMA ストリームの優先順位を設定します。
5. DMA_SxCR レジスタの DIR ビットをセットし、アプリケーション条件に応じて 1/2 転送終了後または転送完了後に、割り込みを設定します。
6. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

DMA コントローラのレジスタにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT/EOT_1）信号を送信し、DMA は DMA チャネル割り込みベクタ上に割り込みを生成します（有効な場合）。

注： DMA が送信に使用される場合、I2C_CR2 レジスタの ITBUFEN ビットを有効にしないでください。

DMA を使用した受信

DMA モードの受信を有効にするには、I2C_CR2 レジスタの DMAEN ビットをセットします。データは、データバイトが受信されるたびに、DMA ペリフェラルを使用して設定されたメモリ領域（DMA 仕様を参照）に I2C_DR レジスタからロードされます。I²C 受信に DMA ストリーム x（ここで x はストリーム番号）を割り当てるには、次のシーケンスを実行します：

1. I2C_DR レジスタのアドレスを DMA_SxPAR レジスタにセットします。データは、毎回の RxNE イベント後に、このアドレスからメモリに移動されます。
2. DMA_SxMA0R レジスタに（およびダブルバッファモードの場合は DMA_SxMA1R レジスタ）メモリアドレスを設定します。データは、毎回の RxNE イベント後に、I2C_DR レジスタからこのメモリ領域にロードされます。
3. 転送されるバイトの総数を DMA_SxNDTR レジスタに設定します。毎回の RxNE イベント後に、この値はデクリメントされます。
4. DMA_SxCR レジスタの PL[0:1] ビットを使用して、DMA ストリームの優先順位を設定します。
5. DMA_SxCR レジスタの DIR ビットをリセットし、アプリケーション条件に応じて 1/2 転送終了後または転送完了後に、割り込みを設定します。
6. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

DMA コントローラのレジスタにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT/EOT_1）信号を送信し、DMA は DMA チャネル割り込みベクタ上に割り込みを生成します（有効な場合）。

注： DMA が受信に使用される場合、I2C_CR2 レジスタの ITBUFEN ビットを有効にしないでください。

24.3.9 パケットエラーチェック

通信の信頼性を高めるため、PEC 計算回路が実装されています。PEC の計算には、 $C(x) = x^8 + x^2 + x + 1$ という CRC-8 多項式を各ビットに連続的に適用します。

- PEC 計算を有効にするには、I2C_CR1 レジスタの ENPEC ビットをセットします。PEC は、アドレスと R/W ビットを含むすべてのメッセージバイトに基づいて計算される CRC-8 コードです。
 - 送信時、最後のバイトに対応する TxE イベントの後、I2C_CR1 レジスタの PEC 転送ビットをセットします。PEC は、最後に送信されたバイトの後で転送されます。
 - 受信時、最後のバイトに対応する RxNE イベントの後、I2C_CR1 レジスタの PEC ビットをセットすることで、次に受信するバイトが内部的に計算した PEC と等しくない場合にレシーバが NACK を送信するようにします。マスタレシーバの場合、チェックの結果にかかわらず、PEC の後には NACK が続く必要があります。スレーブモードでは、CRC 受信の ACK の前に PEC をセットする必要があります。マスタモードでは、ACK がローにセットされたときにセットする必要があります。
- PECERR エラーフラグ／割り込みは、I2C_SR1 レジスタでも使用できます。
- DMA と PEC 計算機能がともに有効な場合：
 - 送信時、I²C インタフェースが DMA コントローラから EOT 信号を受信すると、最終バイトの後で自動的に PEC を送信します。
 - 受信時、I²C インタフェースが DMA コントローラから EOT_1 信号を受信すると、次のバイトを自動的に PEC とみなし、その検査を行います。PEC 受信後に DMA リクエストが生成されます。
- 中間での PEC 転送を可能にするため、I2C_CR2 レジスタには制御ビット（LAST ビット）が用意されており、本当に最後の DMA 転送かどうかを判定できます。マスタレシーバの最後の DMA リクエストである場合、最後に受信されたバイトの後で自動的に NACK が送信されます。
- PEC 計算は、アービトレーションロスによって損なわれます。



24.4 I²C 割り込み

次の表に、I²C 割り込みリクエストの一覧を示します。

表 124. I²C 割り込みリクエスト

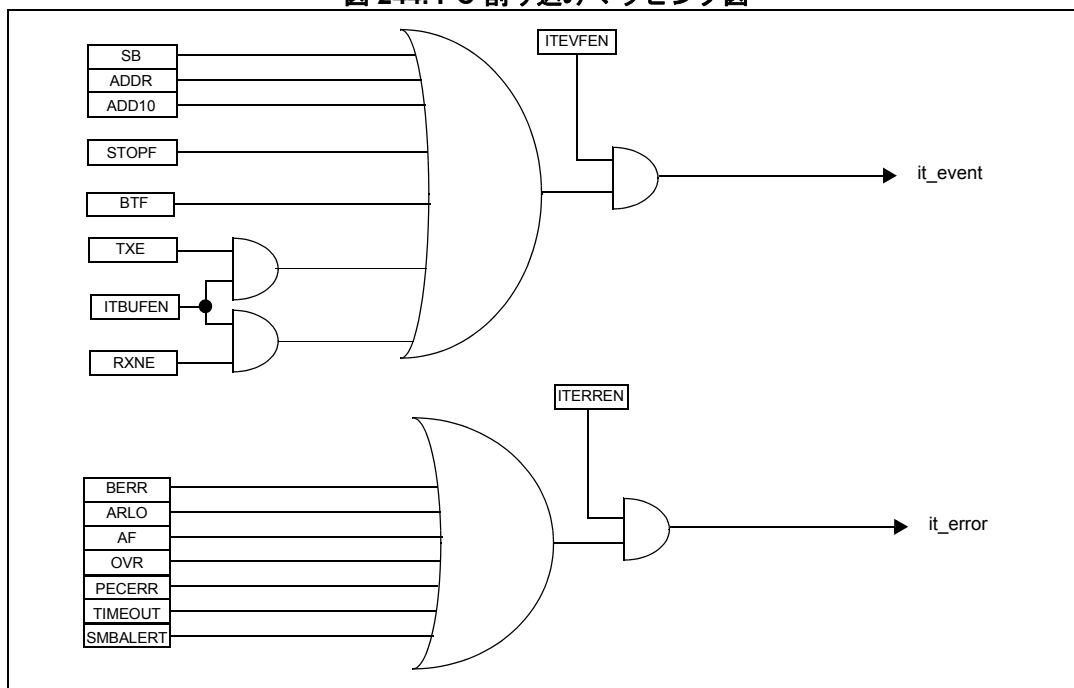
割り込みイベント	イベントフラグ	イネーブル制御ビット
START ビット送信 (マスタ)	SB	ITEVFEN
アドレス送信 (マスタ) またはアドレス一致 (スレーブ)	ADDR	
10 ビットヘッダ送信 (マスタ)	ADD10	
STOP 受信 (スレーブ)	STOPF	
データバイト転送終了	BTF	
受信バッファノットエンプティ	RXNE	ITEVFEN および ITBUFEN
送信バッファエンプティ	TXE	
バスエラー	BERR	ITERREN
アービトラションロスト (マスタ)	ARLO	
確認応答障害	AF	
オーバーラン/アンダーラン	OVR	
PEC エラー	PECERR	
タイムアウト/T lowエラー	TIMEOUT	
SMBus アラート	SMBALERT	

注：SB、ADDR、ADD10、STOPF、BTF、RxNE、およびTxEの各ビットは、同じ割り込みチャネル上で論理和がとられます。

BERR、ARLO、AF、OVR、PECERR、TIMEOUT、およびSMBALERTの各ビットは、同じ割り込みチャネル上で論理和がとられます。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

図 244. I²C 割り込みマッピング図



24.5 I²C デバッグモード

マイクロコントローラがデバッグモードに入ると（FPU 搭載 Cortex®-M4 コア 停止）、DBG モジュールの DBG_I2Cx_SMBUS_TIMEOUT 設定ビットに応じて、SMBUS タイムアウトは、通常の動作を続行するか、あるいは停止します。詳細については、[セクション 30.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

24.6 I²C レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#)を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

24.6.1 I²C 制御レジスタ 1 (I2C_CR1)

アドレスオフセット : 0x00
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW RST	Res.	ALERT	PEC	POS	ACK	STOP	START	NO STRET CH	ENG C	ENPEC	ENARP	SMB TYPE	Res.	SM BUS	PE
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

ビット 15 **SWRST** : ソフトウェアリセット

このビットがセットされると、I2C はリセット状態になります。このビットをリセットする前に、I2C ラインが解放されていて、バスが空いていることを確認してください。

0 : I²C ペリフェラルはリセット状態ではありません。

1 : I²C ペリフェラルはリセット状態です。

注 : このビットを使って、エラー状態またはロック状態の後にペリフェラルを再初期化することができます。例えば、BUSY ビットがセットされ、バス上のグリッチのためにロックされたままになった場合、SWRST ビットを使ってこの状態から抜け出すことができます。

ビット 14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13 **ALERT** : SMBus アラート

このビットは、ソフトウェアによってセット／クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : SMBA ピンをハイレベルに解放します。アラート応答アドレスヘッダの後には NACK が続きます。

1 : SMBA ピンをローレベルに駆動します。アラート応答アドレスヘッダの後には ACK が続きます。

ビット 12 **PEC** : パケットエラーチェック

このビットは、ソフトウェアによってセット／クリアされ、PEC が転送されたとき、START/STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

0 : PEC は転送されません。

1 : PEC が転送されます（送信または受信モード）。

注 : PEC 計算は、アービトレーションロスによって損なわれます。

ビット 11 POS : 確認応答/PEC 位置 (データ受信)

このビットは、ソフトウェアによってセット/クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : ACK ビットは、シフトレジスタに受信されている現在のバイトの (N)ACK を制御します。PEC ビットは、シフトレジスタ内の現在のバイトが PEC であることを示します。

1 : ACK ビットは、シフトレジスタに受信される次のバイトの (N)ACK を制御します。PEC ビットは、シフトレジスタ内の次のバイトが PEC であることを示します。

注: POS ビットは、マスタモードの 2 バイト受信設定でのみ使用する必要があります。また、マスタレシーバで推奨する 2 バイト受信手順に説明されているように、データ受信の開始前に設定する必要があります。

ビット 10 ACK : 確認応答イネーブル

このビットは、ソフトウェアによってセット/クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : 確認応答は返されません。

1 : バイトの受信後 (アドレスまたはデータの一致)、確認応答が返されます。

ビット 9 STOP : STOP 生成

このビットは、ソフトウェアによってセット/クリアされ、STOP コンディションの検出時にハードウェアによってクリアされ、タイムアウトエラーの検出時にハードウェアによってセットされます。

マスタモード :

0 : STOP 生成なし。

1 : 現在のバイト転送後、または現在の START コンディションの送信後に、STOP が生成されます。

スレーブモード :

0 : STOP 生成なし。

1 : 現在のバイト転送後に SCL ラインと SDA ラインを解放します。

ビット 8 START : START 生成

このビットは、ソフトウェアによってセット/クリアされ、START の送信時または PE=0 のときにハードウェアによってクリアされます。

マスタモード :

0 : START は生成されません。

1 : 反復 START が生成されます。

スレーブモード :

0 : START は生成されません。

1 : バスが空いていれば、START が生成されます。

ビット 7 NOSTRETCH : クロック伸長ディセーブル (スレーブモード)

このビットは、スレーブモードで ADDR または BTF フラグがセットされているとき、ソフトウェアによってリセットされるまで、クロック伸長を無効にするために使用されます。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

ビット 6 ENGC : 同報イネーブル

0 : 同報は無効です。アドレス 00h は NACK されます。

1 : 同報は有効です。アドレス 00h は ACK されます。

ビット 5 ENPEC : PEC イネーブル

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

ビット 4 ENARP : ARP イネーブル

0 : ARP は無効です。

1 : ARP は有効です。

SMBTYPE=0 の場合、SMBus デバイスのデフォルトアドレスが認識されます。

SMBTYPE=1 の場合、SMBus ホストのアドレスが認識されます。

ビット 3 **SMBTYPE** : SMBus タイプ

0 : SMBus デバイス

1 : SMBus ホスト

ビット 2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **SMBUS** : SMBus モード

0 : I²C モード

1 : SMBus モード

ビット 0 **PE** : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注 : 通信中にこのビットがリセットされた場合、現在の通信の最後にアイドル状態に戻ったとき、ペリフェラルは無効になります。

通信の最後には PE=0 になるので、すべてのビットがリセットされます。

マスタモードでは、通信が終るまで、このビットをリセットしてはなりません。

注 : STOP、START、または PEC ビットがセットされると、このビットがハードウェアによってクリアされるまで、ソフトウェアは I2C_CR1 に書き込みアクセスを実行してはいけません。さもなければ、第 2 の STOP、START、または PEC リクエストを設定する危険性があります。

24.6.2 I²C 制御レジスタ 2 (I2C_CR2)

アドレスオフセット : 0x04															
リセット値 : 0x0000															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	LAST	DMA EN	ITBUF EN	ITEVT EN	ITERR EN	Res.	Res.	FREQ[5:0]					
			rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値のままにしておかなければなりません。

ビット 12 **LAST** : DMA 最終転送

0 : 次の DMA EOT は最終転送ではありません。

1 : 次の DMA EOT は最終転送です。

注 : このビットは、最後の受信データでの NACK の生成を可能にするために、マスタレシーバモードで使用されます。

ビット 11 **DMAEN** : DMA リクエストイネーブル

0 : DMA リクエストは無効です。

1 : TxE=1 または RxNE =1 のとき、DMA リクエストは有効です。

ビット 10 **ITBUFEN** : バッファ割り込みイネーブル

0 : TxE = 1 または RxNE = 1 で、割り込みは生成されません。

1 : TxE = 1 または RxNE = 1 で、イベント割り込みを生成します (DMAEN ビットの状態には依存しません)。

ビット 9 **ITEVTEN** : イベント割り込みイネーブル

0 : イベント割り込みは無効です。

1 : イベント割り込みは有効です。

この割り込みは、以下の場合に生成されます。

- SB = 1 (マスタ)
- ADDR = 1 (マスタ/スレーブ)
- ADD10 = 1 (マスタ)
- STOPF = 1 (スレーブ)
- BTF = 1 (ただし、TxE または RxNE イベントが生成してしないとき)
- ITBUFEN = 1 で TxE イベントが 1 になったとき
- ITBUFEN = 1 で RxNE イベントが 1 になったとき

ITERREN : エラー割り込みイネーブル

0 : エラー割り込みは無効です。

1 : エラー割り込みは有効です。

この割り込みは、以下の場合に生成されます。

- BERR = 1
- ARLO = 1
- AF = 1
- OVR = 1
- PECERR = 1
- TIMEOUT = 1
- SMBALERT = 1

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5:0 **FREQ[5:0]** : ペリフェラルクロック周波数

ペリフェラルクロック周波数は、入力 APB クロック周波数 (I2C ペリフェラルは APB に接続) を使用して設定する必要があります。FREQ フィールドは、ペリフェラルによって使用され、I2C 仕様に準拠したデータセットアップとホールド時間を生成します。最小許容周波数は 2 MHz であり、最大周波数は最大 APB 周波数で制限され、50 MHz (ペリフェラルの本質的な上限) を超えることはできません。

0b000000 : 設定禁止

0b000001 : 設定禁止

0b000010 : 2 MHz

...

0b110010 : 50 MHz

0b101010 を超える場合 : 設定禁止

24.6.3 I²C 自己アドレスレジスタ 1 (I2C_OAR1)

アドレスオフセット : 0x08
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD MODE	Res.	Res.	Res.	Res.	Res.	ADD[9:8]		ADD[7:1]							ADD0
r/w						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 15 **ADDMODE** アドレスモード (スレーブモード)
- 0 : 7 ビットスレーブアドレス (10 ビットアドレスは確認応答されません)
 - 1 : 10 ビットスレーブアドレス (7 ビットアドレスは確認応答されません)

ビット 14 ソフトウェアによって常に 1 に保持してください。

ビット 13:10 予約済みであり、リセット値のままにしておかなければなりません。

- ビット 9:8 **ADD[9:8]** : インタフェースアドレス
- 7 ビットアドレッシングモード : 無視されます。
 - 10 ビットアドレッシングモード : アドレスのビット 9:8。

ビット 7:1 **ADD[7:1]** : インタフェースアドレス
アドレスのビット 7:1

- ビット 0 **ADD0** : インタフェースアドレス
- 7 ビットアドレッシングモード : 無視されます。
 - 10 ビットアドレッシングモード : アドレスのビット 0。

24.6.4 I²C 自己アドレスレジスタ 2 (I2C_OAR2)

アドレスオフセット : 0x0C
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD2[7:1]							EN DUAL
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:8 予約済みであり、リセット値のままにしておかなければなりません。

ビット 7:1 **ADD2[7:1]** : インタフェースアドレス
デュアルアドレスモードでアドレスのビット 7:1

- ビット 0 **ENDUAL** : デュアルアドレスモードイネーブル
- 0 : 7 ビットアドレスモードで、OAR1 のみが認識されます。
 - 1 : 7 ビットアドレスモードで、OAR1 と OAR2 の両方が認識されます。

24.6.5 I²C データレジスタ (I2C_DR)

アドレスオフセット : 0x10
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値のままにしておかなければなりません。

ビット 7:0 **DR[7:0]** 8 ビットデータレジスタ

- 受信されたバイト、またはバスに送信されるバイト。
- トランスマッタモード：データレジスタにバイトが書き込まれると、バイト送信が自動的に始まりま
す。送信が始まり (TxNE=1)、次に送信されるデータがデータレジスタに書き込まれた場合、連続的な
送信ストリームを維持できます。
 - レシーバモード：受信したバイトはデータレジスタにコピーされます (RxNE=1)。データレジスタが
読み出されてから次のデータバイトが受信された場合 (RxNE=1)、連続的な送信ストリームを維持で
きます。

注： スレープモードでは、アドレスはデータレジスタにコピーされません。

書き込み衝突は管理されません (TxNE=0 の場合、データレジスタは書き込み可能です)。

**ACK パルスで ARLO イベントが発生した場合、受信したバイトはデータレジスタにコピーされ
ないため、読み出しできません。**

24.6.6 I²C ステータスレジスタ 1 (I2C_SR1)

アドレスオフセット : 0x14
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB ALERT	TIMEO UT	Res.	PEC ERR	OVR	AF	ARLO	BERR	TXE	RXNE	Res.	STOPF	ADD10	BTf	ADDR	SB
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

ビット 15 SMBALERT : SMBus アラート

- SMBus ホストモードで、
 - 0 : SMBALERT はありません。
 - 1 : SMBALERT イベントがピンに発生しました。
- SMBus スレーブモードで、
 - 0 : SMBALERT 応答アドレスヘッダはありません。
 - 1 : SMBALERT LOW への SMBALERT 応答アドレスヘッダを受信しました。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 14 TIMEOUT : タイムアウトまたはタイマロー期間エラー

- 0 : タイムアウトエラーは発生していません。
- 1 : SCL は 25 ms の間ローレベルを維持しました (タイムアウト)。
- または
- マスタのクロックロー累積延長時間が 10 ms を超えました (Tlow:mext)。
- または
- スレーブのクロックロー累積伸長時間が 25 ms を超えました (Tlow:sext)。
- スレーブモードに設定した場合 : スレーブは通信をリセットし、ラインはハードウェアによって解放されます。
- マスタモードに設定した場合 : STOP コンディションがハードウェアによって送信されます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

注 : この機能は SMBus モードでのみ使用できます。

ビット 13 予約済みであり、リセット値のままにしておかなければなりません。**ビット 12 PECERR : 受信時の PEC エラー**

- 0 : PEC エラーなし。レシーバは PEC 受信後に ACK を返します (ACK=1 の場合)。
- 1 : PEC エラーあり。レシーバは PEC 受信後に NACK を返します (ACK は任意)。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。
- 注 : 受信 CRC が正しくないとき、CRC 受信が終わる前に PEC 制御ビットがセットされない場合には、スレーブモードでの PECERR はセットされません。しかしながら、PEC 値を読み出すと、受信 CRC が正しいか否かが分かります。

ビット 11 OVR : オーバーラン/アンダーラン

- 0 : オーバーラン/アンダーランなし。
- 1 : オーバーランまたはアンダーランあり。
- スレーブモードで NOSTRETCH=1 の場合、次の条件でハードウェアによってセットされます。
- 受信時、新しいバイト (ACK パルスを含む) が受信され、DR レジスタがまだ読み出されていないとき。新しい受信バイトは失われます。
- 送信時、新しいバイトを送信すべきときに、DR レジスタへの書き込みがまだ行われていないとき。同じバイトが 2 回送信されます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

注 : DR への書き込みが SCL の立ち上がりエッジのごく近くで行われた場合、送信されるデータは不定で、ホールドタイミングエラーが発生します。

ビット 10 AF : 確認応答障害

- 0 : 確認応答障害なし。
- 1 : 確認応答障害あり。
- 確認応答が返されないとき、ハードウェアによってセットされます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 9 **ARLO** : アービトレーションロスト (マスタモード)

0 : アービトレーションロストは検出されていません。

1 : アービトレーションロストが検出されました。

インタフェースがバスのアービトレーションを他のマスタに開放したとき、ハードウェアによってセットされます。

- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ARLO イベントの後、インタフェースは自動的にスレーブモード (MSL=0) に切り替わります。

注 : **SMBUS では、スレーブモードでのデータのアービトレーションは、データフェーズ中または確認応答の送信中にのみ行われます (アドレスの確認応答では行われません)。**

ビット 8 **BERR** : バスエラー

0 : START または STOP コンディションの誤配置はありません。

1 : START または STOP コンディションが誤配置されました。

- バイト転送中に SCL がハイである間、SDA の立ち上がりまたは立ち下がりエッジが有効でない位置に発生していることをインタフェースが検出したとき、ハードウェアによってセットされます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 7 **TxE** : データレジスタエンプティ (トランスミッタ)

0 : データレジスタは空ではありません。

1 : データレジスタは空です。

- 送信中に DR が空になったときにセットされます。TxE はアドレスフェーズではセットされません。
 - DR レジスタへのソフトウェア書き込みによってクリアされるか、もしくは START/STOP コンディションの後または PE=0 のときにハードウェアによってクリアされます。
- NACK が受信された場合、または次に送信されるバイトが PEC である場合 (PEC=1)、TxE はセットされません。

注 : **送信される最初のデータを書き込んだり、BTF がセットされているときにデータを書き込んだりしても、TxE はクリアされません。いずれの場合も、データレジスタはまだ空だからです。**

ビット 6 **RxNE** : データレジスタノットエンプティ (レシーバ)

0 : データレジスタは空です。

1 : データレジスタは空ではありません。

- レシーバモードで、データレジスタが空でないときにセットされます。RxNE は、アドレスフェーズではセットされません。
 - ソフトウェアで DR レジスタへの読み書きを行ってクリアするか、PE=0 のときにハードウェアによってクリアされます。
- ARLO イベントの場合、RxNE はセットされません。

注 : **BTF がセットされているとき、データレジスタはまだ満杯であるため、データを読み出しても RxNE はクリアされません。**

ビット 5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **STOPF** : STOP 検出 (スレーブモード)

0 : STOP コンディションは検出されていません。

1 : STOP コンディションが検出されました。

- 確認応答後に、スレーブがバス上で STOP コンディションを検出したとき、ハードウェアによってセットされます (ACK=1 の場合)。
- ソフトウェアで SR1 レジスタを読み出し、続いて CR1 レジスタに書き込むとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

注 : **STOPF ビットは、NACK 受信後にはセットされません。**

STOPF がセットされた後、完全なクリアシーケンス (READ SR1、続いて WRITE CR1) を実行することを推奨します。図 241 : スレーブ受信の転送シーケンス図 (713 ページ) を参照してください。

ビット 3 ADD10 : 10 ビットヘッダの送信 (マスタモード)

- 0 : ADD10 イベントは発生していません。
- 1 : マスタは最初のアドレスバイト (ヘッダ) を送信しました。

- マスタが 10 ビットアドレスモードで最初のバイトを送信すると、ハードウェアによってセットされます。
- ソフトウェアで SR1 レジスタを読み出し、続いて DR レジスタに 2 番目のアドレスバイトを書き込むとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

注： *ADD10 ビットは、NACK 受信後にはセットされません。*

ビット 2 BTF : バイト転送終了

- 0 : データバイト転送は完了していません。
- 1 : データバイト転送が正常に終了しました。

- NOSTRETCH=0 の場合、次の条件でハードウェアによってセットされます。
- 受信時、新しいバイト (ACK パルスを含む) が受信され、DR レジスタがまだ読み出されていないとき (RxNE=1)。
- 送信時、新しいバイトを送信すべきときに、DR レジスタへの書き込みがまだ行われていないとき (TxE=1)。
- DR レジスタへの読み書きによりソフトウェアでクリアされます。または、送信時の START/STOP コンディションの後や PE=0 のときに、ハードウェアによってクリアされます。

注： *BTF ビットは、NACK 受信後にはセットされません。*

BTF ビットは、次に送信されるバイトが PEC である場合 (I2C_SR2 レジスタの TRA=1 かつ I2C_CR1 レジスタの PEC=1)、セットされません。

ビット 1 ADDR : アドレス送信 (マスタモード) / アドレス一致 (スレーブモード)

このビットは、ソフトウェアで SR1 レジスタを読み出し、続いて SR2 レジスタを読み出すことでクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

アドレス一致 (スレーブ)

- 0 : アドレスは受信されていないか、一致しませんでした。
- 1 : 受信したアドレスが一致しました。

- 受信したスレーブアドレスが OAR レジスタの内容と一致したとき、あるいは同報 / SMBus デバイスデフォルトアドレスまたは SMBus ホスト / SMBus アラートが認識されたとき、ハードウェアによってセットされます。(設定に応じて有効化されている場合)。

注： *スレーブモードでは、ADDR がセットされた後、完全なクリアシーケンス (READ SR1、続いて READ SR2) を実行することを推奨します。図 241 : スレーブ受信の転送シーケンス図 (713 ページ) を参照してください。*

アドレス送信 (マスタ)

- 0 : アドレス送信は終了していません。
- 1 : アドレス送信は終了しました。

- 10 ビットアドレッシングでは、2 番目のバイトの ACK 後にビットがセットされます。
- 7 ビットアドレッシングでは、バイトの ACK 後にビットがセットされます。

注： *ADDR は、NACK 受信後にはセットされません。*

ビット 0 SB : スタートビット (マスタモード)

- 0 : START コンディションなし。
- 1 : START コンディションが生成されました。

- START コンディションが生成されたときにセットされます。
- ソフトウェアで SR1 レジスタを読み出し、続いて DR レジスタへの書き込みを行うとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

24.6.7 I²C ステータスレジスタ 2 (I2C_SR2)

アドレスオフセット : 0x18
リセット値 : 0x0000

注 : たとえ I2C_SR1 の読み出し後に ADDR フラグがセットされた場合でも、I2C_SR1 の読み出し後に I2C_SR2 を読み出すと、ADDR フラグはクリアされます。したがって、I2C_SR1 の ADDR がセットされていると分かったとき、または STOPF ビットがクリアされているときにのみ、I2C_SR2 を読み出す必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMB HOST	SMB DEFAU LT	GEN CALL	Res.	TRA	BUSY	MSL
r	r	r	r	r	r	r	r	r	r	r	r		r	r	r

ビット 15:8 **PEC[7:0]** パケットエラーチェックレジスタ
ENPEC=1 のとき、このレジスタは内部 PEC を保持しています。

ビット 7 **DUALF** : デュアルフラグ (スレーブモード)
0 : 受信アドレスが OAR1 と一致
1 : 受信アドレスが OAR2 と一致
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。

ビット 6 **SMBHOST** : SMBus ホストヘッダ (スレーブモード)
0 : SMBus ホストアドレスは受信されていません。
1 : SMBTYPE=1 かつ ENARP=1 のとき、SMBus ホストアドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。

ビット 5 **SMBDEFAULT** : SMBus デバイスデフォルトアドレス (スレーブモード)
0 : SMBus デバイスデフォルトアドレスは受信されていません。
1 : ENARP=1 のとき、SMBus デバイスデフォルトアドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。

ビット 4 **GENCALL** : 同報アドレス (スレーブモード)
0 : 同報アドレスは受信されていません。
1 : ENGC=1 のとき、同報アドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。

ビット 3 予約済みであり、リセット値のままにしておかなければなりません。

- ビット 2 **TRA** : トランスミッタ/レシーバ
- 0 : データバイトが受信されました。
 - 1 : データバイトが送信されました。
- このビットは、全アドレスフェーズの最後に、アドレスバイトの R/W ビットに応じてセットされます。
また、STOP コンディション (STOPF=1)、繰り返し START コンディション、バスアービトラーションロスト (ARLO=1) の検出後、または PE=0 のときに、ハードウェアによってクリアされます。
- ビット 1 **BUSY** : バスビジー
- 0 : バス上で通信は行われていません。
 - 1 : バス上で通信が行われています。
- SDA または SCL のローレベル検出時に、ハードウェアによってセットされます。
 - STOP コンディションの検出時に、ハードウェアによってクリアされます。
- バス上で通信が進行中であることを表示します。この情報は、インタフェースが無効である (PE=0) と
きにも更新されます。
- ビット 0 **MSL** : マスタ/スレーブ
- 0 : スレーブモード
 - 1 : マスタモード
- インタフェースがマスタモードになると (SB=1)、ハードウェアによってセットされます。
 - バス上での STOP コンディションまたはアービトラーションロスト (ARLO=1) を検出した後、または
PE=0 のときに、ハードウェアによってクリアされます。

注 : *たとえ I2C_SR1 の読み出し後に ADDR フラグがセットされた場合でも、I2C_SR1 の読み出し後に
I2C_SR2 を読み出すと、ADDR フラグはクリアされます。したがって、I2C_SR1 の ADDR がセット
されていると分かったとき、または STOPF ビットがクリアされているときにのみ、I2C_SR2 を読み
出す必要があります。*

24.6.8 I²C クロック制御レジスタ (I2C_CCR)

アドレスオフセット : 0x1C
リセット値 : 0x0000

注 : *標準モード I²C 周波数を実現するためには、f_{PCLK1} は最小 2 MHz 以上である必要があります。高速
モード I²C 周波数を実現するためには、最小 4 MHz 以上である必要があります。400 kHz の最大 I²C
高速 モードクロックに到達するためには、10MHz の整数倍である必要があります。*

CCR レジスタは、I2C が無効 (PE=0) の場合にのみ設定する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Res.	Res.	CCR[11:0]											
r/w	r/w			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 15 **F/S** : I2C マスタモード選択
- 0 : 標準モード I2C
 - 1 : 高速モード I2C

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 14 **DUTY** : 高速モードデューティサイクル

- 0 : 高速 モード $t_{low}/t_{high} = 2$
- 1 : 高速 モード $t_{low}/t_{high} = 16/9$ (CCR を参照)

ビット 13:12 予約済みであり、リセット値のままにしておかなければなりません。

ビット 11:0 **CCR[11:0]** : 高速/標準モード (マスタモード) でのクロック制御レジスタ

マスタモードにおける SCL クロックを制御します。

標準モードまたは SMBus :

$$T_{high} = CCR * T_{PCLK1}$$

$$T_{low} = CCR * T_{PCLK1}$$

高速モード :

DUTY = 0 の場合 :

$$T_{high} = CCR * T_{PCLK1}$$

$$T_{low} = 2 * CCR * T_{PCLK1}$$

DUTY = 1 の場合 : (400 kHz に到達するには)

$$T_{high} = 9 * CCR * T_{PCLK1}$$

$$T_{low} = 16 * CCR * T_{PCLK1}$$

たとえば、標準モードで 100 kHz の SCL 周波数を生成するには、

FREQR = 08、 $T_{PCLK1} = 125$ ns の場合、CCR には 0x28 をプログラムする必要があります。

$$(0x28 \Leftrightarrow 40d \times 125 \text{ ns} = 5000 \text{ ns})$$

注 : 最小許容値は 0x04 です (最小許容値が 0x01 の FAST DUTY モードを除く)

$t_{high} = t_{r(SCL)} + t_{w(SCLH)}$ 。パラメータの定義については、デバイスのデータシートを参照してください。

$t_{low} = t_{f(SCL)} + t_{w(SCLL)}$ 。パラメータの定義については、デバイスのデータシートを参照してください。

I2C 通信速度、 $f_{SCL} \sim 1/(t_{high} + t_{low})$ 。アナログノイズフィルタの入力遅延によって、実際の周波数は異なる場合があります。

CCR レジスタは、I²C が無効 (PE=0) の場合にのみ設定する必要があります。

24.6.9 I²C TRISE レジスタ (I2C_TRISE)

アドレスオフセット : 0x20

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRISE[5:0]					
										r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5:0 **TRISE[5:0]** : 高速/標準モード (マスタモード) における最大立ち上がり時間

これらのビットは、マスタモードで SCL フィードバックループの最大時間を提供します。SCL 立ち上がりエッジ時間にかかわらず、安定した SCL 周波数を保持する目的があります。

これらのビットには、I²C バス仕様で定められている SCL 最大立ち上がり時間より 1 だけ大きな値をプログラミングする必要があります。

たとえば、標準モードで許容される SCL の最大立ち上がり時間は 1000 ns です。

I2C_CR2 レジスタの FREQ[5:0] ビットの値が 0x08 で、 $T_{PCLK1} = 125$ ns である場合、TRISE[5:0] ビットには 09h をプログラミングする必要があります。

$$(1000 \text{ ns} / 125 \text{ ns} = 8 + 1)$$

フィルタ値も TRISE[5:0] に加算することができます。

結果が整数でない場合、 t_{HIGH} パラメータを考慮して、その整数部を TRISE[5:0] にプログラミングする必要があります。

注 : TRISE[5:0] は、I2C が無効 (PE=0) の場合にのみ設定する必要があります。



24.6.10 I²C FLTR レジスタ (I2C_FLTR)

アドレスオフセット : 0x24
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANOFF	DNF[3:0]			
											rw	rw	rw	rw	rw

ビット 15:5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **ANOFF** : アナログノイズフィルタ OFF

0 : アナログノイズフィルタ有効
1 : アナログノイズフィルタ無効

注 : **ANOFF** は、**I2C** が無効 (**PE=0**) の場合にのみ設定する必要があります。

ビット 3:0 **DNF[3:0]** : デジタルノイズフィルタ

これらのビットを使って、SDA 入力と SCL 入力のデジタルノイズフィルタを設定します。デジタルフィルタは、最大 DNF[3:0] * TPCLK1 の幅を持つスパイクを除去します。

0000 : デジタルノイズフィルタを無効化
0001 : デジタルノイズフィルタを有効化し、最大 1 * TPCLK1 の除去能力を持ちます
...
1111 : デジタルノイズフィルタを有効化し、最大 15 * TPCLK1 の除去能力を持ちます

注 : **DNF[3:0]** は、**I2C** が無効 (**PE=0**) の場合にのみ設定する必要があります。アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

24.6.11 I²C のレジスタマップ

次の表に、I²C のレジスタマップとリセット値を示します。

表 125. I²C レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWRST	Res.	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENG	ENPEC	ENARP	SMBTYPE	Res.	SMBUS	PE			
	リセット値																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x04	I2C_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Res.	Res.	FREQ[5:0]									
	リセット値																			0	0	0	0	0	0			0	0	0	0	0	0	0		
0x08	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDMODE	Res.	Res.	Res.	Res.	Res.	ADD[9:8]		ADD[7:1]						ADD0				
	リセット値																	0						0	0	0	0	0	0	0	0	0	0	0		
0x0C	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD2[7:1]						ENDUAL					
	リセット値																									0	0	0	0	0	0	0	0	0	0	
0x10	I2C_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[7:0]											
	リセット値																									0	0	0	0	0	0	0	0	0	0	
0x14	I2C_SR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMBALERT	TIMEOUT	PECERR	OVR	AF	ARLO	BERR	TXE	RXNE	Res.	STOPF	ADD10	BTF	ADDR	SR				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	I2C_SR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]					DUALF					SMBHOST	SMBDEFAULT	GENCALL	Res.	TRA	BUSY	MSL		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	I2C_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F/S	DUTY	Res.	Res.	CCR[11:0]														
	リセット値																	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	I2C_TRISE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRISE[5:0]								
	リセット値																											0	0	0	0	0	1	0	0	
0x24	I2C_FLTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANOFF	DNF[3:0]						
	リセット値																											0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2](#) を参照してください。

25 USART (Universal synchronous asynchronous receiver transmitter)

25.1 USART の概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟に全二重データ交換を行うことができます。USART は、フラクショナルボーレートジェネレータを使用して、非常に広範囲のボーレートを提供します。

USART は、同期式単方向通信と半二重単線通信をサポートします。また、LIN (Local Interconnection Network)、スマートカードプロトコルおよび IrDA (infrared data association) SIR ENDEC 仕様、およびモデム動作 (CTS/RTS) もサポートします。マルチプロセッサ通信も可能です。

また、マルチバッファ設定用の DMA を使用して、高速データ通信が可能です。

25.2 USART の主な機能

- 全二重、非同期通信
- NRZ 標準フォーマット（マーク／スペース）
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロックの柔軟な許容誤差を実現
- フラクショナルボーレートジェネレータシステム
 - 共通のプログラム可能な送信および受信ボーレート（最大 APB 周波数におけるボーレート値については、データシートを参照）。
- プログラム可能なデータワード長：8 または 9 ビット
- 設定可能なストップビット：1 個または 2 個のストップビットをサポート
- LIN マスタの同期ブレイク送信機能と LIN スレーブのブレイク検出機能
 - USART が LIN 用にハードウェア設定されている場合、13 ビットのブレイク生成と 10/11 ビットのブレイク検出
- 同期送信用のトランスミッタクロック出力
- IrDA SIR エンコーダデコーダ
 - 通常モード用に 3/16 ビット持続時間のサポート
- スマートカードエミュレーション機能
 - スマートカードインタフェースは、ISO 7816-3 規格で定義された非同期プロトコルのスマートカードをサポート
 - スマートカード動作に 0.5 個、1.5 個のストップビット
- 単線半二重通信
- DMA (Direct Memory Access) を使用した設定可能なマルチバッファ通信
 - 集中型 DMA を使用して、送受信バイトを専用 SRAM にバッファリング
- トランスミッタとレシーバ用に個別の有効ビット
- 転送検出フラグ：
 - 受信バッファフル
 - 送信バッファエンプティ
 - 送信終了フラグ
- パリティ制御：
 - パリティビットの送信
 - 受信したデータバイトのパリティ検査
- 4 つのエラー検出フラグ：
 - オーバーランエラー
 - ノイズ検出
 - フレームエラー
 - パリティエラー
- 10 種の割り込みソース（フラグ付き）
 - CTS の変化
 - LIN ブレイク検出
 - 送信データレジスタエンプティ
 - 送信完了
 - 受信データレジスタフル

- アイドルラインの受信
- オーバーランエラー
- フレーミングエラー
- ノイズエラー
- パリティエラー
- マルチプロセッサ通信：アドレス不一致の場合、ミュートモードに遷移
- ミュートモードからのウェイクアップ（アイドルライン検出またはアドレスマーク検出による）
- 2種のレシーバウェイクアップモード：アドレスビット（MSB、9番目のビット）、アイドルライン

25.3 USART の実装

このセクションでは、USART1 に実装されているすべての機能について説明しています。USART インスタンス間での違いについては、表 126：USART の機能を参照してください。

表 126. USART の機能

USART のモード／機能 ⁽¹⁾	USART1、USART2、USART3、USART6
モデムのハードウェアフロー制御 ⁽²⁾	X
DMA を使用した連続通信	X
マルチプロセッサ通信	X
同期モード ⁽²⁾	X
スマートカードモード	X
単線半二重通信	X
Ir SIR ENDEC ブロック	X
LIN モード	X
USART データ長	8 または 9 ビット

1. X：サポートされています。
2. この機能はパッケージによって使用できない場合があります。詳細については、製品データシートを参照してください。

25.4 USART の機能詳細

インタフェースは、3 個のピンによって他のデバイスに外部接続されます（図 245 を参照）。USART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力（RX）と送信データ出力（TX）です。

RX：受信データ入力はシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用され、有効な入力データとノイズを区別しています。

TX：送信データ出力です。トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TX ピンはハイレベルになります。単線モードとスマートカードモードでは、この入出力を使用してデータを送受信します（USART レベルでは、データは続いて SW_RX で受信されます）。



通常の USART モードでは、フレームは以下の部分で構成され、これらのピンを通じてシリアルデータが送受信されます。

- 送受信前のアイドルライン
- スタートビット
- データワード (8 または 9 ビット)、LSB ファースト
- フレームの完了を示す 0.5 個、1 個、1.5 個、2 個のストップビット
- このインタフェースはフラクショナルボーレートジェネレータを使用します - 12 ビットの整数部と 4 ビットの小数部
- ステータスレジスタ (USART_SR)
- データレジスタ (USART_DR)
- ボーレートレジスタ (USART_BRR) - 12 ビットの整数部と 4 ビットの小数部
- ガードタイムレジスタ (USART_GTPR) - スマートカードモードの場合

各ビットの定義については、[セクション 25.6 : USART レジスタ](#)を参照してください。

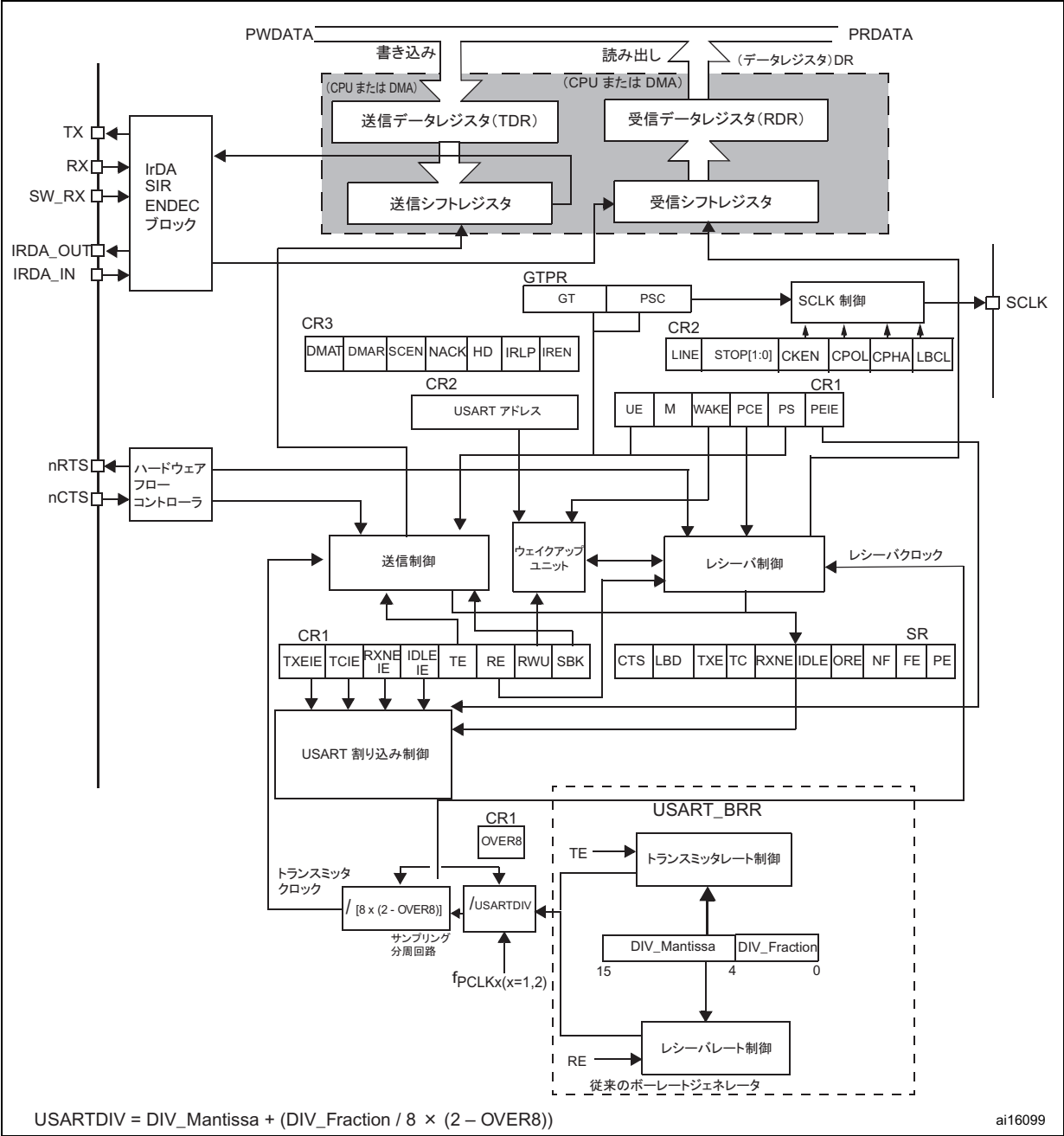
同期モードでのインタフェースには次のピンが必要になります。

- **SCLK** : トランスミッタクロック出力。このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します (スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します)。これと並行して、RX でデータを同期受信できます。これを使用して、シフトレジスタを持つペリフェラル (例 : LCD ドライバ) を制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。スマートカードモードでは、SCLK はスマートカードヘクロックを供給できます。

ハードウェアフロー制御モードでは次のピンが必要になります。

- **nCTS** : Clear To Send は、ハイレベルのとき、現在の転送の終わりにデータ送信をブロックします。
- **nRTS** : Request To Send は、ローレベルのとき、USART がデータを受信できることを示します。

図 245. USART のブロック図



25.4.1 USART キャラクタの説明

ワード長は、USART_CR1 レジスタの M ビットをプログラムすることによって、8 または 9 ビットとして選択できます (図 246 を参照)。

TX ピンは、スタートビットの処理中にはロー状態です。また、ストップビットの処理中にはハイ状態です。

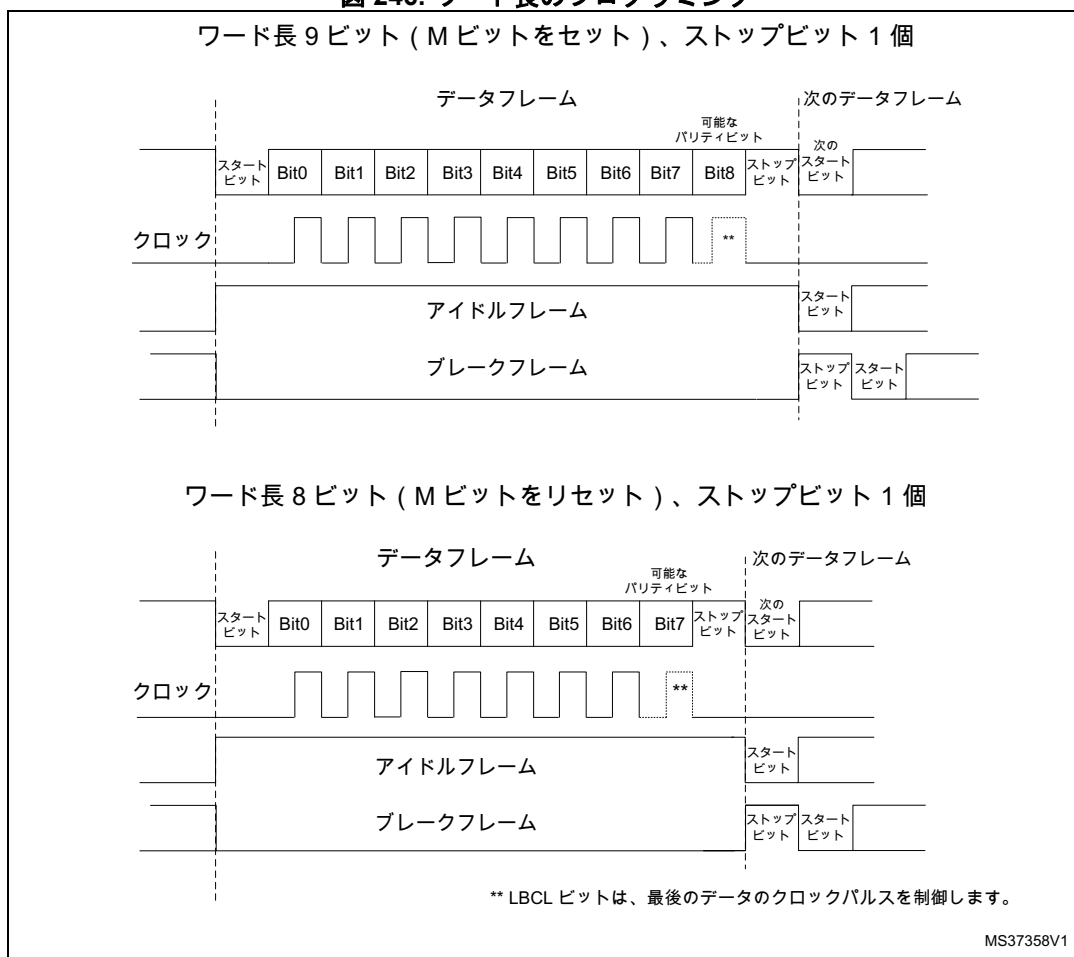
アイドルキャラクタは、1 フレーム分の「1」であると解釈され、この後にデータが格納された次のフレームのスタートビットが続きます (「1」の数にはストップビットの数も含まれます)。

ブ레이크キャラクタは、フレーム周期中に「0」を受信することと解釈されます。トランスミッタは、ブ레이크フレームの終わりに 1 個または 2 個のストップビット (論理「1」ビット) を挿入して、スタートビットを承認します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 246. ワード長のプログラミング



25.4.2 トランスミッタ

トランスミッタは、M ビットのステータスに応じて、8 または 9 ビットのデータワードを送信できます。送信有効ビット (TE) がセットされているとき、送信シフトレジスタ内のデータは TX ピンに出力され、対応するクロックパルスは SCLK ピンに出力されます。

キャラクタ送信

USART 送信時、データは LSB ファーストで TX ピンにシフトアウトされます。このモードでは、USART_DR レジスタは、内部バスと送信シフトレジスタの間のバッファ (TDR) で構成されます (図 245 を参照)。

各キャラクタの前には、スタートビット (1 ビット周期、論理レベルロー) があります。キャラクタは、設定可能な数のストップビットで終端されます。

USART では、0.5、1、1.5、および 2 個のストップビットがサポートされます。

注： データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ボーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中のデータは失われます。

TE ビットが有効になると、アイドルフレームが送信されます。

設定可能なストップビット

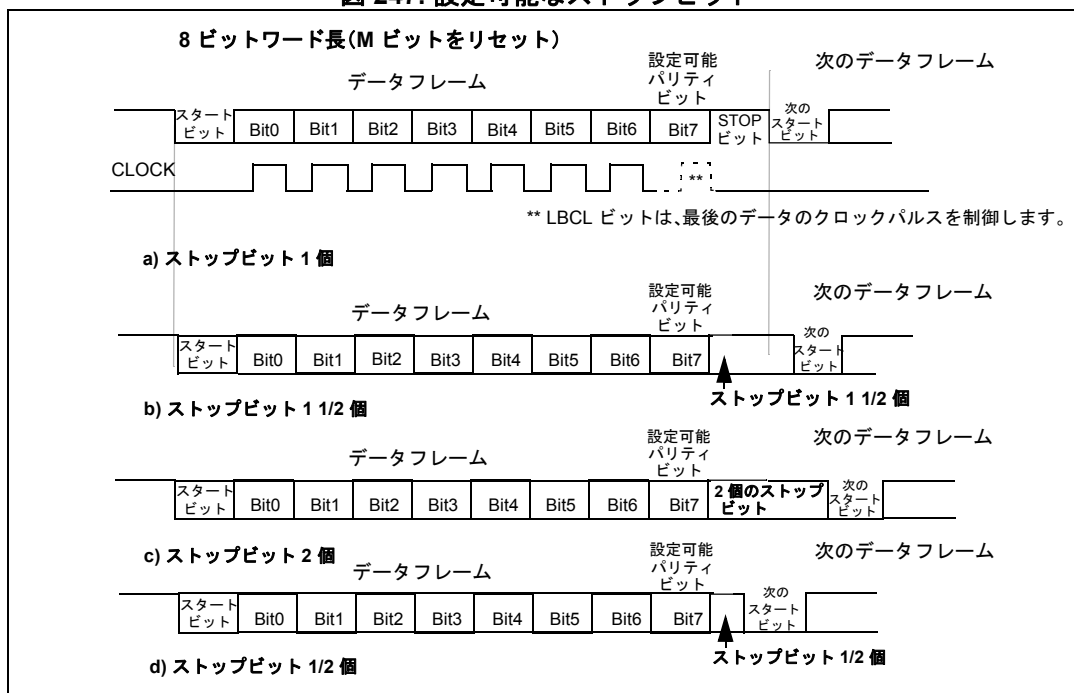
各キャラクタとともに送信されるストップビットの数は、制御レジスタ 2 のビット 13、12 でプログラミングできます。

- **1 個のストップビット：** ストップビット数のデフォルト値です。
- **2 個のストップビット：** 通常の USART モード、単線モード、およびモデムモードでサポートされます。
- **0.5 個のストップビット：** スマートカードモードでのデータの受信に使用されます。
- **1.5 個のストップビット：** スマートカードモードでのデータの送受信に使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信では、10 個 (m = 0 の場合) または 11 個 (m = 1 の場合) のローレベルビットの後に、設定された数のストップビットが続きます。ローレベルビット 10/11 個を超える長いブ레이크を送信することはできません。

図 247. 設定可能なストップビット



手順 :

1. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
2. USART_CR1 の M ビットをプログラミングして、ワード長を定義します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. マルチバッファ通信を行う場合は、USART_CR3 レジスタの DMA 有効 (DMAT) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
5. USART_BRR レジスタを使用して、希望するボーレートを選択します。
6. USART_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
7. 送信するデータを USART_DR レジスタに書き込みます (これによって TXE ビットがクリアされます)。シングルバッファの場合、送信される各データにこれを繰り返します。
8. USART_DR レジスタに最後のデータを書き込んだ後、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。これは、たとえば最後の送信内容の破壊を避けるために、USART が無効になったり HALT モードに入ったりする場合に必要です。

1 バイト通信

TXE ビットは、常にデータレジスタへの書き込みによってクリアされます。

TXE ビットは、ハードウェアによってセットされ、次を示します。

- データが TDR からシフトレジスタに移動し、データ送信が開始しています。
- TDR レジスタは空です。
- 次のデータを、前のデータに上書きすることなく、USART_DR レジスタに書き込みます。

TXEIE ビットがセットされている場合、このフラグは割り込みを生成します。

送信が行われている場合、USART_DR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、さらに、現在の送信の最後にシフトレジスタにコピーされます。

送信が行われていないときには、USART_DR レジスタへの書き込み命令によってデータがシフトレジスタに直接格納され、データ送信が開始され、TXE ビットがただちにセットされます。

フレームが送信され（ストップビットの後）、TXE ビットがセットされると、TC ビットはハイレベルになります。USART_CR1 レジスタの TCIE ビットがセットされると、割り込みが生成されます。

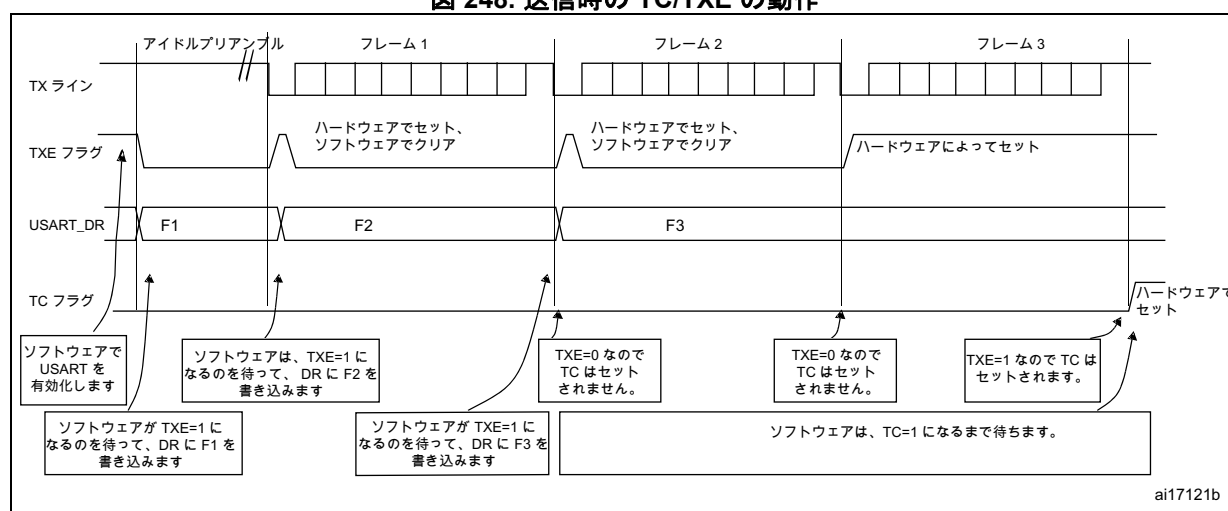
USART_DR レジスタに最後のデータを書き込んだ後は、USART を無効にしたり、マイクロコントローラを低電力モードにする前に TC=1 になるまで待つ必要があります（[図 248：送信時の TC/TXE の動作](#)を参照）。

TC ビットは、次のソフトウェアシーケンスによってクリアされます。

1. USART_SR レジスタからの読み出し
2. USART_DR レジスタへの書き込み

注： TC ビットは、“0”を書き込むことによってもクリアされます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

図 248. 送信時の TC/TXE の動作



ブ레이크キャラクタ

SBK ビットをセットすると、ブ레이크キャラクタが送信されます。ブ레이크フレーム長は、M ビットに依存します（[図 246](#)を参照）。

SBK ビットが“1”にセットされると、現在のキャラクタ送信が完了してから、TX ラインにブ레이크キャラクタが送信されます。このビットは、ブ레이크キャラクタが完了したとき（ブ레이크キャラクタのストップビット）、ハードウェアによってリセットされます。USART は、次のフレームのスタートビットが正しく認識されるように、最後のブ레이크フレームの終わりに論理 1 ビットを挿入します。

注： ソフトウェアがブ레이크送信の開始前に SBK ビットをリセットすると、ブ레이크キャラクタは送信されません。2 つの連続したブ레이크の場合は、前のブ레이크のストップビットの後に SBK ビットをセットしてください。

アイドルキャラクタ

TE ビットをセットすると、USART は、最初のデータフレームの前にアイドルフレームを送信します。

25.4.3 レシーバ

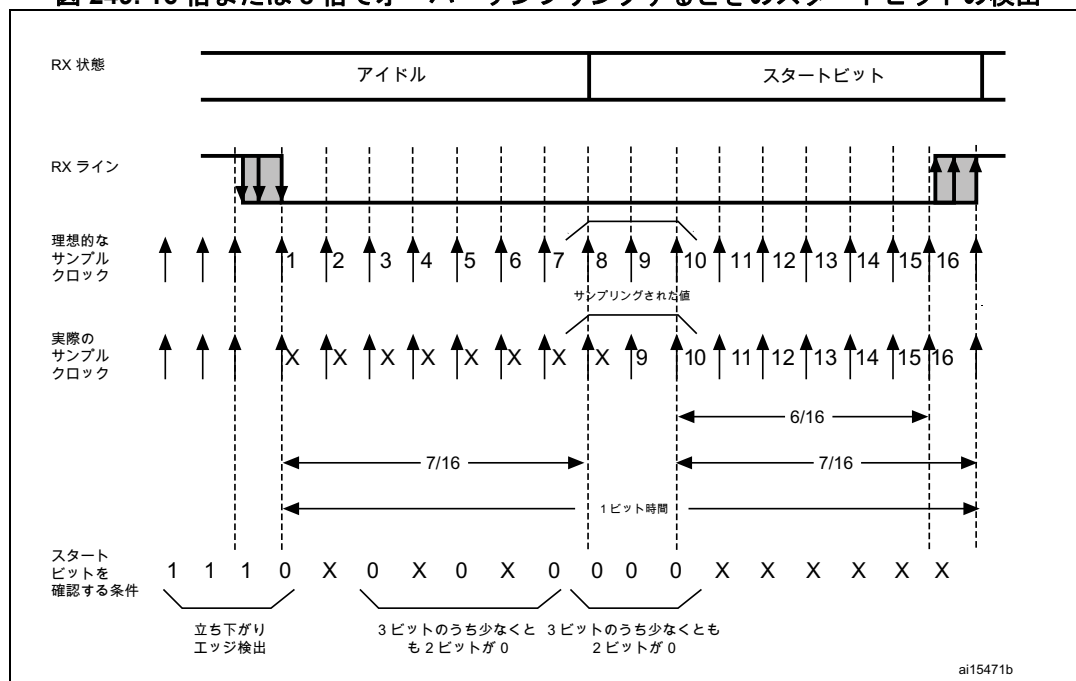
USART は、USART_CR1 レジスタの M ビットに応じて、8 ビットまたは 9 ビットのデータワードを受信できます。

スタートビット検出

スタートビット検出シーケンスは、16 倍または 8 倍でオーバーサンプリングするときと同じです。

USART では、サンプルの特定シーケンスが認識されると、スタートビットが検出されます。このシーケンスは、1110X0X0X0000 です。

図 249. 16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出



注： シーケンスが完了していない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻って（フラグはセットされません）、立ち下がりエッジを待ちます。

サンプリングされた 3 つのビットが 0 の場合（第 3、第 5、および第 7 ビットでの最初のサンプリングで 3 つのビットが 0 であり、第 8、第 9、および第 10 ビットでの 2 回目のサンプリングでも 3 つのビットが 0）、スタートビットが確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）。

両方のサンプリングでサンプリングされた 3 つのビットのうち、少なくとも 2 つが 0 の場合（第 3、第 5、および第 7 ビットでのサンプリングと、第 8、第 9、および第 10 ビットでのサンプリング）、スタートビットは確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）、NE ノイズフラグがセットされます。この条件が満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります（フラグはセットされません）。

いずれか一方のサンプリングで（第 3、第 5、および第 7 ビットでのサンプリングまたは第 8、第 9、および第 10 ビットでのサンプリング）、3 つのビットのうち 2 つが 0 の場合、スタートビットは確認されますが、NE ノイズフラグビットがセットされます。

キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファーストでシフトインされます。このモードでは、USART_DR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

手順：

1. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
2. USART_CR1 の M ビットをプログラミングして、ワード長を定義します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. マルチバッファ通信を行う場合には、USART_CR3 レジスタの DMA 有効 (DMAR) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。STEP 3
5. ボーレートレジスタ USART_BRR を使用して、希望するボーレートを選択します。
6. USART_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- RXNE ビットがセットされます。これは、シフトレジスタの内容が RDR レジスタに転送されたことを示します。言い換えると、データは受信され、読み出し可能です (関連するエラーフラグも同様です)。
- RXNEIE ビットがセットされていた場合、割り込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。
- マルチバッファでは、RXNE ビットはバイトを受信するたびにセットされ、データレジスタへの DMA 読み出しによってクリアされます。
- シングルバッファモードでは、RXNE ビットは USART_DR レジスタへのソフトウェア読み出しによってクリアされます。RXNE フラグは、0 を書き込むとクリアできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。

注： データの受信中には、RE ビットをリセットしないでください。受信中に RE ビットが無効にされた場合、現在のバイトの受信が中止されます。

ブレークキャラクタ

ブレークキャラクタを受信すると、USART はブレークキャラクタをフレーミングエラーとして処理します。

アイドルキャラクタ

アイドルフレームが検出された場合、データ受信キャラクタと同じ手順があり、IDLEIE ビットがセットされている場合は割り込みが生成されます。

オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。

RXNE フラグは、バイトを受信するたびにセットされます。次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RDR の内容は失われません。USART_DR への読み出しが行われると、前回のデータが読み出されます。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXNEIE ビットがセットされた場合、または EIE ビットと DMAR ビットの両方がセットされた場合は、割り込みが生成されます。
- ORE ビットは、USART_SR レジスタの読み出しと、それに続く USART_DR レジスタの読み出し動作によってリセットされます。

注： ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。この場合、次の 2 つの可能性がります。

- RXNE=1 の場合、有効な最後のデータは、受信レジスタ RDR に格納され、読み出しが可能です。
- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものがないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。また、読み出しシーケンス中 (USART_SR レジスタと USART_DR レジスタの読み出しアクセスの間) に新しいデータが受信された場合にも発生します。

適切なオーバーサンプリング方式の選択

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています（同期モードの場合を除く）。

オーバーサンプリング方式は、USART_CR1 レジスタの OVER8 ビットをプログラムすることによって選択でき、ボーレートクロックの 16 倍または 8 倍にすることができます（[図 250](#) および [図 251](#)）。

アプリケーションに応じて、

- 高速（最大 $f_{\text{CLK}}/8$ ）を達成するには、8 倍（OVER8=1）のオーバーサンプリングを選択します。この場合、クロック偏差に対するレシーバの最大許容誤差は軽減されます（[セクション 25.4.5：クロック偏差に対する USART レシーバの許容誤差](#)を参照）。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍（OVER8=0）のオーバーサンプリングを選択します。この場合、最大速度は $f_{\text{CLK}}/16$ に制限されます。

論理レベルの評価方法を選択するには、USART_CR3 レジスタの ONEBIT ビットをプログラミングします。これには 2 つのオプションがあります。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NF ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。

アプリケーションに応じて、

- ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式（ONEBIT=0）を選択します。そしてノイズが検出された場合（[図 127](#)を参照）は、サンプリング中にグリッチが発生していることになりしますので、そのデータを除去します。
- ラインがノイズフリーであるときは、1 つのサンプルによる方式（ONEBIT=1）を選択し、クロック偏差に対するレシーバの許容誤差を増やします（[セクション 25.4.5：クロック偏差に対する USART レシーバの許容誤差](#)を参照）。この場合、NF ビットはセットされません。

フレーム内でノイズが検出された場合：

- RXNE ビットの立ち上がりエッジで、NF ビットがセットされます。
- 無効なデータがシフトレジスタから USART_DR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USART_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

NF ビットは、USART_SR レジスタの読み出し動作と、それに続く USART_DR レジスタの読み出し動作によってリセットされます。

注： 8 倍のオーバーサンプリングは、スマートカード、IrDA、および LIN の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって“0”に固定されています。

図 250. データサンプリング (16 倍のオーバーサンプリング)

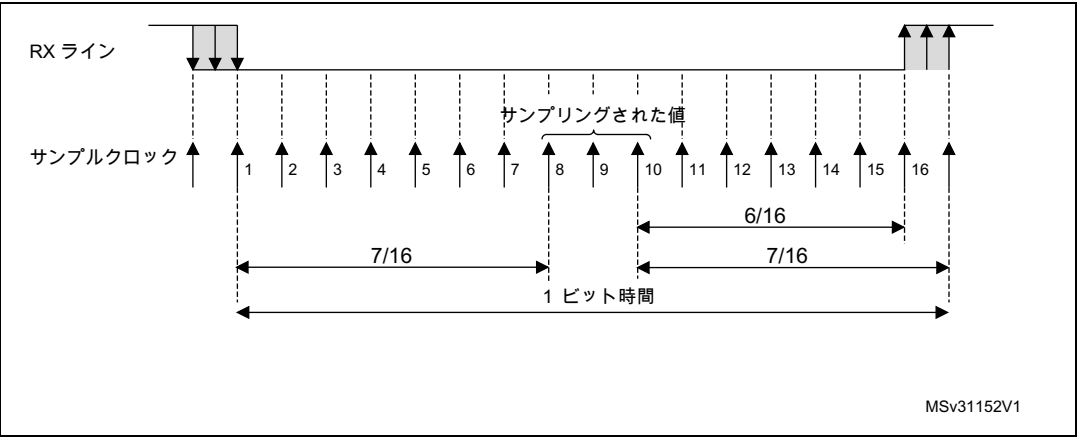


図 251. データサンプリング (8 倍のオーバーサンプリング)

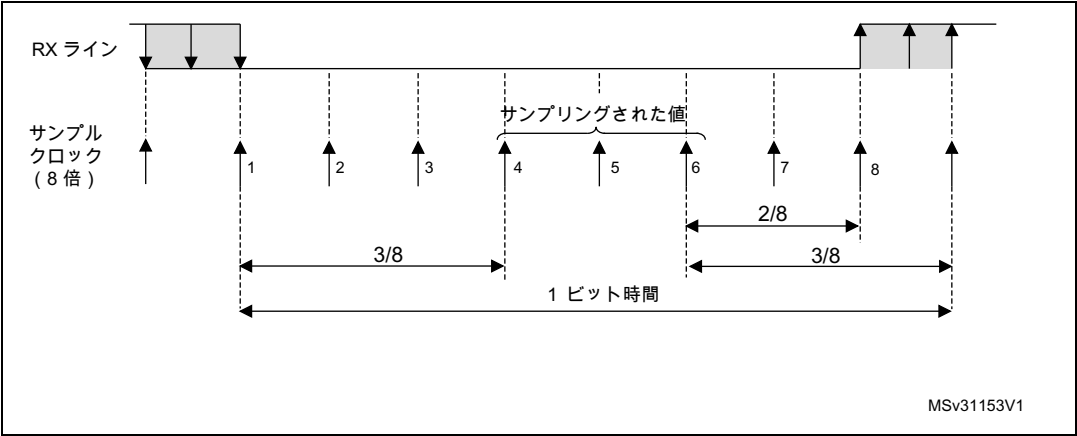


表 127. サンプリングされたデータからのノイズ検出

サンプリングされた値	NE ステータス	受信ビットの値
000	0	0
001	1	0
010	1	0
011	1	1

表 127. サンプリングされたデータからのノイズ検出 (続き)

サンプリングされた値	NE ステータス	受信ビットの値
100	1	0
101	1	1
110	1	1
111	0	1

フレーミングエラー

フレーミングエラーは、次の場合に検出されます。

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USART_DR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USART_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

FE ビットは、USART_SR レジスタの読み出し動作と、それに続く USART_DR レジスタの読み出し動作によってリセットされます。

受信時の設定可能なストップビット

受信するストップビット数は、制御レジスタ 2 の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

1. **0.5 個のストップビット (スマートカードモードでの受信) :** 0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
2. **1 個のストップビット :** ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
3. **1.5 個のストップビット (スマートカードモード) :** スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USART_CR1 レジスタの RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE ビットとともに FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボックロック周期後に) 行われます。1.5 個のストップビットは、2 つの部分に分解できます。すなわち、何も起こらない 0.5 ボックロック周期と、途中でサンプリングが行われる通常の 1 ストップビット周期です。詳細については、[セクション 25.4.11](#) を参照してください。
4. **2 個のストップビット :** ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。最初のストップビットでフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。2 番目のストップビットでは、フレーミングエラーの検査は行われません。RXNE フラグは、最初のストップビットの最後でセットされます。

25.4.4 フラクショナルボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートには、USARTDIV の整数部と小数部にプログラミングされたものと同じ値が設定されます。

式1：標準 USART のボーレート (SPI モードを含む)

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

式2：スマートカード、LIN、および IrDA モードのボーレート

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{16 \times \text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USART_BRR レジスタでコード化されます。

- OVER8=0 のとき、小数部は 4 ビットでコード化され、USART_BRR レジスタの DIV_fraction[3:0] ビットによってプログラミングされます。
- OVER8=1 のとき、小数部は 3 ビットでコード化され、USART_BRR レジスタの DIV_fraction[2:0] ビットによってプログラミングされます。ビット DIV_fraction[3] はクリアされたままである必要があります。

注： ボーカウンタは、USART_BRR への書き込み後、ボーレジスタの新しい値によって更新されます。したがって、通信中はボーレートレジスタの値を変更しないでください。

OVER8=0 のとき、USART_BRR レジスタの値から USARTDIV を得る方法

例 1：

DIV_Mantissa = 0d27 かつ DIV_Fraction = 0d12 (USART_BRR = 0x1BC) の場合、

整数部 (USARTDIV) = 0d27

小数部 (USARTDIV) = 12/16 = 0d0.75

したがって、USARTDIV = 0d27.75

例 2：

USARTDIV = 0d25.62 をプログラミングするには

次のようにします。

DIV_Fraction = 16*0d0.62 = 0d9.92

最も近い実数は 0d10 = 0xA

DIV_Mantissa = 整数部 (0d25.620) = 0d25 = 0x19

USART_BRR = 0x19A であるため、USARTDIV = 0d25.625

例 3：

USARTDIV = 0d50.99 をプログラミングするには

次のようにします。

DIV_Fraction = 16*0d0.99 = 0d15.84

最も近い実数は $0d16 = 0x10 \Rightarrow \text{DIV_frac}[3:0]$ のオーバーフロー \Rightarrow キャリーを整数部に加算する必要があります。

$\text{DIV_Mantissa} = \text{整数部} (0d50.990 + \text{キャリー}) = 0d51 = 0x33$

$\text{USART_BRR} = 0x330$ であるため、 $\text{USARTDIV} = 0d51.000$

OVER8=1 のとき、USART_BRR レジスタの値から USARTDIV を得る方法

例 1 :

$\text{DIV_Mantissa} = 0x27$ かつ $\text{DIV_Fraction}[2:0] = 0d6$ ($\text{USART_BRR} = 0x1B6$) の場合、

整数部 (USARTDIV) = $0d27$

小数部 (USARTDIV) = $6/8 = 0d0.75$

したがって、 $\text{USARTDIV} = 0d27.75$

例 2 :

$\text{USARTDIV} = 0d25.62$ をプログラミングするには

次のようにします。

$\text{DIV_Fraction} = 8 \times 0d0.62 = 0d4.96$

最も近い実数は $0d5 = 0x5$

$\text{DIV_Mantissa} = \text{整数部} (0d25.620) = 0d25 = 0x19$

これによって、 $\text{USART_BRR} = 0x195 \Rightarrow \text{USARTDIV} = 0d25.625$

例 3 :

$\text{USARTDIV} = 0d50.99$ をプログラミングするには

次のようにします。

$\text{DIV_Fraction} = 8 \times 0d0.99 = 0d7.92$

最も近い実数は $0d8 = 0x8 \Rightarrow \text{DIV_frac}[2:0]$ のオーバーフロー \Rightarrow キャリーを仮数部に加算する必要があります。

$\text{DIV_Mantissa} = \text{整数部} (0d50.990 + \text{キャリー}) = 0d51 = 0x33$

これによって、 $\text{USART_BRR} = 0x0330 \Rightarrow \text{USARTDIV} = 0d51.000$

表 128. プログラミングされたボーレートの誤差計算 ($f_{\text{PCLK}} = 8 \text{ MHz}$ または $f_{\text{PCLK}} = 12 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート 7		$f_{\text{PCLK}} = 8 \text{ MHz}$			$f_{\text{PCLK}} = 12 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値)B レート / 目標のB レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0



表 128. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 12 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾ (続き)

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート 7		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 12 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値)B レート / 目標のB レート	実際	ボーレート レジスタに プログラミング されている値	誤差
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 129. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 12 \text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 12 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値)B レート / 目標の B レート	実際	ボーレートレジ スタにプログラ ミングされてい る値	誤差
1	1.2 KBps	1.2 KBps	833.375	0	1.2 KBps	1250	0
2	2.4 KBps	2.4 KBps	416.625	0.01	2.4 KBps	625	0
3	9.6 KBps	9.604 KBps	104.125	0.04	9.6 KBps	156.25	0
4	19.2 KBps	19.185 KBps	52.125	0.08	19.2 KBps	78.125	0
5	38.4 KBps	38.462 KBps	26	0.16	38.339 KBps	39.125	0.16
6	57.6 KBps	57.554 KBps	17.375	0.08	57.692 KBps	26	0.16
7	115.2 KBps	115.942 KBps	8.625	0.64	115.385 KBps	13	0.16
8	230.4 KBps	228.571 KBps	4.375	0.79	230.769 KBps	6.5	0.16
9	460.8 KBps	470.588 KBps	2.125	2.12	461.538 KBps	3.25	0.16
10	921.6 KBps	888.889 KBps	1.125	3.55	923.077 KBps	1.625	0.16

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 130. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16 \text{ MHz}$ または $f_{PCLK} = 24 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 16 \text{ MHz}$			$f_{PCLK} = 24 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) B レート / 目標の B レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 Kbps	1.2 Kbps	833.3125	0	1.2	1250	0
2	2.4 Kbps	2.4 Kbps	416.6875	0	2.4	625	0
3	9.6 Kbps	9.598 Kbps	104.1875	0.02	9.6	156.25	0
4	19.2 Kbps	19.208 Kbps	52.0625	0.04	19.2	78.125	0
5	38.4 Kbps	38.369 Kbps	26.0625	0.08	38.4	39.0625	0
6	57.6 Kbps	57.554 Kbps	17.375	0.08	57.554	26.0625	0.08
7	115.2 Kbps	115.108 Kbps	8.6875	0.08	115.385	13	0.16
8	230.4 Kbps	231.884 Kbps	4.3125	0.64	230.769	6.5	0.16
9	460.8 Kbps	457.143 Kbps	2.1875	0.79	461.538	3.25	0.16
10	921.6 Kbps	941.176 Kbps	1.0625	2.12	923.077	1.625	0.16

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 131. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16 \text{ MHz}$ または $f_{PCLK} = 24 \text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 16 \text{ MHz}$			$f_{PCLK} = 24 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) B レート / 目標の B レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 Kbps	1.2 Kbps	1666.625	0	1.2 Kbps	2500	0
2	2.4 Kbps	2.4 Kbps	833.375	0	2.4 Kbps	1250	0
3	9.6 Kbps	9.598 Kbps	208.375	0.02	9.6 Kbps	312.5	0
4	19.2 Kbps	19.208 Kbps	104.125	0.04	19.2 Kbps	156.25	0
5	38.4 Kbps	38.369 Kbps	52.125	0.08	38.4 Kbps	78.125	0
6	57.6 Kbps	57.554 Kbps	34.75	0.08	57.554 Kbps	52.125	0.08
7	115.2 Kbps	115.108 Kbps	17.375	0.08	115.385 Kbps	26	0.16
8	230.4 Kbps	231.884 Kbps	8.625	0.64	230.769 Kbps	13	0.16
9	460.8 Kbps	457.143 Kbps	4.375	0.79	461.538 Kbps	6.5	0.16
10	921.6 Kbps	941.176 Kbps	2.125	2.12	923.077 Kbps	3.25	0.16
11	2 MBps	2000 Kbps	1	0	2000 Kbps	1.5	0
12	3 MBps	NA	NA	NA	3000 Kbps	1	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

**表 132. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 16 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾**

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 16 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	2.4 Kbps	2.400 Kbps	208.3125	0.00%	2.400 Kbps	416.6875	0.00%
2.	9.6 Kbps	9.604 Kbps	52.0625	0.04%	9.598 Kbps	104.1875	0.02%
3.	19.2 Kbps	19.185 Kbps	26.0625	0.08%	19.208 Kbps	52.0625	0.04%
4.	57.6 Kbps	57.554 Kbps	8.6875	0.08%	57.554 Kbps	17.3750	0.08%
5.	115.2 Kbps	115.942 Kbps	4.3125	0.64%	115.108 Kbps	8.6875	0.08%
6.	230.4 Kbps	228.571 Kbps	2.1875	0.79%	231.884 Kbps	4.3125	0.64%
7.	460.8 Kbps	470.588 Kbps	1.0625	2.12%	457.143 Kbps	2.1875	0.79%
8.	896 Kbps	NA	NA	NA	888.889 Kbps	1.1250	0.79%
9.	921.6 Kbps	NA	NA	NA	941.176 Kbps	1.0625	2.12%

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

**表 133. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 16 \text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾**

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 16 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	2.4 Kbps	2.400 Kbps	416.625	0.01%	2.400 Kbps	833.375	0.00%
2.	9.6 Kbps	9.604 Kbps	104.125	0.04%	9.598 Kbps	208.375	0.02%
3.	19.2 Kbps	19.185 Kbps	52.125	0.08%	19.208 Kbps	104.125	0.04%
4.	57.6 Kbps	57.557 Kbps	17.375	0.08%	57.554 Kbps	34.750	0.08%
5.	115.2 Kbps	115.942 Kbps	8.625	0.64%	115.108 Kbps	17.375	0.08%
6.	230.4 Kbps	228.571 Kbps	4.375	0.79%	231.884 Kbps	8.625	0.64%
7.	460.8 Kbps	470.588 Kbps	2.125	2.12%	457.143 Kbps	4.375	0.79%
8.	896 Kbps	888.889 Kbps	1.125	0.79%	888.889 Kbps	2.250	0.79%
9.	921.6 Kbps	888.889 Kbps	1.125	3.55%	941.176 Kbps	2.125	2.12%
10.	1.792 MBps	NA	NA	NA	1.7777 MBps	1.125	0.79%
11.	1.8432 MBps	NA	NA	NA	1.7777 MBps	1.125	3.55%

- CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

**表 134. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30\text{ MHz}$ または $f_{PCLK} = 60\text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾⁽²⁾**

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 30\text{ MHz}$			$f_{PCLK} = 60\text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	2.4 KBps	2.400 KBps	781.2500	0.00%	2.400 KBps	1562.5000	0.00%
2.	9.6 KBps	9.600 KBps	195.3125	0.00%	9.600 KBps	390.6250	0.00%
3.	19.2 KBps	19.194 KBps	97.6875	0.03%	19.200 KBps	195.3125	0.00%
4.	57.6 KBps	57.582 KBps	32.5625	0.03%	57.582 KBps	65.1250	0.03%
5.	115.2 KBps	115.385 KBps	16.2500	0.16%	115.163 KBps	32.5625	0.03%
6.	230.4 KBps	230.769 KBps	8.1250	0.16%	230.769 KBps	16.2500	0.16%
7.	460.8 KBps	461.538 KBps	4.0625	0.16%	461.538 KBps	8.1250	0.16%
8.	896 KBps	909.091 KBps	2.0625	1.46%	895.522 KBps	4.1875	0.05%
9.	921.6 KBps	909.091 KBps	2.0625	1.36%	923.077 KBps	4.0625	0.16%
10.	1.792 MBps	1.1764 MBps	1.0625	1.52%	1.8182 MBps	2.0625	1.36%
11.	1.8432 MBps	1.8750 MBps	1.0000	1.73%	1.8182 MBps	2.0625	1.52%
12.	3.584 MBps	NA	NA	NA	3.2594 MBps	1.0625	1.52%
13.	3.6864 MBps	NA	NA	NA	3.7500 MBps	1.0000	1.73%

- CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

**表 135. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30\text{ MHz}$ または $f_{PCLK} = 60\text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾⁽²⁾**

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 30\text{ MHz}$			$f_{PCLK} = 60\text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	2.4 KBps	2.400 KBps	1562.5000	0.00%	2.400 KBps	3125.0000	0.00%
2.	9.6 KBps	9.600 KBps	390.6250	0.00%	9.600 KBps	781.2500	0.00%
3.	19.2 KBps	19.194 KBps	195.3750	0.03%	19.200 KBps	390.6250	0.00%
4.	57.6 KBps	57.582 KBps	65.1250	0.16%	57.582 KBps	130.2500	0.03%
5.	115.2 KBps	115.385 KBps	32.5000	0.16%	115.163 KBps	65.1250	0.03%
6.	230.4 KBps	230.769 KBps	16.2500	0.16%	230.769 KBps	32.5000	0.16%



表 135. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30 \text{ MHz}$ または $f_{PCLK} = 60 \text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾⁽²⁾ (続き)

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 30 \text{ MHz}$			$f_{PCLK} = 60 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
7.	460.8 KBps	461.538 KBps	8.1250	0.16%	461.538 KBps	16.2500	0.16%
8.	896 KBps	909.091 KBps	4.1250	1.46%	895.522 KBps	8.3750	0.05%
9.	921.6 KBps	909.091 KBps	4.1250	1.36%	923.077 KBps	8.1250	0.16%
10.	1.792 MBps	1.7647 MBps	2.1250	1.52%	1.8182 MBps	4.1250	1.46%
11.	1.8432 MBps	1.8750 MBps	2.0000	1.73%	1.8182 MBps	4.1250	1.36%
12.	3.584 MBps	3.7500 MBps	1.0000	4.63%	3.5294 MBps	2.1250	1.52%
13.	3.6864 MBps	3.7500 MBps	1.0000	1.73%	3.7500 MBps	2.0000	1.73%
14.	7.168 MBps	NA	NA	NA	7.5000 MBps	1.0000	4.63%
15.	7.3728 MBps	NA	NA	NA	7.5000 MBps	1.0000	1.73%

- CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

表 136. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42 \text{ MHz}$ または $f_{PCLK} = 84 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾⁽²⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 42 \text{ MHz}$			$f_{PCLK} = 84 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 KBps	1.2 KBps	2187.5	0	1.2 KBps	4375	0
2	2.4 KBps	2.4 KBps	1093.75	0	2.4 KBps	2187.5	0
3	9.6 KBps	9.6 KBps	273.4375	0	9.6 KBps	546.875	0
4	19.2 KBps	19.195 KBps	136.75	0.02	19.2 KBps	273.4375	0
5	38.4 KBps	38.391 KBps	68.375	0.02	38.391 KBps	136.75	0.02
6	57.6 KBps	57.613 KBps	45.5625	0.02	57.613 KBps	91.125	0.02
7	115.2 KBps	115.068 KBps	22.8125	0.11	115.226 KBps	45.5625	0.02
8	230.4 KBps	230.769 KBps	11.375	0.16	230.137 KBps	22.8125	0.11
9	460.8 KBps	461.538 KBps	5.6875	0.16	461.538 KBps	11.375	0.16
10	921.6 KBps	913.043 KBps	2.875	0.93	923.076 KBps	5.6875	0.93
11	1.792 MBps	1.826 MBps	1.4375	1.9	1.787 MBps	2.9375	0.27

表 136. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42 \text{ MHz}$ または $f_{PCLK} = 84 \text{ MHz}$)、
16 倍のオーバーサンプリング⁽¹⁾⁽²⁾ (続き)

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 42 \text{ MHz}$			$f_{PCLK} = 84 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
12	1.8432 MBps	1.826 MBps	1.4375	0.93	1.826 MBps	2.875	0.93
13	3.584 MBps	NA	NA	NA	3.652 MBps	1.4375	1.9
14	3.6864 MBps	NA	NA	NA	3.652 MBps	1.4375	0.93

- CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

表 137. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42 \text{ MHz}$ または $f_{PCLK} = 84 \text{ MHz}$)、
8 倍のオーバーサンプリング⁽¹⁾⁽²⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 42 \text{ MHz}$			$f_{PCLK} = 84 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	1.2 KBps	1.2 KBps	4375	0	1.2 KBps	8750	0
2.	2.4 KBps	2.4 KBps	2187.5	0	2.4 KBps	4375	0
3.	9.6 KBps	9.6 KBps	546.875	0	9.6 KBps	1093.75	0
4.	19.2 KBps	19.195 KBps	273.5	0.02	19.2 KBps	546.875	0
5.	38.4 KBps	38.391 KBps	136.75	0.02	38.391 KBps	273.5	0.02
6.	57.6 KBps	57.613 KBps	91.125	0.02	57.613 KBps	182.25	0.02
7.	115.2 KBps	115.068 KBps	45.625	0.11	115.226 KBps	91.125	0.02
8.	230.4 KBps	230.769 KBps	22.75	0.11	230.137 KBps	45.625	0.11
9.	460.8 KBps	461.538 KBps	11.375	0.16	461.538 KBps	22.75	0.16
10.	921.6 KBps	913.043 KBps	5.75	0.93	923.076 KBps	11.375	0.93
11.	1.792 MBps	1.826 MBps	2.875	1.9	1.787 Mbps	5.875	0.27
12.	1.8432 MBps	1.826 MBps	2.875	0.93	1.826 MBps	5.75	0.93
13.	3.584 MBps	3.5 MBps	1.5	2.34	3.652 MBps	2.875	1.9
14.	3.6864 MBps	3.82 MBps	1.375	3.57	3.652 MBps	2.875	0.93
15.	7.168 MBps	NA	NA	NA	7 MBps	1.5	2.34
16.	7.3728 MBps	NA	NA	NA	7.636 MBps	1.375	3.57
18.	9 MBps	NA	NA	NA	9.333 MBps	1.125	3.7
20.	10.5 MBps	NA	NA	NA	10.5 MBps	1	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
2. USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

**表 138. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 100$ MHz または $f_{PCLK} = 50$ MHz)、
16 倍のオーバーサンプリング⁽¹⁾⁽²⁾**

16 倍のオーバーサンプリング (OVER16=1)							
ボーレート		$f_{PCLK} = 100$ MHz			$f_{PCLK} = 50$ MHz		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	9.600 KBps	9.601 KBps	651	0.006	9.601 KBps	325.5	0.006
2.	19.200 KBps	19.201 KBps	325	0.006	19.201 KBps	162.75	0.006
3.	38.400 KBps	38.402 KBps	162.75	0.006	38.402 KBps	81.375	0.006
4.	57.600 KBps	57.603 KBps	108.5	0.006	57.603 KBps	54.25	0.006
5.	115.200 KBps	115.207 KBps	54.25	0.006	115.207 KBps	27.125	0.006
6.	230.400 KBps	230.414 KBps	27.125	0.006	230.414 KBps	13.5625	0.006
7.	460.800 KBps	460.829 KBps	13.5625	0.006	462.962 KBps	6.75	0.47
8.	921.600 KBps	925.925 KBps	6.75	0.470	925.925 KBps	3.375	0.47
9.	3.125 MBps	3.125 MBps	2	0	3.125 MBps	1	0
10.	4.000 MBps	4.000 MBps	1.5625	0	NA	NA	NA
11.	6.250 MBps	6.250 MBps	1	0	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
2. USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

**表 139. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 100$ MHz または $f_{PCLK} = 50$ MHz)、
8 倍のオーバーサンプリング⁽¹⁾⁽²⁾**

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 100$ MHz			$f_{PCLK} = 50$ MHz		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	9.600 KBps	9.601 KBps	1302	0.006	9.601 KBps	651	0.006
2.	19.200 KBps	19.201 KBps	651	0.006	19.201 KBps	325.5	0.006
3.	38.400 KBps	38.402 KBps	325.5	0.006	38.402 KBps	162.75	0.006
4.	57.600 KBps	57.603 KBps	217	0.006	57.603 KBps	108.5	0.006
5.	115.200 KBps	115.207 KBps	108.5	0.006	115.207 KBps	54.25	0.006
6.	230.400 KBps	230.414 KBps	54.25	0.006	230.414 KBps	27.125	0.006

表 139. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 100\text{ MHz}$ または $f_{PCLK} = 50\text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾⁽²⁾ (続き)

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 100\text{ MHz}$			$f_{PCLK} = 50\text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 (%) = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
7.	460.800 Kbps	460.829 Kbps	27.125	0.006	462.962 Kbps	13.5	0.470
8.	921.600 Kbps	925.925 Kbps	13.5	0.470	925.925 Kbps	6.75	0.470
9.	4.000 MBps	4 MBps	3.125	0.000	4.167 MBps	1.5	4.170
10.	6.250 MBps	6.25 MBps	2	0.000	6.250 MBps	1	0.000
11.	12.500 MBps	12.500 MBps	1	0.000	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
2. USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

25.4.5 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバが正しく機能するのは、クロックシステムの合計偏差が USART レシーバ許容誤差を下回る場合に限りです。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$DTRA + DQUANT + DREC + DTCL < \text{USART レシーバ許容誤差}$

データを正しく受信するための USART レシーバ許容誤差は、許容される最大偏差と等しく、以下の選択に依存します。

- USART_CR1 レジスタの M ビットによって定義される 10 または 11 ビットのキャラクタ長
- USART_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- フラクショナルボーレートを使用するかどうか
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USART_CR3 レジスタの ONEBIT ビットの値に依存)。

表 140. USART レシーバ許容誤差 (DIV_Fraction が 0 の場合)

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%



表 141. USART レシーバ許容誤差 (DIV_Fraction が 0 でない場合)

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

注：表 140 と表 141 に示されている数値は、受信したフレームが正確に 10 ビット時間 (M=0 の場合) または 11 ビット時間 (M=1 の場合) のアイドルフレームを含んでいる場合、若干異なることがあります。

25.4.6 マルチプロセッサ通信

複数の USART をネットワークに接続すると、マルチプロセッサ通信を行える可能性があります。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。別の USART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモードでは、

- 受信ステータスビットはセットできません。
- 受信割り込みはすべて禁止されます。
- USART_CR1 レジスタの RWU ビットが 1 にセットされます。RWU ビットは、ハードウェアによって自動的に制御できますが、特定の条件下ではソフトウェアによる書き込みが可能です。

USART は、USART_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

アイドルライン検出 (WAKE=0)

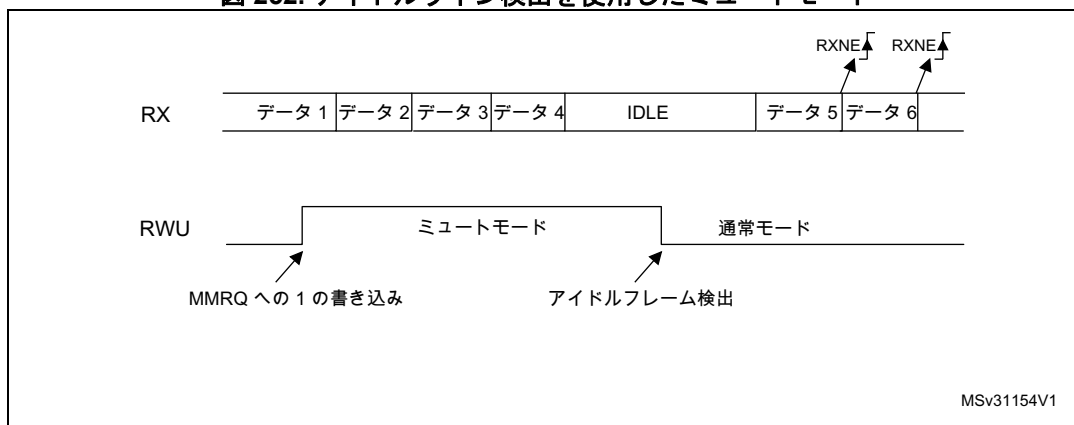
RWU ビットに 1 が書き込まれると、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USART_SR レジスタの IDLE ビットはセットされません。RWU ビットには、ソフトウェアによって 0 を書き込むこともできます。

アイドルライン検出を使用したミュートモードの動作例を [図 252](#) に示します。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス (株) が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス (株) は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

図 252. アイドルライン検出を使用したミュートモード



アドレスマーク検出 (WAKE=1)

このモードでは、MSB が“1”のバイトはアドレスとして、そうでないバイトはデータとして、それぞれ認識されます。アドレスバイトでは、ターゲットレシーバのアドレスは下位 4 ビットになります。レシーバは、この 4 ビットワードを、USART_CR2 レジスタの ADD ビットにプログラミングされている自分のアドレスと比較します。

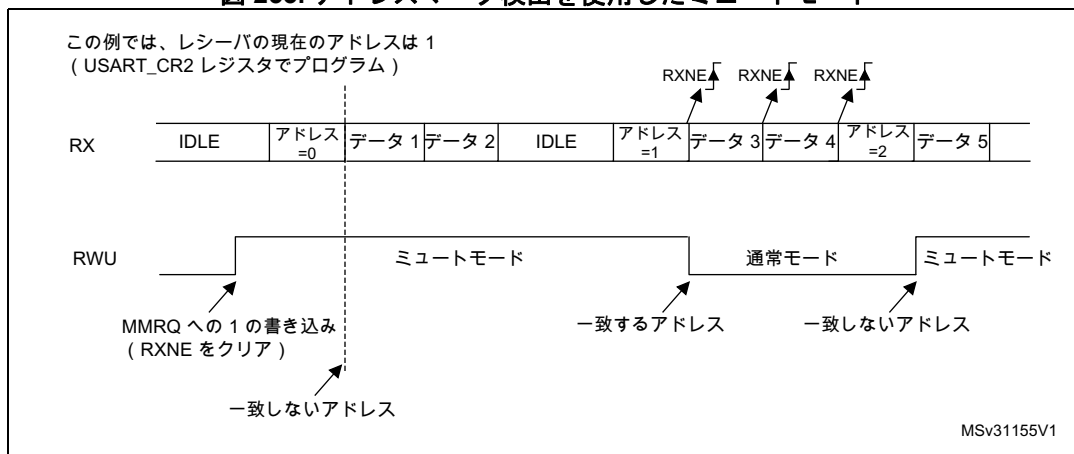
プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART はミュートモードに入っているため、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

プログラミングされたアドレスと一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE ビットがセットされます。

レシーババッファにデータが存在しない場合 (USART_SR レジスタの RXNE=0)、RWU ビットに 0 または 1 を書き込むことができます。そうでない場合、書き込みは無視されます。

アドレスマーク検出を使用したミュートモードの動作例を 図 253 に示します。

図 253. アドレスマーク検出を使用したミュートモード



25.4.7 パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティ検査）を有効にするには、USART_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な USART フレームフォーマットを 表 142 に示します。

表 142. フレームフォーマット

M ビット	PCE ビット	USART フレーム ⁽¹⁾
0	0	SB 8 ビットデータ STB
0	1	SB 7 ビットデータ PB STB
1	0	SB 9 ビットデータ STB
1	1	SB 8 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。

偶数パリティ

下位 7 ビットまたは 8 ビット（M=0 または M=1 に応じて）とパリティビットから構成されるフレーム内で、「1」の数が偶数になるように、パリティビットが計算されます。

例：データ = 00110101、4 ビットがセット => 偶数パリティが選択された場合（USART_CR1 の PS=0）、パリティビットは 0 になります。

奇数パリティ

下位 7 ビットまたは 8 ビット（M=0 または M=1 に応じて）とパリティビットから構成されるフレーム内で、「1」の数が奇数になるように、パリティビットが計算されます。

例：データ = 00110101、4 ビットがセット => 奇数パリティが選択された場合（USART_CR1 の PS=1）、パリティビットは 1 になります。

受信中のパリティチェック

パリティチェックに失敗した場合、USART_SR レジスタの PE フラグがセットされ、USART_CR1 レジスタの PEIE ビットがセットされている場合は割込みが生成されます。PE フラグは、ソフトウェアシーケンスによってクリアされます（ステータスレジスタの読み出しと、それに続く USART_DR データレジスタへの読み出しまたは書き込みアクセス）。

注： アドレスマークによるウェイクアップの場合、アドレス（パリティビットではありません）の特定にはデータの最上位ビットが考慮されます。そしてレシーバは、アドレスデータのパリティをチェックしません（パリティエラーの場合、PE はセットされません）。

送信中のパリティ生成

USART_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます（偶数パリティが選択された場合（PS=0）は偶数個の「1」、奇数パリティが選択された場合（PS=1）は奇数個の「1」）。

注： 送信を管理するソフトウェアルーチンは、PE フラグをクリアするソフトウェアシーケンス（ステータスレジスタの読み出しに続くデータレジスタへの読み出しまたは書き込みアクセス）を起動できます。半二重モードでは、ソフトウェアによっては PE フラグが不意にクリアされることがあります。

25.4.8 LIN (Local Interconnection Network) モード

LIN モードを選択するには、USART_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの STOP[1:0] および CLKEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

LIN 送信

LIN マスタの送信では、通常の USART 送信の手順ではなく、[セクション 25.4.2](#) で説明したものと同じ手順を適用する必要がありますが、次の点が異なります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBK ビットをセットすると、13 個の“0”ビットがブレークキャラクタとして送信されます。その後、値“1”のビットが送信され、次の START 検出が可能になります。

LIN 受信

USART インタフェースにはブレーク検出回路が実装されています。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生の際にブレークが検出できます。

レシーバが有効になると (USART_CR1 レジスタの RE=1)、RX 入力の START 信号を探します。スタートビットの検出方法は、ブレークキャラクタやデータの検索方法と同じです。スタートビットが検出された後、データの場合とまったく同様に次のビットがサンプリングされます (8、9、および 10 番目のサンプル)。10 個 (USART_CR2 レジスタの LBDL=0) または 11 個 (USART_CR2 レジスタの LBDL=1) の連続した“0”ビットが検出され、その後、デリミタキャラクタが続く場合、USART_SR レジスタの LBD フラグがセットされます。LBDIE ビットが 1 の場合、割り込みが生成されます。ブレークを確認する前に、RX ラインがハイレベルに戻ったことを知らせるデリミタが検査されます。

この 10 または 11 が発生する前に“1”がサンプリングされた場合、ブレーク検出回路は現在の検出をキャンセルし、再びスタートビットを検索します。

LIN モードが無効にされた場合 (LINEN=0)、レシーバは、ブレーク検出を考慮することなく、通常の USART として機能し続けます。

LIN モードが有効にされた場合 (LINEN=1)、フレーミングエラーが発生 (つまり、ブレークフレームの場合と同様に、“0”の位置でストップビットを検出) すると、レシーバは停止し、ブレーク検出回路が“1” (ブレークワードが完全でなかった場合) またはデリミタキャラクタ (ブレークが検出された場合) を受信するまで停止状態を維持します。

ブレーク検出回路ステートマシンの動作とブレークフラグを [図 254](#) に示します。

ブレークフレームの例を [図 255](#) に示します。LBDL=1 (11 ビットブレーク長)、M=0 (8 ビットデータ) を想定しています。

図 254. LIN モードでのブレーク検出（11 ビットブレーク長、LBDL=1）

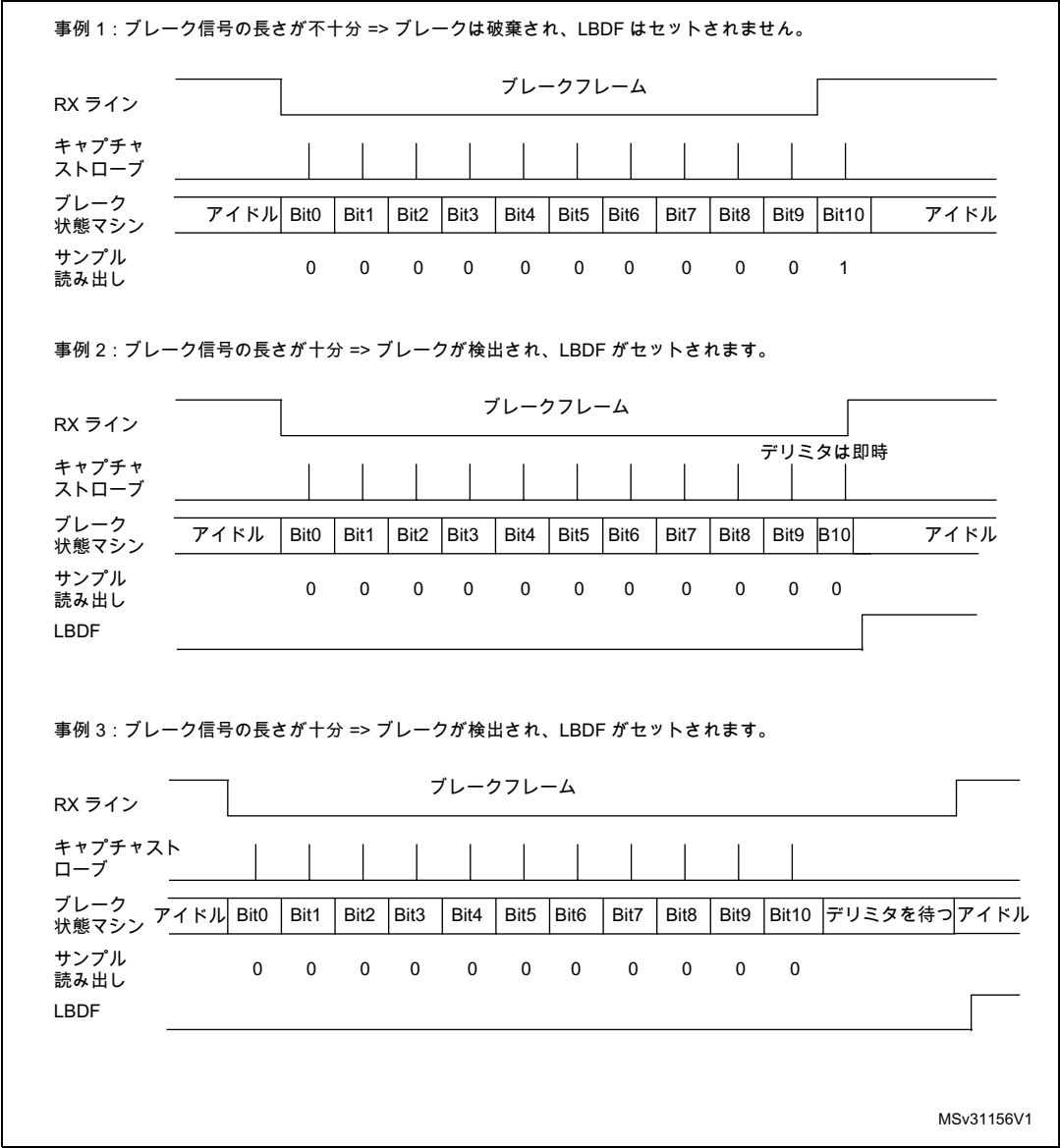
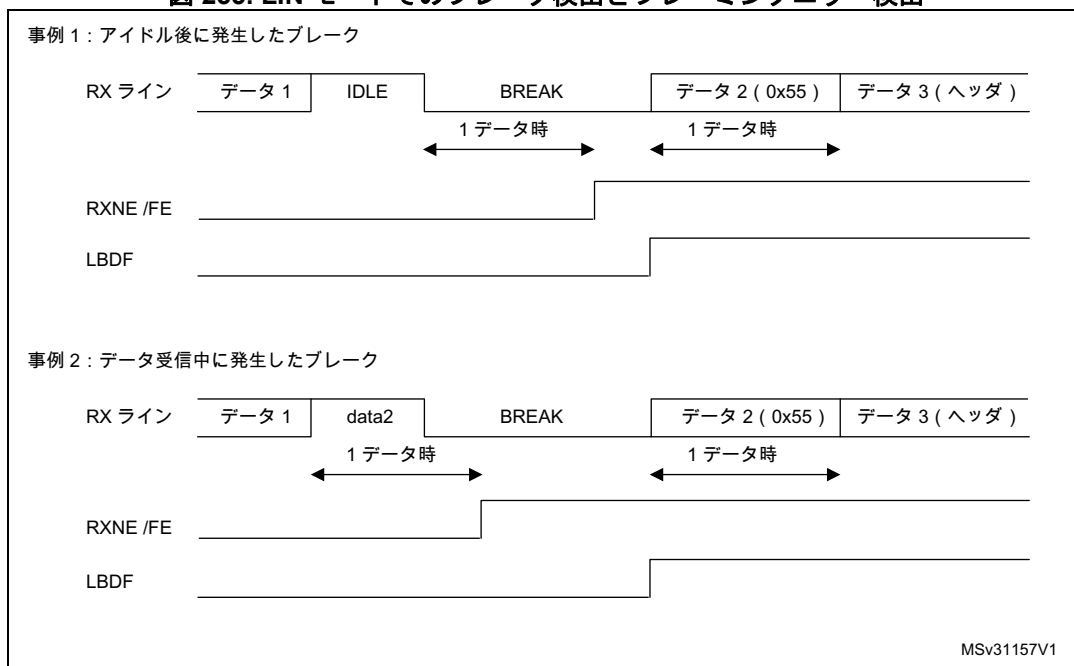


図 255. LIN モードでのブレーク検出とフレーミングエラー検出



25.4.9 USART 同期モード

同期モードを選択するには、USART_CR2 レジスタの CLKEN ビットに 1 を書き込みます。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

USART では、マスタモードで双方向の同期シリアル通信を制御できます。SCLK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、SCLK ピンにクロックパルスは送信されません。USART_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USART_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用でき、USART_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用できます（[図 256](#)、[図 257](#)、および [図 258](#) を参照）。

アイドル時、プリアンブル処理時、およびブレーク送信時には、外部 SCLK クロックは起動されません。

同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、CPOL と CPHA に基づいて SCLK が TX と同期すると、TX 上のデータが同期します。

このモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE=1 の場合、データはオーバーサンプリングなしで、SCLK (CPOL と CPHA に応じて立上りまたは立下りエッジ) でサンプリングされます。ボーレート (1/16 ビット持続時間) に応じたセットアップ時間とホールド時間を守る必要があります。

注： SCLK ピンはTX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (データレジスタ USART_DR への書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。

トランスミッタとレシーバが無効にされた場合 (TE=RE=0)、クロックパルスが正常に機能するには、LBCL、CPOL、および CPHA の各ビットを選択する必要があります。これらのビットは、トランスミッタまたはレシーバが有効である間は、変更しないでください。

レシーバのセットアップ時間とホールド時間を最小限に抑えるため、TE ビットと RE ビットを同じ命令でセットすることをお勧めします。

USART は、マスタモードのみをサポートし、入力クロックに関連するデータを送受信することはできません (SCLK は常に出力です)。

図 256. USART の同期送信の例

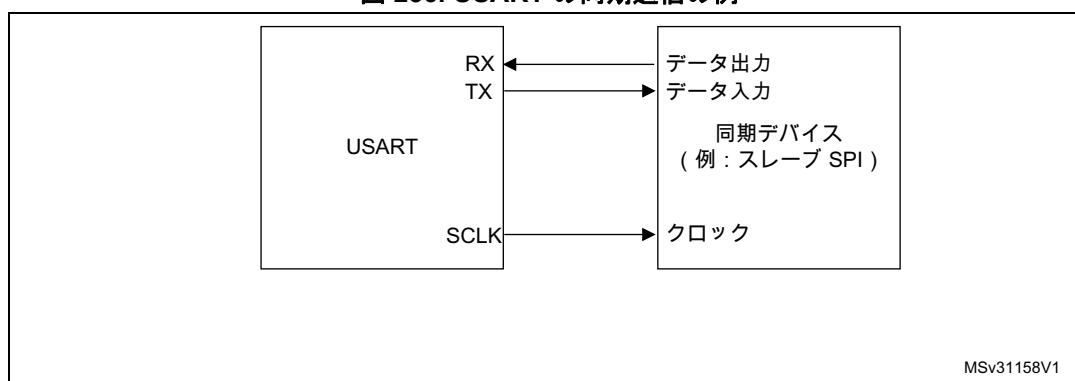


図 257. USART データクロックタイミング図 (M=0)

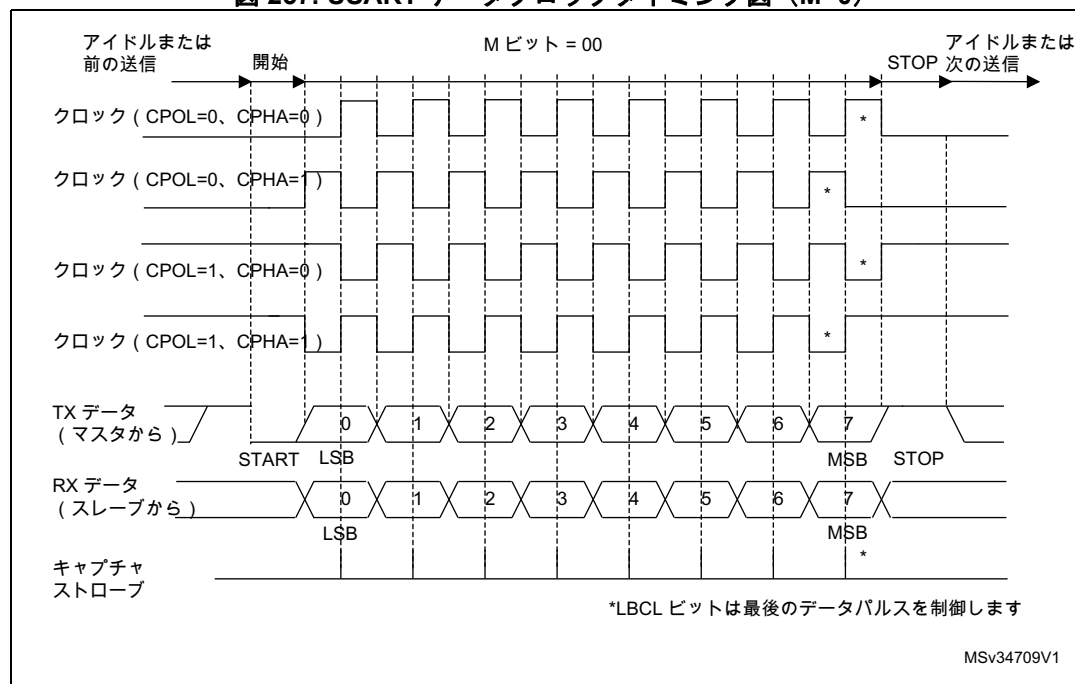


図 258. USART データクロックタイミング図 (M=1)

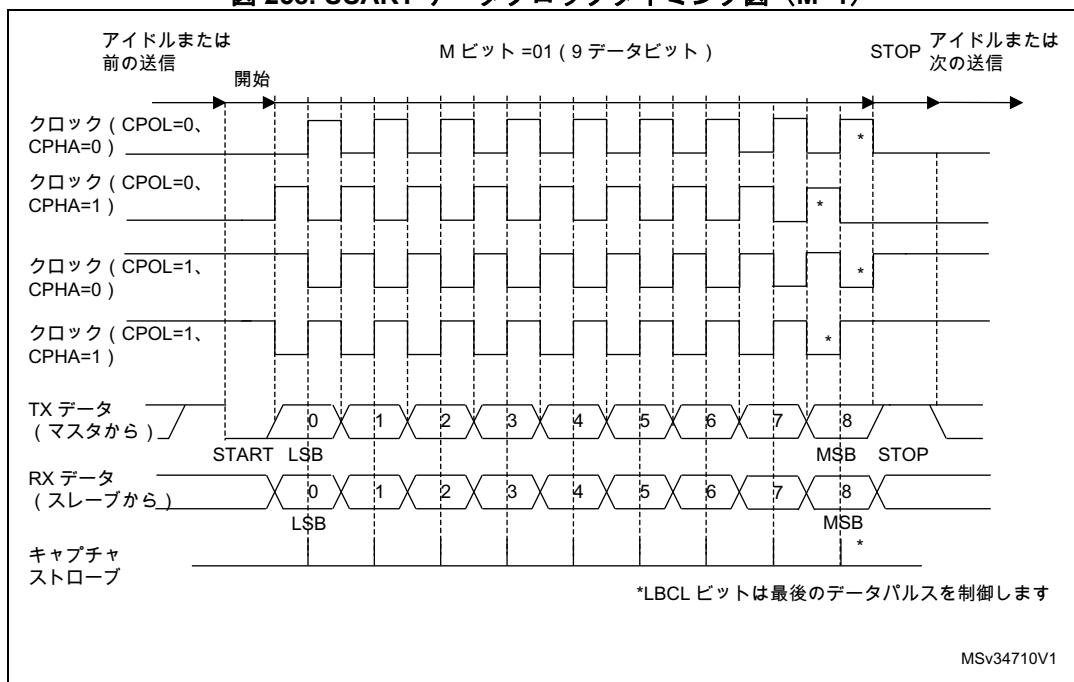
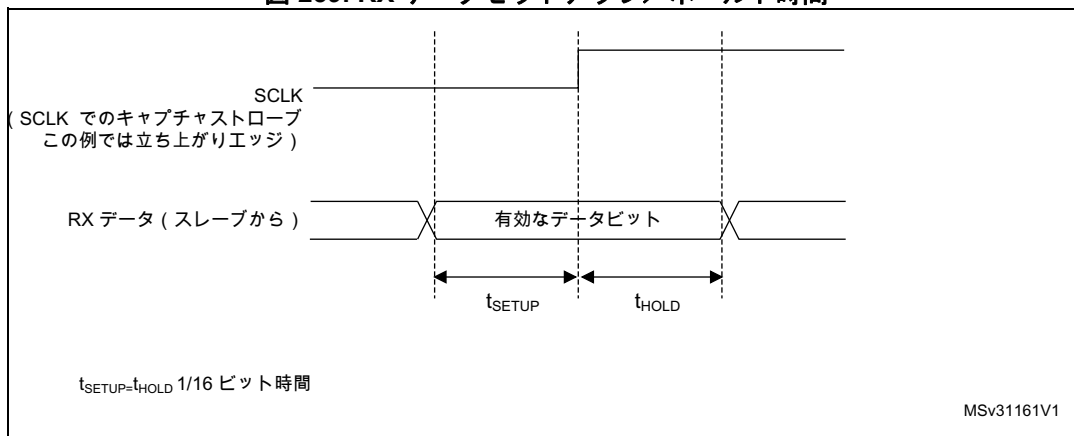


図 259. RX データセットアップ/ホールド時間



注: スマートカードモードでは SCLK の機能が異なります。詳細については、スマートカードモードの章を参照してください。

25.4.10 単線半二重通信

単線半二重モードを選択するには、USART_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN および CLKEN ビット
- USART_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、制御ビット“HALF DUPLEX SEL” (USART_CR3 レジスタの HDSEL) で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が USART によって駆動されないときはフローティング入力（または出力ハイのオープンドレイン）として設定されるように、入出力を設定する必要があります。

この点を除くと、通信は通常の USART モードの場合と同様です。ラインの競合はソフトウェアによって管理する必要があります（たとえば、集中型アービタを使用）。特に、TE ビットがセットされている間は、送信はハードウェアによってブロックされることはなく、データがデータレジスタに書き込まれると送信を続行します。

25.4.11 スマートカード

スマートカードモードを選択するには、USART_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの HDSEL および IREN ビット

さらに、スマートカードにクロックを供給するために CLKEN ビットをセットすることもできます。

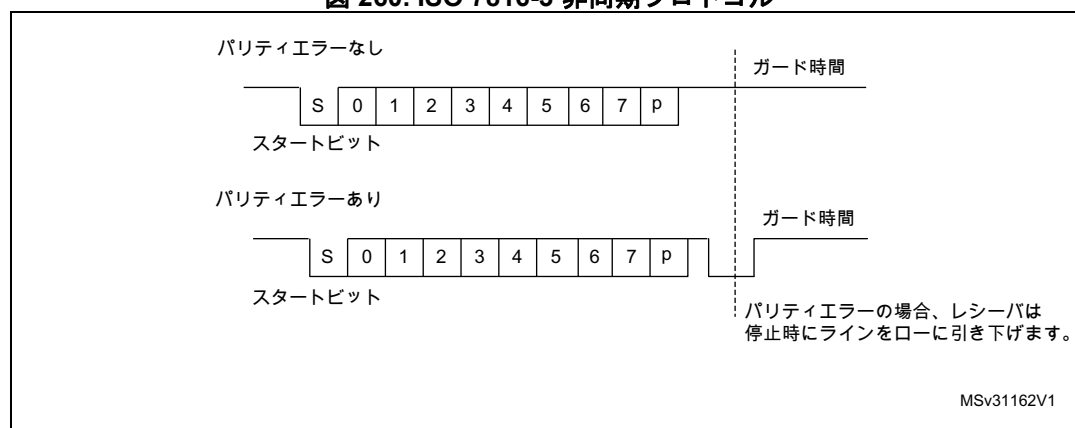
スマートカードインタフェースは、ISO 7816-3 標準で定義された非同期プロトコルスマートカードをサポートするように設計されています。USART は次のように設定してください。

- 8 ビット+パリティ：M ビット=1 および USART_CR1 レジスタの PCE=1
- 送受信時の 1.5 個の STOP ビット：USART_CR2 レジスタの STOP=11

注： 受信にはストップビット 0.5 個を選択することも可能ですが、2 つの設定間の切り替えを避けるために、送信と受信の両方にストップビット 1.5 個を使用することを推奨します。

図 260 に、パリティエラーの有無によるデータラインの状況の変化の例を示します。

図 260. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、やはりスマートカードによって駆動される双方向ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードは、単線半二重通信プロトコルです。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されます。通常動作では、満杯の送信シフトレジスタは、次のボークロックエッジでシフト動作を開始

します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。

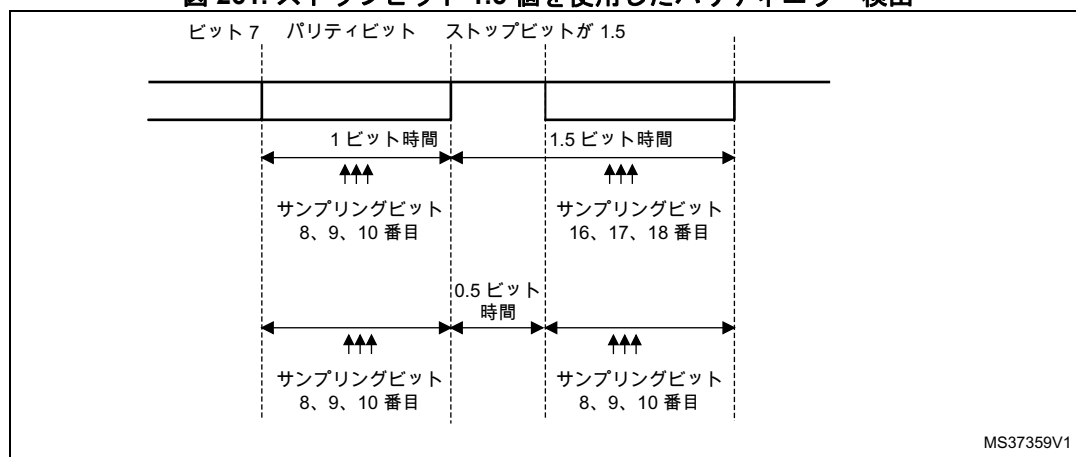
- ストップビット 0.5 または 1.5 周期でプログラミングされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボークロック期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。この NACK 信号（1 ボークロックの間、送信ラインをローに引き下げ）は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。アプリケーションは、プロトコルに基づいて、データの再送信に対処できます。NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって“NACK”されます。そうでない場合、NACK は送信されません。
- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ側でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボークロック周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

注： スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

図 261 に、USART による NACK 信号のサンプリング方法を示します。この例では、USART はデータを送信中であり、ストップビットが 1.5 個組み込まれています。データと NACK 信号の整合性を検査するために、USART のレシーバ部が有効にされます。

図 261. ストップビット 1.5 個を使用したパリティエラー検出



USART は、SCLK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、SCLK は通信に関係せず、5 ビットのプリスケアラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、プリスケアラレジスタ USART_GTPR で設定されます。SCLK 周波数は、 $f_{CK}/2$ から $f_{CK}/62$ までの範囲でプログラムでき、 f_{CK} はペリフェラル入力クロックです。

25.4.12 Ir SIR ENDEC ブロック

IrDA モードを選択するには、USART_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USART_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します (図 262 を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbps のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力が高レベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである (つまり、USART が IrDA エンコーダにデータを送信している) 場合、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。
- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます (図 263 を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。
- IrDA 仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ端のグリッチ検出回路は、PSC 2 周期 (PSC は IrDA 低電力ポーレレジスタ USART_GTPR にプログラミングされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上 2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低電力トランスミッタと通信できます。
- IrDA モードでは、USART_CR2 レジスタのストップビットを「1 ストップビット」に設定する必要があります。

IrDA 低電力モード

トランスミッタ:

低電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低電力ボーレート（最小で 1.42 MHz）の 3 倍となります。一般に、この値は 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz) です。低電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

レシーバ:

低電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は 1/PSC よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低電力ボーレート (USART_GTPR の PSC 値) の 2 周期分を超える場合にのみ受け入れられます。

注: PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。

レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 262. IrDA SIR ENDEC- ブロック図

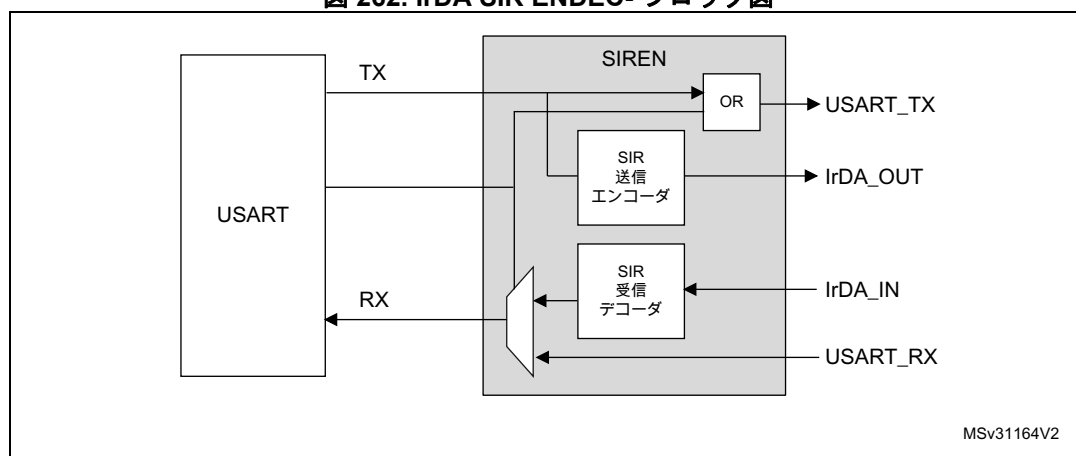
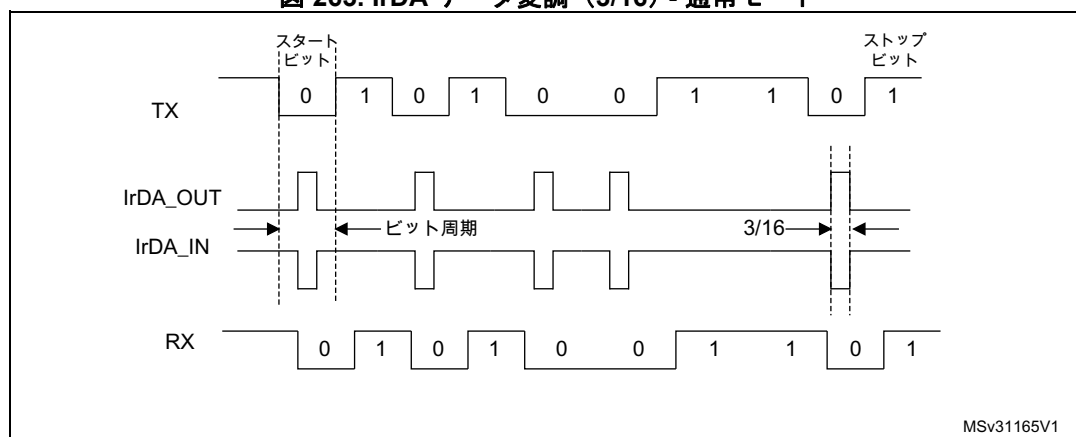


図 263. IrDA データ変調 (3/16) - 通常モード



25.4.13 DMA を使用した連続通信

USART は、DMA を使用して通信を継続することができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

DMA を使用した送信

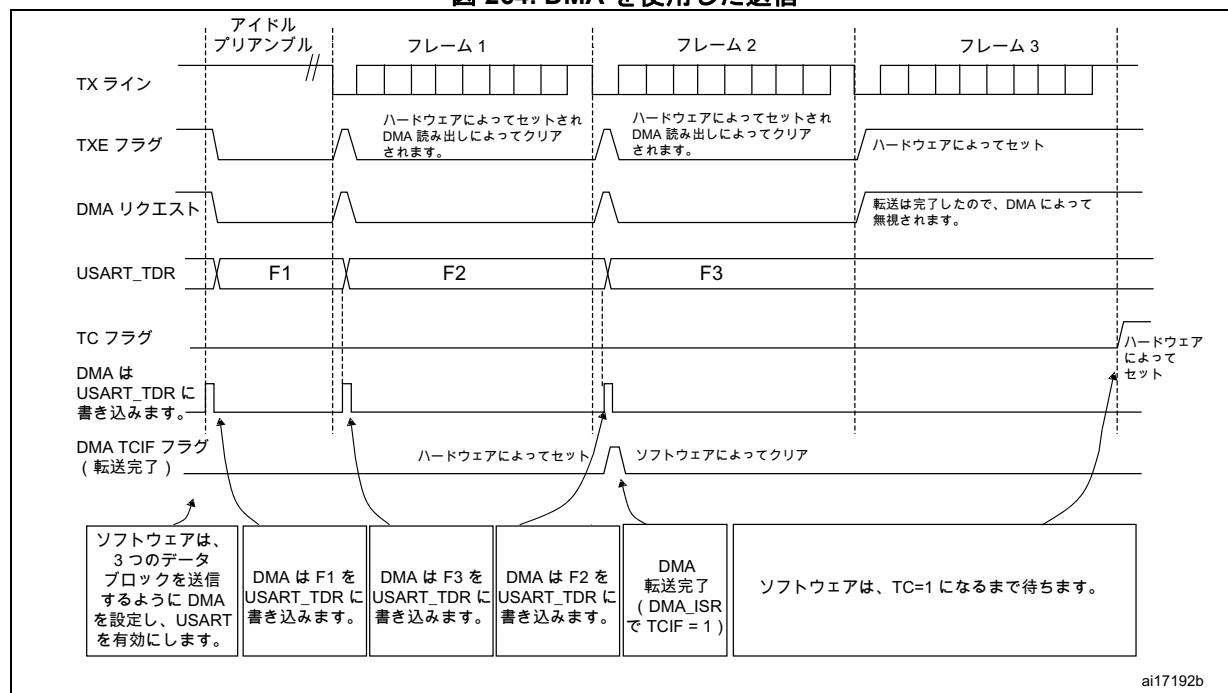
DMA モードでの送信を有効にするには、USART_CR3 レジスタの DMAT ビットをセットします。TXE ビットがセットされるたびに、データは、DMA ペリフェラル（DMA 仕様を参照）を使用して設定された SRAM 領域から USART_DR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します（x はチャンネル番号を示します）

1. DMA 制御レジスタに USART_DR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE イベント後にメモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE イベント後に、このメモリ領域から USART_DR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. SR レジスタの TC ビットに 0 を書き込んでクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと（DMA_ISR レジスタの TCIF フラグがセットされます）、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にしたり STOP モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、データ転送中はクリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

図 264. DMA を使用した送信



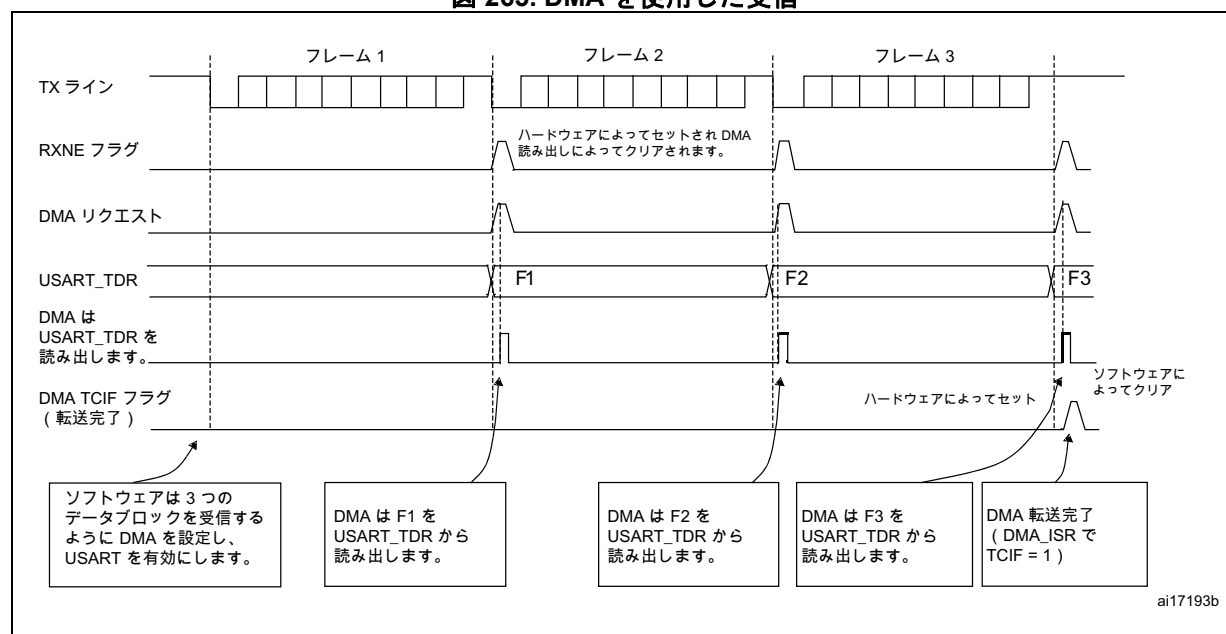
DMA を使用した受信

DMA モードでの受信を有効にするには、USART_CR3 レジスタの DMAR ビットをセットします。データは、データバイトが受信されるたびに、USART_DR レジスタから DMA ペリフェラル (DMA 仕様を参照) を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USART_DR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE イベントの後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE イベントの後に、USART_DR レジスタからこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。割り込み処理サブルーチンでは、USART_CR3 レジスタの DMAR ビットをソフトウェアでクリアする必要があります。

図 265. DMA を使用した受信



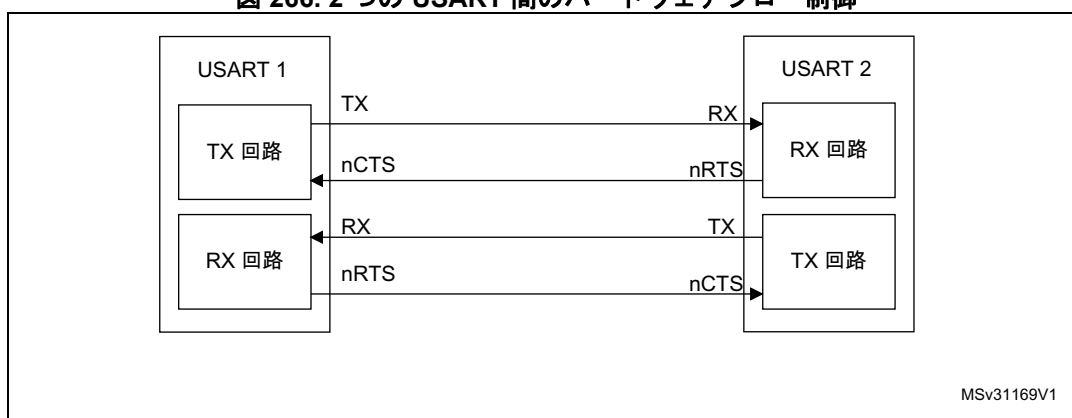
マルチバッファ通信における割り込み生成とエラーフラグ

マルチバッファ通信でトランザクションの処理中にエラーが発生した場合、現在のバイトの後でエラーフラグがアサートされます。割り込み有効フラグがセットされている場合、割り込みが生成されます。1 バイト受信において RXNE とともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別個のエラーフラグ割り込み有効ビット (USART_CR3 レジスタの EIE ビット) が用意されており、これがセットされている場合、いずれかのエラーを持つ現在のバイトの後で割り込みが発行されます。

25.4.14 ハードウェアフロー制御

nCTS 入力と nRTS 出力を使用すると、2 つのデバイス間でシリアルデータフローを制御できます。[図 266](#) に、このモードで 2 つのデバイスを接続する方法を示します。

図 266. 2 つの USART 間のハードウェアフロー制御

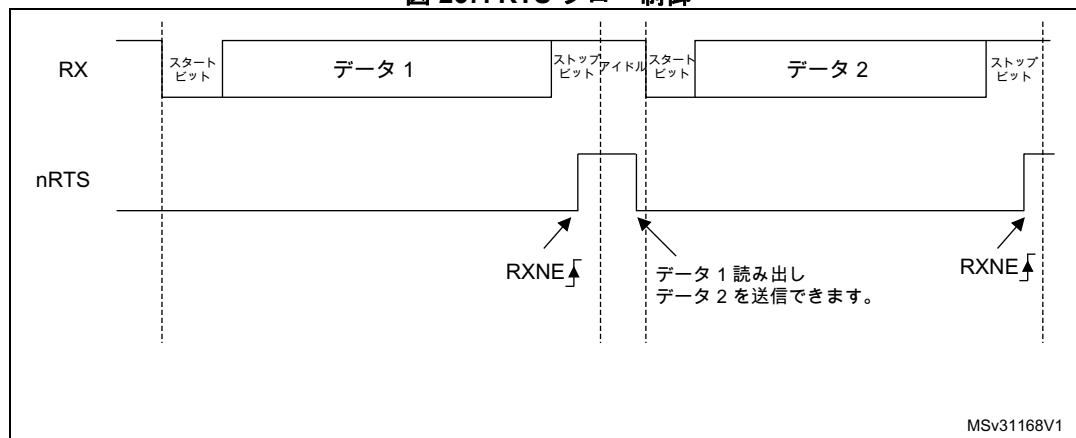


RTS と CTS のフロー制御は、USART_CR3 レジスタの RTSE ビットと CTSE ビットにそれぞれ 1 を書き込むことによって、個別に有効にできます。

RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、nRTS がアサートされます (ローレベル接続)。受信レジスタが満杯になると nRTS がネゲートされ、現在のフレームの終わりに送信が停止する予定であることを示します。図 267 に、RTS フロー制御が有効な場合の通信例を示します。

図 267. RTS フロー制御

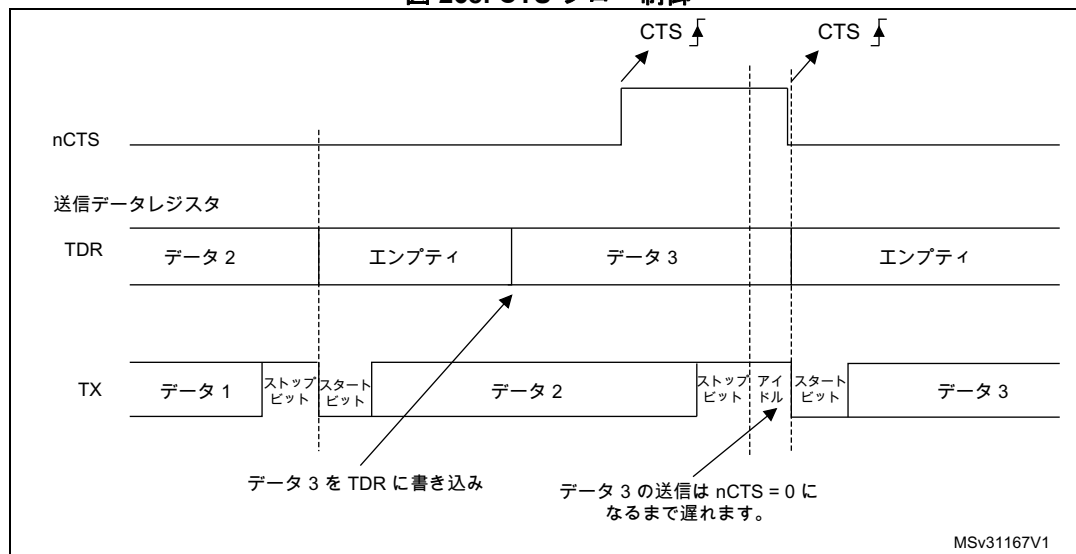


CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、nCTS 入力を検査してから、次のフレームを送信します。nCTS がアサートされていれば (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE=0 の場合)。そうでない場合、送信は行われません。送信中に nCTS がネゲートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、nCTS 入力にトグルすると、CTSIF ステータスビットはハードウェアによって自動的にセットされます。このビットは、レシーバの通信準備ができていないかどうかを示します。USART_CR3 レジスタの CTSIE ビットがセットされている場合、割り込みが生成されます。次の図に、CTS フロー制御が有効な場合の通信例を示します。

図 268. CTS フロー制御



注： ブレークフレームの特殊な動作：CTS フローが有効になると、トランスミッタは、ブレークを送信するために nCTS 入力の状態をチェックしません。

25.5 USART 割り込み

表 143. USART 割り込みリクエスト

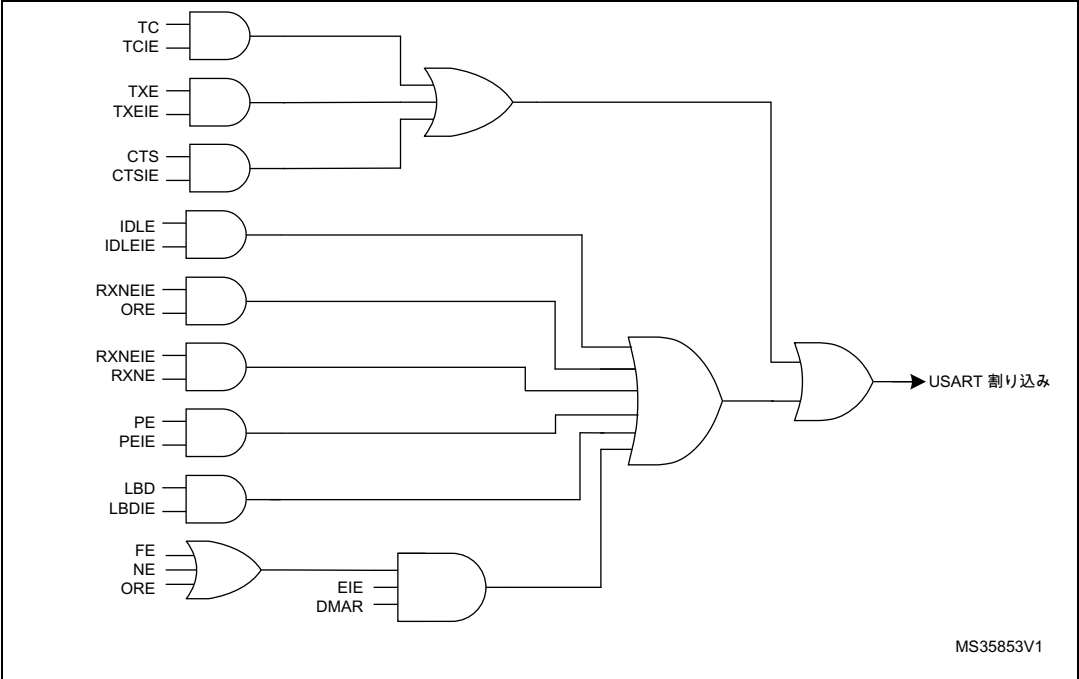
割り込みイベント	イベントフラグ	イネーブル制御ビット
送信データレジスタエンプティ	TXE	TXEIE
CTS フラグ	CTS	CTSIE
送信完了	TC	TCIE
受信データ読み出し可能	RXNE	RXNEIE
オーバーランエラー検出	ORE	
アイドルライン検出	IDLE	IDLEIE
パリティエラー	PE	PEIE
ブレークフラグ	LBD	LBDIE
ノイズフラグ、オーバーランエラー、およびフレーミングエラー (マルチバッファ通信)	NF または ORE または FE	EIE

USART 割り込みイベントは、割り込みベクタに接続されます (図 269 を参照)。

- 送信時：送信完了割り込み、CTS 割り込み、または送信データレジスタエンプティ割り込み
- 受信時：アイドルライン検出、オーバーランエラー、受信データレジスタノットエンプティ、パリティエラー、LIN ブレーク検出、ノイズフラグ (マルチバッファ通信時のみ)、およびフレーミングエラー (マルチバッファ通信時のみ)

これらのイベントは、対応する有効制御ビットがセットされている場合に割り込みを生成します。

図 269. USART 割り込みマッピング図



25.6 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

25.6.1 ステータスレジスタ (USART_SR)

アドレスオフセット：0x00

リセット値：0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

ビット 31:10 予約済みであり、リセット値のままにしておかなければなりません。

ビット 9 **CTS** : CTS フラグ

このビットは、CTSE ビットがセットされていた場合、nCTS 入力がトグルしたときにハードウェアによってセットされます。ソフトウェアで 0 を書き込むことによってクリアされます。また、USART_CR3 レジスタで CTSIE=1 であれば、割り込みが生成されます。

0 : nCTS ステータスラインでの変更はありません。

1 : nCTS ステータスラインで変更がありました。

注： このビットは UART4 および UART5 では使用できません。

ビット 8 **LBD** : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。ソフトウェアで 0 を書き込むことによってクリアされます。USART_CR2 レジスタの LBDIE=1 である場合、割り込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

注： LBDIE=1 の場合、LBD=1 になると割り込みが生成されます。

ビット 7 **TXE** : 送信データレジスタエンプティ

このビットは、TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。USART_CR1 レジスタの TXEIE ビット=1 の場合、割り込みが生成されます。このビットは、USART_DR レジスタへの書き込みによってクリアされます。

0 : データはシフトレジスタに転送されていません。

1 : データはシフトレジスタに転送されます。

注： このビットは、シングルバッファ送信時に使用されます。

ビット 6 **TC** : 送信完了

データを含むフレームの送信が完了し、TXE がセットされている場合、このビットはハードウェアによってセットされます。USART_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。このビットは USART_SR レジスタを読み出し、その後に USART_DR レジスタへの書き込みを行うソフトウェアシーケンスによってクリアされます。TC ビットは、“0”を書き込むことによってもクリアできます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

0 : 送信は完了していません。

1 : 送信は完了しています。

ビット 5 **RXNE** : 読み出しデータレジスタノットエンプティ

このビットは、RDR シフトレジスタの内容が USART_DR レジスタに転送されると、ハードウェアによってセットされます。USART_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。このビットは、USART_DR レジスタへの読み出しによってクリアされます。RXNE フラグは、0 を書き込むとクリアできます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 **IDLE** : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USART_CR1 レジスタの IDLEIE=1 である場合、割り込みが生成されます。このビットは USART_SR レジスタを読み出し、その後に USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注： RXNE ビットがセットされる（つまり、新しいアイドルラインが出現する）まで、IDLE ビットが再びセットされることはありません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE=1 のときに、シフトレジスタで現在受信中のワードを RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USART_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

注： このビットがセットされると、RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。マルチバッファ通信で EIE ビットがセットされた場合、ORE フラグに基づいて割り込みが生成されます。

ビット 2 NF : ノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

注： このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。マルチバッファ通信で EIE ビットがセットされている場合は、NF フラグに基づいて割り込みが生成されます。

注： ラインがノイズフリーであるとき、NF フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます (セクション 25.4.5 : クロック偏差に対する USART レシーバの許容誤差 (765 ページ) を参照)。

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注： このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。現在転送中のワードによってフレームエラーとオーバーランエラーが発生すると、そのワードが転送され、ORE ビットのみがセットされます。

マルチバッファ通信で EIE ビットがセットされている場合は、FE フラグに基づいて割り込みが生成されます。

ビット 0 PE : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。このビットはステータスレジスタを読み出し、その後 USART_DR データレジスタへの読み出しまたは書き込みアクセスを行うソフトウェアシーケンスによってクリアされます。ソフトウェアは、RXNE フラグがセットされるまで待たなければ、PE ビットをクリアできません。

USART_CR1 レジスタの PEIE = 1 の場合、割り込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

25.6.2 データレジスタ (USART_DR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8:0 **DR[8:0]** : データ値

読み出ししか書き込みかに応じて、受信または送信のデータキャラクタが含まれます。
データレジスタは、送信用 (TDR) と受信用 (RDR) の 2 個のレジスタから構成されるため、2 つの機能 (読み出しと書き込み) を実行します。
TDR レジスタは、内部バスと出力シフトレジスタとの間に並列インタフェースを提供します (図 1 を参照)。
RDR レジスタは、入力シフトレジスタと内部バスとの間に並列インタフェースを提供します。
パリティを有効にして (USART_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。
パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

25.6.3 ボーレートレジスタ (USART_BRR)

注 : **TE または RE ビットがそれぞれ無効になると、ボーカウンタはカウントを停止します。**

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:4 **DIV_Mantissa[11:0]** : USARTDIV の整数部

これらの 12 ビットは、USART 分周比 (USARTDIV) の整数部を定義します。

ビット 3:0 **DIV_Fraction[3:0]** : USARTDIV の小数部

これらの 4 ビットは、USART 分周比 (USARTDIV) の小数部を定義します。OVER8=1 のとき、DIV_Fraction3 ビットは考慮されないため、クリアされた状態に保つ必要があります。

25.6.4 制御レジスタ 1 (USART_CR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15 **OVER8** : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

注 : 8 倍のオーバーサンプリングは、スマートカード、IrDA、および LIN の各モードでは使用できません。SCEN=1、IREN=1、または LINEN=1 であるとき、OVER8 はハードウェアによって“0”に固定されます。

ビット 14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13 **UE** : USART 有効

このビットがクリアされると、USART プリスケアラと出力は停止され、現在のバイト転送を最後にして、

消費電力を低減することができます。このビットは、ソフトウェアによってセット／クリアされます。

0 : USART プリスケアラと出力は無効です。

1 : USART は有効です。

ビット 12 **M** : ワード長

このビットは、ワード長を決定します。ソフトウェアによってセット／クリアされます。

0 : スタートビット 1 個、データビット 8 個、ストップビット n 個

1 : スタートビット 1 個、データビット 9 個、ストップビット n 個

注 : M ビットは、送受信のデータ転送中に変更することはできません。

ビット 11 **WAKE** : ウェイクアップ方法

このビットは、USART のウェイクアップ方法を決定し、ソフトウェアによってセット／クリアされます。

0 : アイドルライン

1 : アドレスマーク

ビット 10 **PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御（生成と検出）を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置（M=1 の場合はビット 9、M=0 の場合はビット 8）に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

ビット 9 **PS** : パリティ選択

このビットは、パリティの生成／検出が有効である（PCE ビットがセットされている）とき、奇数パリティ／偶数パリティを選択します。ソフトウェアによってセット／クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ



ビット 8 PEIE : PE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USART_SR レジスタで PE=1 の時に必ず USART 割り込みが生成されます。

ビット 7 TXEIE : TXE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USART_SR レジスタで TXE=1 のときに必ず USART 割り込みが生成されます。

ビット 6 TCIE : 転送完了割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USART_SR レジスタで TC=1 のときに必ず USART 割り込みが生成されます。

ビット 5 RXNEIE : RXNE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USART_SR レジスタで RXNE=1 または ORE=1 のときに必ず USART 割り込みが生成されます。

ビット 4 IDLEIE : IDLE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USART_SR レジスタで IDLE=1 のときに必ず USART 割り込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : 1 : スマートカードモードの場合を除いて、送信中に TE ビットに「0」パルスを与える（「0」に続けて「1」を書き込む）と、現在のワードの後にプリアンブル（アイドルライン）が送信されます。

2 : TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 RWU : レシーバウェイクアップ

このビットは、USART がミュートモードかどうかを決定します。ソフトウェアによってセット／クリアされ、ウェイクアップシーケンスが認識されるとハードウェアによってクリアできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注 : 1 : ミュートモードを選択する（RWU ビットをセットする）前に、USART はまずデータバイトを受信する必要があります。受信しない場合、ミュートモードでアイドルライン検出によるウェイクアップを伴う動作はできません。

2 : アドレスマーク検出のウェイクアップ設定（WAKE=1）では、RXNE ビットがセットされている限り、ソフトウェアで RWU ビットを変更することはできません。

ビット 0 SBK : ブレーク送信

このビットは、ブレークキャラクタを送信するために使用されます。ソフトウェアによってセット／クリアできます。ソフトウェアによってセットされ、ブレークのストップビットの処理中にハードウェアによってリセットされます。

0 : ブレークキャラクタは送信されません。

1 : ブレークキャラクタは送信されます。

25.6.5 制御レジスタ 2 (USART_CR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]	CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]				
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w		r/w	r/w	r/w	r/w

ビット 31:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14 **LINEN** : LIN モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : LIN モードは無効です。

1 : LIN モードは有効です。

LIN モードでは、USART_CR1 レジスタの SBK ビットを使用して LIN 同期ブ레이크 (下位 13 ビット) を送信し、LIN 同期ブ레이크を検出することができます。

ビット 13:12 **STOP** : STOP ビット

このビットは、ストップビットのプログラミングに使用します。

00 : ストップビットが 1

01 : ストップビットが 0.5

10 : ストップビットが 2

11 : ストップビットが 1.5

注 : 0.5 のストップビットおよび 1.5 のストップビットは、UART4 および UART5 では使用できません。

ビット 11 **CLKEN** : クロック有効

このビットによって、SCLK ピンを有効にできます。

0 : SCLK ピンは無効です。

1 : SCLK ピンは有効です。

このビットは UART4 および UART5 では使用できません。

ビット 10 **CPOL** : クロック極性

このビットによって、同期モードにおける SCLK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック/データ関係になるようにします。

0 : 送信ウィンドウの外で、SCLK ピンはローレベルを維持します。

1 : 送信ウィンドウの外で、SCLK ピンはハイレベルを維持します。

このビットは UART4 および UART5 では使用できません。

ビット 9 **CPHA** : クロック位相

このビットによって、同期モードにおける SCLK ピンのクロック出力の位相を選択できます。CPOL ビットと連携して動作し、希望するクロック/データ関係になるようにします (図 257~258 を参照)。

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

注 : このビットは UART4 および UART5 では使用できません。

- ビット 8 **LBCL** : 最終ビットのクロックパルス
- このビットによって、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、SCLK ピンに出力する必要があるかどうかを選択します。
- 0 : 最終データビットのクロックパルスは、SCLK ピンに出力されません。
- 1 : 最終データビットのクロックパルスは、SCLK ピンに出力されます。
- 注 :** **1 : 最終ビットは、USART_CR1 レジスタの M ビットによって選択された 8 または 9 ビットフォーマットに応じて送信された 8 番目または 9 番目のデータビットです。**
- 2 : このビットは UART4 および UART5 では使用できません。**
- ビット 7 予約済みであり、リセット値のままにしておかなければなりません。
- ビット 6 **LBDIE** : LIN ブレーク検出割込み有効
- ブレーク割込みマスクです (ブレークデリミタを使用したブレーク検出)。
- 0 : 割り込みは禁止されています。
- 1 : USART_SR レジスタで LBD=1 になるたび、割込みが生成されます。
- ビット 5 **LBDL** : *lin* ブレーク検出長
- このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。
- 0 : 10 ビットブレーク検出
- 1 : 11 ビットブレーク検出
- ビット 4 予約済みであり、リセット値のままにしておかなければなりません。
- ビット 3:0 **ADD[3:0]** : USART ノードのアドレス
- このビットフィールドは、USART ノードのアドレスを示します。
- これは、ミュートモードでのマルチプロセッサ通信において、アドレスマーク検出によるウェイクアップに使用されます。
- 注 :** **CPOL、CPHA、LBCL の 3 ビットには、トランスミッタが有効である間は書き込まないでください。**

25.6.6 **制御レジスタ 3 (USART_CR3)**

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:12 予約済みであり、リセット値のままにしておかなければなりません。
- ビット 11 **ONEBIT** : 1 サンプルビット方式有効
- このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NF) が無効になります。
- 0 : 3 サンプルビット方式
- 1 : 1 サンプルビット方式
- ビット 10 **CTSIE** : CTS 割り込み有効
- 0 : 割り込みは禁止されています。
- 1 : USART_SR レジスタで CTS=1 になるたびに割り込みが生成されます。
- 注 :** **このビットは UART4 および UART5 では使用できません。**

ビット 9 CTSE : CTS 有効

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは nCTS 入力のアサート (0 に関係) されている場合にのみ転送されます。データ転送中に nCTS 入力にネグートされると、転送は停止前に完了します。nCTS がアサートされている間にデータがデータレジスタに書き込まれると、転送は nCTS がアサートされるまで延期されます。

注： このビットは UART4 および UART5 では使用できません。

ビット 8 RTSE : RTS 有効

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 割り込みが有効です。データは受信バッファに空きがある場合にのみリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。nRTS 出力は、データが受信可能な時にアサートされます (0 に関係)。

注： このビットは UART4 および UART5 では使用できません。

ビット 7 DMAT : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが転送に有効です。

0 : DMA モードが転送に無効です。

ビット 6 DMAR : DMA 有効レシーバ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

ビット 5 SCEN : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

0 : スマートカードモードが無効です。

1 : スマートカードモードが有効です。

注： このビットは UART4 および UART5 では使用できません。

ビット 4 NACK : スマートカード NACK 有効

0 : パリティエラーの際の NACK 転送が無効です。

1 : パリティエラー時の NACK 転送が有効です。

注： このビットは UART4 および UART5 では使用できません。

ビット 3 HDSEL : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

ビット 2 IRLP : IrDA 低電力

このビットは、通常と低電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低電力モード

ビット 1 IREN : IrDA モード有効

このビットは、ソフトウェアによってセット / クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

ビット 0 EIE : エラー割り込みイネーブル

エラー割り込み有効ビットは、マルチバッファ通信 (USART_CR3 レジスタの DMAR=1) において、フレーミングエラー、オーバーランエラー、またはノイズフラグ (USART_SR レジスタの FE=1、ORE=1、または NF=1) の場合に割り込み生成を有効にするために必要です。

0 : 割り込みは禁止されています。

1 : USART_CR3 レジスタで DMAR=1 になるたび、および USART_SR レジスタで FE=1、ORE=1、または NF=1 になるたび、割り込みが生成されます。

25.6.7 ガード時間およびプリスケアラレジスタ (USART_GTPR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:8 **GT[7:0]** : ガード時間値

このビットフィールドは、ガード時間値をポーロック数で表示します。

これはスマートカードモードで使用します。このガード時間値の後は転送完了フラグがセットされます。

注 : このビットは **UART4 および UART5 では使用できません。**

ビット 7:0 **PSC[7:0]** : プリスケーラ値

– **IrDA 低電力モード :**

PSC[7:0] = IrDA 低電力ポーレート

システムクロックを分周して低電力周波数を得るためのプリスケアラのプログラミングに使用します。ソースクロックは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

...

– **IrDA 通常モード :** PSC には 00000001 を設定する必要があります。

– **スマートカードモード :**

PSC[4:0] : プリスケーラ値

システムクロックを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、ソースクロック周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

...

注 : 1 : スマートカードモードが使用される場合、ビット [7:5] は無効です。

2 : このビットは **UART4 および UART5 では使用できません。**

25.6.8 USART レジスタマップ

次の表に、USART のレジスタマップとリセット値を示します。

表 144. USART レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	USART_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE						
	リセット値																							0	0	1	1	0	0	0	0	0	0						
0x04	USART_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[8:0]															
	リセット値																								0	0	0	0	0	0	0	0	0	0					
0x08	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIV_Mantissa[15:4]																DIV_Fraction [3:0]					
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0C	USART_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVER8	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x10	USART_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDE	LBDE	Res.	ADD[3:0]									
	リセット値																		0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					
0x14	USART_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE							
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x18	USART_GTPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]								PSC[7:0]												
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。



26 シリアルペリフェラルインタフェース/I²S (SPI/I²S)

26.1 概要

SPI/I²S インタフェースを使用して、SPI プロトコルまたは I²S オーディオプロトコルに基づき外部デバイスと通信することができます。SPI または I²S モードはソフトウェアによって選択可能です。デバイスのリセット後は、デフォルトで SPI モードが選択されます。

SPI (シリアルペリフェラルインタフェース) プロトコルは、外部デバイスとの半二重、全二重、および単方向の同期シリアル通信をサポートしています。このインタフェースはマスタとして設定することも可能で、その場合、外部スレーブデバイスに通信クロック (SCK) を供給します。このインタフェースは、マルチマスタ設定で動作することもできます。

インター IC サウンド (I²S) プロトコルも、同期シリアル通信インタフェースです。スレーブまたはマスタモードで、レシーバまたはトランスミッタとして動作することができます。

フィリップス I²S 規格、MSB/LSB 詰め規格、PCM 規格など、4 つのオーディオ規格に対応できます。

警告： 一部の SPI1 ピンは JTAG インタフェースによって使用されるピンに配置されるため、他のピンに SPI/I²S を配置したり、(アプリケーションのデバッグ時) JTAG を無効化して、SPI I/O として列挙されたピンを設定する前に SWD インタフェースを使用したり、(スタンドアロンアプリケーションの場合) JTAG/SWD インタフェースを無効化したりできません。JTAG/SWD インタフェースピンの設定の詳細については、[セクション 7.3.2 : I/O ピンマルチプレクサとマッピング](#)を参照してください。

26.1.1 SPI の主な機能

- マスタまたはスレーブ動作
- 3本のラインでの全二重同期転送
- 2本のラインでの半二重同期転送（双方向データライン有り）
- 2本のラインでの単方向同期転送（単方向データライン有り）
- 8または16ビットの転送フレームフォーマット選択
- マルチマスタモード機能
- 8個のマスタモードボーレートプリスケアラ（最大周波数 $f_{PCLK}/2$ ）
- スレーブモード周波数（最大周波数 $f_{PCLK}/2$ ）
- マスタとスレーブの両方に対するハードウェア/ソフトウェアによる NSS 管理：マスタ/スレーブ動作の動的切り替え
- クロックの極性と位相をプログラム可能
- データ順序をプログラム可能（MSB ファースト/LSB ファーストのシフト）
- 専用の送受信フラグ（割り込み機能付き）
- SPI バスビジステータスフラグ
- SPI モトローラモードをサポート
- ハードウェア CRC 機能による信頼性の高い通信：
 - Tx モードでは CRC 値を最終バイトとして送信可能
 - 最終受信バイトに対する CRC エラーの自動チェック
- マスタモードの障害、オーバーランの各フラグ（割り込み機能付き）
- CRC エラーフラグ
- 1バイト/ワードの送受信バッファ（DMA 機能付き：送受信リクエスト）

26.1.2 SPI の拡張機能

- SPI TI モードをサポート

26.1.3 I²Sの機能

- 全二重通信
- 半二重通信（トランスミッタまたはレシーバのみ）
- マスタまたはスレーブ動作
- 正確なオーディオサンプリング周波数（8～192 kHz）を実現するプログラム可能な 8 ビットのリニアプリスケアラ
- 16、24、または 32 ビットのデータフォーマット
- パケットフレームはオーディオチャネルによって 16 ビット（16 ビットデータフレーム）または 32 ビット（16、24、32 ビットデータフレーム）に固定。
- プログラム可能なクロック極性（定常状態）
- スレーブ送信モードのアンダーランフラグ、受信モード（マスタおよびスレーブ）のオーバーランフラグ、受信モードと送信モード（スレーブの場合のみ）のフレームエラーフラグ
- 送受信用の 16 ビットレジスタ（両チャンネルサイドに対して 1 個のデータレジスタ）
- 以下の I²S プロトコルをサポート。
 - － フィリップス I²S 規格
 - － MSB 詰め規格（左詰め）
 - － LSB 詰め規格（右詰め）
 - － PCM 規格（16 ビットチャンネルフレーム、または 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームでの、ショートおよびロングフレーム同期付き）
- データ方向は常に MSB ファースト。
- 送受信（16 ビット幅）用の DMA 機能
- 外部オーディオコンポーネントを駆動するためのマスタクロックを出力可能。周波数比は、 $256 \times F$ （ F_S はオーディオサンプリング周波数）に固定。
- I²S（I2S1、I2S2、I2S3、I2S4 および I2S5）クロックは、I2S_CKIN ピンに割り当てられた外部クロックから取得可能

26.2 SPI/I2S の実装

このマニュアルでは、SPI1、SPI2、SPI3、SPI4 および SPI5 に実装されているすべての機能について説明しています。

表 145. STM32F412 SPI 実装

SPI の機能 ⁽¹⁾	SPI1	SPI2	SPI3	SPI4	SPI5
ハードウェア CRC 計算	X	X	X	X	X
I2S モード	X	X	X	X	X
TI モード	X	X	X	X	X

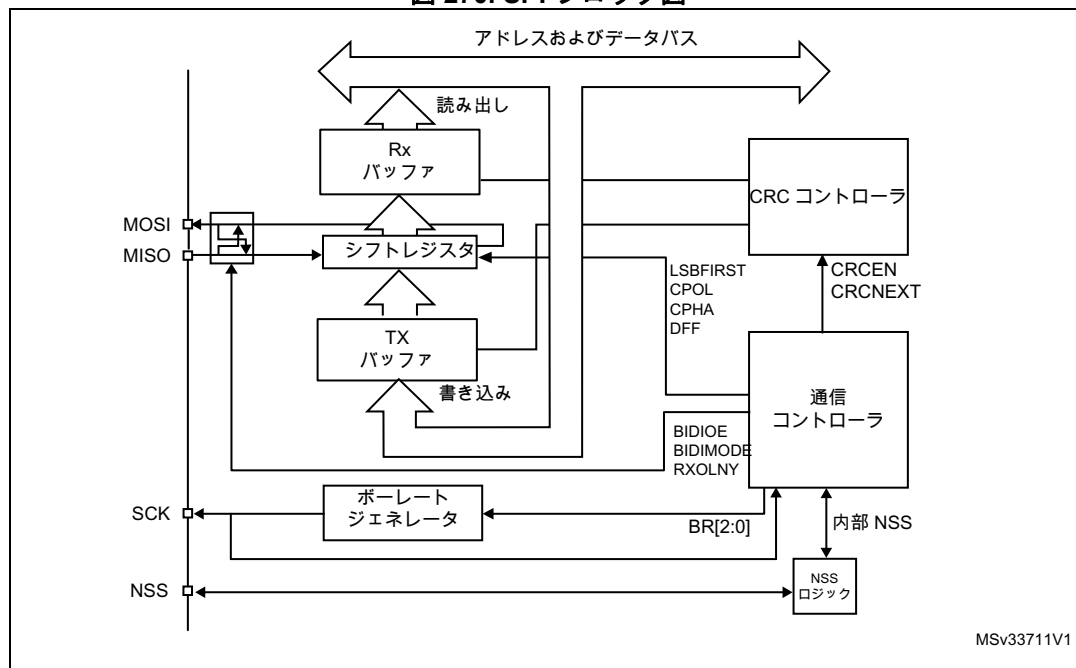
1. X：サポートされています。

26.3 SPI の機能説明

26.3.1 概要

SPI では、MCU と外部デバイス間の同期シリアル通信が可能です。アプリケーションソフトウェアは、ステータスフラグをポーリングするか、または専用の SPI 割り込みを使用することで、通信を管理することができます。SPI の主要要素およびそれらの相互作用を以下のブロック図 (図 270 に示します。

図 270. SPI ブロック図



4 本の I/O ピンが外部デバイスとの SPI 通信専用に使われます。

- **MISO** : マスターイン/スレーブアウトデータ。一般に、このピンは、スレーブモードではデータの送信に、マスタモードではデータの受信に使われます。
- **MOSI** : マスターアウト/スレーブインデータ。一般に、このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使われます。
- **SCK** : SPI マスタではシリアルクロックの出力に、SPI スレーブでは入力に使われます。
- **NSS** : スレーブ選択用のピンです。このピンは、SPI および NSS の設定に応じて、以下のいずれかに使用できます。
 - 個々の通信用スレーブデバイスを選択する
 - データフレームを同期させる
 - 複数のマスタ間での競合を検出する

詳細は、[セクション 26.3.4 : マルチマスタ通信](#) を参照してください。

SPI バスを使用することで、1 つのマスタデバイスと 1 つ以上のスレーブデバイスとの間で通信することができます。バスは 2 本以上の線から成り、1 本はクロック信号用、その他はデータの同期転送用です。SPI ノード間でのデータ交換とそれらのスレーブ選択信号管理に応じて、その他の信号を追加することができます。

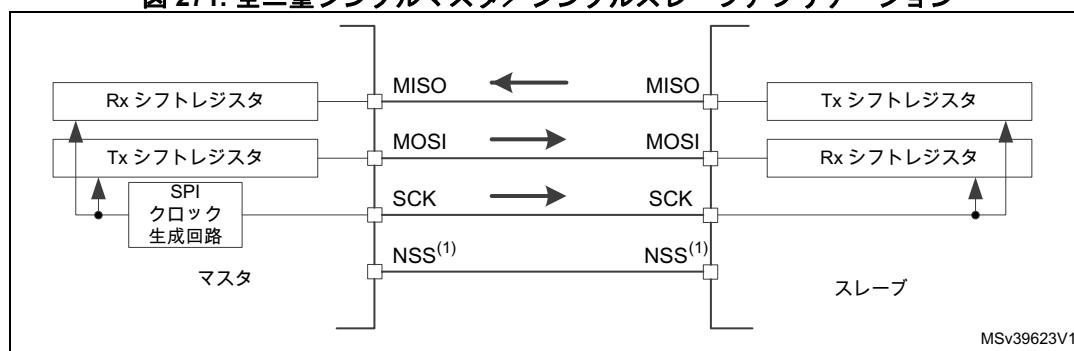
26.3.2 マスタとスレーブの 1 対 1 の通信

SPI を使用することで、MCU は対象となるデバイスやアプリケーション要件に応じたさまざまな設定で通信ができます。これらの設定には、2 または 3 本の線（ソフトウェア NSS 管理あり）、あるいは 3 または 4 本の線（ハードウェア NSS 管理あり）が使われます。通信は常にマスタによって開始されます。

全二重通信

SPI は、デフォルトで全二重通信に設定されます。この設定では、マスタおよびスレーブのシフトレジスタは、MOSI ピンと MISO ピンの間に 2 本の単方向ラインを介してリンクされます。SPI 通信の間、データはマスタから供給される SCK クロックのエッジに同期してシフトされます。マスタは、送信すべきデータを MOSI ライン経由でスレーブに送信し、MISO ライン経由でスレーブからデータを受信します。データフレーム転送が完了した（すべてのビットがシフトされた）時点で、マスタとスレーブの間で情報が交換されます。

図 271. 全二重シングルマスタ/シングルスレーブアプリケーション

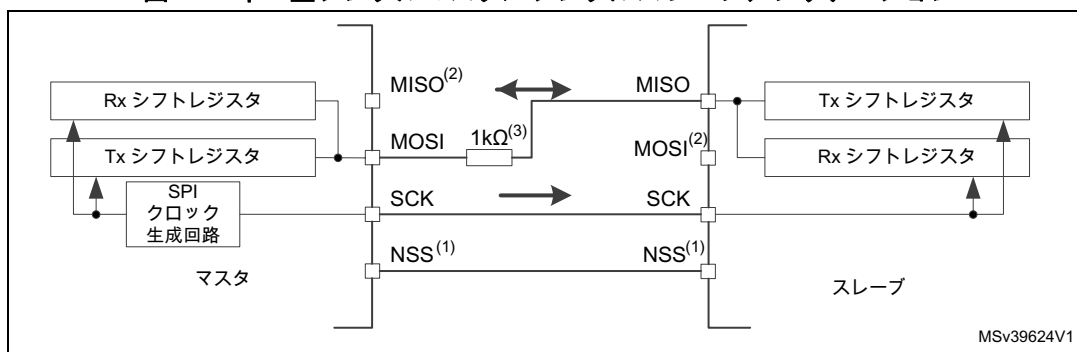


1. NSS ピンを使用して、マスタとスレーブの間のハードウェア制御フローを提供できます。オプションとして、ピンはペリフェラルで未使用のままにしておくことができます。その際、マスタおよびスレーブに対するフローを内部的に処理する必要があります。詳細については、[セクション 26.3.5 : スレーブ選択 \(NSS\) ピンの管理](#)を参照してください。

半二重通信

SPIx_CR1 レジスタの BIDIMODE ビットをセットすることで、SPI は半二重モードで通信できます。この設定では、1 本の交差接続ラインを使用して、マスタとスレーブのシフトレジスタを互いにリンクさせます。この通信中に、データは SCK クロックのエッジに同期して、シフトレジスタ間でシフトされます。シフトの方向は、マスタとスレーブの両方が SPIx_CR1 レジスタの BDIOE ビットを使用して相互に選択された転送方向となります。この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、他のアプリケーションで使用でき、GPIO として機能します。

図 272. 半二重シングルマスタ/シングルスレーブアプリケーション



1. NSS ピンを使用して、マスタとスレーブの間のハードウェア制御フローを提供できます。オプションとして、ピンはペリフェラルで未使用のままにしておくことができます。その際、マスタおよびスレーブに対するフローを内部的に処理する必要があります。詳細については、[セクション 26.3.5 : スレーブ選択 \(NSS\) ピンの管理](#)を参照してください。
2. この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、GPIO として使用できます。
3. 双方向モードで動作している 2 つのノード間で同期されず、通信方向が変更されて、新しいトランスミッタが共通のデータラインにアクセスし、前のトランスミッタが逆の値をライン上に保持している場合（値は SPI 設定および通信データによります）、重大な事態が発生する可能性があります。そのとき、次のノードが対応して方向設定を変更するまで、共通のライン上に一時的に逆の出力レベルが提供され、両ノードの競合も発生します。このモードでは、MISO および MOSI ピンの間に直列抵抗を挿入して、出力を保護し、この状況での電流上昇を制限することをお奨めします。

単方向通信

SPI は、SPIx_CR2 レジスタの RXONLY ビットを使用して送信専用または受信専用に設定することにより、単方向モードで通信できます。この設定では、マスタとスレーブのシフトレジスタ間の転送に使用するのは 1 ラインのみです。残りの MISO ピンと MOSI ピンのペアは通信には使用されず、標準の GPIO として使用できます。

- **送信専用モード (RXONLY = 0) の場合** : 設定は全二重の場合と同じです。アプリケーションは、未使用の入力ピンでキャプチャされた情報を無視する必要があります。このピンは標準の GPIO として使用できます。
- **受信専用モード (RXONLY = 1) の場合** : アプリケーションにて、RXONLY ビットをセットすることによって、SPI 出力機能を無効にできます。スレーブ設定では、MISO 出力が無効化され、ピンを GPIO として使用することができます。スレーブ選択信号がアクティブな間は、スレーブは MOSI ピンからデータを受信し続けます ([26.3.4 : マルチマスタ通信](#)を参照)。データバッファの設定に応じて、受信データイベントが出現します。マスタ設定では、MOSI 出力が無効化され、ピンを GPIO として使用することができます。SPI が有効である間はクロック信号が生成され続けます。クロックを停止させる唯一の方法は、クロックの設定に応じて、RXONLY ビットまたは SPE ビットをクリアし、MISO ピンからの受信パターンが終了し、データバッファ構造への書き込みが行われるまで待つことです。

MSv39625V1

- 注: すべての単方向通信は、トランザクション方向の設定を固定して (双方向モードは BDIO ビットが変化しない限り有効)、別の半二重通信に置き換えることができます。

2つ以上の独立したスレーブがある設定の場合、マスタは GPIO ピンを使用して、各スレーブのチップセレクトラインを管理します (図 274 を参照)。マスタは、スレーブの NSS 入力に接続されている GPIO をローレベルにプルダウンすることによって、スレーブの 1 つを選択する必要があります。これを行うことにより、標準マスタと専用スレーブの通信が確立します。

1. NSS ピンはこの設定のマスタ側では使用しません。内部的に管理 (SSM=1、SSI=1) し、あらゆる MODF エラーを防ぐ必要があります。
2. スレーブの MISO ピンは相互接続されているので、すべてのスレーブにおいて、その MISO ピンの GPIO 設定をオルタネート機能オープンドレインとしてセットする必要があります (セクション 7.3.7 : I/O オルタネート機能の入力/出力 (169 ページ) を参照)。

26.3.4 マルチマスタ通信

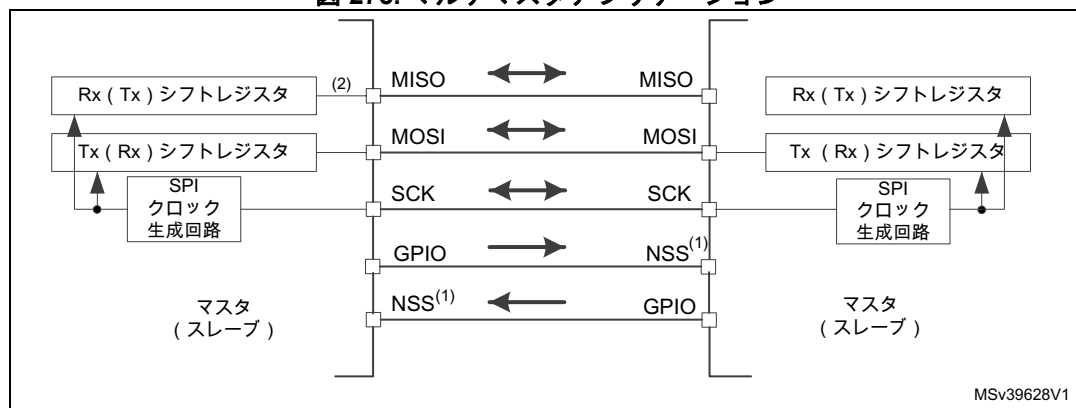
SPI バスが主にマルチマスタ機能向けに設計されていない限り、同時にバスをマスタ化しようとする 2 つのノード間での電位競合を検出する組み込み機能を使用できます。この検出では、NSS ピンをハードウェア入力モードで設定して使用されます。

2 つ以上の SPI ノードの接続での同時マスタ動作は、共通のデータラインで一度に出力を適用できるノードは 1 つだけであるため、不可能です。

ノードが非アクティブであるとき、両ノードはデフォルトでスレーブモードのままとなります。1 つのノードがマスタに切り替わろうとした場合、自分自身でマスタモードに切り替わり、専用の GPIO ピンを介して他のノードのスレーブ選択入力で有効なレベルを適用します。セッションが完了すると、有効なスレーブ選択信号は解放され、バスを一時的にマスタ化していたノードは、次のセッション開始に向けて待機する受動的なスレーブモードに戻ります。

両ノードのマスタリングリクエストが同時に上がった場合、バス競合イベントが発生します（モードフォールト MODF イベントを参照）。そのとき、いくつかのシンプルなアービトレーション処理（例：両ノードで適用される異なるタイムアウトを事前に定義することで次の試行を延期する）を適用できます。

図 275. マルチマスタアプリケーション



1. NSS ピンは両ノードのハードウェア入力モードで設定されます。アクティブレベルにすることで、パッシブノードがスレーブとして設定されるため、MISO ライン出力の制御が可能になります。

26.3.5 スレーブ選択 (NSS) ピンの管理

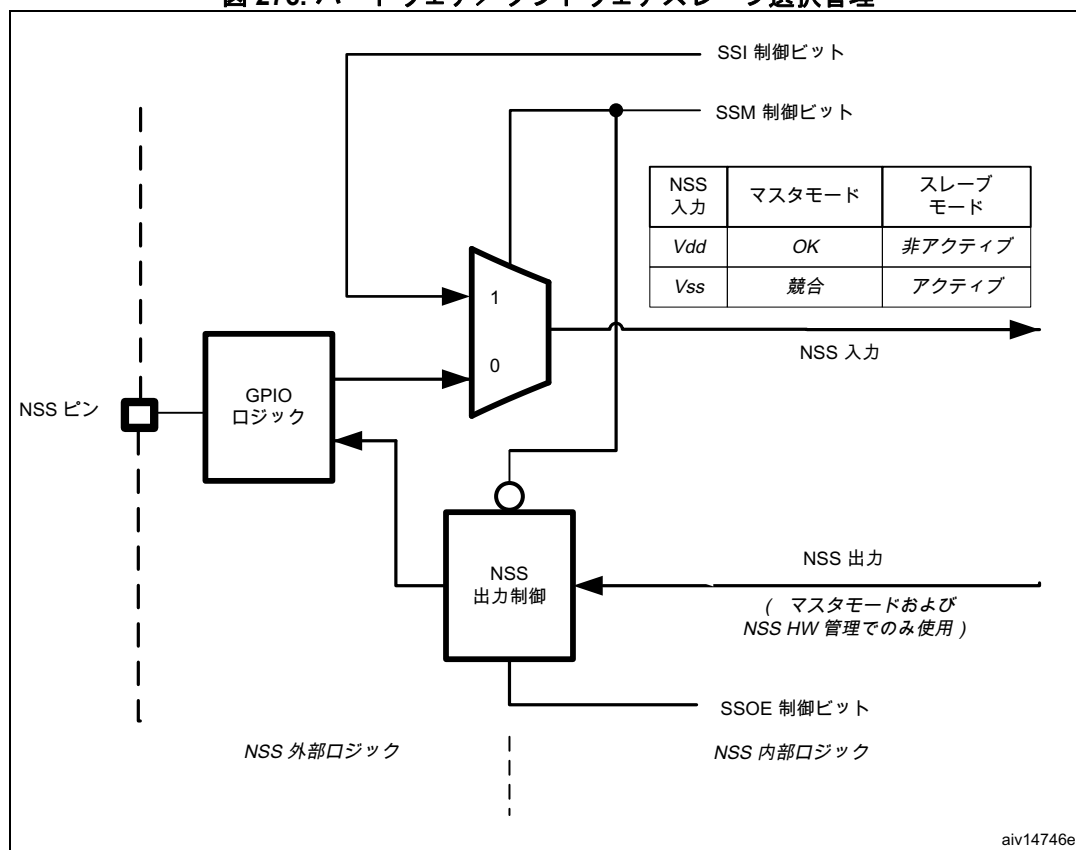
スレーブモードでは、NSS は標準の「チップセレクト」入力として機能し、スレーブをマスタと通信させます。マスタモードでは、NSS は出力としても入力としても使用できます。入力として使用する場合、NSS はマルチマスタのバスの衝突を未然に防ぎ、出力として使用する場合は 1 つのスレーブのスレーブ選択信号を駆動させることができます。

ハードウェアまたはソフトウェアのスレーブ選択管理は、SPIx_CR1 レジスタの SSM ビットを使用して、以下のようにセットすることができます。

- **ソフトウェア NSS 管理 (SSM = 1) :** この設定では、スレーブ選択情報は SPIx_CR1 レジスタの SSI ビットの値によって内部で駆動されます。外部 NSS ピンは他のアプリケーションで使用できます。
- **ハードウェア NSS 管理 (SSM = 0) :** この場合、2 通りの設定が可能です。次のどちらの設定を使用するかは、NSS 出力設定 (SPIx_CR1 レジスタの SSOE ビット) によって決まります。

- **NSS 出力が有効な場合 (SSM = 0, SSOE = 1)** : この設定は、MCU がマスタとしてセットされている場合にのみ使用します。NSS ピンはハードウェアによって管理されます。NSS 信号は、SPI がマスタモードで有効になる (SPE = 1) とすぐにローレベルに駆動され、SPI が無効化される (SPE = 0) までローレベルに保たれます。
- **NSS 出力が無効な場合 (SSM = 0, SSOE = 0)** : マイクロコントローラがバスでマスタとして機能している場合、この設定によりマルチマスタ機能が可能になります。このモードで、NSS ピンがローレベルにプルダウンされた場合、SPI はマスタモードのフォールト状態に入り、デバイスが自動的にスレーブモードに再設定されます。スレーブモードでは、NSS ピンは標準の「チップセレクト」入力として機能し、NSS ラインがローレベルの間はスレーブが選択されます。

図 276. ハードウェア/ソフトウェアスレーブ選択管理



aiv14746e

26.3.6 通信フォーマット

SPI 通信中は受信と送信の操作が同時に行われます。シリアルクロック (SCK) は、データライン上で行われる情報のシフトとサンプリングを同期させます。通信フォーマットは、クロック位相、クロック極性、およびデータフレームフォーマットに応じて決定されます。マスタデバイスとスレーブデバイスの通信を可能にするには、双方が同じ通信フォーマットに従う必要があります。

クロックの位相および極性の制御

SPIx_CR1 レジスタの CPOL ビットと CPHA ビットを使用することによって、考えられる4つのタイミングの関係をソフトウェアで選択できます。CPOL (クロック極性) ビットは、データが転送されていないときのクロックのアイドル状態の値を制御します。このビットは、マスタモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA ビットがセットされると、SCK ピンの2番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがリセットされていれば立ち下がりエッジ、CPOL ビットがセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。CPHA ビットがリセットされている場合、SCK ピンの1番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがセットされていれば立ち下がりエッジ、CPOL ビットがリセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。

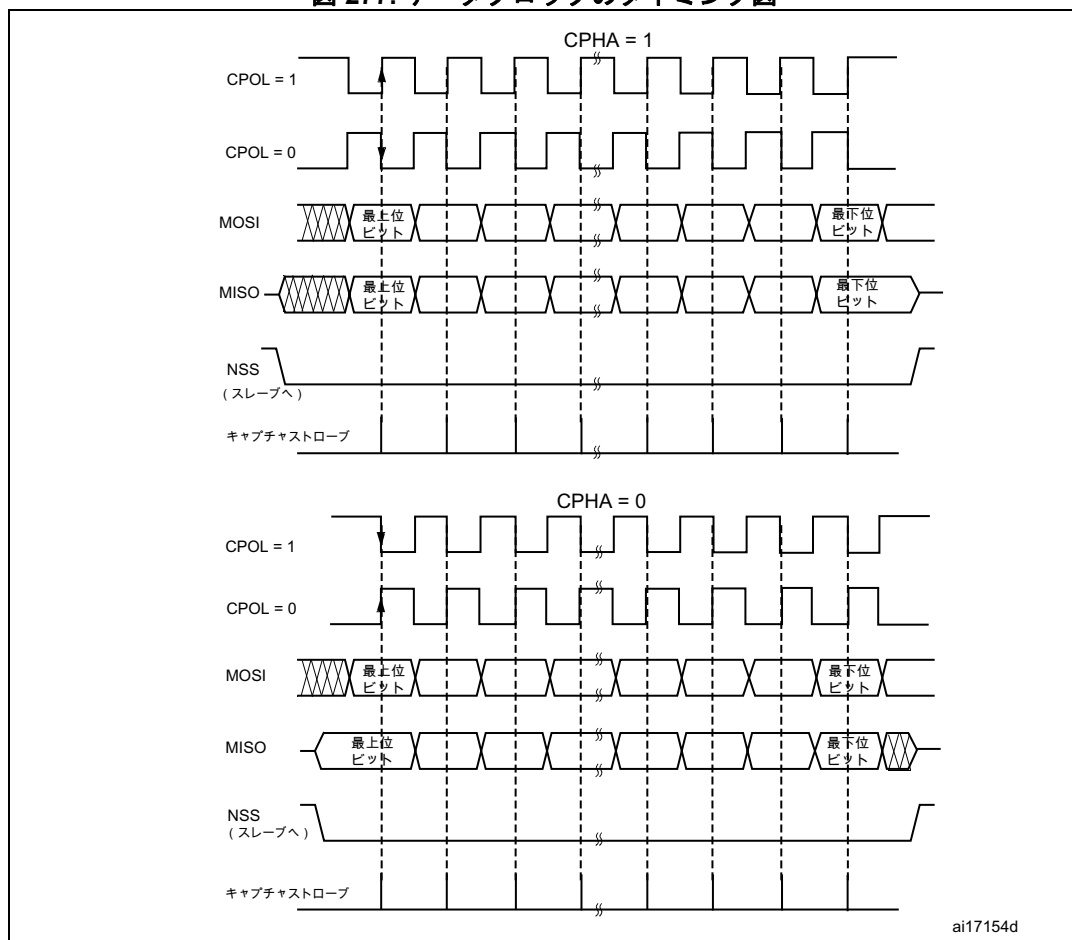
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジを選択できます。

 277 は、CPHA ビットと CPOL ビットの4つの組み合わせによる SPI 全二重転送を示しています。

注： *CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。*

SCK のアイドル状態は、SPIx_CR1 レジスタで (CPOL = 1 なら SCK のプルアップ、CPOL = 0 なら SCK のプルダウンによって) 選択された極性に一致する必要があります。

図 277. データクロックのタイミング図



注： データビットの順序は **LSBFIRST** ビットの設定値に依存します。

データフレームフォーマット

LSBFIRST ビットの値に応じて、SPI シフトレジスタを設定することで、シフトをMSB ファーストまたは LSB ファーストに設定することができます。各データフレームの長さは、SPI_CR1 レジスタの DFF ビットを使用してデータのサイズをプログラムすることにより、8 または 16 ビットに設定できます。選択されたデータフレームフォーマットは、送受信のどちらにも適用できます。

26.3.7 SPI の設定

設定手順は、マスタとスレーブではほぼ同じです。特定のモードの設定については、それぞれのモードに関する章を参照してください。標準通信を初期化する必要があるときは、以下の手順を実行します。

- 適切な GPIO レジスタに書き込みを行います。MOSIピン、MISOピン、SCKピンの GPIO 設定を行います。
- SPI_CR1 レジスタに書き込みを行います。
 - BR[2:0] ビットを使用して、シリアルクロックボーレートを設定します（[注3](#)を参照）。
 - CPOL ビットと CPHA ビットの組み合わせを設定して、データ転送とシリアルクロックの4つの関係のうちの1つを定義します（[注2](#)を参照）。
 - RXONLY または BIDIMODE、および BIDIOE を設定することによって（RXONLY と BIDIMODE は同時にセットできません）、単方向または半二重モードを選択します。
 - LSBFIRST ビットを設定して、フレームフォーマットを定義します（[注2](#)を参照）。
 - CRC が必要な場合は（SCK クロック信号がアイドル状態のとき）、CRCEN ビットを設定します。
 - SSM と SSI を設定します（[注2](#)を参照）。
 - MSTR ビットを設定します（マルチマスタ NSS 設定では、MODF エラーを防ぐためにマスタが設定されている場合、NSS での競合を避けること）。
 - DFF ビットをセットして、データフレームフォーマット（8 または 16 ビット）を設定します。
- 以下のように、SPI_CR2 レジスタに書き込みを行います。
 - SSOE を設定します（[注1](#) および [注2](#)を参照）。
 - TI プロトコルが必要な場合は、FRF ビットをセットします。
- SPI_CRCPR レジスタに書き込みを行います。必要に応じて CRC 多項式を設定します。
- 適切な DMA レジスタに書き込みを行います。DMA ストリームが使用されている場合は、DMA レジスタに SPI Tx および Rx 専用の DMA ストリームを設定します。

注：

- (1) このステップはスレーブモードでは必要ありません。
- (2) このステップは TI モードでは必要ありません。
- (3) このステップは、スレーブモードにて TI モードで動作している場合を除き、スレーブモードでは必要ありません。

26.3.8 SPI を有効にする手順

マスタがクロックを送信する前に、SPI スレーブを有効にすることを推奨します。そうしないと、望ましくないデータ送信が発生することがあります。スレーブデータレジスタは、マスタとの通信を開始する前に、送信データをすでに格納していなければなりません（通信クロックの1番目のエッジに、またはクロック信号が連続的なときは現在の通信の最後の前に）。SPI スレーブが有効になる前に、SCK 信号を選択された極性に対応するアイドル状態のレベルに安定させる必要があります。

全二重（または送信専用モード）では、SPI が有効になり、送信するデータがTxバッファに書き込まれた時点で、マスタは通信を開始します。

あらゆるマスタ受信専用モードにおいて（RXONLY = 1 または BIDIMODE = 1、および BIDIOE = 0）、SPI が有効になるとすぐに、マスタは通信を開始し、クロックは動作を開始します。

スレーブは、マスタから正しいクロック信号を受信した時点で通信を開始します。SPI マスタが転送を開始する前に、スレーブのソフトウェアは送信データを書き込んでおく必要があります。

DMA の処理方法については、[セクション 26.3.11 : DMA \(ダイレクトメモリアクセス\) を使用する通信](#)を参照してください。

26.3.9 データの送受信手順

RxバッファとTxバッファ

受信の場合、データは受信されたあと、内部のRxバッファに格納されます。一方、送信の場合、データはまず内部のTxバッファに格納されてから、送信されます。SPI_DR レジスタへの読み出しアクセスでは、Rxバッファに格納された値が返されます。一方、SPI_DR への書き込みアクセスでは、書き込まれたデータがTxバッファに格納されます。

Txバッファの処理

データフレームは、最初のビットの送信中にTxバッファからシフトレジスタへロードされます。その後、ビットは、LSBFIRST ビットの設定に応じて、シフトレジスタから専用の出力ピンへ順次シフトアウトされます。データがTxバッファからシフトレジスタへ転送された時点で、TXE フラグ (Txバッファエンプティ) がセットされます。このフラグは、内部のTxバッファに次のデータをロードする準備ができていることを示します。SPI_CR2 レジスタの TXEIE ビットがセットされている場合は、割り込みを生成できます。TXE ビットは、SPI_DR レジスタへの書き込みによってクリアされます。

前のフレーム送信がまだ進行中の間は、次に送信されるデータがTxバッファに格納されていれば、連続した送信ストリームを実現できます。TXE フラグがセットされていないときにソフトウェアがTxバッファへの書き込みを行った場合、トランザクション待ちのデータが上書きされます。

Rxバッファの処理

データがシフトレジスタからRxバッファに転送されると、最後のサンプリングクロックエッジでRXNE フラグ (Rxバッファノートエンプティ) がセットされます。このフラグは、SPI_DR レジスタからデータを読み出す準備ができていることを示します。SPI_CR2 レジスタの RXNEIE ビットがセットされている場合は、割り込みを生成できます。RXNE ビットは、SPI_DR レジスタの読み出しによってクリアされます。

前に送信されたデータバイトが原因でデバイスが RXNE ビットをクリアしなかった場合、次の値がバッファされた時点でオーバーランの条件が発生します。OVR ビットがセットされ、ERRIE ビットがセットされている場合は割り込みが生成されます。

データ交換を管理するもう一つの方法は、DMA を使用することです ([セクション 9.2 : DMA の主な機能](#)を参照)。

シーケンス処理

現在のデータフレームトランザクションが進行中の場合に、BSY ビットがセットされます。クロック信号が流れ続けているときは、BSY フラグはマスタ側のデータフレーム間でセットされたままになります。一方、スレーブ側では、BSY フラグは、各データフレーム転送間で最小の SPI クロック1サイクルの間ローレベルになります。

設定によっては、最後のデータ転送時に BSY フラグを使用して、転送の完了まで待つことができます。

マスタ側で、半二重 (BIDIMODE = 1、BIDIOE = 0) または単方向 (BIDIMODE = 0、RXONLY = 1) のいずれかの設定で、受信専用モードが設定されている場合、SPI が有効になるとすぐにマスタは受信シーケンスを開始します。次に、マスタによってクロック信号が供給されますが、この信号はマスタが SPI または受信専用モードを無効にするまで停止しません。マスタは、クロック信号が停止するまでデータフレームを受信し続けます。

マスタはあらゆるトランザクションを連続モードで供給できる (SCK 信号は連続的) 一方で、データフローおよびその内容をいつでも処理できるスレーブ機能を優先する必要があります。必要に応じて、マスタは通信速度を下げ、より低速のクロックか、または十分な遅延を含む個別のフレーム/データセッションを供給する必要があります。次の 2 点に注意してください。SPI モードで動作する

スレーブに対してはアンダーフローエラー信号はありません。また、スレーブからのデータは、たとえスレーブがそれらのデータを時間内に適切に準備できない場合でも、常にマスタによってトランザクション処理されます。スレーブが DMA を使用することが特に望ましいのは、データフレームが短く、バスが高速の場合です。

マルチスレーブシステムにおいて通信用のスレーブを 1 つだけ選択するには、各シーケンスを NSS パルス内に収める必要があります。シングルスレーブシステムでは、スレーブの制御に NSS を使用する必要はありません。ただし、NSS パルスを使用して、スレーブを各データ転送シーケンスの開始と同期させることができます。NSS は、ソフトウェアまたはハードウェアによって管理できます（[セクション 26.3.4：マルチマスタ通信](#)を参照）。

マスタ/全二重モードおよびスレーブ/全二重モードでの連続転送については、[図 278](#) および [図 279](#) を参照してください。

図 278. マスタ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0, RXONLY = 0) 連続転送の場合

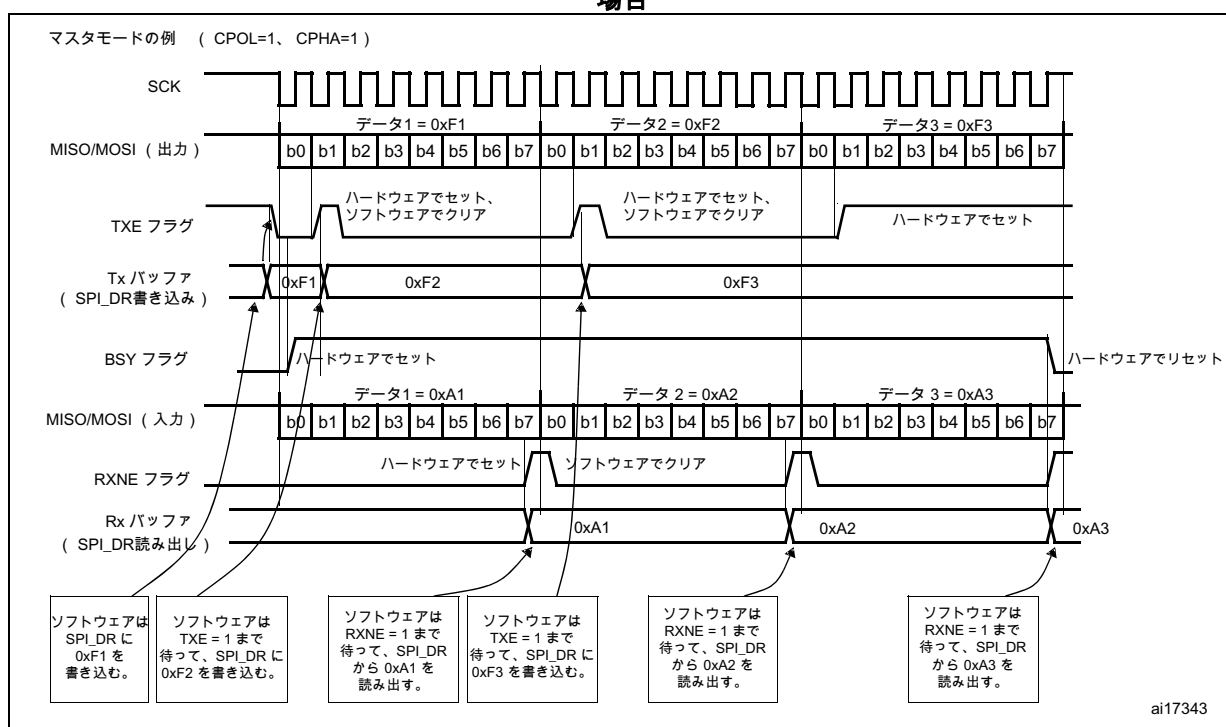
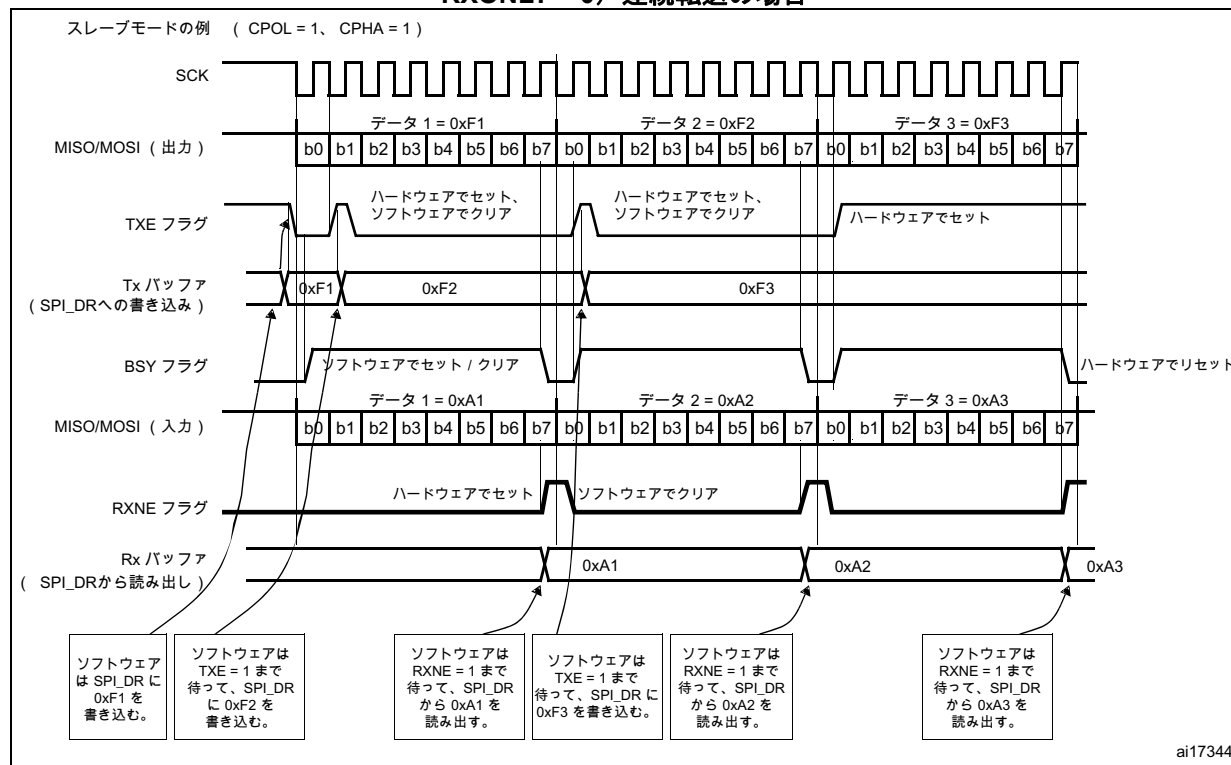


図 279. スレーブ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合



26.3.10 SPI を無効にする手順

SPI を無効にする場合は、本項に記載されている無効化手順に従ってください。この手順は、ペリフェラルクロックが停止し、システムが低電力モードに入る前に行うことが重要です。この場合、進行中のトランザクションが破壊されることがあります。モードによっては、この無効化手順が連続通信を停止させる唯一の方法です。

全二重または送信専用モードでは、マスタは、転送するデータの供給を停止した時点でいかなるトランザクションも終了することができます。この場合、クロックは最後のデータトランザクション後に停止します。

標準的な無効化手順は、送信セッションが完全に終わったかどうかをチェックするために、TXE フラグおよび BSY フラグの状態をポーリングして行われます。このチェックは、たとえば以下に示すように、進行中のトランザクションの終わりを識別する必要があるような特別な場合にも行うことができます。

- NSS 信号が任意の GPIO トグルによって管理されており、マスタはスレーブに適切な NSS パルスの終わりを提供する場合
- 最後のデータフレームまたは CRC フレームのトランザクションがまだペリフェラルバスで進行している間に、DMA からのトランザクションのストリームが完了した場合

正しい無効化手順を以下に示します (受信専用モードが使用されている場合を除く)。

1. RXNE = 1 になるまで待ってから、最後のデータを受信します。
2. TXE = 1 になるまで待ち、さらに BSY = 0 になるまで待ってから、SPI を無効にします。
3. 受信データを読み出します。

注： 不連続通信時には、SPI_DR レジスタへの書き込み動作と BSY ビットが設定される間に、2 APB クロック周期分の遅延があります。従って、最後のデータを書き込んだ後、まず TXE がセットされるまで待ち、さらに BSY がクリアされるまで待つ必要があります。

受信専用モードの場合の正しい無効化手順を以下に示します。

1. 最後のデータフレームの進行中に特定の時間枠内で SPI を無効 (SPE = 0) にすることにより、受信フローへの割り込みを行います。
2. BSY = 0 (最後のデータフレームが処理される) まで待ちます。
3. 受信データを読み出します。

注： 連続受信シーケンスを停止させるには、最後のデータフレームの受信中に、ある特定の時間枠を順守する必要があります。その時間枠は、最初のビットがサンプリングされたときに開始され、最後のビット転送が開始される前に終了します。

26.3.11 DMA (ダイレクトメモリアクセス) を使用する通信

最高速度で動作し、オーバーランを回避するために必要なデータレジスタの読み出し/書き込み処理を容易にするために、SPI は簡単なリクエスト/確認応答プロトコルを実現する DMA 機能を備えています。

SPIx_CR2 レジスタの TXE または RXNE イネーブルビットをセットすると、DMA アクセスがリクエストされます。Tx バッファと Rx バッファには、別々のリクエストを発行する必要があります。

- 送信では、TXE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx_DR レジスタに書き込みます。
- 受信では、RXNE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx_DR レジスタを読み出します。

DMA 送受信波形については、[図 280](#) および [図 281](#) を参照してください。

SPI がデータの送信にのみ使用される場合、SPI Tx DMA チャンネルのみを有効にすることができます。この場合、受信したデータは読み出されないため、OVR フラグがセットされます。SPI がデータの受信にのみ使用される場合、SPI Rx DMA チャンネルのみを有効にすることができます。

送信モードで、DMA がすべての送信データを書き込んだとき (DMA_ISR レジスタの TCIF フラグがセットされます)、BSY フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、または STOP モードに入る前にこの操作を行う必要があります。ソフトウェアは、まず TXE = 1 になるまで待ち、さらに BSY = 0 になるまで待つ必要があります。

DMA を使用して通信を開始する場合、DMA チャンネルの管理によるエラーイベントを防ぐために、以下の手順を順に実行する必要があります。

1. DMA Rx が使用される場合、SPI_CR2 レジスタの RXDMAEN ビットの DMA Rx バッファを有効にします。
2. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを有効にします。
3. DMA Tx が使用される場合、SPI_CR2 レジスタの TXDMAEN ビットにて DMA Tx バッファを有効にします。
4. SPE ビットをセットして、SPI を有効にします。

通信を終了するには、以下の手順を順に実行する必要があります。

1. ストリームが使用される場合、DMA レジスタにて TxおよびRxの DMA ストリームを無効にします。
2. SPI 無効化手順に従って SPI を無効にします。
3. DMA Tx および(または) DMA Rx が使用されている場合、SPI_CR2 レジスタの TXDMAEN ビットおよびRXDMAEN ビットをクリアすることにより、DMA Tx バッファおよびRx バッファを無効にします。

図 280. DMA を使用した送信

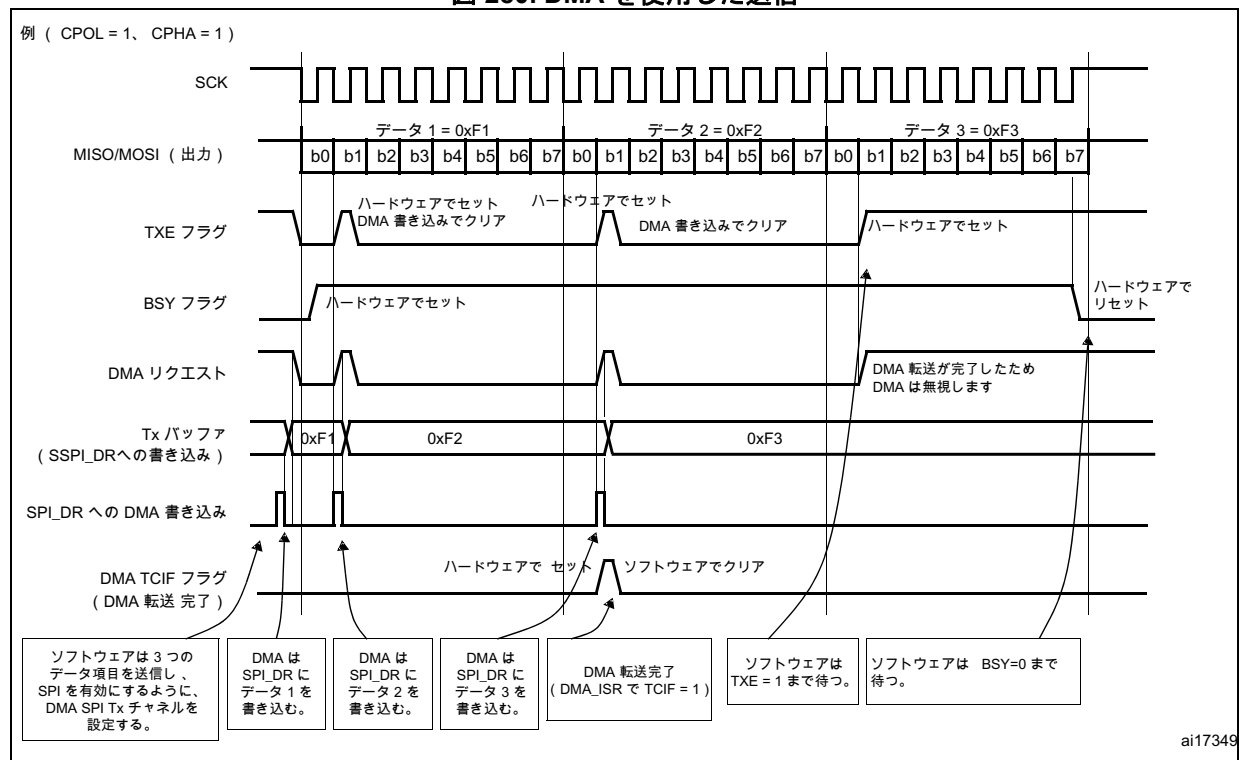
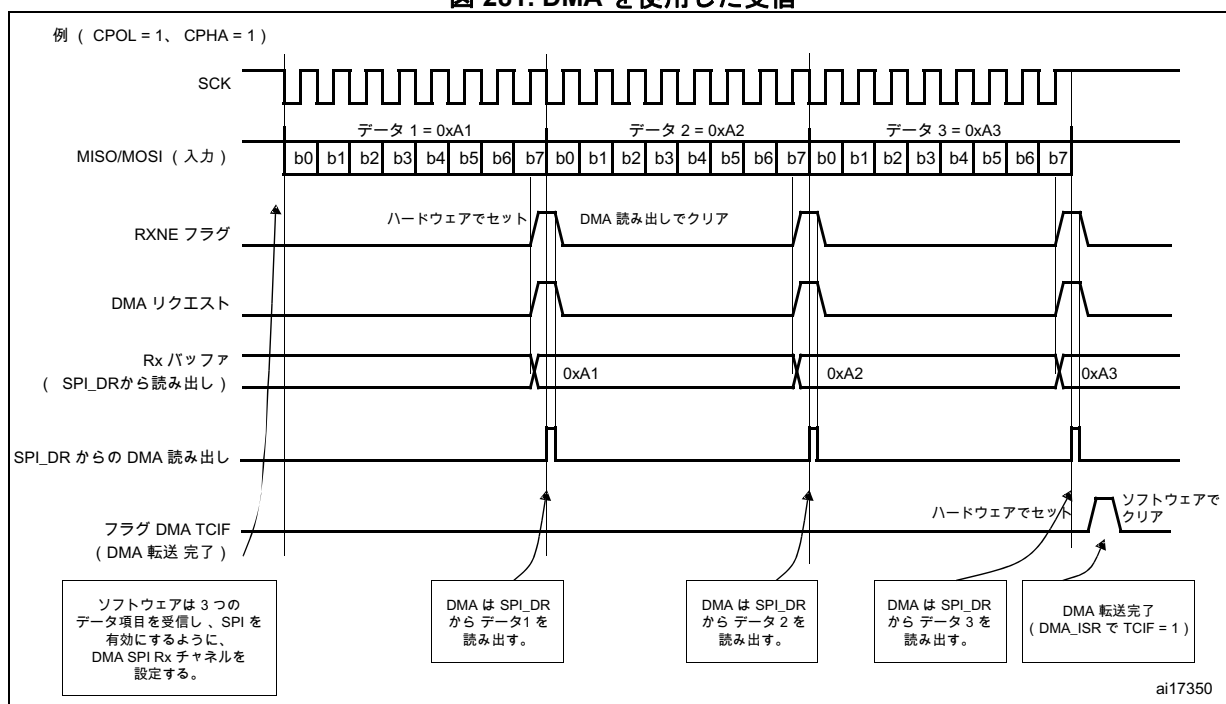


図 281. DMA を使用した受信



26.3.12 SPI ステータスフラグ

アプリケーションが SPI バスの状態を完全に監視できるように、3つのステータスフラグが用意されています。

Tx バッファエンプティフラグ (TXE)

このフラグがセットされると、Txバッファが空であり、次に送信するデータをバッファにロードできることを示します。TXE フラグは、SPI_DR レジスタへの書き込みによってクリアされます。

Rx バッファノットエンプティ (RXNE)

このフラグがセットされると、Rxバッファに有効な受信データがあることを示します。このフラグは、SPI_DR レジスタからの読み出しによってクリアされます。

ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。

BSY フラグがセットされると、SPI上 でデータ転送が進行中であることを示します (SPI バスはビジー)。1つ例外があり、マスタの双方向受信モード (MSTR = 1, BDM = 1、および BDOE = 0) では、BSY フラグが受信時にローレベルに保持されます。

BSY フラグはいくつかのモードで転送の終わりを検出するのに使用できます。それにより、低電力モードに入る前に SPI ペリフェラルクロックが無効化される場合、または NSS パルスの終わりがソフトウェアで処理される場合に、最後の転送の内容が破壊されるのを防ぐことができます。

BSY フラグは、マルチマスタシステムでの書き込み衝突の回避にも役立ちます。

BSY フラグは次のいずれかの条件下でクリアされます。

- SPI が正常に無効にされたとき
- マスタモードで、障害が検出 (MODF ビットが 1 にセットされます) されたとき
- マスタモードで、データ送信が終了し、送信準備ができていない新しいデータがないとき
- スレーブモードで、各データ転送間で少なくとも SPI の 1 クロックサイクルの間、BSY フラグが 0 にセットされているとき

注： *必ず (BSY フラグを使用する代わりに) TXE フラグと RXNE フラグを使用して、データの送受信の処理を行うことを推奨します。*

26.3.13 SPI エラーフラグ

次のいずれかのエラーフラグがセットされていて、ERRIE ビットをセットすることにより割り込みが有効になっている場合、SPI 割り込みが生成されます。

オーバーランフラグ (OVR)

Rxバッファからの前のフレームの読み出し動作が完了していない (RXNE フラグがセットされている) うちに、マスタまたはスレーブが次のデータフレームの受信を完了すると、オーバーラン条件が発生します。

この場合、Rxバッファの内容は新しい受信データによって更新されません。SPI_DR レジスタからの読み出し動作によって、前に受信されたフレームが返されます。その後に送信されたすべてのデータは失われます。

OVR ビットをクリアするには、SPI_DR レジスタを読み出し、続けて SPI_SR レジスタを読み出しアクセスを行います。

モードフォールト (MODF)

モードフォールトは、マスタデバイスが内部 NSS 信号 (NSS ハードウェアモードでは NSS ピン、NSS ソフトウェアモードでは SSI ビット) をローレベルにプルダウンしたときに発生します。これにより、MODF ビットが自動的にセットされます。マスタモードフォールトは、SPI インタフェースに次のような影響を与えます。

- MODF ビットがセットされ、ERRIE ビットがセットされている場合は SPI 割り込みが生成されます。
- SPE ビットがクリアされます。これによって、デバイスからのすべての出力がブロックされ、SPI インタフェースが無効になります。
- MSTR ビットがクリアされ、デバイスは強制的にスレーブモードになります。

MODF ビットをクリアするには、次のソフトウェアシーケンスを実行します。

1. MODF ビットがセットされている間、SPIx_SR レジスタを読み出し／書き込みアクセスを行います。
2. 次に、SPIx_CR1 レジスタに書き込みを行います。

複数の MCU で構成されるシステムでスレーブ間の競合を避けるには、MODF ビットをクリアするシーケンス中、NSS ピンをハイレベルにプルアップする必要があります。このクリアシーケンスの後、SPE ビットと MSTR ビットは、元の状態に戻すことができます。安全のため、MODF ビットがセットされている間、ハードウェアは SPE ビットと MSTR ビットのセットを許可しません。スレーブデバイスでは、MODF ビットはセットできません。ただし、前回のマルチマスタ競合の結果としてセットする場合は例外です。

CRC エラー (CRCERR)

このフラグを使用して、SPIx_CR1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。シフトレジスタに受信された値が、レシーバである SPIx_RXCRC の値と一致しなかった場合、SPIx_SR レジスタの CRCERR フラグがセットされます。フラグはソフトウェアによってクリアされます。

TI モードフレームフォーマットエラー (FRE)

SPI がスレーブモードで動作し、かつ TI モードプロトコルに準拠した設定となっている場合、通信の進行中に NSS パルスが発生すると、TI モードフレームフォーマットエラーが検出されます。このエラーが発生すると、SPIx_SR レジスタの FRE フラグがセットされます。エラー発生時には SPI は無効にされず、この NSS パルスは無視されます。SPI は次の NSS パルスを待ってから新規の転送を開始します。このエラーの検出により 2 バイトのデータが失われるため、データは破壊される可能性があります。

SPIx_SR レジスタを読み出すと、FRE フラグがクリアされます。ERRIE ビットがセットされている場合、NSS エラー検出時に割り込みが生成されます。この場合、データの一貫性が保証されなくなるため、SPI を無効にする必要があり、またスレーブ SPI が再び有効化された場合は、マスタによって通信を再起動する必要があります。

26.4 SPI の特殊機能

26.4.1 TI モード

マスタモードでの TI プロトコル

SPI インタフェースは TI プロトコルと互換性があります。SPIx_CR2 レジスタの FRF ビットを使って、SPI をこのプロトコルに準拠させるように設定することができます。

SPIx_CR1 レジスタにセットされる値によらず、クロックの極性と位相は TI プロトコル要件に必ず適合します。NSS 管理も TI プロトコルに固有なものになります。これにより、この場合の SPIx_CR1 レジスタと SPIx_CR2 レジスタによる NSS 管理の設定 (SSM, SSI, SSOE) ができなくなります。

スレーブモードでは、SPI ボーレートプリスケラを使用して、現在のトランザクションが終了した時点で、MISO ピンの状態がハイインピーダンスに変化するタイミングを制御します (図 282 を参照)。任意のボーレートが使用できるため、このタイミングを非常に柔軟に決定することができます。ただし、ボーレートは外部マスタクロックボーレートに設定されるのが一般的です。MISO 信号がハイインピーダンス (t_{release}) になるまでの遅延は、内部再同期と SPIx_CR1 レジスタの BR[2:0] ビットで設定されたボーレート値によって変わります。この値は次式で求められます：

$$\frac{t_{\text{baud_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud_rate}}}{2} + 6 \times t_{\text{pclk}}$$

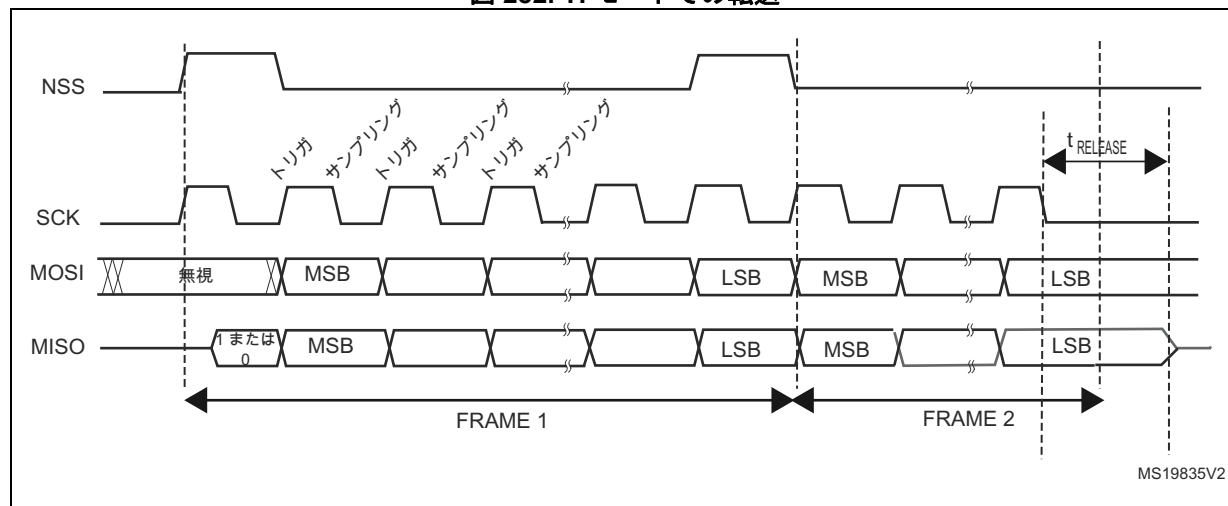
スレーブがデータフレームトランザクション中に NSS パルスの誤配置を検出すると、TIFRE フラグがセットされます。

この機能はモトローラの SPI 通信には使用できません (FRF ビットを 0 に設定)。

注： エラー割り込み (ERRIE = 1) を使ってスレーブ送信専用モードで TI フレームエラーを検出するには、SPI_CR1 レジスタで BIDIMODE と BIDIOE を 1 に設定して SPI を 2 線単方向モードに設定する必要があります。BIDIMODE を 0 に設定すると、OVR が 1 にセットされます。これは、データレジスタは決して読み出されることがなく、エラー割り込みが常に生成されるためです。これに対して BIDIMODE を 1 に設定した場合は、データが受信されず、OVR がセットされることはありません。

図 282：TI モードでの転送 TI モードが選択されているときの SPI の通信波形を示します。

図 282. TI モードでの転送



26.4.2 CRC 計算

2 つの CRC 計算機 (送信および受信データフロー用) が、送信データおよび受信データの信頼性をチェックするために実装されています。SPI は、DFF ビットを介して選択されたデータフォーマットに応じて、CRC8 または CRC16 を計算します。CRC は、SPI_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

CRC の原理

CRC 計算は、SPI が有効 (SPE = 1) になる前に、SPIx_CR1 レジスタの CRCEN ビットをセットすることによって有効になります。CRC 値は、各ビットに対して奇数のプログラム可能な多項式の値を使用して計算されます。計算は、SPIx_CR1 レジスタの CPHA ビットと CPOL ビットによって定義されるサンプリングクロックエッジで行われます。計算された CRC 値は、データブロックの最後のみならず、CPU または DMA によって管理される転送に関しても、自動的にチェックされます。受信データをもとに内部で計算された CRC 値とトランスミッタが送信した CRC 値の間に不一致が検出された場合、データ破壊エラーを示すために CRCERR フラグがセットされます。CRC 計算を処理する正しい手順は、SPI の設定および選択された転送管理によって変わります。

注： 多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。

CPU によって管理される CRC 転送

通信が開始され、最後のデータフレームが SPIx_DR レジスタで送信または受信されるまで正常に続きます。次に、CRC フレームトランザクションが現在処理中のデータフレームトランザクションの後に行われることを示すために、SPIx_CR1 レジスタの CRCNEXT ビットをセットする必要があります。CRCNEXT ビットは、最後のデータフレームトランザクションの終了前にセットしてください。CRC のトランザクション中は CRC 計算は行われません。

受信した CRC 値は、他のデータフレームと同じように Rx バッファに格納されます。

CRC フォーマットトランザクションは、データシーケンスの最後に通信を行うためにデータフレームをもう一つ受け取ります。

最後の CRC データを受信すると、受信した値と SPIx_RXCRC レジスタの値を比較する自動チェックが行われます。ソフトウェアは、SPIx_SR レジスタの CRCERR フラグをチェックして、データ転送の内容が破壊されているか否かを判断する必要があります。ソフトウェアは、CRCERR フラグに“0”を書き込んでクリアします。

CRC 受信後、CRC 値はRxバッファに格納され、RXNE フラグをクリアするために SPIx_DR レジスタを読み出す必要があります。

DMA によって管理される CRC 転送

SPI 通信が CRC 通信と DMA モードで有効化される場合、CRC の送受信は通信の最後に自動で行われます（ただし、受信専用モードで CRC データを読み出す場合を除く）。CRCNEXT ビットはソフトウェアで処理する必要はありません。SPI 送信用DMA チャンネルのカウンタは、転送するデータフレーム数からCRC フレームを除いてセットする必要があります。レシーバ側では、受信した CRC 値はトランザクションの終了時に DMA によって自動的に処理されますが、SPI_DR から受け取る CRC フレームは常にレシーバにロードされますので、ユーザは注意してそれを一掃してください。

転送中に内容の破壊が生じた場合、データと CRC の転送の最後に SPIx_SR レジスタの CRCERR フラグがセットされます。

SPIx_TXCRC および SPIx_RXCRC の値のリセット

SPIx_TXCRC および SPIx_RXCRC の値は、CRC 計算が有効になった時点で自動的にクリアされます。

CRC 機能を有効にした状態で SPI がスレーブモードに設定されると、NSS ピンにハイレベルが入力された場合でも、CRC 計算が行われます。これは、たとえば、通信マスタが複数のスレーブに交互に対処するマルチスレーブ環境の場合に起こることがあります。

スレーブの選択解除（NSS のハイレベル）と新しいスレーブの選択（NSS のローレベル）の間では、マスタとスレーブのそれぞれの CRC 計算を再同期するために、マスタ側とスレーブ側の両方でCRC 値をクリアしてください。

CRC をクリアするには、次の手順に従います。

1. SPI を無効にします。
2. CRCEN ビットをクリアします。
3. CRCEN ビットを有効にします。
4. SPI を有効にします。

注： SPI がスレーブモードのとき、SPE ビットの値に関係なく、CRCEN がセットされるとすぐに、CRC 計算機は SCK スレーブ入力クロックの影響を受けます。CRC 計算の間違いを回避するために、ソフトウェアは、クロックが安定している（定常状態にある）ときのみ、CRC 計算を有効にする必要があります。SPI インタフェースがスレーブとして設定されている場合、データフェーズと CRC フェーズの間で NSS 内部信号をローレベルに維持する必要があります。

26.5 SPI 割り込み

SPI 通信中、割り込みは次のイベントによって生成できます。

- ロードする準備ができているTxバッファの送信
- Rxバッファのデータ受信
- マスタモードフォールト
- オーバーランエラー
- TI フレームフォーマットエラー

割り込みは個別に有効または無効にできます。

表 146. SPI 割り込みリクエスト

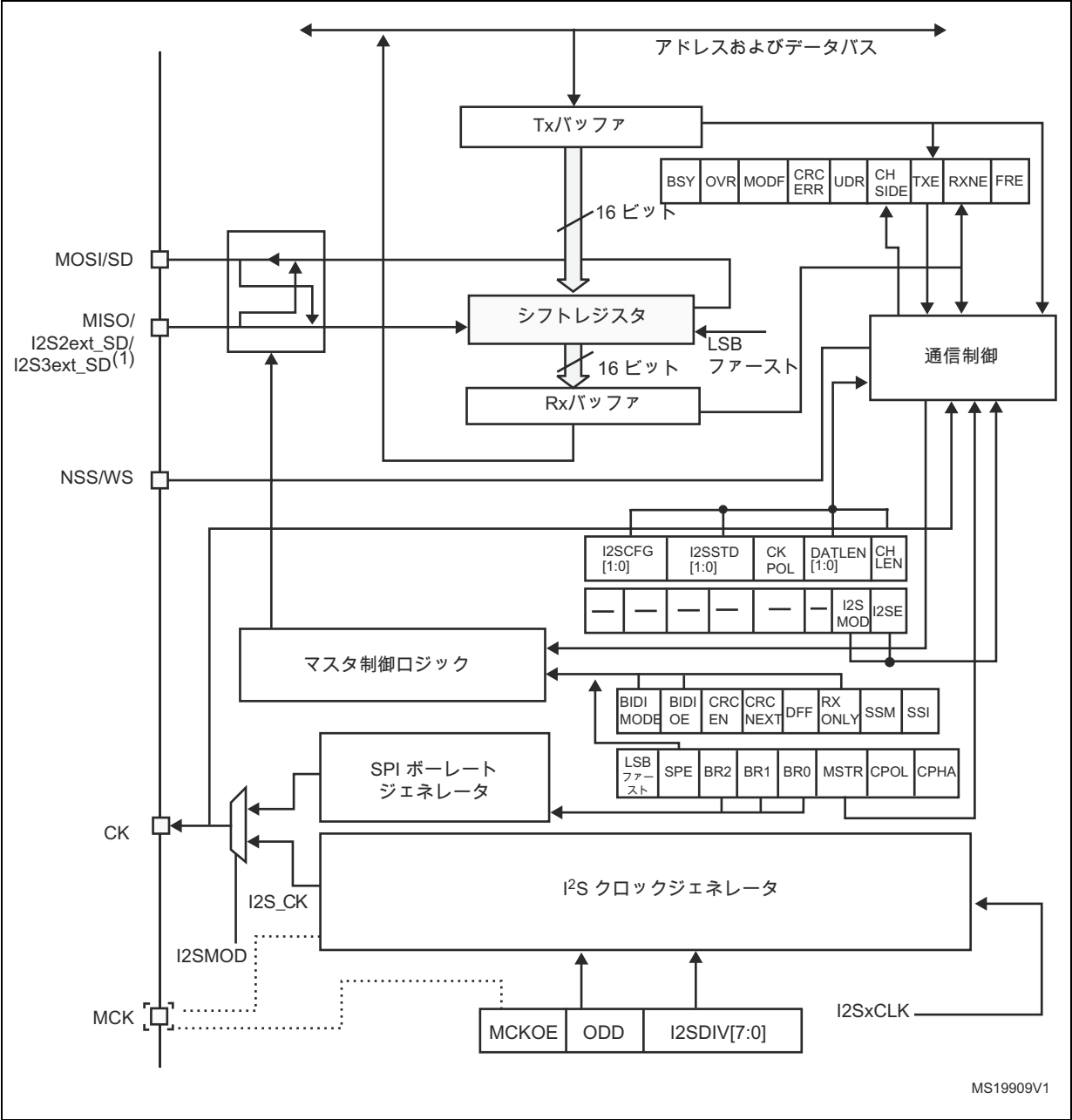
割り込みイベント	イベントフラグ	イネーブル制御ビット
ロードする準備ができているTxバッファの送信	TXE	TXEIE
Rxバッファのデータ受信	RXNE	RXNEIE
マスタモードフォールトイベント	MODF	ERRIE
オーバーランエラー	OVR	
CRC エラー	CRCERR	
TI フレームフォーマットエラー	FRE	

26.6 I²S の機能説明

26.6.1 I²S の概要

I²S のブロック図を 図 283 に示します。

図 283. I²S ブロック図



1. I2S2ext_SD と I2S3ext_SD は、I2S 全二重 モードを制御する拡張 SD ピンです。
2. MCK はMISO ピンに配置されます。

SPI はオーディオ I²S インタフェースとして機能することができます。それには、SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能を有効にしてください。このインタフェースは、主に SPI と同じピン、フラグ、および割り込みを使用します。

I²S と SPI は、以下の3つのピンを共用します。

- SD : MOSI ピンに配置され、2つの時間多重化データチャネルを送受信します（半二重モードのみ）。
- WS : NSS ピンに配置され、マスタモードではデータ制御信号の出力、スレーブモードでは入力です。
- CK : SCK ピンに配置され、マスタモードではシリアルクロック出力、スレーブモードではシリアルクロック入力です。

外部オーディオデバイスにマスタクロック出力が必要な場合、追加のピンを使用できます。

- MCK : 別々に配置され、I²S がマスタモードに設定されている（かつ、SPIx_I2SPR レジスタの MCKOE ビットがセットされている）とき、 $256 \times f_s$ と等しい（ f_s はオーディオサンプリング周波数）設定済みの周波数で生成されたこの追加クロックを出力するために使用されます。

I²S は、マスタモードに設定されているとき、専用のクロックジェネレータを使用して通信クロックを生成します。このクロックジェネレータは、マスタクロック出力のソースでもあります。I²S モードでは、2つの追加レジスタを使用できます。1つはクロックジェネレータ設定 SPIx_I2SPR にリンクされ、もう1つは汎用 I²S 設定レジスタ SPIx_I2SCFGR（オーディオ規格、スレーブ/マスタモード、データフォーマット、パケットフレーム、クロック極性など）です。

SPIx_CR1 レジスタとすべての CRC レジスタは、I²S モードでは使用されません。同様に、SPIx_CR2 レジスタの SSOE ビットと SPIx_SR レジスタの MODF および CRCERR ビットも使用されません。

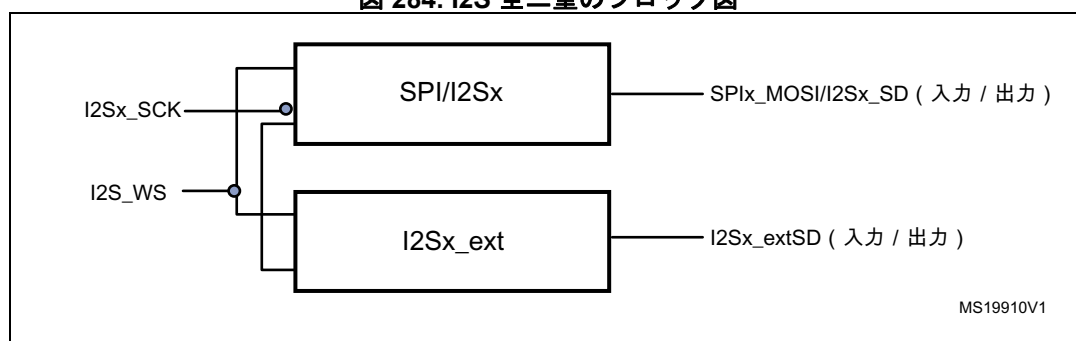
I²S は、16ビット幅モードでのデータ転送に同じ SPI レジスタ（SPIx_DR）を使用します。

26.6.2 I2S 全二重

I2S 全二重モードをサポートするため、I2S2 と I2S3 の他に拡張 I2S（I2S2_ext、I2S3_ext）と呼ぶ2つの追加 I²S インスタンスが利用可能です（[図 284](#) を参照）。そのため、最初の I2S 全二重インタフェースは I2S2 と I2S2_ext に、2つ目の全二重インタフェースは I2S3 と I2S3_ext に、それぞれ基づいています。

注 : I2S2_ext と I2S3_ext は全二重モードでのみ使用されます。

図 284. I2S 全二重のブロック図



1. x は 2 または 3 です。

I2Sx はマスタモードで動作することができます。その結果 :

- 半二重モードでは、I2Sx だけが SCK と WS を出力することができます。
- 全二重モードでは、I2Sx だけが I2S2_ext と I2S3_ext への SCK と WS を供給することができます。

拡張 I2S (I2Sx_ext) は、全二重モードでのみ使用することができます。I2Sx_ext は、常にスレーブモードで動作します。

I2Sx と I2Sx_ext はどちらも、トランスミッタまたはレシーバとして設定することができます。

26.6.3 サポートされるオーディオプロトコル

3 線バスでは、一般に 2 つのチャンネル（右チャンネルと左チャンネル）で時間多重化されたオーディオデータのみを処理する必要があります。しかしながら、送受信用には 1 つの 16 ビットレジスタしかありません。したがって、各チャンネルサイドに対応する適切な値をデータレジスタに書き込んだり、データレジスタからデータを読み出して SPIx_SR レジスタの CHSIDE ビットをチェックして対応するチャンネルを識別したりすることは、ソフトウェアでの処理になります。左チャンネルは常に最初に送信され、その後で右チャンネルが送信されます（CHSIDE は PCM プロトコルには無関係です）。

4 つのデータおよびパケットフレームを使用できます。データは次のフォーマットで送信されます。

- 16 ビットフレームにパックされた 16 ビットデータ
- 16 ビットフレームにパックされた 32 ビットデータ
- 24 ビットフレームにパックされた 32 ビットデータ
- 32 ビットフレームにパックされた 32 ビットデータ

32 ビットパケットに拡張された 16 ビットデータを使用するとき、最初の 16 ビット（MSB）は上位ビットであり、LSB の 16 ビットは、ソフトウェア動作や DMA リクエストを必要とせずに（1 回の読み込み／書き込み動作のみで）強制的に 0 にされます。

24 ビットと 32 ビットのデータフレームは、SPIx_DR レジスタとの間で 2 回の CPU 読み出しまたは書き込み動作（あるいはアプリケーションにとって DMA が望ましい場合は 2 回の DMA 動作）を必要とします。特に 24 ビットのデータフレームの場合、8 つの下位ビットはハードウェアによって 0 のビットで 32 ビットに拡張されます。

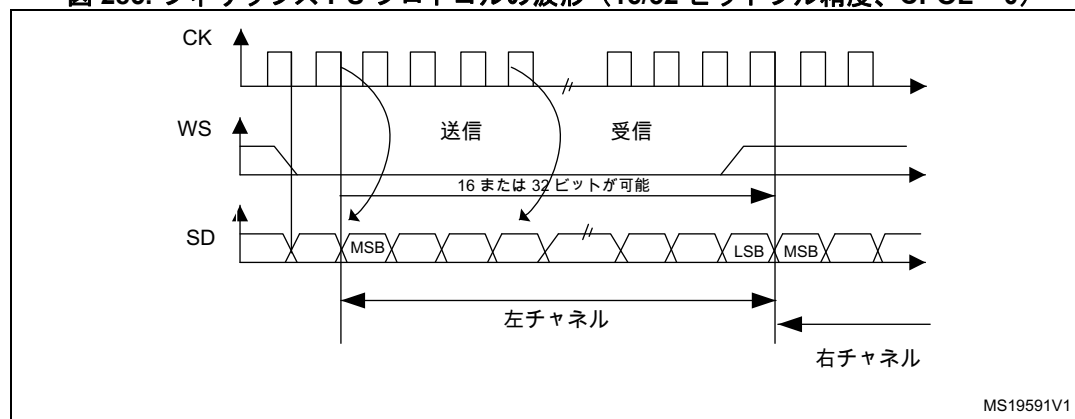
すべてのデータフォーマットと通信規格に対して、最上位ビットは常に最初に送信されます（MSB ファースト）。

I²S インタフェースは、SPIx_I2SCFGR レジスタの I2SSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ規格をサポートします。

フィリップス I²S 規格

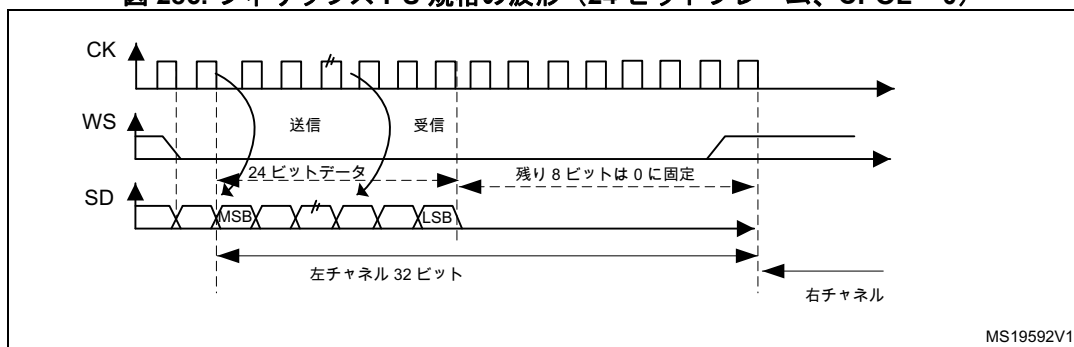
この規格では、どのチャンネルが送信されているかを示すために WS 信号を使用します。この信号が有効になってから 1 CK クロックサイクル後に最初のビット（MSB）が使用可能になります。

図 285. フィリップス I²S プロトコルの波形（16/32 ビットフル精度、CPOL = 0）



データは、CK の立ち下がりエッジでラッチされ（トランスミッタの場合）、立ち上がりエッジで読み出されます（レシーバの場合）。WS 信号も CK の立ち下がりエッジでラッチされます。

図 286. フィリップス I²S 規格の波形（24 ビットフレーム、CPOL = 0）

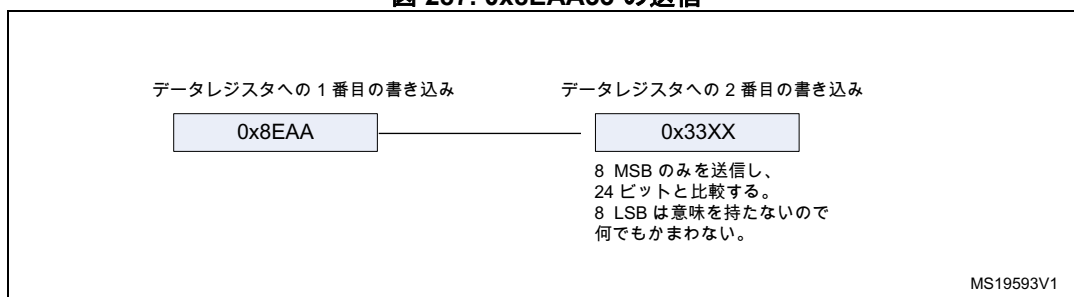


このモードでは、SPIx_DR レジスタに対して 2 回の書き込みまたは読み出し動作が必要です。

- 送信モードの場合

0x8EAA33 を送信する必要がある場合（24 ビット）：

図 287. 0x8EAA33 の送信



- 受信モードの場合

データ 0x8EAA33 が受信される場合：

図 288. 0x8EAA33 の受信

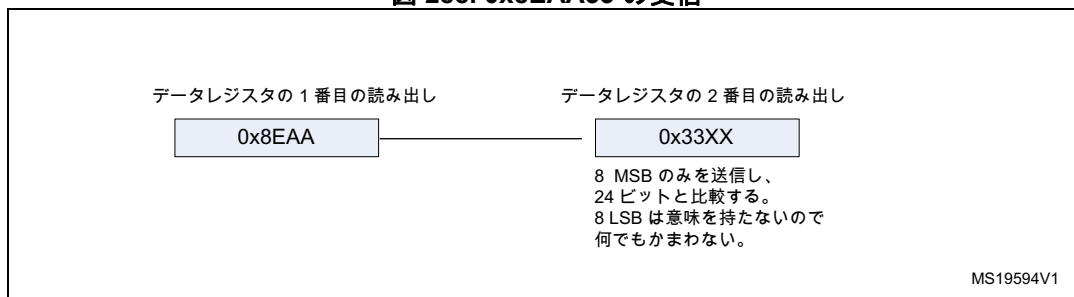
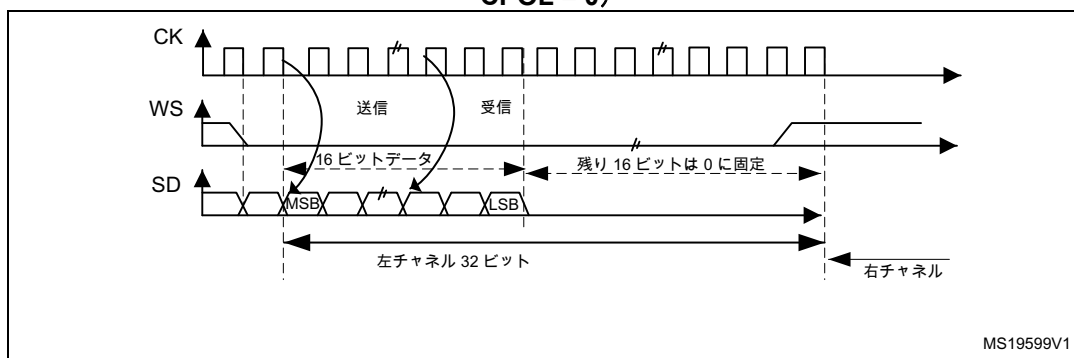


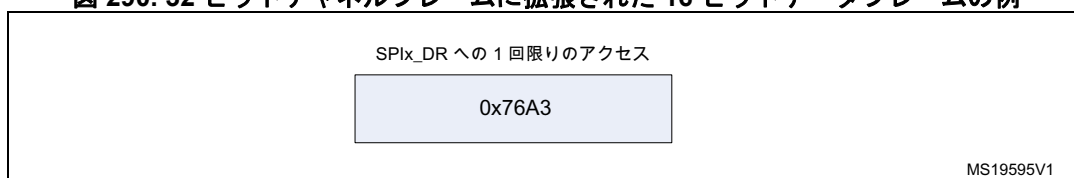
図 289. I²S フィリップス標準 (32 ビットパケットフレームに拡張された 16 ビット、CPOL = 0)



I²S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x76A30000) であるとき、[図 290](#) に示す動作が要求されます。

図 290. 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームの例



送信では、SPIx_DR に MSB が書き込まれるたびに、TXE フラグがセットされ、可能ならば、SPIx_DR レジスタに新しい送信値をロードするために割り込みが生成されます。これはハードウェアによって行われるため、0x0000 がまだ送信されていなくても行われます。

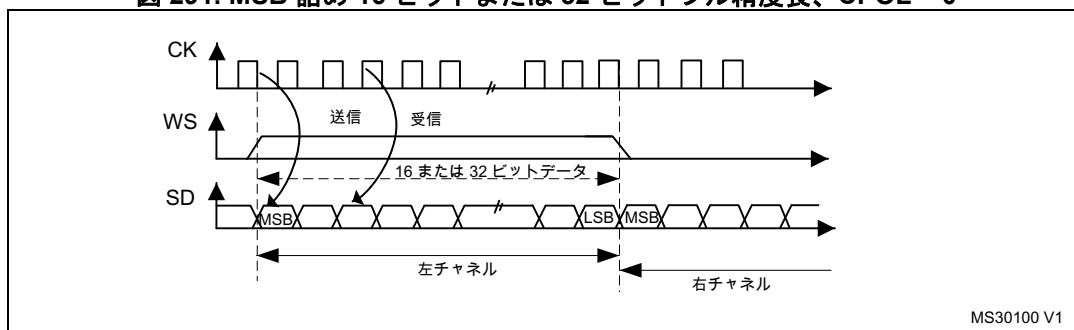
受信では、RXNE フラグがセットされ、可能ならば、最初の 16 MSB ハーフワードの受信時にその割り込みが生成されます。

このように、2 回の書き込みまたは読み出し動作の間にはより多くの時間が設けられるため、アンダーランまたはオーバーラン状態（データ転送の方向に依存）を避けられます。

MSB 詰め規格

この規格では、WS 信号は最初のデータビット（最上位ビット）と同時に生成されます。

図 291. MSB 詰め 16 ビットまたは 32 ビットフル精度長、CPOL = 0



データは、CK の立ち下がりエッジでラッチされ（トランスミッタの場合）、立ち上がりエッジで読み出されます（レシーバの場合）。

図 292. MSB 詰め 24 ビットフレーム長、CPOL = 0

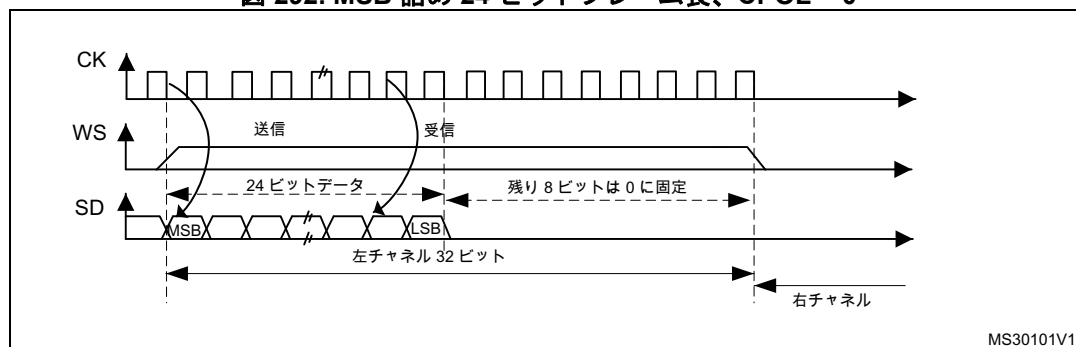
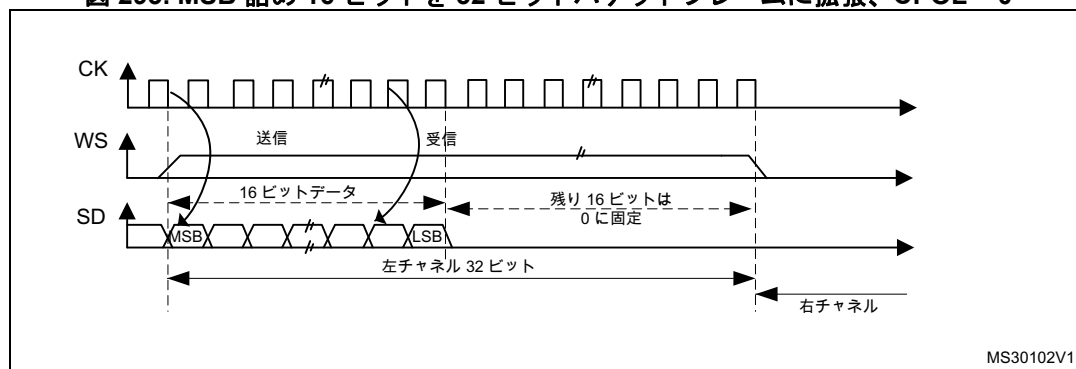


図 293. MSB 詰め 16 ビットを 32 ビットパッケージフレームに拡張、CPOL = 0



LSB 詰め規格

この規格は、MSB 詰め規格と似ています（16 ビットと 32 ビットのフル精度フレームフォーマットに違いはありません）。

図 294. LSB 詰め 16 ビットまたは 32 ビットフル精度、CPOL = 0

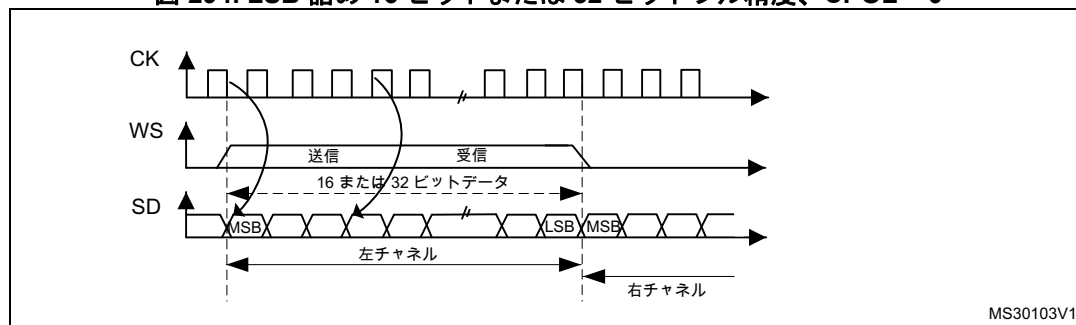
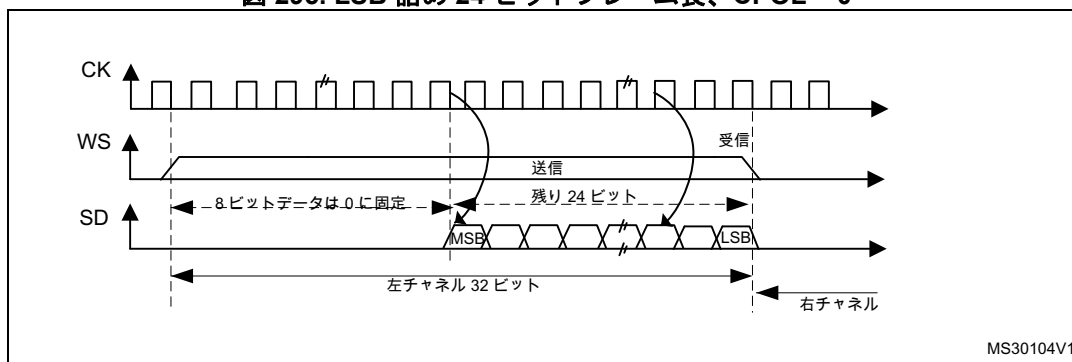


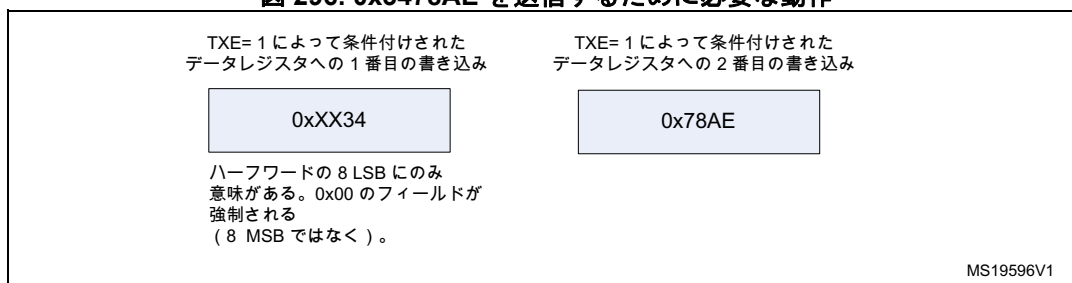
図 295. LSB 詰め 24 ビットフレーム長、CPOL = 0



● 送信モードの場合

データ 0x3478AE を送信する必要がある場合、ソフトウェアまたは DMA によって SPIx_DR レジスタへの 2 回の書き込み動作が必要です。この動作を次に示します。

図 296. 0x3478AE を送信するために必要な動作



● 受信モードの場合

データ 0x3478AE が受信される場合、RXNE イベントごとに、SPIx_DR レジスタから連続する 2 回の読み出し動作が必要です。

図 297. 0x3478AE の受信に必要な動作

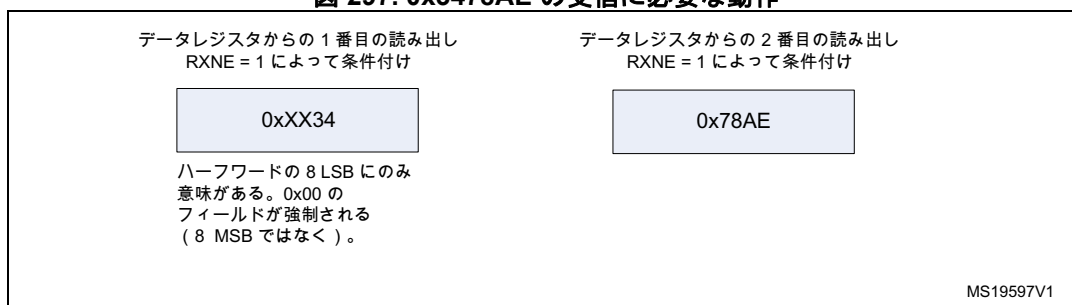
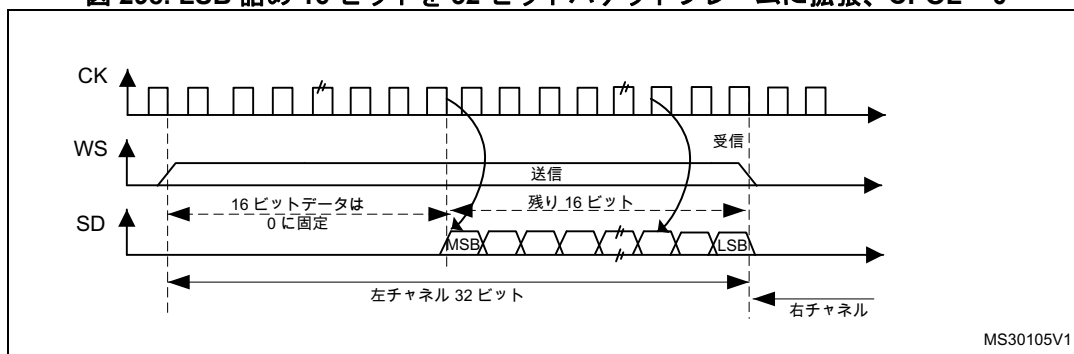


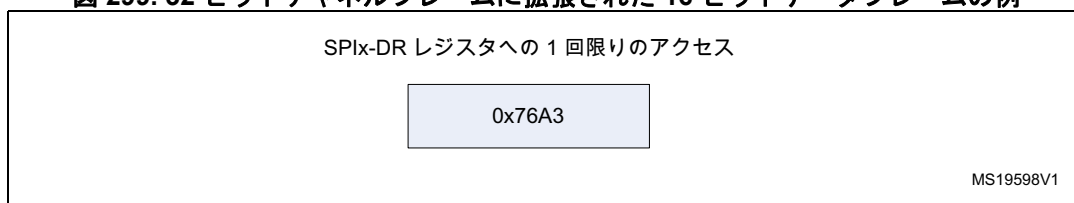
図 298. LSB 詰め 16 ビットを 32 ビットパケットフレームに拡張、CPOL = 0



I²S 設定フェーズで、32 ビットチャネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。この場合、それはハーフワード MSB に相当します。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x0000 76A3) であるとき、図 299 に示す動作が要求されます。

図 299. 32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例



送信モードでは、TXE イベントが発生すると、アプリケーションは送信するデータ（この場合は 0x76A3）を書き込む必要があります。0x000 フィールドが最初に送信されます（32 ビット拡張部）。有効なデータ（0x76A3）が SD に送信されると、すぐに TXE フラグが再びセットされます。

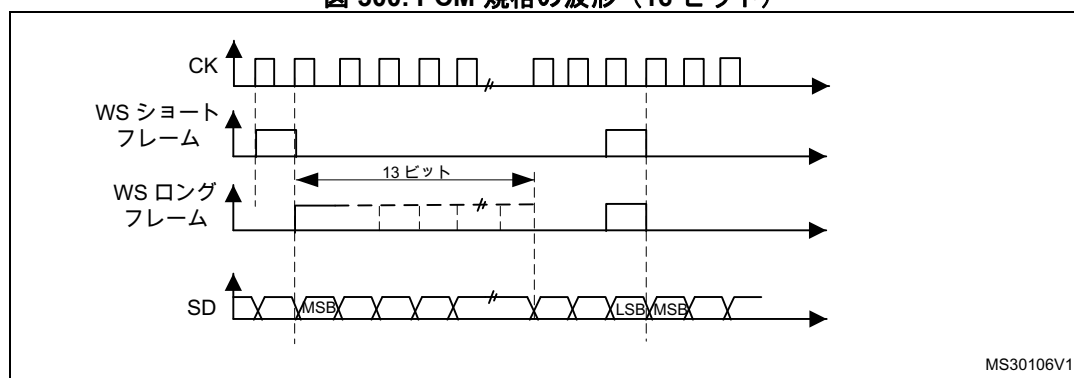
受信モードでは、0x0000 フィールドではない有効なハーフワードが受信されると、すぐに RXNE がアサートされます。

このように、アンダーランやオーバーランの状態を防ぐために、2 回の書き込みまたは読み出し動作の間には、より多くの時間が設けられています。

PCM 規格

PCM 規格では、チャネルサイド情報を使用する必要はありません。SPIx_I2SCFGR レジスタの PCMSYNC ビットを使用して、2 つの PCM モード（ショートおよびロングフレーム）の使用および設定が可能です。

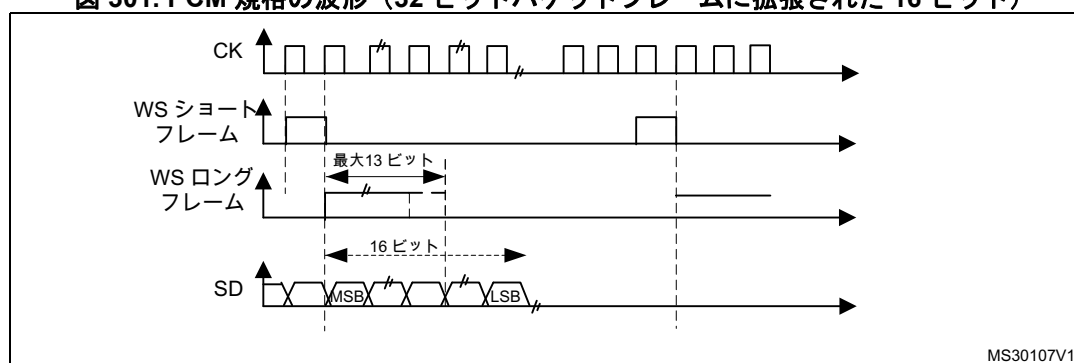
図 300. PCM 規格の波形 (16 ビット)



ロングフレーム同期では、WS 信号のアサーション時間はマスタモードで 13 ビットに固定されています。

ショートフレーム同期では、WS 同期信号の長さは、わずか 1 サイクルです。

図 301. PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)



注： 2 つのモード (マスタとスレーブ) と 2 つの同期 (ショートとロング) に関しては、スレーブモードでも、連続した 2 つのデータ (したがって 2 つの同期信号) 間のビット数を (SPIx_I2SCFGR レジスタの DATLEN および CHLEN ビットで) 指定する必要があります。

26.6.4 クロックジェネレータ

The I²S ビットレートにより、I²S データライン上のデータフローと I²S クロック信号周波数が決まります。

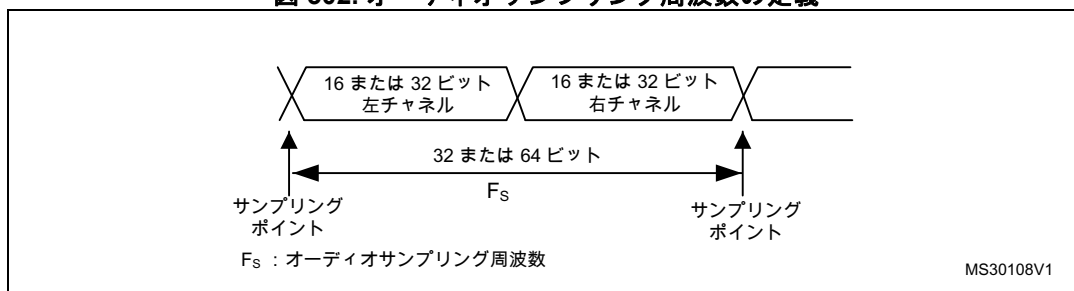
I²S ビットレート = チャンネルあたりのビット数 × チャンネル数 × オーディオサンプリング周波数

16 ビットオーディオ、左チャンネルおよび右チャンネルの場合、I²S ビットレートは次のように算出されます。

$$I^2S \text{ ビットレート} = 16 \times 2 \times f_s$$

パケット長が 32 ビットの場合、I²S ビットレート = 32 × 2 × f_s となります。

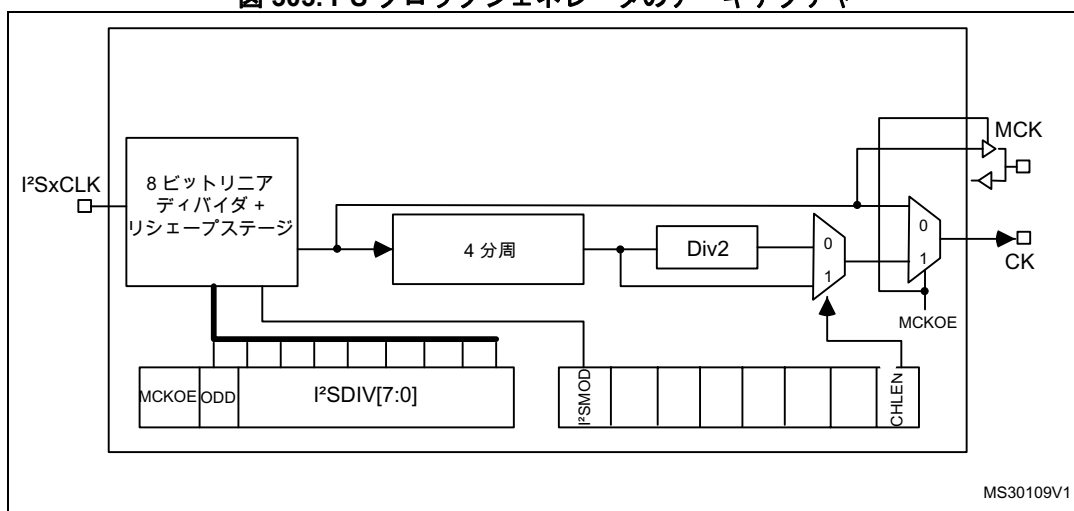
図 302. オーディオサンプリング周波数の定義



マスタモードが設定された場合、希望するオーディオ周波数で通信するために、特定の措置を講じてリニアディバイダを適切にプログラムする必要があります。

図 303 に、通信クロックのアーキテクチャを示します。I2Sx クロックは常にシステムクロックです。

図 303. I²S クロックジェネレータのアーキテクチャ



1. x = 2

オーディオサンプリング周波数の取り得る値は、192 KHz、96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz、8 kHz（またはその他のこの範囲内にある値）。希望する周波数を達成するには、次の式に基づいてリニアディバイダをプログラムする必要があります。

マスタクロックが生成される（SPIx_I2SPR レジスタの MCKOE ビットをセットされる）場合

$$f_s = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$

$$f_s = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

マスタクロックが無効にされた場合（MCKOE ビットをクリア）

$$f_s = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$

$$f_s = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

表 147 に、さまざまなクロック設定に対する精度値の例を示します。

注： 最適クロック精度を可能にするその他の設定も可能です。

表 147. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度⁽¹⁾

SYSCCLK (MHz)	データ長	I2SDIV	I2SODD	MCLK	ターゲット f_s (Hz)	実際の f_s (Hz)	誤差
48	16	8	0	なし	96000	93750	2.3438%
48	32	4	0	なし	96000	93750	2.3438%
48	16	15	1	なし	48000	48387.0968	0.8065%
48	32	8	0	なし	48000	46875	2.3438%
48	16	17	0	なし	44100	44117.647	0.0400%
48	32	8	1	なし	44100	44117.647	0.0400%
48	16	23	1	なし	32000	31914.8936	0.2660%
48	32	11	1	なし	32000	32608.696	1.9022%
48	16	34	0	なし	22050	22058.8235	0.0400%
48	32	17	0	なし	22050	22058.8235	0.0400%
48	16	47	0	なし	16000	15957.4468	0.2660%
48	32	23	1	なし	16000	15957.447	0.2660%
48	16	68	0	なし	11025	11029.4118	0.0400%
48	32	34	0	なし	11025	11029.412	0.0400%
48	16	94	0	なし	8000	7978.7234	0.2660%
48	32	47	0	なし	8000	7978.7234	0.2660%
48	16	2	0	あり	48000	46875	2.3430%
48	32	2	0	あり	48000	46875	2.3430%
48	16	2	0	あり	44100	46875	6.2925%
48	32	2	0	あり	44100	46875	6.2925%
48	16	3	0	あり	32000	31250	2.3438%
48	32	3	0	あり	32000	31250	2.3438%
48	16	4	1	あり	22050	20833.333	5.5178%
48	32	4	1	あり	22050	20833.333	5.5178%
48	16	6	0	あり	16000	15625	2.3438%
48	32	6	0	あり	16000	15625	2.3438%
48	16	8	1	あり	11025	11029.4118	0.0400%
48	32	8	1	あり	11025	11029.4118	0.0400%
48	16	11	1	あり	8000	8152.17391	1.9022%
48	32	11	1	あり	8000	8152.17391	1.9022%

1. この表の値は、さまざまなクロック設定に対するほんの一例です。最適クロック精度を可能にするその他の設定も可能です。

26.6.5 I²S マスタモード

I²S は次のように設定することができます：

- 送信用または受信用のマスタモード (I2Sx を使用した半二重モード)
- 送受信用のマスタモード (I2Sx および I2Sx_ext を使用した全二重モード)。

つまり、シリアルクロックは、ワードセレクト信号 WS だけでなく、CK ピン上でも生成されます。マスタクロック (MCK) は、SPIx_I2SPR レジスタの MCKOE ビットで、出力するかしないかを制御できます。

手順

1. 適切なオーディオサンプリング周波数に到達するシリアルクロックボーレートを定義するため、SPIx_I2SPR レジスタの I2SDIV[7:0] ビットを選択します。SPIx_I2SPR レジスタの ODD ビットも定義する必要があります。
2. 通信クロックの一定したレベルを定義するために、CKPOL ビットを選択します。外部の ADC オーディオコンポーネントにマスタクロック MCK を供給する必要がある場合、SPIx_I2SPR レジスタの MCKOE ビットをセットします (I2SDIV と ODD の値は、MCK 出力の状態に応じて計算する必要があります。詳細については、[セクション 26.6.4 : クロックジェネレータ](#)を参照)。
3. SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能を有効にし、I2SSTD[1:0] と PCMSYNC ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してチャンネルあたりのビット数を、それぞれ選択します。SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットにより I²S マスタモードと方向 (トランスミッタまたはレシーバ) も選択します。
4. 必要の場合は、SPIx_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
5. SPIx_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

WS と CK は出力モードに設定されます。SPIx_I2SPR の MCKOE ビットがセットされている場合、MCK も出力です。

送信シーケンス

送信シーケンスは、Txバッファにハーフワードが書き込まれたときに開始されます。

Txバッファに書き込まれる最初のデータは、左チャンネルのデータに対応すると想定します。データが Txバッファからシフトレジスタに転送されると、TXE がセットされ、右チャンネルに対応するデータを Txバッファに書き込む必要があります。CHSIDE フラグは、どのチャンネルが送信されるかを示します。CHSIDE フラグは TXE がハイレベルになったときに更新されるため、このフラグは TXE フラグがセットされた場合に意味を持ちます。

完全なフレームとは、左チャンネルのデータ送信と、それに続く右チャンネルのデータ送信であるとみなす必要があります。左チャンネルのみが送信される部分的フレームは実現できません。

データハーフワードは、最初のビット送信時に 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

選択された I²S 規格モードに応じた書き込み動作の詳細については、[セクション 26.6.3 : サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx_DR レジスタに書き込む必要があります。

I2SE をクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。



受信シーケンス

動作モードは、ポイント 3 を除いて、送信モードの場合と同じです（[セクション 26.6.5: I²S マスタモード](#)に示す手順を参照）。ポイント 3 では、I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットの packets によって受信されます。つまり、Rxバッファがフルになるたびに RXNE フラグがセットされ、SPIx_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

RXNE ビットは、SPIx_DR レジスタの読み出しによってクリアされます。

CHSIDE は毎回の受信後に更新されます。CHSIDE は、I²S セルによって生成される WS 信号に反応します。

選択された I²S 規格モードに応じた読み出し動作の詳細については、[セクション 26.6.3: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx_CR2 レジスタの ERRIE ビットがセットされている場合、割り込みが生成されてエラーを知らせます。

I²S をスイッチオフするには、I²S が新しいデータ転送を開始することなく転送サイクルを適切に完了できるように、特定の動作が要求されます。そのシーケンスは、データ長とチャンネル長の設定、および選択したオーディオプロトコルモードに依存します。ケース別の説明

- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00、CHLEN = 1)、LSB 詰めモードを使用 (I2SSTD = 10)
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 17 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00 および CHLEN = 1)、MSB 詰め、I²S または PCM モード (それぞれ、I2SSTD = 00、I2SSTD = 01、または I2SSTD = 11)
 - a) 最後の RXNE を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- DATLEN と CHLEN のその他すべての組み合わせについては、I2SSTD ビットを通じて選択したオーディオモードが何であれ、次のシーケンスを実行して I²S をスイッチオフします。
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。

注： 転送時、BSY フラグはローレベルに保持されます。

26.6.6 I²S スレーブモード

I²S は次のように設定することができます：

- 送信用または受信用のスレーブモード (I2Sx を使用した半二重モード)
- 送受信用のスレーブモード (I2Sx および I2Sx_ext を使用した全二重モード)。

動作モードは、主として I²S マスタ設定に関して述べたものと同じ規則に従います。スレーブモードでは、I²S インタフェースによって生成されるクロックはありません。クロックと WS 信号は、I²S インタフェースに接続された外部マスタから入力されます。したがって、ユーザがクロックを設定する必要はありません。

設定ステップを次に示します。

1. SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S モードを選択し、I2SSTD[1:0] ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してフレームのチャンネルあたりのビット数を、それぞれ選択します。SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットにより、スレーブのモード (送信または受信) も選択します。
2. 必要の場合は、SPIx_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
3. SPIx_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

送信シーケンス

送信シーケンスは、外部マスタデバイスがクロックを送信したときと、NSS_WS 信号がデータの転送を要求したときに開始されます。スレーブを有効にしなければ、外部マスタは通信を開始しません。I²S データレジスタは、マスタが通信を開始する前に、ロードする必要があります。

I²S、MSB 詰め、および LSB 詰めモードの場合、データレジスタに書き込まれる最初のデータは、左チャンネルのデータに対応します。通信が開始されると、データはTxバッファからシフトレジスタに転送されます。次に、右チャンネルのデータを I²S データレジスタに書き込むように要求するために、TXE フラグがセットされます。

CHSIDE フラグは、どのチャンネルが送信されるかを示します。スレーブモードでは、マスタ送信モードに比べて、CHSIDE は外部マスタからの WS 信号に反応します。つまり、スレーブが最初のデータの送信準備をしなければ、マスタはクロックを生成できません。WS アサーションは、最初に送信される左チャンネルに対応します。

注： *I2SE は、マスタの最初のクロックが CK ラインに到達するよりも、少なくとも 2 PCLK サイクル前に書き込まれる必要があります。*

データハーフワードは、最初のビット送信時に内部バスから 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

なお、Txバッファへの書き込みの前に、TXE フラグが 1 であることを確認する必要があります。

選択された I²S 規格モードに応じた書き込み動作の詳細については、[セクション 26.6.3: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx_DR レジスタに書きこむ必要があります。次のデータ通信の最初のクロックエッジよりも前にデータが SPIx_DR レジスタに書き込まれない場合、アンダーランフラグがセットされ、割り込みが生成されることがあります。これによって、転送データに誤りがあることがソフトウェアに知らされます。SPIx_CR2 レジスタの ERRIE ビットがセットされた場合、SPIx_SR レジスタの UDR フラグがハイレベルになると割り込みが生成されます。この場合、I²S をスイッチオフし、左チャネルからデータ転送をリスタートする必要があります。

I2SE をクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

受信シーケンス

動作モードは、ポイント 1 を除いて、送信モードの場合と同じです（[セクション 26.6.6: I²S スレーブモード](#)に示す手順を参照）。ポイント 1 では、SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャネルの長さに関係なく、オーディオデータは 16 ビットのパケットによって受信されます。つまり、Rxバッファがフルになるたびに SPIx_SR レジスタの RXNE フラグがセットされ、SPIx_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャネルまたは左チャネルに受信されることがあります。

CHSIDE フラグは、SPIx_DR レジスタから読み出されるデータが受信されるたびに更新されます。このフラグは、外部マスタコンポーネントによって管理される外部 WS ラインに反応します。

RXNE ビットは、SPIx_DR レジスタの読み出しによってクリアされます。

選択された I²S 規格モードに応じた読み出し動作の詳細については、[セクション 26.6.3: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx_CR2 レジスタの ERRIE ビットがセットされた場合、エラーを示すために割り込みが生成されます。

受信モードで I²S をスイッチオフするには、最後の RXNE = 1 を受信した直後に I2SE をクリアする必要があります。

注： 外部マスタコンポーネントには、オーディオチャネルを介して 16 ビットまたは 32 ビットのパケットでデータを送受信する機能が必要です。

26.6.7 I²S ステータスフラグ

アプリケーションが I²S バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます（このフラグへの書き込みは無効）。このフラグは I²S の通信層の状態を示します。

BSY がセットされると、I²S が通信中でビジーであることを示します。マスタ受信モード (I2SCFG=11) には 1 つの例外があり、BSY フラグは受信時にローレベルに保持されます。

ソフトウェアが I²S を無効にする必要がある場合、BSY フラグは転送の終わりを検出するために役立ちます。これによって、最後の転送データの破壊を回避します。このため、下記の手順を厳守する必要があります。

BSY フラグは転送が開始されるとセットされます。ただし、I²S がマスタ受信モードにある場合を除きます。

BSY フラグをクリアするタイミング

- 転送が完了したとき（ただし、通信が連続的だと思われるマスタ送信モードの場合を除く）
- I²S が無効化されたとき

通信が連続的な場合

- マスタ送信モードでは、BSY フラグはすべての転送期間を通じてハイレベルに保持されます。
- スレーブモードでは、BSY フラグは、各転送間で I²S の 1 クロックサイクルの間ローレベルになります。

注： 各データの送受信の処理には BSY フラグを使用しないでください。代わりに、TXE フラグと RXNE フラグを使用することをお勧めします。

Txバッファエンプティフラグ (TXE)

このフラグがセットされると、Txバッファはエンプティ（空）であり、次に送信するデータをバッファにロードできることを示します。送信されるデータがすでにTxバッファに格納されているとき、TXE フラグはリセットされます。TXE フラグは、I²S が無効にされている（I2SE ビットがリセット）ときにもリセットされます。

Rxバッファノットエンプティ (RXNE)

このフラグがセットされると、Rxバッファに有効な受信データがあることを示します。このフラグは、SPIx_DR レジスタが読み出されるとリセットされます。

チャネルサイドフラグ (CHSIDE)

送信モードでは、このフラグは TXE がハイレベルになるとリフレッシュされます。このフラグは、SD 上の転送データが属するチャネルサイドを示します。スレーブ送信モードでアンダーランエラーイベントが発生した場合、このフラグは信頼できないため、通信を再開する前に、I²S をスイッチオフし、さらにスイッチオンする必要があります。

受信モードでは、このフラグは SPIx_DR にデータが受信されるとリフレッシュされます。このフラグは、どちらのチャネルサイドからデータが受信されたかを示します。なお、エラー（OVR など）が発生した場合、このフラグは無意味になるため、I²S を無効にし、さらに有効にすることによってリセットする必要があります（変更が必要な場合は設定する）。

このフラグは、PCM 規格では意味を持ちません（ショートとロングフレームの両モード）。

SPIx_SR の OVR または UDR フラグがセットされ、SPIx_CR2 の ERRIE ビットもセットされると、割り込みが生成されます。この割り込みをクリアするには、割り込みソースをいったんクリアした後で、SPIx_SR ステータスレジスタを読み出します。

26.6.8 I²S エラーフラグ

I²S セルには 3 つのエラーフラグがあります。

アンダーランフラグ (UDR)

スレーブ送信モードでは、ソフトウェアが SPIx_DR にまだ値をロードしていない間に、データ送信用の最初のクロックが現れると、このフラグがセットされます。このフラグは、SPIx_I2SCFGR レジスタの I2SMOD ビットがセットされると使用できます。SPIx_CR2 レジスタの ERRIE ビットがセットされている場合は、割り込みを生成できます。

UDR ビットは、SPIx_SR レジスタの読み出し動作によってクリアされます。



オーバーランフラグ (OVR)

このフラグがセットされるのは、データが受信され、前のデータが SPIx_DR レジスタからまだ読み出されていないときです。結果として、受信データは失われます。SPIx_CR2 の ERRIE ビットがセットされている場合、割り込みが生成されることがあります。

この場合、受信バッファの内容は、送信側のデバイスからの新しい受信データによって更新されません。SPIx_DR レジスタへの読み出し動作によって、前に正しく受信されたデータが返されます。それ以降に送信されたすべてのハーフワードは失われます。

OVR ビットをクリアするには、SPIx_DR レジスタを読み出し、続けて SPIx_SR レジスタを読み出しアクセスを行います。

フレームエラーフラグ (FRE)

このフラグは、I2S がスレーブモードに設定された場合にのみハードウェアによってセットすることができます。このフラグは、スレーブが WS ラインの変化を想定していないときに外部マスタが WS ラインを変化させようとしている場合にセットされます。同期が失われた場合、この状態から回復し、外部マスタデバイスと I2S スレーブデバイスを再同期させるためには、次のステップに従います。

- 1. I2S を無効にします。
- 2. 正しいレベルが WS ラインで検出されたとき (WS ラインは I2S モードではハイレベル、MSB 詰めまたは LSB 詰めあるいは PCM モードではローレベル)、再度有効にします。

マスタデバイスとスレーブデバイスの間の同期外れは、SCK 通信クロック上または WS フレーム同期ライン上のノイズの多い環境に起因する可能性があります。ERRIE ビットをセットすれば、エラー割り込みを生成させることができます。同期外れフラグ (FRE) は、ステータスレジスタを読み出すときに、ソフトウェアでクリアされます。

26.6.9 I2S 割り込み

表 148 に I2S 割り込みのリストを示します。

表 148. I2S 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信バッファエンptyフラグ	TXE	TXEIE
受信バッファノットエンptyフラグ	RXNE	RXNEIE
オーバーランエラー	OVR	ERRIE
アンダーランエラー	UDR	
フレームエラーフラグ	FRE	

26.6.10 DMA の機能

I2S モードでは、DMA は SPI モードとまったく同じように機能します。異なる点は、I2S モードではデータ転送保護システムがないため、CRC 機能を使用できない点のみです。



26.7 SPI および I²S レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。さらに、SPI_DR へは8 ビット単位でアクセスできます。

レジスタの説明で使用されている略語のリストについては、[セクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

26.7.1 SPI 制御レジスタ 1 (SPI_CR1) (I²S モードでは使用しません)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **BIDIMODE** : 双方向データモードイネーブル

このビットは、共通の双方向データラインを 1 本使用して、半二重通信を有効にします。双方向モードがアクティブのときは、RXONLY ビットをクリアされたままにします。

0 : 2 線単方向データモードを選択します。

1 : 1 線双方向データモードを選択します。

注 : このビットは I²S モードでは使用しません。

ビット 14 **BIDIOE** : 双方向モードでの出力イネーブル

双方向モードでの転送方向は、このビットと BIDIMODE ビットを組み合わせて選択します。

0 : 出力は無効です（受信専用モード）。

1 : 出力は有効です（送信専用モード）。

注 : このビットは I²S モードでは使用しません。

MISO ピンがスレーブモードで使用されている間、マスタモードでは、**MOSI** ピンが使用されます。

ビット 13 **CRCEN** : ハードウェア CRC 計算イネーブル

0 : CRC 計算は無効です。

1 : CRC 計算は有効です。

注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。

このフラグは I²S モードでは使用しません。

ビット 12 **CRCNEXT** : CRC 転送指示

0 : データフェーズです（CRC フェーズなし）。

1 : 次の転送は CRC です（CRC フェーズ）。

注 : SPI が全二重モードまたは送信専用モードに設定されている場合、最後のデータが SPI_DR に書き込まれた直後に、CRCNEXT ビットを書き込む必要があります。

SPI が受信専用モードに設定されている場合、最後から 2 番目のデータが受信された後に CRCNEXT ビットをセットする必要があります。

転送が DMA で管理される場合、このビットはクリアされた状態に保持する必要があります。

このフラグは I²S モードでは使用しません。

ビット 11 DFF : データフレームフォーマット

- 0 : 送受信に 8 ビットデータフレームフォーマットが選択されます。
- 1 : 送受信に 16 ビットデータフレームフォーマットが選択されます。

注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。
このフラグは I²S モードでは使用しません。

ビット 10 RXONLY : 受信専用モードイネーブル

このビットは、データ受信専用の単方向ラインを 1 本使用して、単方向通信を有効にします。受信専用モードがアクティブのときは、BIDIMODE ビットをクリアされたままにします。

このビットはマルチスレーブシステムでも役立ちます。そのシステムでは、この特定のスレーブはアクセスされず、アクセスされたスレーブからの出力は破壊されません。

- 0 : 全二重 (送受信)
- 1 : 出力は無効です (受信専用モード)。

注 : このビットは I²S モードでは使用しません。

ビット 9 SSM : ソフトウェアスレーブ管理

SSM ビットがセットされているとき、NSS ピンの入力 SSI ビットからの値に置き換えられます。

- 0 : ソフトウェアスレーブ管理は無効です。
- 1 : ソフトウェアスレーブ管理は有効です。

注 : このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 8 SSI : 内部スレーブ選択

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値は強制的に NSS ピンに設定され、NSS ピンの IO 値は無視されます。

注 : このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 7 LSBFIRST : フレームフォーマット

- 0 : MSB が最初に送信されます。
- 1 : LSB が最初に送信されます。

注 : このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 6 SPE : SPI イネーブル

- 0 : ペリフェラルは無効です。
- 1 : ペリフェラルは有効です。

注 : このビットは I²S モードでは使用しません。

SPI を無効にするときは、セクション 26.3.10 : SPI を無効にする手順に記載されている手順に従ってください。

ビット 5:3 BR[2:0] : ボーレート制御

- 000 : f_{PCLK} /2
- 001 : f_{PCLK} /4
- 010 : f_{PCLK} /8
- 011 : f_{PCLK} /16
- 100 : f_{PCLK} /32
- 101 : f_{PCLK} /64
- 110 : f_{PCLK} /128
- 111 : f_{PCLK} /256

注 : これらのビットは、通信中には変更しないでください。
これらのビットは I²S モードでは使用されません。

- ビット 2 **MSTR** : マスタ選択
- 0 : スレーブ設定
 - 1 : マスタ設定
- 注:** このビットは、通信中には変更しないでください。
このフラグは I²S モードでは使用しません。
- Bit1 **CPOL** : クロック極性
- 0 : アイドル時に CK は 0 となります。
 - 1 : アイドル時に CK は 1 となります。
- 注:** このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。
- ビット 0 **CPHA** : クロック位相
- 0 : 最初のクロック遷移が最初のデータキャプチャエッジです。
 - 1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。
- 注:** このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。

26.7.2 SPI 制御レジスタ 2 (SPI_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEIE	RXNEIE	ERRIE	FRF	Res.	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw	rw		rw	rw	rw

- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TXEIE** : Txバッファエンプティ割り込みイネーブル
- 0 : TXE 割り込みはマスクされます。
 - 1 : TXE 割り込みはマスクされません。TXE フラグがセットされたとき、割り込みリクエストの生成に使用されます。
- ビット 6 **RXNEIE** : Rxバッファノットエンプティ割り込みイネーブル
- 0 : RXNE 割り込みはマスクされます。
 - 1 : RXNE 割り込みはマスクされません。RXNE フラグがセットされたとき、割り込みリクエストの生成に使用されます。
- ビット 5 **ERRIE** : エラー割り込みイネーブル
- このビットは、エラー条件が発生したとき (SPI モードでは OVR、CRCERR、MODF、FRE ; I²S モードでは UDR、OVR、FRE)、割り込みの生成を制御します。
- 0 : エラー割り込みはマスクされます。
 - 1 : エラー割り込みは有効です。
- ビット 4 **FRF** : フレームフォーマット
- 0 : SPI モトローラモード
 - 1 : SPI TI モード
- 注:** このビットは I²S モードでは使用しません。
- ビット 3 予約済み。ハードウェアによって 0 に固定されています。

ビット 2 **SSOE** : SS 出力イネーブル

- 0 : マスタモードで SS 出力は無効にされ、セルはマルチマスタ設定で機能できます。
1 : セルが有効であるとき、マスタモードで SS 出力は有効です。セルはマルチマスタ環境では機能できません。

注 : このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 1 **TXDMAEN** : Txバッファ DMA イネーブル

- このビットがセットされると、TXE フラグがセットされるたびに DMA リクエストが行われます。
0 : Txバッファ DMA は無効です。
1 : Txバッファ DMA は有効です。

ビット 0 **RxDMAEN** : Rxバッファ DMA イネーブル

- このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが行われます。
0 : Rxバッファ DMA は無効です。
1 : Rxバッファ DMA は有効です。

26.7.3 SPI ステータスレジスタ (SPI_SR)

アドレスオフセット : 0x08

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
							r	r	r	r	rc_w0	r	r	r	r

ビット 15:9 予約済み。ハードウェアによって 0 に固定されています。

ビット 8 **FRE** : フレームエラー

- 0 : フレームエラーはありません。
1 : フレームエラーが発生しました。
このビットは、ハードウェアによってセットされ、SPIx_SR レジスタを読み出すとソフトウェアでクリアされます。
このビットは、選択したオーディオプロトコルに関係なく、SPI TI モードまたは I2S モードで使用されます。また、スレーブモードにおいて予想されないタイミングで起こる NSS または WS ライン上の変化を検出し、外部マスタデバイスとスレーブの間の同期外れに関する情報を提供します。

ビット 7 **BSY** : ビジーフラグ

- 0 : SPI (または I2S) はビジー状態ではありません。
1 : SPI (または I2S) が通信ビジー状態であるか、または Tx バッファが空ではありません。
このフラグはハードウェアによってセット/クリアされます。

注 : **BSY フラグを使用する際は注意が必要です。セクション 26.3.12 : SPI ステータスフラグ およびセクション 26.3.10 : SPI を無効にする手順を参照してください。**

ビット 6 **OVR** : オーバーランフラグ

- 0 : オーバーランは発生していません。
1 : オーバーランが発生しました。
このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション 26.3.13 : SPI エラーフラグ](#)を参照してください。

ビット 5 **MODF** : モードフォールト

- 0 : モードフォールトは発生していません。
1 : モードフォールトが発生しました。
このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション 26.4 \(814 ページ\)](#)を参照してください。

注 : このビットは I²S モードでは使用しません。

ビット 4 CRCERR : CRC エラーフラグ

0 : 受信した CRC 値が SPI_RXCRCR 値と一致します。

1 : 受信した CRC 値が SPI_RXCRCR 値と一致しません。

このフラグは、ハードウェアによってセットされ、ソフトウェアによって 0 を書き込むことでクリアされます。

注： このビットは I²S モードでは使用しません。

ビット 3 UDR : アンダーランフラグ

0 : アンダーランは発生していません。

1 : アンダーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション 26.6.8 : I²S エラーフラグ](#)を参照してください。

注： このビットは SPI モードでは使用しません。

ビット 2 CHSIDE : チャネルサイド

0 : 左チャネルを送信する必要があるか、または受信が行われました。

1 : 右チャネルを送信する必要があるか、または受信が行われました。

注： このビットは、SPI モードに対しては使用しません。また、PCM モードでは意味を持ちません。

ビット 1 TXE : 送信バッファエンプティ

0 : Txバッファは空ではありません。

1 : Txバッファは空です。

ビット 0 RXNE : 受信バッファノットエンプティ

0 : Rxバッファは空です。

1 : Rxバッファは空ではありません。

26.7.4 SPI データレジスタ (SPI_DR)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DR[15:0]** : データレジスタ

受信したデータまたは送信されるデータ

データレジスタは 2 つのバッファに分割されます。一方は書き込み用 (送信バッファ) であり、他方は読み出し用 (受信バッファ) です。データレジスタへの書き込みは、Tx バッファへの書き込みになります。データレジスタからの読み出しでは、Rx バッファに保持されている値が返されます。

注: 以下の注記は SPI モードに適用されます。

送受信されるデータは、データフレームフォーマットの選択ビット (SPI_CR1 レジスタの DFF ビット) に応じて、8 ビットまたは 16 ビットになります。正しい動作のためには、この選択を行ってから、SPI を有効にする必要があります。

8 ビットデータフレームの場合、バッファは 8 ビットであり、レジスタの LSB (SPI_DR[7:0]) のみが送受信に使用されます。受信モードのとき、レジスタの MSB (SPI_DR[15:8]) は強制的に 0 にされます。

16 ビットデータフレームの場合、バッファは 16 ビットであり、レジスタ全体 (SPI_DR[15:0]) が送受信に使用されます。

26.7.5 SPI CRC 多項式レジスタ (SPI_CRCPR) (I²S モードでは使用しません)

アドレスオフセット : 0x10

リセット値 : 0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CRCPOLY[15:0]** : CRC 多項式レジスタ

このレジスタは、CRC 計算用の多項式を格納します。

CRC 多項式 (0007h) は、このレジスタのリセット値です。必要に応じて、別の多項式を設定することができます。

注: これらのビットは I²S モードでは使用しません。

26.7.6 SPI RX CRC レジスタ (SPI_RXCRCR) (I²S モードでは使用しません)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **RXCRC[15:0]** : Rx CRC レジスタ

CRC 計算が有効なとき、RxCRC[15:0] ビットには、その後に受信したバイトから算出された CRC 値が格納されています。このレジスタは、SPI_CR1 レジスタの CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPI_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPI_CR1 の DFF ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPI_CR1 レジスタの DFF ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注: BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。これらのビットは I²S モードでは使用しません。

26.7.7 SPI TX CRC レジスタ (SPI_TXCRCR) (I²S では使用しません)

アドレスオフセット : 0x18

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **TXCRC[15:0]** : Tx CRC レジスタ

CRC 計算が有効なとき、TxCRC[7:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。このレジスタは、SPI_CR1 の CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPI_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。データフレームフォーマットが 8 ビットデータに設定された場合 (SPI_CR1 の DFF ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPI_CR1 レジスタの DFF ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注: BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。これらのビットは I²S モードでは使用しません。

26.7.8 SPI_I2S 設定レジスタ (SPI_I2SCFGR)

アドレスオフセット : 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ASTRE N	I2SMOD	I2SE	I2SCFG		PCMSY NC	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
				rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **ASTREN** : 非同期開始イネーブル

0 : 非同期開始は無効です。I2S がスレーブモードで有効な場合、I2S スレーブは I2S クロックを受信し、WS 信号で適切な遷移（選択されたプロトコルによる）を検出すると、転送を開始します。

1 : 非同期開始は有効です。I2S がスレーブモードで有効な場合、I2S スレーブは I2S クロックをマスタから予測される WS 信号の遷移をチェックせずに受信すると、即座に転送を開始します。

注 : *注 : WS 信号での適切な遷移は、I2S フィリップス規格が使用されている場合は立ち下がリエッジ、それ以外の規格の場合は立ち上がりエッジです。*

ビット 11 **I2SMOD** : I2S モード選択

0 : SPI モードが選択されます。
1 : I2S モードが選択されます。

注 : *このビットは、SPI または I2S が無効なときに設定してください。*

ビット 10 **I2SE** : I2S イネーブル

0 : I2S ペリフェラルは無効です。
1 : I2S ペリフェラルは有効です。

注 : *このビットは SPI モードでは使用しません。*

ビット 9:8 **I2SCFG** : I2S 設定モード

00 : スレーブ - 送信
01 : スレーブ - 受信
10 : マスタ - 送信
11 : マスタ - 受信

注 : *このビットは、I2S が無効なときに設定してください。
SPI モードでは使用しません。*

ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期
1 : ロングフレーム同期

注 : *このビットは、I2SSTD = 11 (PCM 規格使用) の場合にのみ意味を持ちます。
SPI モードでは使用しません。*

ビット 6 予約済みであり、ハードウェアによって 0 に固定されています。

ビット 5:4 **I2SSTD** : I2S 規格選択

- 00 : フィリップス I²S 規格
- 01 : MSB 詰め規格 (左詰め)
- 10 : LSB 詰め規格 (右詰め)
- 11 : PCM 規格

I²S 規格に関する詳細は、[セクション 26.6.3 \(820 ページ\)](#) を参照してください。SPI モードでは使用しません。

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。

ビット 3 **CKPOL** : 定常状態クロック極性

- 0 : I²S クロックの定常状態はローレベルです。
- 1 : I²S クロックの定常状態はハイレベルです。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。

このビットは SPI モードでは使用しません。

ビット 2:1 **DATLEN** : 転送されるデータ長

- 00 : 16 ビットデータ長
- 01 : 24 ビットデータ長
- 10 : 32 ビットデータ長
- 11 : 設定禁止

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。

このビットは SPI モードでは使用しません。

ビット 0 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

- 0 : 16 ビット幅
- 1 : 32 ビット幅

ビット書き込み動作は、DATLEN = 00 のときにのみ意味を持ちます。そうでない場合、書き込まれた値とは無関係に、チャネル長はハードウェアによって 32 ビットに固定されます。SPI mode では使用しません。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。

26.7.9 SPI_I2S プリスケアラレジスタ (SPI_I2SPR)

アドレスオフセット : 0x20

リセット値 : 0000 0010 (0x0002)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV							
						rw	rw	rw							

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 MCKOE : マスタクロック出カインーブル

- 0 : マスタクロック出力は無効です。
- 1 : マスタクロック出力は有効です。

注: このビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときのみ使用します。
このビットは SPI モードでは使用しません。

ビット 8 ODD : プリスケアラの奇数分周比

- 0 : 真のディバイダ値 = I2SDIV * 2
- 1 : 真のディバイダ値 = (I2SDIV * 2) + 1

セクション 26.6.4 (826 ページ) を参照してください。SPI モードでは使用しません。

注: このビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときのみ使用します。

ビット 7:0 I2SDIV : I2S リニアプリスケアラ

I2SDIV [7:0] = 0 または I2SDIV [7:0] = 1 は禁止されている値です。

セクション 26.6.4 (826 ページ) を参照してください。SPI モードでは使用しません。

注: これらのビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときのみ使用します。



26.7.10 SPI レジスタマップ

次の表に、SPI レジスタマップとリセット値を示します。

表 149. SPI レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	SPI_CR1	Res.																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	SPI_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEIE	RXNEIE	ERRIE	FRF	Res.	SFOE	TXDMAEN	RXDMAEN					
	リセット値																								0	0	0	0		0	0	0	0				
0x08	SPI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE					
	リセット値																							0	0	0	0	0	0	0	1	0					
0x0C	SPI_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	SPI_CRCPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCPOLY[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1				
0x14	SPI_RXCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RxCRC[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x18	SPI_TXCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TxCRC[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	SPI_I2SCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ASTREN	I2SMOD	I2SE	I2SCFG	PCMSSYNC	Res.	I2SSD	CKPOL	DATLEN	CHLEN									
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x20	SPI_I2SPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV											
	リセット値																						0	0	0	0	0	0	0	0	1	0	0				

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。

27 セキュアデジタル入力／出カインタフェース (SDIO)

27.1 SDIO の主な特長

SD/SDIO MMC カードホストインタフェース (SDIO) は、APB2 ペリフェラルバスと MultiMediaCard (MMC)、SD メモリカード、および SDIO カード間のインタフェースを提供します。

MMCA 技術委員会から発行された MultiMediaCard システム仕様書は、JEDECのウェブサイトから入手できます。

SD メモリカードと SD I/O カードシステム仕様書は、SD アソシエーションのウェブサイトから入手できます。

SDIO には以下の機能があります。

- MultiMediaCard System Specification Version 4.2 に完全準拠。3 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）、4 ビットおよび 8 ビット
- 以前のバージョンの MultiMediaCard との完全互換（上位互換性）
- SD Memory Card Specifications Version 2.0 に完全準拠
- SD I/O Card Specification Version 2.0 完全準拠：2 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）および 4 ビット
- 8 ビットモードで最大 50 MHz のデータ転送
- 外部双方向ドライバを制御するデータおよびコマンド出力有効信号

- 注：
- 1 SDIO は SPI 互換の通信モードを備えていません。
 - 2 SD メモリカードプロトコルは、MultiMediaCard system specification V2.11 で定義されている MultiMediaCard プロトコルのスーパーセットです。SD メモリデバイスに必要なコマンドの中には、SD I/O 専用カードやコンボカードの I/O 部分ではサポートされないものもあります。消去コマンドなど、SD I/O デバイスで使用されないコマンドは、SDIO プロトコルでサポートされません。また、コマンドの中には SD メモリカードと SD I/O カードで異なるものがあり、このようなコマンドは SDIO プロトコルでサポートされません。詳細については、SD I/O card Specification Version 1.0 を参照してください。

MultiMediaCard/SD バスは、カードをコントローラに接続します。

現行バージョンの SDIO は、1 度に 1 枚の SD/SDIO/MMC 4.2 カードと、MMC4.1 以前のスタックだけをサポートします。

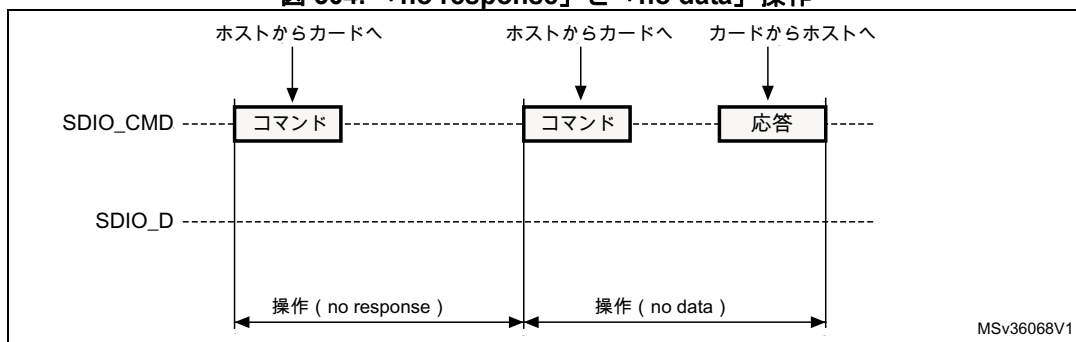
27.2 SDIO バストポロジ

バス上の通信は、コマンドとデータ転送に基づいて行われます。

MultiMediaCard/SD/SD I/O バス上の基本的なトランザクションは、コマンド/レスポンストランザクションです。これらの種類のバストランザクションでは、コマンドやレスポンスの決められた通信フォーマットでカードの情報を直接転送します。また、データトークンを持つ操作もあります。

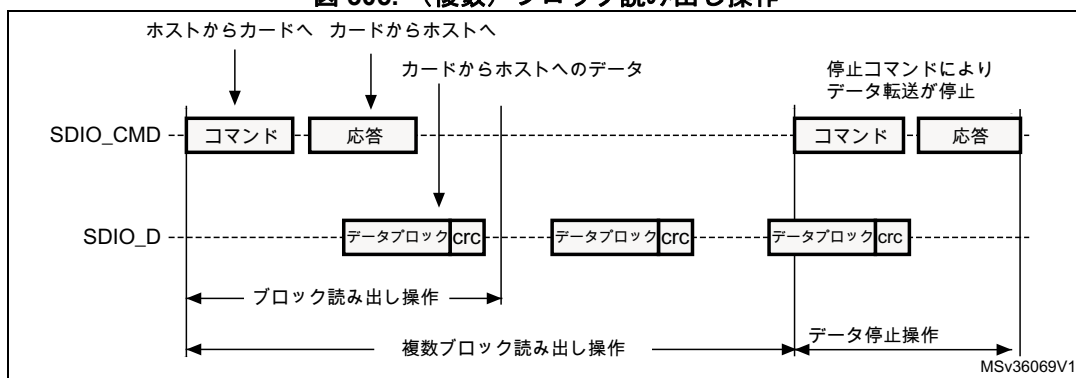
SD/SDIO メモリカードに対するデータ転送は、データブロック単位で行われます。MMC に対するデータ転送は、データブロックまたはストリーム単位で行われます。

図 304. 「no response」と「no data」操作



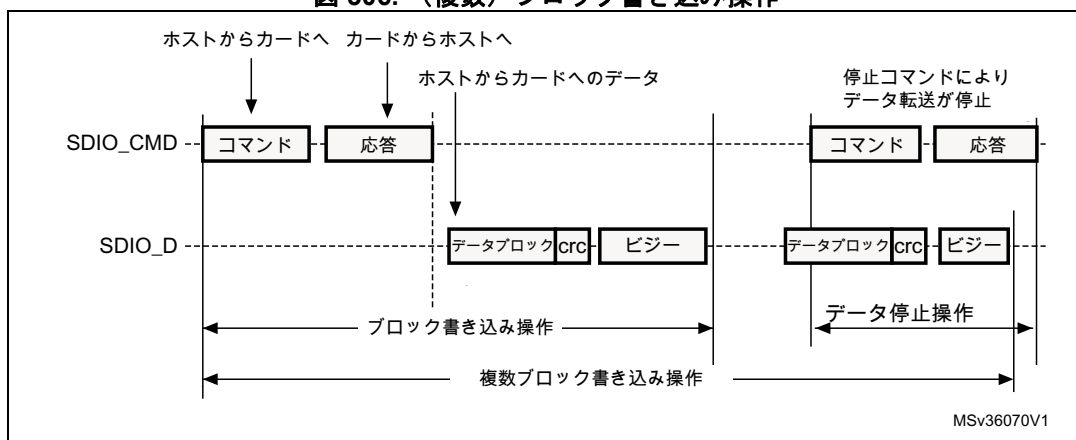
MSv36068V1

図 305. (複数) ブロック読み出し操作



MSv36069V1

図 306. (複数) ブロック書き込み操作



MSv36070V1

注: SDIO は、ビジー信号がアサートされている限り (SDIO_D0 をローへ固定)、データを一切送信しません。

図 307. 連続読み出し操作

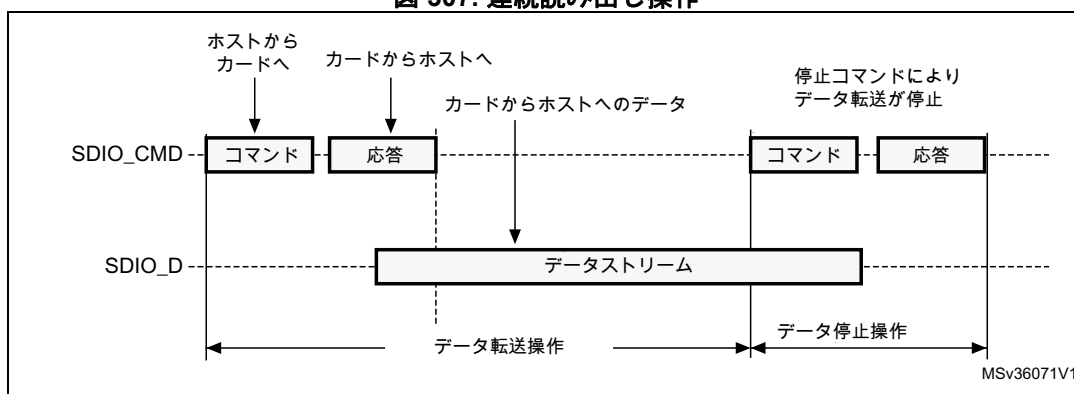
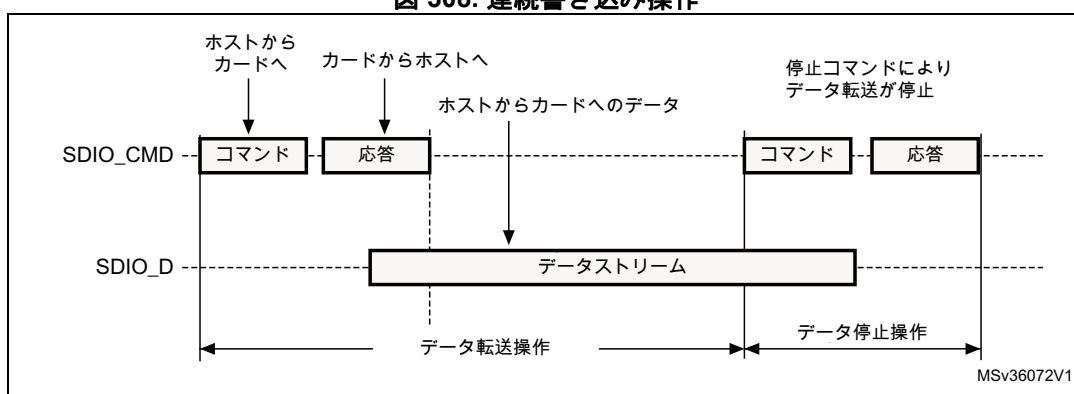


図 308. 連続書き込み操作

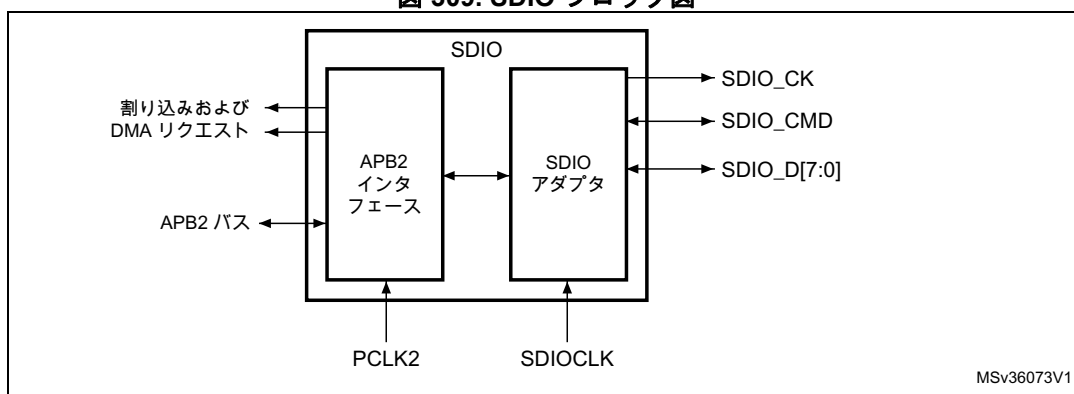


27.3 SDIO機能詳細

SDIO は、2 つの部分で構成されています。

- SDIO アダプタブロックは、クロック生成ユニット、コマンド、およびデータ転送など、MMC/SD/SD I/O カードに固有のすべての機能を提供します。
- APB2 インタフェースは、SDIO アダプタレジスタにアクセスして、割り込みおよび DMA リクエスト信号を生成します。

図 309. SDIO ブロック図



デフォルトでは、SDIO_D0 はデータ転送に使用されます。初期化後、ホストはデータバス幅を変更することができます。

MultiMediaCard がバスに接続されている場合、SDIO_D0、SDIO_D[3:0]、または SDIO_D[7:0] はデータ転送に使用できます。MMC V3.31 以前では、1 ビットのデータのみがサポートされるため、使用できるのは SDIO_D0 だけです。

SD または SD I/O カードがバスに接続されている場合、ホストはデータ転送に SDIO_D0 または SDIO_D[3:0] を使用するよう設定できます。すべてのデータラインはプッシュプルモードで動作します。

SDIO_CMD には、2 つの動作モードがあります。

- 初期化の場合は、オープンドレイン (MMCV3.31 以前)
- コマンド転送の場合は、プッシュプル (SD/SD I/O カード MMC4.2 は、初期化にもプッシュプルドライバを使用)

SDIO_CK は、カードに供給されるクロックです。各クロックサイクルで、コマンドラインとデータラインの両方に 1 ビットが転送されます。

SDIO は 2 つのクロック信号を使用します。

- SDIO アダプタクロック (SDIOCLK = 50 MHz)
- APB2 バスクロック (PCLK2)

PCLK2 および SDIO_CK クロック周波数は、次の条件を満足している必要があります。

$$\text{Frequency(PCLK2)} > ((3 \times \text{Width}) / 32) \times \text{Frequency(SDIO_CK)}$$

表 150 に示す信号は、MultiMediaCard/SD/SD I/O カードバス上で使用されます。

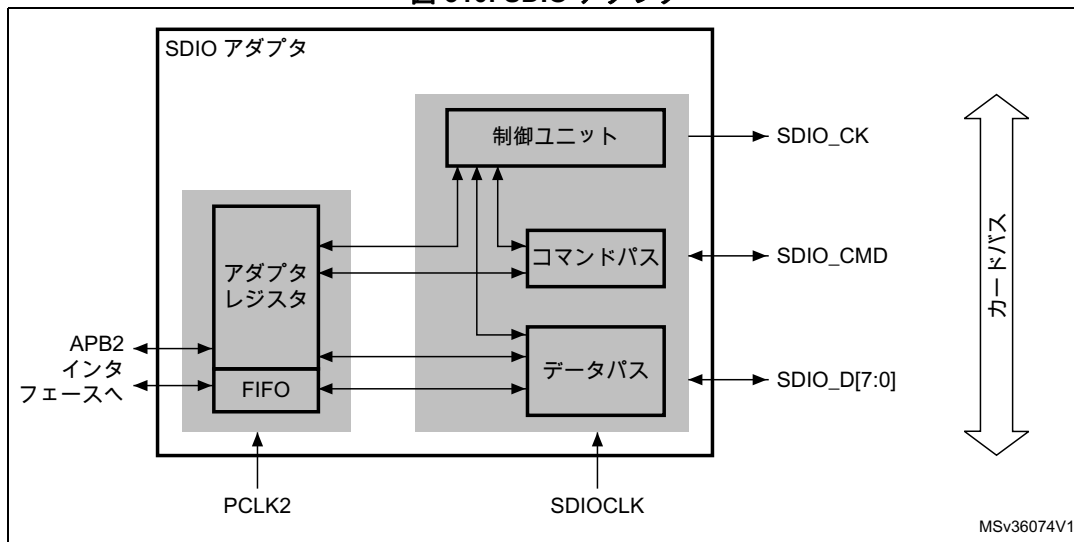
表 150. SDIO I/O 定義

ピン	方向	説明
SDIO_CK	出力	MultiMediaCard/SD/SDIO カードクロック。このピンは、ホストからカードへのクロックです。
SDIO_CMD	双方向	MultiMediaCard/SD/SDIO カードコマンド。このピンは、双方向のコマンド／レスポンス信号です。
SDIO_D[7:0]	双方向	MultiMediaCard/SD/SDIO カードデータ。これらのピンは、双方向データバスです。

27.3.1 SDIO アダプタ

図 310 は、SDIO アダプタの簡略化されたブロック図です。

図 310. SDIO アダプタ



SDIO アダプタは、マルチメディアカードスタックやセキュアデジタルメモリカードとのインタフェースを提供するマルチメディア/セキュアデジタルメモリカードバスマスタです。5つのサブユニットで構成されています。

- アダプタレジスタブロック
- 制御ユニット
- コマンドバス
- データバス
- データ FIFO

注： アダプタレジスタと FIFO は、APB2 バスクロックドメイン (PCLK2) を使用します。制御ユニット、コマンドバス、およびデータバスは、SDIO アダプタクロックドメイン (SDIOCLK) を使用します。

アダプタレジスタブロック

アダプタレジスタブロックは、すべてのシステムレジスタを含んでいます。このブロックは、マルチメディアカードのスタティックフラグをクリアする信号も生成します。クリア信号は、SDIO クリアレジスタの対応するビット位置に 1 が書き込まれると生成されます。

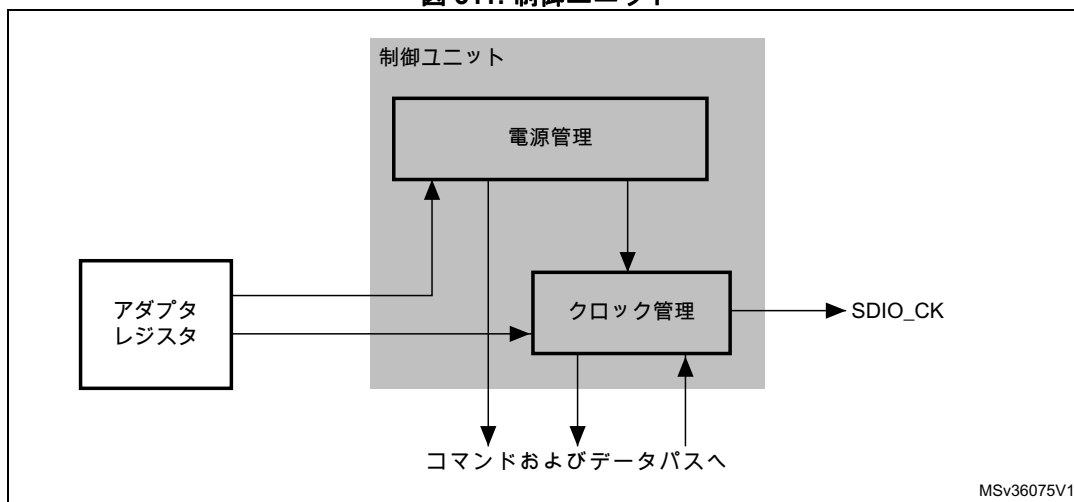
制御ユニット

制御ユニットには、電源管理機能とメモリカードクロックのクロック分周器が含まれています。

3つの電源フェーズがあります。

- パワーオフ
- パワーアップ
- パワーオン

図 311. 制御ユニット



制御ユニットは、図 311 に示されます。電源管理サブユニットとクロック管理サブユニットで構成されています。

電源管理サブユニットは、パワーオフおよびパワーアップフェーズの間、カードバス出力信号を無効にします。

クロック管理サブユニットは、SDIO_CK 信号を生成し、制御します。SDIO_CK 出力は、クロック分周またはクロックバイパスモードを使用することができます。クロック出力は、次の場合にインアクティブです。

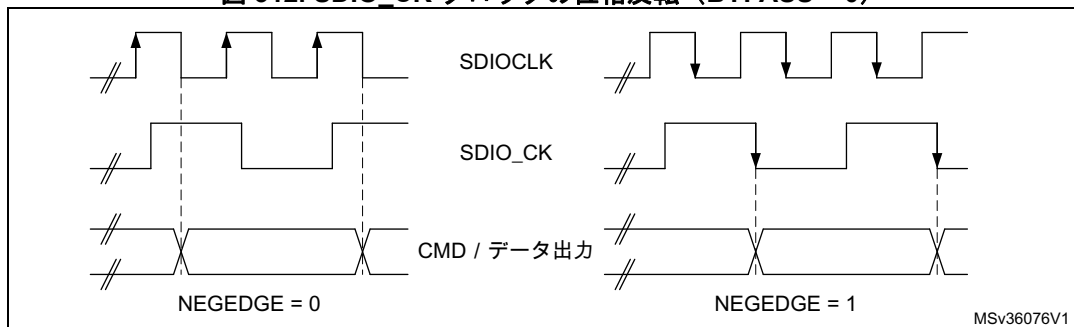
- リセット後
- パワーオフまたはパワーアップフェーズ中
- 節電モードが有効であり、カードバスがアイドル状態の場合（コマンドおよびデータバスサブユニットがアイドルフェーズに入ってから 8 クロック周期後）

クロック管理サブユニットは、SDIO_CK位相をずらす制御をします。バイパスモードでない場合、SDIO コマンドとデータ出力は SDIO_CK の立ち上がりエッジの次の SDIO CLK 立ち下がりエッジで生成されます。SDIO_CK 立ち上がりエッジは、SDIO_CLKCR[13] ビットがリセット (NEGEDGE = 0) された場合に SDIOCLK 立ち上がりエッジで発生します。SDIO_CLKCR[13] ビットがセットされると (NEGEDGE = 1)、SDIO_CK 立ち下がりエッジで SDIO コマンドとデータが変更されます。

SDIO_CLKCR[10] がセットされると (BYPASS = 1)、SDIOCLK 立ち上がりエッジで SDIO_CK 立ち上がりエッジが発生します。NEGEDGE 値にかかわらず、SDIOCLK 立ち下がりエッジでデータとコマンドが変更されます。

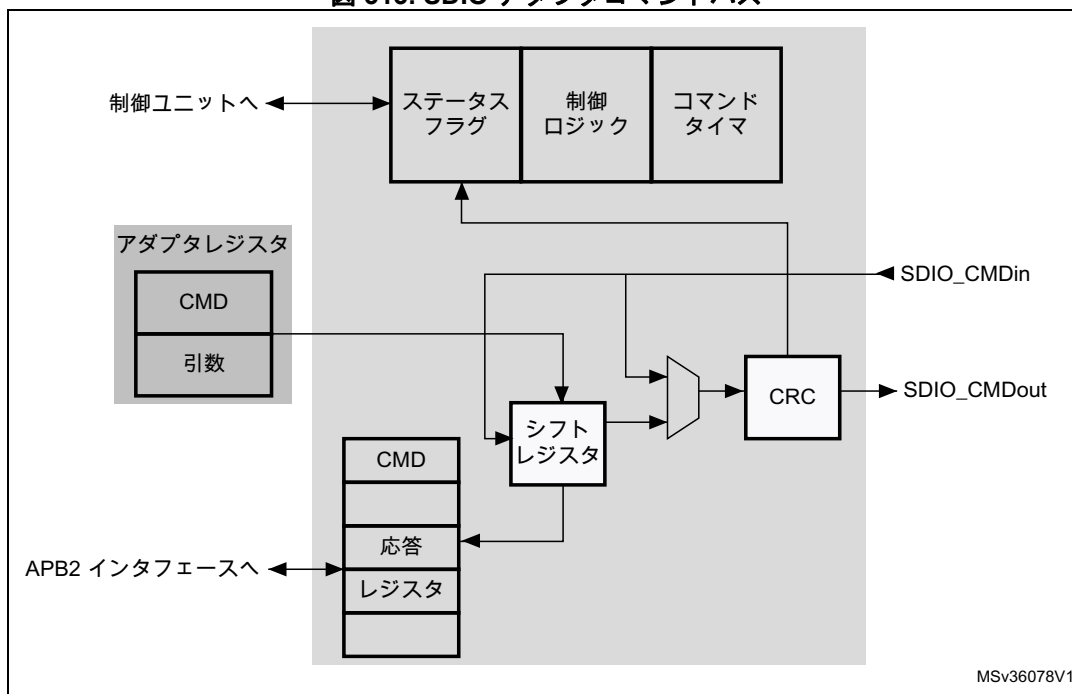
データおよびコマンドレスポンスは、SDIO_CK 立ち上がりエッジを使用してラッチされます。

図 312. SDIO_CK クロックの位相反転 (BYPASS = 0)



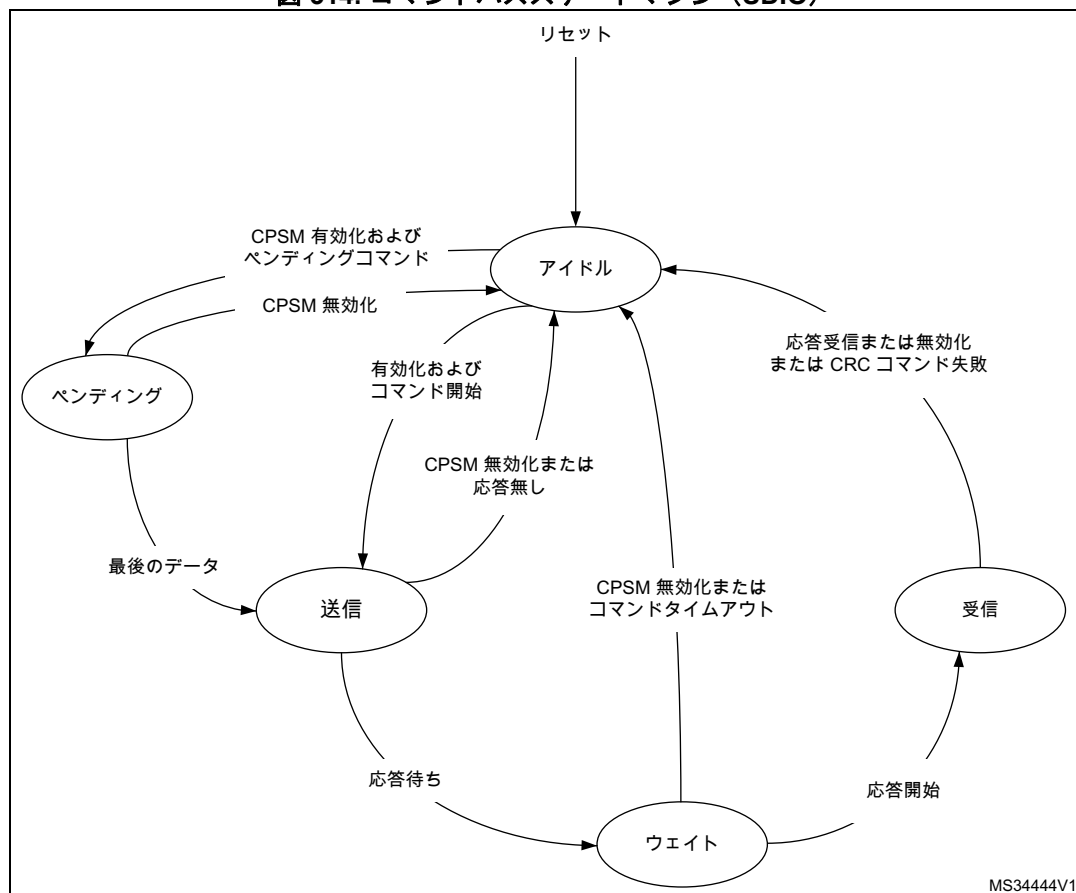
コマンドパスユニットは、カードにコマンドを送信し、カードからレスポンスを受信します。

図 313. SDIO アダプタコマンドパス



- コマンドパスステートマシン (CPSM)
 - コマンドレジスタが書き込まれて、有効ビットがセットされると、コマンド転送が開始します。コマンドが送信されると、コマンドパスステートマシン (CPSM) は、ステータスフラグをセットし、レスポンスが不要な場合はアイドル状態になります。レスポンスが必要な場合は、レスポンスを待ちます (図 314 (853 ページ) を参照)。レスポンスが受信されると、受信した CRC コードと内部生成されたコードを比較して適切なステータスフラグをセットします。

図 314. コマンドパスステートマシン (SDIO)



MS34444V1

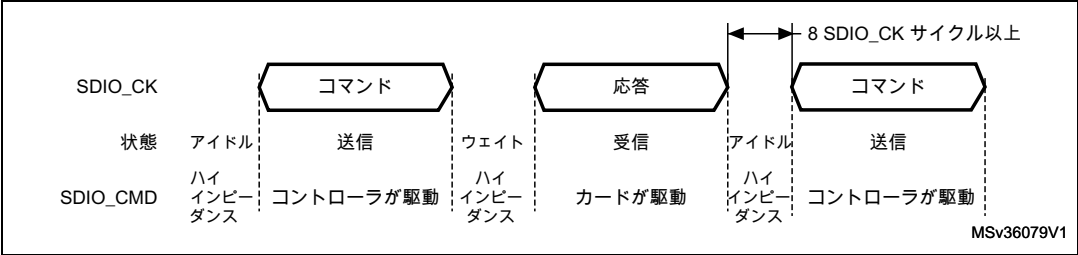
ウェイト状態に入ると、コマンドタイムが作動します。CPSM が受信状態になる前にタイムアウトに達した場合には、タイムアウトフラグがセットされ、アイドル状態に入ります。

注： コマンドタイムアウトは、64 SDIO_CLK クロック周期の固定値です。

コマンドレジスタの割込みビットがセットされると、タイマは無効になり、CPSM はカードの 1 枚からの割込みリクエストを待ちます。コマンドレジスタのペンディングビットがセットされると、CPSM はペンディング状態に入り、データパスサブユニットからの CmdPend 信号を待ちます。CmdPend が検出されると、CPSM は送信状態に移行します。これによって、停止コマンドの送信をトリガするデータカウンタが有効になります。

注： CPSM は、 N_{CC} および N_{RC} タイミング制約を満たすために、少なくとも 8 SDIO_CLK 周期の間、アイドル状態を保ちます。 N_{CC} は、2 つのホストコマンド間の最小遅延であり、 N_{RC} は、ホストコマンドとカードレスポンスの間の最小遅延です。

図 315. SDIO コマンド転送



- コマンドフォーマット
 - － コマンド：コマンドは、動作を開始するトークンです。コマンドは、ホストから 1 枚のカード（アドレス付きコマンド）または接続されたすべてのカードに送信されます（ブロードキャストコマンドは MMCV3.31 以前で使用可能）。コマンドは、CMD ライン上で連続的に転送されます。すべてのコマンドは、固定長 48 ビットです。MultiMediaCard、SD メモリカード、および SDIO カードのコマンドトークンの一般的なフォーマットを表 151 に示します。
コマンドパスは、コマンドとレスポンスを送受信できるように半二重モードで動作します。CPSM が送信状態でない場合、SDIO_CMD 出力は図 315 (854 ページ) に示すようにハイインピーダンス状態になります。SDIO_CMD のデータは、SDIO_CLK の立ち上がりエッジと同期します。表 151 に、コマンドフォーマットを示します。

表 151. コマンドフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	1	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7
0	1	1	終了ビット

- － レスポンス：レスポンスは、以前に受信したコマンドへの応答として、アドレス指定されたカードから（MMC V3.31 以前の場合はすべてのカードから同期して）ホストに送信されるトークンです。レスポンスは、CMD ライン上で連続的に転送されます。

SDIO は、2 種類のレスポンスをサポートします。どちらも CRC エラーチェックを使用します。

- 48 ビットショートレスポンス
- 136 ビットロングレスポンス

注： レスポンスが CRC を含まない場合（CMD1 レスポンス）、デバイスドライバは CRC 障害ステータスを無視する必要があります。

表 152. ショートレスポンスフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7 (または 1111111)
0	1	1	終了ビット

表 153. ロングレスポンスフォーマット

ビット位置	幅	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	111111	予約済み
[127:1]	127	-	CID または CSD (内部 CRC7 を含む)
0	1	1	終了ビット

コマンドレジスタは、コマンドインデックス（カードに送信される 6 ビット）とコマンドタイプを含みます。これらによって、コマンドがレスポンスを必要とするかどうかと、レスポンスが 48 ビット長か 136 ビット長かが決まります（[セクション 27.8.4 \(890 ページ\)](#) を参照）。コマンドパスは、[表 154](#) に示すようにステータスフラグを実装します。

表 154. コマンドパスステータスフラグ

フラグ	説明
CMDREND	レスポンス CRC が OK の場合にセットされます。
CCRCFAIL	レスポンス CRC が失敗した場合にセットされます。
CMDSENT	コマンド（レスポンスを必要としない）が送信されたときにセットされます。
CTIMEOUT	レスポンスタイムアウトです。
CMDACT	コマンド転送中です。

CRC 生成回路は、CRC コードの前のすべてのビットについて CRC チェックサムを計算します。これには、スタートビット、送信ビット、コマンドインデックス、およびコマンド引数（またはカードステータス）が含まれます。CRC チェックサムは、ロングレスポンスフォーマットの場合、CID または CSD の最初の 120 ビットについて計算されます。スタートビット、送信ビット、および 6 つの予約ビットは、CRC 計算では使用されませんので注意してください。

CRC チェックサムは 7 ビットの値です。

$$\text{CRC}[6:0] = \text{剰余} [(M(x) * x^7) / G(x)]$$

$$G(x) = x^7 + x^3 + 1$$

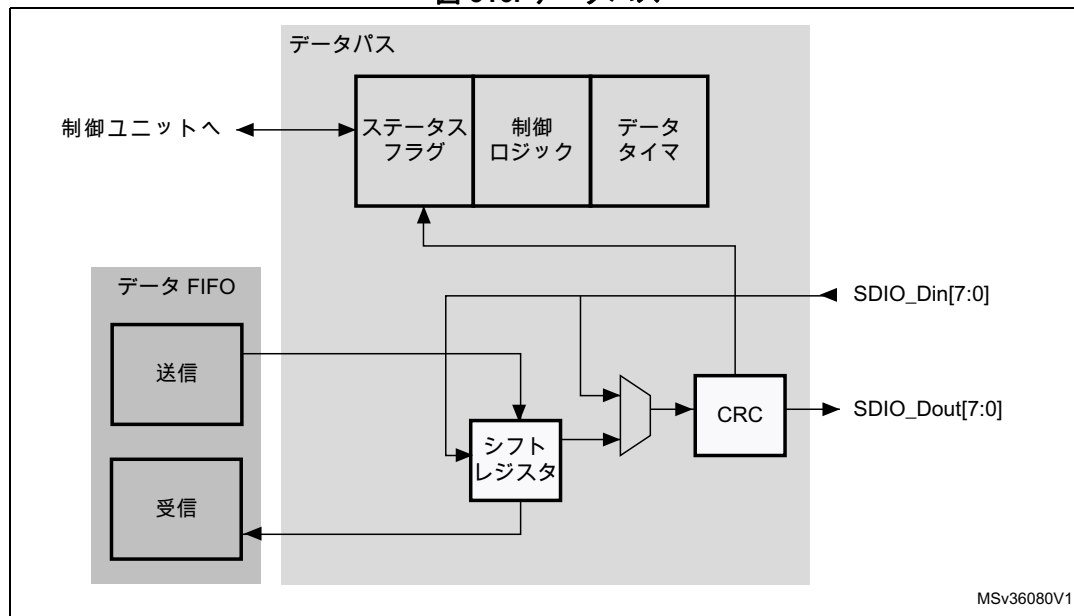
$$M(x) = (\text{スタートビット}) * x^{39} + \dots + (\text{CRC 前の最後のビット}) * x^0, \text{または}$$

$$M(x) = (\text{スタートビット}) * x^{119} + \dots + (\text{CRC 前の最後のビット}) * x^0$$

データパス

データパスサブユニットは、カードとの間でデータ転送を行います。[図 316](#) に、データパスのブロック図を示します。

図 316. データパス



カードデータバス幅は、クロック制御レジスタを使用してプログラムできます。4 ビット幅バスモードが有効な場合、データは 4 つのデータ信号すべて (SDIO_D[3:0]) で、クロックサイクルごとに 4 ビットで転送されます。8 ビット幅バスモードが有効な場合、データは 8 つのデータ信号すべて (SDIO_D[7:0]) で、クロックサイクルごとに 8 ビットで転送されます。ワイドバスモードが有効でない場合は、SDIO_D0 上でクロックサイクルごとに 1 ビットだけが転送されます。

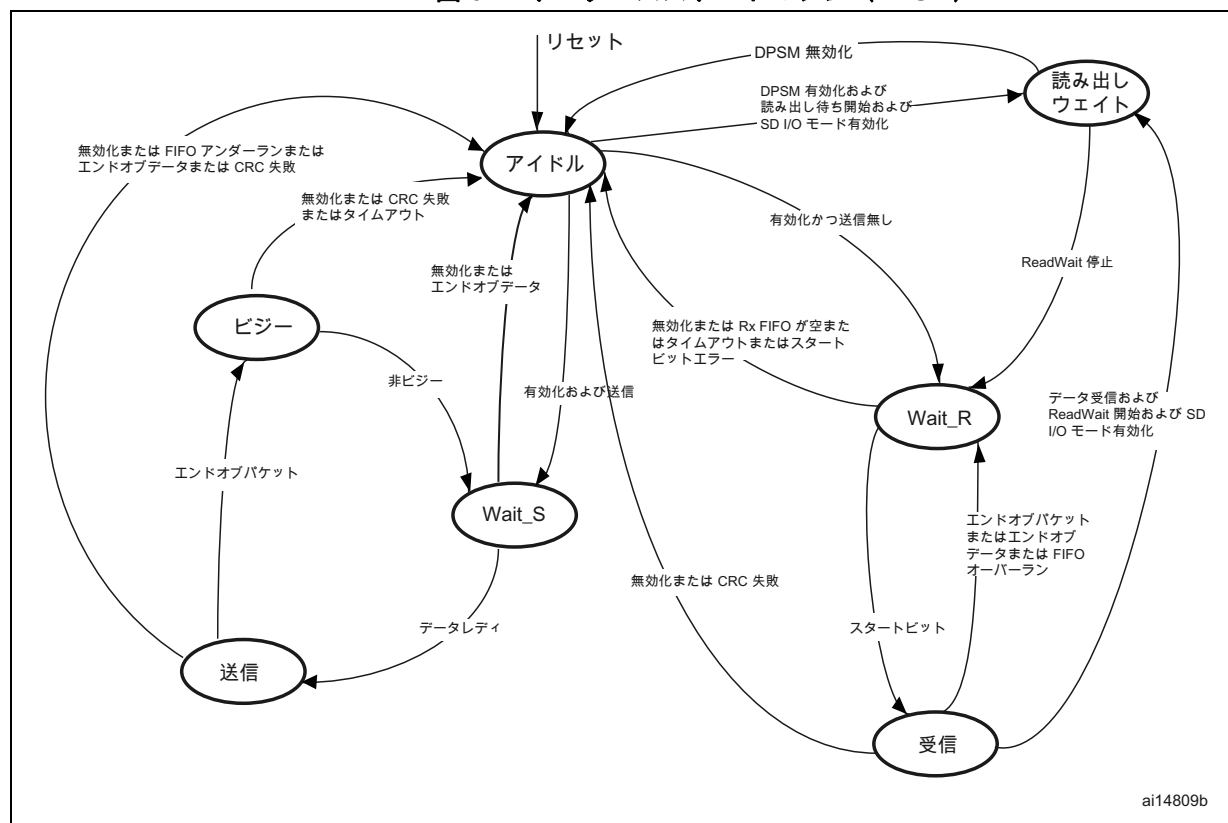
転送の方向 (送信または受信) によって、データパスステートマシン (DPSM) は、有効になったときに Wait_S または Wait_R 状態へ移行します。

- 送信 : DPSM は Wait_S 状態へ移行します。送信 FIFO にデータがある場合、DPSM は送信状態へ移行し、データパスサブユニットはカードへのデータ転送を開始します。
- 受信 : DPSM は Wait_R 状態へ移行して、スタートビットを待ちます。スタートビットを受信すると、DPSM は受信状態へ移行し、データパスサブユニットはカードからのデータ受信を開始します。

データパスステートマシン (DPSM)

DPSM は、SDIO_CK 周波数で動作します。カードバス信号上のデータは、SDIO_CK の立ち上がりエッジと同期しています。[図 317 : データパスステートマシン \(DPSM\)](#) に示すように、DPSM には、6 つの異なる状態があります。

図 317. データパスステートマシン (DPSM)



- アイドル：データパスはインアクティブであり、SDIO_D[7:0] 出力はハイインピーダンスです。データ制御レジスタが書き込まれ、有効ビットがセットされると、DPSM はデータカウンタに新しい値をロードし、データ方向ビットに応じて Wait_S または Wait_R 状態へ移行します。
- Wait_R：データカウンタがゼロに等しい場合、DPSM は受信 FIFO が空になると、アイドル状態へ移行します。データカウンタがゼロでない場合、DPSM は SDIO_D 上のスタートビットを待ちます。タイムアウト前にスタートビットを受信した場合、DPSM は受信状態に移行し、データブロックカウンタをロードします。スタートビットを検出する前にタイムアウトに達した場合、アイドル状態へ移行して、タイムアウトステータスフラグをセットします。
- 受信：カードから受信したシリアルデータは数バイトにまとめられ、データ FIFO に書き込まれます。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
 - － ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は CRC コードを受信するまで待ちます。受信したコードが内部生成された CRC コードと一致する場合、DPSM は Wait_R 状態へ移行します。そうでない場合、CRC 失敗ステータスフラグをセットして、DPSM はアイドル状態へ移行します。
 - － ストリームモードでは、DPSM はデータカウンタがゼロでない間、データを受信します。カウンタがゼロになると、シフトレジスタ内の残りのデータがデータ FIFO に書き込まれ、DPSM は Wait_R 状態へ移行します。

FIFO オーバーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。

- Wait_S：データカウンタがゼロの場合、DPSM はアイドル状態へ移行します。そうでない場合は、データ FIFO エンプティフラグがネゲートされるまで待ってから、送信状態へ移行します。

- 注： **DPSM は、 N_{WR} タイミングの要件を満たすため、少なくとも 2 クロック周期の間、Wait_S 状態を保ちます。ここで、 N_{WR} は、カードレスポンスの受信とホストからのデータ転送の開始との間のクロックサイクル数です。**
- **送信：** DPSM はカードへのデータの送信を開始します。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
 - － ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は内部生成された CRC コードと終了ビットを送信して、ビジー状態へ移行します。
 - － ストリームモードでは、DPSM は、有効ビットがハイであり、データカウンタがゼロでない間、カードへデータを送信します。その後、アイドル状態へ移行します。FIFO アンダーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。
 - **ビジー：** DPSM は、CRC ステータスフラグを待ちます。
 - － 良好な CRC ステータスを受信しなかった場合、DPSM はアイドル状態へ移行し、CRC 失敗ステータスフラグをセットします。
 - － 良好な CRC ステータスを受信した場合、SDIO_D0 がローでない（カードがビジーでない）場合は Wait_S 状態へ移行します。DPSM がビジー状態の間にタイムアウトエラーが発生した場合には、DPSM はデータタイムアウトフラグをセットして、アイドル状態へ移行します。
データタイマは、DPSM が Wait_R またはビジー状態のときに有効になり、以下の場合にデータタイムアウトを生成します。
 - － データ送信中、DPSM がプログラムされたタイムアウト時間より長く、ビジー状態にとどまっていた場合。
 - － データ受信時、データの終わりが真でなく、DPSM がプログラムされたタイムアウト時間より長く Wait_R 状態にとどまっていた場合、タイムアウトが発生します。
 - **データ：** データは、カードからホストへ、またはホストからカードへ転送できます。データは、データラインを介して転送されます。32 ワードの FIFO に格納され、各ワードは 32 ビット幅です。

表 155. データトークンフォーマット

説明	スタートビット	データ	CRC16	終了ビット
ブロックデータ	0	-	あり	1
ストリームデータ	0	-	なし	1

DPSM フラグ

データパスサブユニットの転送ステータスはいくつかのステータスフラグによって報告されます。

表 156. DPSM フラグ

フラグ	説明
DBCKEND	データブロックの送受信 CRC チェックに成功すると、ハイにセットされます。 SDIO マルチバイト転送モードでは、このフラグは転送終了時にセットされます（ホストでは、マルチバイト転送は単一のブロック転送とみなされます）。
DATAEND	SDIO_DCOUNT レジスタがデクリメントされ、0 に達するとハイにセットされます。 DATAEND は、SDIO データラインでの転送終了を示します。
DTIMEOUT	データタイムアウト期間に達すると、ハイにセットされます。 DPSM が Wait_R またはビジー状態のときにデータタイムは 0 に達すると、タイムアウトがセットされます。DPSM がプログラムされた期間より長くビジー状態にとどまっている場合、DATAEND の後で DTIMEOUT を セットできます。
DCRCFAIL	データブロックの送受信 CRC チェックに失敗すると、ハイにセットされます。

データ FIFO

データ FIFO（ファーストインファーストアウト）サブユニットは、送受信ユニットを持つデータバッファです。

FIFO には、32 ビット幅、32 ワードの深さのデータバッファと、送受信ロジックがあります。データ FIFO は APB2 クロックドメイン（PCLK2）で動作するため、SDIO クロックドメイン（SDIOCLK）のサブユニットからの信号はすべて再同期されます。

TXACT および RXACT フラグに応じて、FIFO は無効、送信有効、または受信有効にできます。TXACT と RXACT は、データパスサブユニットによって駆動され、相互排他的となっています。

- 送信 FIFO は、TXACT がアサートされると、送信ロジックとデータバッファを参照します。
- 受信 FIFO は、RXACT がアサートされると、受信ロジックとデータバッファを参照します。
- 送信 FIFO :
データは、SDIO が送信有効なときに、APB2 インタフェースを通じて送信 FIFO に書かれます。
送信 FIFO は、32 個の連続アドレスを介してアクセス可能です。送信 FIFO には、読み出しポインタが示すデータワードを保持するデータ出力レジスタが含まれています。データパスサブユニットがシフトレジスタへロードすると、読み出しポインタがインクリメントされ、新しいデータが出力されます。
送信 FIFO が無効な場合、すべてのステータスフラグがネゲートされます。データパスサブユニットは、データを送信するときに TXACT をアサートします。

表 157. 送信 FIFO ステータスフラグ

フラグ	説明
TXFIFOOF	32 個すべての送信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
TXFIFIOE	送信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
TXFIFOHE	8 個以上の送信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
TXDAVL	送信 FIFO が有効なデータを含んでいるとき、ハイにセットされます。このフラグは、TXFIFIOE フラグの反転です。
TXUNDERR	アンダーランエラーが発生すると、ハイにセットされます。このフラグは、SDIO クリアレジスタへの書き込みによってクリアされます。 注： TXUNDERR の場合で SDIO FIFO に書き込むために DMA が使用される場合、ユーザソフトウェアでは DMA ストリームを無効にし、SDIO_DCTRL の DMAEN ビットに '0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。

● 受信 FIFO

データバスサブユニットがワードデータを受信すると、書き込みデータバス上にデータが出力されます。書き込み操作が完了すると、書き込みポインタがインクリメントされます。読み出し側では、読み出しポインタの現在値が示す FIFO ワードの内容が、読み出しバス上に出力されます。受信 FIFO が無効な場合、すべてのステータスフラグがネゲートされ、読み出しおよび書き込みポインタがリセットされます。データバスサブユニットは、データを受信すると、RXACT をアサートします。表 158 に、受信 FIFO ステータスフラグをリストします。受信 FIFO は、32 個の連続アドレスを介してアクセス可能です。

表 158. 受信 FIFO ステータスフラグ

フラグ	説明
RXFIFOOF	32 個すべての受信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
RXFIFIOE	受信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
RXFIFOHF	8 個以上の受信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
RXDAVL	受信 FIFO が空でないとき、ハイにセットされます。このフラグは RXFIFIOE フラグの反転です。
RXOVERR	オーバーランエラーが発生すると、ハイにセットされます。このフラグは、SDIO クリアレジスタへの書き込みによってクリアされます。 注： RXOVERR の場合で SDIO FIFO を読み出すために DMA が使用される場合、ユーザソフトウェアでは DMA ストリームを無効にし、SDIO_DCTRL の DMAEN ビットに '0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。

27.3.2 SDIO APB2 インタフェース

APB2 インタフェースは、割り込みおよび DMA リクエストを生成し、SDIO アダプタレジスタとデータ FIFO にアクセスします。このインタフェースは、データパス、レジスタデコーダ、および割り込み／DMA ロジックで構成されています。

SDIO 割り込み

割り込みロジックは、選択されたステータスフラグの少なくとも 1 つがハイのときにアサートされる割り込みリクエスト信号を生成します。割り込み生成条件が選択できるように、マスクレジスタが提供されています。ステータスフラグは、対応するマスクフラグがセットされている場合に割り込みリクエストを生成します。

SDIO/DMA インタフェース

SDIO APB インタフェースは、ホストとカードの間の転送を実行するために、すべてのサブユニットを制御します。

DMA を使用した読み出し手順の例

次のように、CMD17 (READ_BLOCK) を送信します。

- SDIO データ長レジスタをプログラムします (SDIO データタイマレジスタは、カード識別プロセスの前にすでにプログラムされている必要があります)。
- DMA チャンネルをプログラムします ([SDIO コントローラの DMA の設定](#)を参照)。
- SDIO データ制御レジスタをプログラムします。DTEN は“1” (SDIO カードホストはデータの送信が可能)、DTDIR は“1” (カードからコントローラへ)、DTMODE は“0” (ブロックデータ転送)、DMAEN は“1” (DMA は有効)、DBLOCKSIZE は 0x9 (512 バイト) にプログラムします。他のフィールドは無視します。
- SDIO 引数レジスタに、データが転送されるカードのアドレス位置をプログラムします。
- SDIO コマンドレジスタをプログラムします。CmdIndex は 17 (READ_BLOCK) に、WaitResp は“1” (SDIO カードホストはレスポンスを待つ) に、CPSMEN は“1” (SDIO カードホストはコマンドの送信が可能) にプログラムします。他のフィールドはリセット値です。
- SDIO_STA[6] = CMDREND 割り込みを待ちます (コマンドパス上にエラーがない場合は CMDREND がセットされる)。
- SDIO_STA[10] = DBCKEND を待ちます (CRC 確認が成功するまでエラーがない場合は DBCKEND がセットされる)。
- FIFO がエンプティになるまで待ち、FIFO がエンプティになったら SDIO_STA[5] = RXOVERR の値で読み出しが成功したことを確認してください。

注： 最後の 1~4 バイトで FIFO オーバーランエラーが発生した場合は、DATAEND フラグをセットした 2 APB クロックサイクル後に RXOVERR フラグがセットされている可能性があります。読み出し操作を成功させるには、FIFO がエンプティになってから RXOVERR を必ず確認する必要があります。

DMA を使用した書き込み手順の例

次のように、CMD24 (WRITE_BLOCK) を送信します。

- SDIO データ長レジスタをプログラムします (SDIO データタイムレジスタは、カード識別プロセスの前にすでにプログラムされている必要があります)。
- DMA チャンネルをプログラムします ([SDIO コントローラの DMA の設定](#)を参照)。
- SDIO 引数レジスタに、データが転送されるカードのアドレス位置をプログラムします。
- SDIO コマンドレジスタをプログラムします。CmdIndex は 24 (WRITE_BLOCK) に、WaitResp は“1” (SDIO カードホストはレスポンスを待つ) に、CPSMEN は“1” (SDIO カードホストはコマンドの送信が可能) にプログラムします。他のフィールドはリセット値です。
- SDIO_STA[6] = CMDREND 割り込みを待ち、SDIO データ制御レジスタを次のようにプログラムします。DTEN は“1” (SDIO カードホストはデータの送信が可能)、DTDIR は“0” (コントローラからカードへ)、DTMODE は“0” (ブロックデータ転送)、DMAEN は“1” (DMA は有効)、DBLOCKSIZE は 0x9 (512 バイト) にプログラムします。他のフィールドは無視します。
- SDIO_STA[10] = DBCKEND を待ちます (エラーがない場合は DBCKEND がセットされる)。

SDIO コントローラの DMA の設定

- DMA2 コントローラを有効にして、ペンディング中の割り込みをすべてクリアします。
- DMA2_Stream3(または DMA2_Stream6) Channel4 転送元アドレスレジスタをメモリ位置のベースアドレスでプログラムし、DMA2_Stream3(または DMA2_Stream6) Channel4 転送先アドレスレジスタを SDIO_FIFO レジスタアドレスでプログラムします。
- DMA2_Stream3(または DMA2_Stream6) Channel4 制御レジスタをプログラムします (ペリフェラルインクリメントではなくメモリインクリメント、またペリフェラル幅とソース幅はワードサイズ)。
- DMA2_Stream3(または DMA2_Stream6) Channel4 をプログラムしてペリフェラルをフローコントローラとして選択します (DMA_S3CR または DMA_S6CR 設定レジスタの PFCTRL ビットをセットします)。
- DMA2_Stream3(または DMA2_Stream6) Channel4 で、インクリメンタルバースト転送を 4 ビート (最低でもペリフェラル側から) に設定します。
- DMA2_Stream3(または DMA2_Stream6) Channel4 を有効にします。

注： SDIO ホストでは、ペリフェラルフローコントローラモードでのみ DMA を使用できます。SDIO で使用する DMA ストリームは、ペリフェラルフローコントローラモードで設定する必要があります。

SDIO では、DMA コントローラに DMA バーストリクエストのみを生成します。DMA は、ペリフェラルサイドでは、インクリメントバーストモードで設定する必要があります。

27.4 カード機能詳細

27.4.1 カード識別モード

カード識別モードでは、ホストはすべてのカードをリセットし、動作電圧範囲を検証し、カードを識別し、バス上の各カードの相対カードアドレス (RCA) をセットします。カード識別モードでは、すべてのデータ通信にコマンドライン (CMD) のみを使用します。

27.4.2 カードリセット

GO_IDLE_STATE コマンド (CMD0) は、ソフトウェアリセットコマンドであり、MultiMediaCard と SD メモリをアイドル状態にします。IO_RW_DIRECT コマンド (CMD52) は、SD I/O カードをリセットします。パワーアップまたは CMD0 の後、すべてのカード出力バスドライバはハイインピーダンス状態となり、カードはデフォルトの相対カードアドレス (RCA=0x0001) とデフォルトのドライバステージレジスタ設定 (最低速度、最高駆動電流) で初期化されます。

27.4.3 動作電圧範囲の検証

すべてのカードは、仕様範囲内の任意の動作電圧で SDIO カードホストと通信できます。サポートされる最小および最大 V_{DD} 値は、カード上の動作条件レジスタ (OCR) で定義されています。

ペイロードメモリにカード識別番号 (CID) とカード固有データ (CSD) を格納するカードは、この情報をデータ転送 V_{DD} 条件下でのみ通信することができます。SDIO カードホストモジュールとカードの V_{DD} 範囲に互換性がないときには、カードは識別サイクルを完了できず、CSD データを送信できません。このため、SDIO カードホストに必要な V_{DD} 範囲に一致しないカードを識別して拒否するメカニズムとして、特殊なコマンド SEND_OP_COND (CMD1)、SD_APP_OP_COND (SD メモリ用 ACMD41)、および IO_SEND_OP_COND (SD I/O 用 CMD5) が用意されています。SDIO カードホストは、これらのコマンドのオペランドとして、必要な V_{DD} 電圧範囲を送信します。指定された範囲でデータ転送を行えないカードは、バスから切断され、インアクティブ状態になります。

電圧範囲をオペランドとして含めずにこれらのコマンドを使用すると、SDIO カードホストは各カードへの問い合わせを行い、共通の電圧範囲を判断した後に、範囲外のカードをインアクティブ状態にできます。この問い合わせは、SDIO カードホストが共通の電圧範囲を選択できるときや、カードが使用不可能であることをユーザに通知する必要があるときに使用されます。

27.4.4 カード識別プロセス

カード識別プロセスは、MultiMediaCard と SD カードで異なります。MultiMediaCard では、識別プロセスは F_{od} クロックレートで開始します。SDIO_CMD ライン出力ドライバはオープンドレインであり、このプロセス中は並行カード操作が可能です。登録プロセスは、次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは、SEND_OP_COND (CMD1) をブロードキャストして、動作条件を受信します。
3. レスポンスは、すべてのカードからの動作条件レジスタのワイヤード AND 操作です。
4. 互換性のないカードはインアクティブ状態となります。
5. SDIO カードホストは、すべてのアクティブカードに ALL_SEND_CID (CMD2) をブロードキャストします。
6. アクティブカードは、それぞれの CID 番号をシリアルに同時送信します。発信 CID ビットがコマンドラインのビットと一致しないカードは、転送を停止して、次の識別サイクルを待たなければなりません。接続されたカードがすべてのCIDを正常に SDIO カードホストに送信すると、識別状態へ移行します。
7. SDIO カードホストは、そのカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードはスタンバイ状態に遷移し、その後の識別サイクルには反応せず、出力はオープンドレインからプッシュプルへ切り替わります。
8. SDIO カードホストは、タイムアウト条件を受信するまで、ステップ 5~7 を繰り返します。

SD カードの場合、識別プロセスは F_{od} クロックレートで開始し、SDIO_CMD ライン出力ドライバはオープンドレインでなくプッシュプルドライバです。登録プロセスは、次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは SD_APP_OP_COND (ACMD41) をブロードキャストします。
3. カードは、それぞれの動作条件レジスタの内容で応答します。
4. 互換性のないカードは、インアクティブ状態へ移行されます。
5. SDIO カードホストは、すべてのアクティブカードに ALL_SEND_CID (CMD2) をブロードキャストします。
6. カードは、一意のカード識別番号 (CID) を送り返し、識別状態に入ります。
7. SDIO カードホストは、アドレスのあるアクティブなカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDIO カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最新の値が割り当てられます。
8. SDIO カードホストは、すべてのアクティブなカードについて、ステップ 5~7 を繰り返します。

SD I/O カードの場合、登録プロセスは次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは IO_SEND_OP_COND (CMD5) を送信します。
3. カードは、それぞれの動作条件レジスタの内容で応答します。
4. 互換性のないカードは、インアクティブ状態に設定されます。
5. SDIO カードホストは、アドレスのあるアクティブなカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDIO カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最新の値が割り当てられます。

27.4.5 ブロック書き込み

ブロック書き込み (CMD24-27) では、1 つ以上のブロックデータがホストからカードに転送され、ホストによって各ブロックの最後に CRC が付加されます。ブロック書き込みをサポートするカードは、WRITE_BL_LEN によって定義されたデータブロックを常に受け入れることができます。CRC が失敗した場合、カードは SDIO_D ライン上で失敗を示し、転送されたデータは破棄されて書き込まれず、それ以降に転送されたブロック (複数ブロック書き込みモード時) はすべて無視されます。

ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合 (CSD パラメータ WRITE_BLK_MISALIGN がセットされていない)、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します。(ステータスレジスタの ADDRESS_ERROR ビットがセットされます)。ホストが書き込み保護されたエリアに書き込みもうとした場合も、書き込み操作がアボートされます。ただし、この場合、カードは WP_VIOLATION ビットをセットします。

CID および CSD レジスタをプログラムする場合、以前のブロック長設定は不要です。転送されるデータも、CRC 保護されます。CSD または CID レジスタの一部が ROM に格納される場合、この変更不能部分が受信バッファの対応する部分と一致しなければなりません。これが一致しない場合、カードはエラーを報告し、レジスタの内容を変更しません。一部のカードでは、ブロックデータの書き込みに時間がかかり、その時間が予測不能ことがあります。ブロックデータの受信と CRC チェックの完了後、カードは書き込みを開始し、書き込みバッファがフルで新しい WRITE_BLOCK コマンドから新しいデータを受け入れられない場合には、SDIO_D ラインをローに保ちます。ホストは、SEND_STATUS コマンド (CMD13) でいつでもカードのステータスをポーリングでき、カードはそれぞれのステータスで応答します。READY_FOR_DATA ステータスビットは、カードが新しいデータを受け入れら

れるかどうか、または書き込み動作がまだ進行中かどうかを示します。ホストは、カードを切断状態にする CMD7 を発行してカードの選択を解除し（別のカードを選択するため）、書き込み操作を中断せずに SDIO_D ラインを解放することができます。カードを再び選択したときにプログラミングがまだ続行中であり、書き込みバッファが使用できない場合には、SDIO_D をローにプルすることによって、ビジー表示を再びアクティブにします。

27.4.6 ブロック読み出し

ブロック読み出しモードでは、データ転送の基本単位はブロックであり、その最大サイズは CSD (READ_BL_LEN) で定義されます。READ_BL_PARTIAL がセットされると、開始アドレスと終了アドレスが 1 つの物理ブロック (READ_BL_LEN で定義) に完全に含まれる小さなブロックを送信することもできます。各ブロックの最後には CRC を付加して、データの整合性を確保します。CMD17 (READ_SINGLE_BLOCK) は、ブロック読み出しを開始し、転送完了後、カードは転送状態に戻ります。

CMD18 (READ_MULTIPLE_BLOCK) は、複数の連続したブロックの転送を開始します。

ホストは、複数ブロック操作内においてはタイプに関係なく、いつでも読み出しをアボートできます。トランザクションは、送信停止コマンドを送信してアボートします。

カードが複数ブロック読み出し操作中にエラー（範囲外、アドレスのミスアラインメント、内部エラーなど）を検出した場合は、データ転送を停止して、データ状態に留まります。その場合、ホストは、送信停止コマンドを送信して操作をアボートする必要があります。読み出しエラーは、送信停止コマンドに対するレスポンスで報告されます。

カードが事前定義されたブロック数による複数ブロック操作で、最後のブロックを送信した後でホストが送信停止コマンドを送信した場合には、カードがすでにデータ状態ではないため、不正なコマンドとして応答されます。ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します（ステータスレジスタの ADDRESS_ERROR ビットがセットされます）。

27.4.7 ストリームアクセス、ストリーム書き込み、およびストリーム読み出し (MultiMediaCard のみ)

ストリームモードでは、データはバイト単位で転送され、各ブロックの最後に CRC は付加されません。

ストリーム書き込み (MultiMediaCard のみ)

WRITE_DAT_UNTIL_STOP (CMD20) は、SDIO カードホストからカードへのデータ転送を指定されたアドレスで開始し、SDIO カードホストが停止コマンドを発行するまで続けます。部分的なブロックが許可される場合 (CSD パラメータ WRITE_BL_PARTIAL がセットされている)、データストリームはカードアドレス空間内の任意のアドレスで開始および停止できます。そうでない場合は、ブロック境界でのみ開始と停止が可能です。転送データ量を事前に決めることができないため、CRC は使用できません。データ送信中にメモリ範囲の終わりに達し、SDIO カードホストから停止コマンドが送信されなかった場合、その後に転送されるデータは破棄されます。

ストリーム書き込み操作の最大クロック周波数は、カード固有データレジスタの次の式で与えられます。

$$\text{Maximumspeed} = \text{MIN}(\text{TRANSPEED}, \frac{(8 \times 2^{\text{writeblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 書き込み最大周波数
- TRANSPEED = データ転送最大速度
- writeblen = 書き込みデータブロック最大長
- NSAC = データ読み出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読み出しアクセス時間 1
- R2WFACTOR = 書き込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータを処理できずにプログラミングを停止し、ステータスレジスタの OVERRUN エラービットをセットすることがあります。この場合、それ以降のデータ転送は無視され、停止コマンドを待ちます（データ受信状態で）。ホストが書き込み保護領域に書き込もうとした場合も、書き込み操作がアボートされます。ただし、この場合、カードは WP_VIOLATION ビットをセットします。

ストリーム読み出し (MultiMediaCard のみ)

READ_DAT_UNTIL_STOP (CMD11) は、ストリーム指向のデータ転送を制御します。

このコマンドは、SDIO カードホストが STOP_TRANSMISSION (CMD12) を送信するまで、指定されたアドレスからデータを送信するようにカードに指示します。停止コマンドには、シリアルコマンド送信による実行の遅延があり、データ転送は停止コマンドの終了ビット後に停止します。データ送信中にメモリ範囲の終わりに達し、SDIO カードホストから停止コマンドが送信されなかった場合、その後に転送されるデータは未定義と考慮されます。

ストリーム読み出し操作の最大クロック周波数は次の式によって与えられ、カード固有データレジスタのフィールドを使用します。

$$\text{Maximumspeed} = \text{MIN}(\text{TRANSPEED}, \frac{(8 \times 2^{\text{readblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 読み出し最大周波数
- TRANSPEED = データ転送最大速度
- readblen = 読み出しデータブロック最大長
- writeblen = 書き込みデータブロック最大長
- NSAC = データ読み出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読み出しアクセス時間 1
- R2WFACTOR = 書き込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータ転送を維持できません。これが発生した場合、カードはステータスレジスタの UNDERRUN エラービットをセットし、送信をアボートし、データ状態で停止コマンドを待ちます。

27.4.8 消去：グループ消去とセクタ消去

MultiMediaCard の消去可能単位は、消去グループです。消去グループは、カードの基本的な書き込み可能単位である書き込みブロックで分けます。消去グループのサイズはカード固有のパラメータであり、CSD で定義されます。

ホストは連続した範囲の消去グループを消去することができます。消去プロセスは、3 ステップのシーケンスで開始します。

まずホストは、ERASE_GROUP_START (CMD35) コマンドを使用して範囲の開始アドレスを定義し、次に、ERASE_GROUP_END (CMD36) コマンドを使用して範囲の最後のアドレスを定義し、最後に、ERASE (CMD38) コマンドを発行して消去プロセスを開始します。消去コマンドのアドレスフィールドは、バイト単位の消去グループアドレスです。カードは、消去グループサイズ未満のすべての LSB を無視して、アドレスを効果的に消去グループの境界まで切り捨てます。

消去コマンドが誤った順序で受信されると、カードはステータスレジスタの ERASE_SEQ_ERROR ビットをセットし、シーケンス全体をリセットします。

シーケンス外 (SEND_STATUS を除き、どちらの消去コマンドでもない) のコマンドを受信した場合、カードはステータスレジスタの ERASE_RESET ステータスビットをセットして消去シーケンスをリセットし、最後のコマンドを実行します。

消去範囲に書き込み保護ブロックが含まれていると、それらのブロックは消去されず、保護されていないブロックだけが消去されます。ステータスレジスタの WP_ERASE_SKIP ステータスビットがセットされます。

カードは、SDIO_D をローに保持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることがあり、ホストは CMD7 を発行して、カードを選択解除できます。

27.4.9 ワイドバス選択または選択解除

ワイドバス (4 ビットバス幅) 操作モードは、SET_BUS_WIDTH (ACMD6) を使用して選択または選択解除されます。パワーアップ後または GO_IDLE_STATE (CMD0) 後のデフォルトのバス幅は、1 ビットです。SET_BUS_WIDTH (ACMD6) は、転送状態でのみ有効です。すなわち、SELECT/DESELECT_CARD (CMD7) によって選択された後でなければバス幅を変更することはできません。

27.4.10 保護管理

SDIO カードホストモジュールでは、カードの書き込み保護方法が 3 種類サポートされています。

1. 内部カード書き込み保護 (カード側の責任)
2. 機械的書き込み保護スイッチ (SDIO カードホストモジュール側のみの責任)
3. パスワードで保護されたカードロック操作

内部カード書き込み保護

カードデータは、書き込みや消去から保護できます。CSD の永久的または一時的書き込み保護ビットをセットすると、製造業者やコンテンツプロバイダによってカード全体を永久的に保護することができます。セクタのグループの書き込み保護をサポートするカードの場合、CSD の WP_GRP_ENABLE ビットをセットするとデータの一部を保護することができ、書き込み保護はアプリケーションによって変更できます。書き込み保護は、CSD で指定された WP_GRP_SIZE セクタ単位です。SET_WRITE_PROT および CLR_WRITE_PROT コマンドは、アドレス指定されたグループの保護を制御します。SEND_WRITE_PROT コマンドは、1 ブロックの読み出しコマンドと似ています。カードは、32 個の書き込み保護ビット (指定されたアドレスから始まる 32 個の書き込み保護グループを表す) に続いて、16 個の CRC ビットを含むデータブロックを送信します。書き込み保護コマンドのアドレスフィールドは、バイト単位のグループアドレスです。



カードは、グループサイズ未満のすべての LSB を無視します。

機械的書き込み保護スイッチ

カードの横にある機械的なスライドタブによって、カードの書き込み保護をセット／クリアできます。スライドタブが開位置にあるときにはカードは書き込み保護され、閉位置にあるときには、カードの内容を変更できます。ソケット側の対応するスイッチは、カードが書き込み保護されていることを SDIO カードホストモジュールに示します。カードを保護するのは、SDIO カードホストモジュールです。書き込み保護スイッチの位置は、カードの内部回路にはわかりません。

パスワード保護

パスワード保護機能によって、SDIO カードホストモジュールはカードをパスワードでロック／アンロックできます。パスワードは 128 ビットの PWD レジスタに格納され、そのサイズは 8 ビットの PWD_LEN レジスタで設定されます。これらのレジスタは不揮発性のため、パワーサイクルでは消去されません。ロックされたカードは、特定のコマンドに応答し、それを実行します。すなわち、SDIO カードホストモジュールは、リセット、初期化、選択、およびステータスの問い合わせを行うことはできますが、カード上のデータにアクセスすることはできません。パスワードが設定されると (PWD_LEN の値がゼロ以外)、カードはパワーアップ後に自動的にロックされます。CSD および CID レジスタの書き込みコマンド同様、ロック／アンロックコマンドは転送状態でのみ使用できます。この状態では、コマンドはアドレス引数を含まず、カードは使用前に選択しなければなりません。カードロック／アンロックコマンドは、通常の 1 ブロック書き込みコマンドの構造とバストランザクシオンタイプとなっています。転送されるデータブロックは、コマンドに必要なすべての情報 (パスワード設定モード、PWD 自体、およびカードのロック／アンロック) を含みます。コマンドデータブロックサイズは、カードロック／アンロックコマンド送信前に、SDIO カードホストモジュールによって定義され、表 172 に示す構造となっています。

ビット設定は、次のとおりです。

- ERASE : セットすると、強制的に消去操作が行われます。他のすべてのビットはゼロでなければならない、コマンドバイトだけが送信されます。
- LOCK_UNLOCK : セットすると、カードをロックします。LOCK_UNLOCK は SET_PWD と同時にセットできますが、CLR_PWD と同時にセットすることはできません。
- CLR_PWD : セットすると、パスワードデータをクリアします。
- SET_PWD : セットすると、パスワードデータをメモリに保存します。
- PWD_LEN : パスワードの長さ (バイト数) を定義します。
- PWD : パスワード (コマンドに応じて、新しいパスワードまたは現在使用中のパスワード)。

以下のセクションでは、パスワードのセット／リセット、カードのロック／アンロック、および強制消去のコマンドシーケンスを説明します。

パスワードの設定

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード、8 ビットの PWD_LEN、および新しいパスワードのバイト数によって定義されます。パスワードの書き換えが完了すると、ブロックサイズでは古いパスワードと新しいパスワードの両方がコマンドとともに送信されることを考慮する必要があります。
3. 16 ビットの CRC を含めた、データライン上の適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (SET_PWD=1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。パスワードの書き換えが完了すると、長さ値 (PWD_LEN) には古いパスワードと新しいパスワードの両方の長さが含まれ、PWD フィールドには、古いパスワード (現在使用中) に続いて新しいパスワードが含まれます。

4. パスワードが一致すると、新しいパスワードとそのサイズがそれぞれ PWD と PWD_LEN フィールドに保存されます。送信された古いパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、パスワードは変更されません。

パスワード長フィールド (PWD_LEN) は、パスワードが現在設定されているかどうかを示します。このフィールドがゼロ以外のときには、パスワードが設定されていて、カードはパワーアップ後に自身をロックします。LOCK_UNLOCK ビットをセットするか (パスワードの設定時に)、カードをロックする追加のコマンドを送信すると、現在のパワーセッションでただちにカードをロックすることが可能です。

パスワードのリセット

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード、8 ビットの PWD_LEN、および現在使用中のパスワードのバイト数によって定義されます。
3. 16 ビットの CRC を含めた、データライン上の適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (CLR_PWD=1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。LOCK_UNLOCK ビットは無視されます。
4. パスワードが一致すると、PWD フィールドはクリアされ、PWD_LEN は 0 にセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、パスワードは変更されません。

カードのロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード (表 172 のバイト 0)、8 ビットの PWD_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上の適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK_UNLOCK = 1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはロックされ、カードステータスレジスタの CARD_IS_LOCKED ステータスビットがセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、ロックは失敗します。

パスワードの設定とカードのロックは、同じシーケンスで行うことが可能です。この場合、SDIO カードホストモジュールは、パスワードの設定に必要なすべてのステップを実行します ([パスワードの設定 \(868 ページ\)](#) を参照)。ただし、新しいパスワードコマンドを送信するときには、ステップ 3 で LOCK_UNLOCK ビットをセットする必要があります。

パスワードが以前に設定されているときには (PWD_LEN が 0 でない)、カードは電源投入リセット後に自動的にロックされます。ロックされたカードをロックしようとしたり、パスワードが設定されていないカードをロックしようすると、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

カードのアンロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード (表 172 のバイト 0)、8 ビットの PWD_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上の適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK_UNLOCK = 0)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはアンロックされ、カードステータスレジスタの CARD_IS_LOCKED ステータスビットがクリアされます。送信されたパスワードのサイズまたは内容が予期されたパスワードと一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、カードはロックされたままとなります。

アンロック機能は、現在のパワーセッションでのみ有効です。PWD フィールドがクリアされていないときには、カードは次のパワーアップ時に自動的にロックされます。

アンロックされたカードをアンロックしようとする、アンロックは失敗し、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

強制消去

ユーザがパスワード (PWD の内容) を忘れた場合は、カード上のすべてのデータをクリアした後で、カードにアクセスすることができます。この強制消去操作では、すべてのカードデータとすべてのパスワードデータが消去されます。

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. ブロック長 (SET_BLOCKLEN、CMD16) を 1 バイトに設定します。8 ビットのカードロック／アンロックバイト (表 172 のバイト 0) だけが送信されます。
3. 16 ビットの CRC を含めた、データライン上の適切なデータバイトとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (ERASE = 1) を示します。他のすべてのビットはゼロでなければなりません。
4. ERASE ビットがデータフィールドでセットされた唯一のビットであるときには、PWD および PWD_LEN フィールドを含め、すべてのカード内容が消去され、カードはロックされなくなります。他にもセットされているビットがあるときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、カードはすべてのデータを保持し、ロックされたままとなります。

アンロックされたカードを強制消去しようすると、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

27.4.11 カードステータスレジスタ

レスポンスフォーマット R1 には、カードステータスと呼ばれる 32 ビットのフィールドがあります。このフィールドは、カードステータス情報（ローカルステータスレジスタに格納可能）をホストに送信するためのものです。特に記載がない限り、ステータスエントリは、常に、以前に発行されたコマンドに関連するものとしします。

表 159 に、さまざまなステータスエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ :

- E : エラービット。
- S : ステータスビット。
- R : 実際のコマンドレスポンスに対して検出され、セットされます。
- X : コマンド実行中に検出され、セットされます。SDIO カードホストは、ステータスコマンドを発行してカードにポーリングし、これらのビットを読み出します。

クリア条件 :

- A : カードの現在の状態によります。
- B : 常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます（1 コマンドの遅延）。
- C : 読み出すとクリアされます。

表 159. カードステータス

ビット	識別子	タイプ	値	説明	クリア条件
31	ADDRESS_OUT_OF_RANGE	E R X	'0' = エラーなし 1 = エラー	コマンドアドレス引数が、このカードの許容範囲を超えています。 複数ブロックまたはストリーム読み出し／書き込み操作は、有効なアドレスから開始していますが、カード容量を超える読み込みまたは書き込みをしようとしています。	C
30	ADDRESS_MISALIGN		'0' = エラーなし 1 = エラー	現在設定されているブロック長にしたがってコマンドアドレス引数が指している最初のデータブロックは、カードの物理ブロックに合っていません。 複数ブロックの読み出し／書き込み操作は、（有効なアドレス／ブロック長の組み合わせから開始している場合でも）カードの物理ブロックに合っていないデータブロックの読み込みまたは書き込みをしようとしています。	C
29	BLOCK_LEN_ERROR		'0' = エラーなし 1 = エラー	SET_BLOCKLEN コマンドの引数がカードの許容する最大値を超えているか、以前に定義されたブロック長が現在のコマンドに対して無効です（ホストが書き込みコマンドを発行し、現在のブロック長がカードの許容する最大値より小さく、部分的なブロックの書き込みが許可されない場合など）。	C
28	ERASE_SEQ_ERROR		'0' = エラーなし 1 = エラー	消去コマンドのシーケンス中にエラーが発生しました。	C
27	ERASE_PARAM	E X	'0' = エラーなし 1 = エラー	消去に対して無効な消去グループが選択されました。	C



表 159. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
26	WP_VIOLATION	E X	'0'= エラーなし 1= エラー	書き込み保護されたブロックをプログラムしようとした。	C
25	CARD_IS_LOCKED	S R	0 = カードがアンロックされています。 1 = カードがロックされています。	セットされると、カードがホストによってロックされていることを示します。	A
24	LOCK_UNLOCK_FAILED	E X	'0'= エラーなし 1= エラー	ロック／アンロックコマンドで、シーケンスエラーまたはパスワードエラーが発生したときにセットされます。	C
23	COM_CRC_ERROR	E R	'0'= エラーなし 1= エラー	以前のコマンドの CRC チェックが失敗しました。	B
22	ILLEGAL_COMMAND	E R	'0'= エラーなし 1= エラー	カードの状態に対して不正なコマンドです。	B
21	CARD_ECC_FAILED	E X	'0' = 成功 '1' = 失敗	カードの内部 ECC が適用されましたが、データの訂正に失敗しました。	C
20	CC_ERROR	E R	'0'= エラーなし 1= エラー	(標準では定義されていません) ホストコマンドに関係ないカードエラーが発生しました。	C
19	ERROR	E X	'0'= エラーなし 1= エラー	(標準では定義されていません) 最後のホストコマンドの実行に関係した一般カードエラーが検出されました (読み出しまたは書き込みの失敗など)。	C
18	予約済み				
17	予約済み				
16	CID/CSD_OVERWRITE	E X	'0'= エラーなし '1'= エラー	次のいずれかのエラーです。 – CID レジスタはすでに書き込まれていて、上書きできません。 – CSD の読み出し専用セクションが、カードの内容と一致しません。 – 不正なコピー (オリジナルとしてセット) または永久的な WP ビット (保護されない) を元に戻そうとしました。	C
15	WP_ERASE_SKIP	E X	0= 保護されていません 1= 保護されています	既存の書き込みが理由で、部分的なアドレス空間だけが消去されたときにセットされます。	C
14	CARD_ECC_DISABLED	S X	0= 有効 1= 無効	内部 ECC を使用せずに、コマンドが実行されました。	A
13	ERASE_RESET		0= クリア 1= セット	シーケンス外の消去コマンド (CMD35、CMD36、CMD38、または CMD13 以外) が受信されたため、実行前に消去シーケンスがクリアされました。	C

表 159. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
12:9	CURRENT_STATE	S R	0 = アイドル 1 = レディ状態 2 = 識別 3 = スタンバイ 4 = 転送 5 = データ 6 = 受信 7 = プログラム 8 = Dis 9 = Btst 10-15 = 予約済み	コマンドを受信したときのカードの状態。コマンドの実行によってカードの状態が変化する 場合、次のコマンドのレスポンスで、ホストに 状態が知らされます。4 ビットは、0 から 15 ま での2 進数として解釈されます。	B
8	READY_FOR_DATA	S R	0 = 非レディ状態 1 = レディ	バス上のバッファ空信号に対応します。	-
7	SWITCH_ERROR	E X	'0' = エラーなし 1 = スイッチエ ラー	セットされた場合に、カードは SWICHTH コマ ンドで要求された予期されたモードに切り替 わりませんでした。	B
6	予約済み				
5	APP_CMD	S R	0 = 無効 1 = 有効	カードは ACMD を期待しているか、コマンド が ACMD として解釈されたという指示を期待 します。	C
4	SD I/O カード用に予約済み				
3	AKE_SEQ_ERROR	E R	'0' = エラーなし 1 = エラー	認証プロセスのシーケンスにエラーがありま す。	C
2	アプリケーション固有のコマンド用に予約済み。				
1	製造業者のテストモード用に予約済み。				
0					

27.4.12 SD ステータスレジスタ

SD ステータスは、SD メモリカード独自の機能に関連するステータスビットを含み、将来のアプリケーション固有の用途で使用できます。SD ステータスのサイズは、512 ビットのデータブロックです。このレジスタの内容は、ACMD13 が送信された場合に（CMD55 に続いて CMD13）、SDIO カードホストに送信されます。ACMD13 は、転送状態のカード（カードが選択されている）にのみ送信できます。

表 160 に、SD ステータスレジスタのさまざまなエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ：

- E：エラービット。
- S：ステータスビット。
- R：実際のコマンドレスポンスに対して検出され、セットされます。
- X：コマンド実行中に検出され、セットされます。SDIO カードホストは、ステータスコマンドを発行してカードをポーリングし、これらのビットを読み出します。

クリア条件：

- A：カードの現在の状態によります。
- B：常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます（1 コマンドの遅延）。
- C：読み出すとクリアされます。

表 160. SD ステータス

ビット	識別子	タイプ	値	説明	クリア条件
511 : 510	DAT_BUS_WIDTH	S R	00= 1（デフォルト） 01= 予約済み 10= 4 ビット幅 11= 予約済み	SET_BUS_WIDTH コマンドによって定義された、現在定義されているバス幅を示します。	A
509	SECURED_MODE	S R	0= セキュアモードではありません。 1= セキュアモードです。	カードはセキュア操作モードになっています（「SD セキュリティ仕様」を参照）。	A
508 : 496	予約済み				
495 : 480	SD_CARD_TYPE	S R	00xxh= 物理仕様 Ver1.01-2.00 で定義された SD メモリカード（“x”= 無視） 現在定義されているカードは、次のとおりです。 0000= 通常の SD RD/WR カード 0001= SD ROM カード	将来は、8 つの LSB を使用して、さまざまな種類の SD メモリカードを定義する予定です（各ビットは異なる SD タイプを定義します）。8 つの MSB は、現在の SD 物理レイヤ仕様に準拠しない SD カードを定義するために使用します。	A
479 : 448	SIZE_OF_PROTECTED_AREA	S R	保護領域のサイズ（下記を参照）	（下記を参照）	A
447 : 440	SPEED_CLASS	S R	カードの速度クラス（下記を参照）	（下記を参照）	A
439 : 432	PERFORMANCE_MOVE	S R	1 [MB/s] ステップが表す移動性能（下記を参照）	（下記を参照）	A
431:428	AU_SIZE	S R	AU のサイズ（下記を参照）	（下記を参照）	A

表 160. SD ステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
427:424	予約済み				
423:408	ERASE_SIZE	S R	一度に消去される AU の数	(下記を参照)	A
407:402	ERASE_TIMEOUT	S R	UNIT_OF_ERASE_AU で指定された領域を消去する際のタイムアウト値	(下記を参照)	A
401:400	ERASE_OFFSET	S R	消去時間に追加される固定オフセット値	(下記を参照)	A
399:312	予約済み				
311:0	製造業者のために予約済み。				

SIZE_OF_PROTECTED_AREA

このフィールドの設定は、標準カードと大容量カードで異なります。標準容量のカードでは、保護領域の容量は次のように計算されます。

保護領域= SIZE_OF_PROTECTED_AREA * MULT * BLOCK_LEN
SIZE_OF_PROTECTED_AREA は、MULT*BLOCK_LEN 単位で指定されます。

大容量のカードでは、保護領域の容量は、このフィールドで指定されます。

保護領域= SIZE_OF_PROTECTED_AREA
SIZE_OF_PROTECTED_AREA は、バイト単位で指定されます。

SPEED_CLASS

この 8 ビットフィールドは速度クラスを示し、値は $P_W/2$ で求めることができます (P_W は書き込み性能)。

表 161. 速度クラスコードフィールド

SPEED_CLASS	値の定義
00h	クラス 0
01h	クラス 2
02h	クラス 4
03h	クラス 6
04h – FFh	予約済み

PERFORMANCE_MOVE

この 8 ビットフィールドは、Pm（移動パフォーマンス）を示し、値は、1 [MB/sec] ステップで設定できます。カードが使用済みの RU（記録ユニット）を移動しない場合は、Pm を無限大であるとみなします。このフィールドを FFh にセットすると、無限を意味します。

表 162. 移動パフォーマンスフィールド

PERFORMANCE_MOVE	値の定義
00h	定義されません。
01h	1 [MB/sec]
02h	02h 2 [MB/sec]
-----	-----
FEh	254 [MB/sec]
FFh	無限

AU_SIZE

この 4 ビットフィールドは、AU サイズを示し、値は、16 KB から 2 の累乗で選択できます。

表 163. AU_SIZE フィールド

AU_SIZE	値の定義
00h	定義されません。
01h	16 KB
02h	32 KB
03h	64 KB
04h	128 KB
05h	256 KB
06h	512 KB
07h	1 MB
08h	2 MB
09h	4 MB
Ah – Fh	予約済み

最大 AU サイズはカード容量に依存し、表 164 に示されています。カードは、RU サイズと最大 AU サイズの間の任意の AU サイズに設定できます。

表 164. 最大 AU サイズ

容量	16 MB-64 MB	128 MB-256 MB	512 MB	1 GB-32 GB
最大 AU サイズ	512 KB	1 MB	2 MB	4 MB

ERASE_SIZE

この 16 ビットフィールドは、NERASE を示します。AU の NERASE 数が消去される時、タイムアウト値は ERASE_TIMEOUT によって指定されます ([ERASE_TIMEOUT](#) を参照)。ホストは、1 回の操作で消去される正しい AU の数を決定する必要があります。それにより、ホストは消去操作の進捗状況を表示することができます。このフィールドが 0 にセットされた場合、消去タイムアウト計算はサポートされません。

表 165. 消去サイズフィールド

ERASE_SIZE	値の定義
0000h	消去タイムアウトの計算はサポートされません。
0001h	1 AU
0002h	2 AU
0003h	3 AU
-----	-----
FFFFh	65535 AU

ERASE_TIMEOUT

この 6 ビットフィールドは、TERASE を示し、値は、ERASE_SIZE によって指定された複数の AU が消去される時のオフセットからの消去タイムアウトを示します。ERASE_TIMEOUT の範囲は、63 秒までの間で定義でき、カード製造業者は実装に応じて、ERASE_SIZE と ERASE_TIMEOUT の任意の組み合わせを選ぶことができます。ERASE_TIMEOUT を決めると、ERASE_SIZE が決まります。

表 166. 消去タイムアウトフィールド

ERASE_TIMEOUT	値の定義
00	消去タイムアウトの計算はサポートされません。
01	1 [sec]
02	2 [sec]
03	3 [sec]
-----	-----
63	63 [sec]

ERASE_OFFSET

この 2 ビットフィールドは、TOFFSET を示し、4 つの値から 1 つを選択できます。このフィールドは、ERASE_SIZE および ERASE_TIMEOUT フィールドが 0 にセットされている場合は意味を持ちません。

表 167. 消去オフセットフィールド

ERASE_OFFSET	値の定義
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

27.4.13 SD I/O モード

SD I/O 割込み

SD インタフェースのピンには、SD I/O カードから MultiMediaCard/SD モジュールへの割込みを可能にする割込み機能が用意されています。ピン 8 は、4 ビット SD モードで動作するときに SDIO_D1 として使用され、MultiMediaCard/SD モジュールへのカード割込みを示します。割込みの使用は、各カードまたはカード内の各機能でオプションとなっています。SD I/O 割込みはレベル対応であるため、割込みラインは MultiMediaCard/SD モジュールによって認識され、割込みが処理されるか、割込み期間の終了によってネゲートされるまで、アクティブ（ロー）に保たれる必要があります。MultiMediaCard/SD モジュールが割込み処理を終えると、SD I/O カードの内部レジスタの該当するビットへの I/O 書き込みによって割込みステータスビットがクリアされます。すべての SD I/O カードの割込み出力はアクティブローであり、アプリケーションはすべてのデータライン (SDIO_D[3:0]) 上に、外部プルアップ抵抗を提供する必要があります。MultiMediaCard/SD モジュールは、割込み期間に限って、割込み検出器にピン 8 (SDIO_D/IRQ) のレベルをサンプリングします。その他の状況では、MultiMediaCard/SD モジュールはこの値を無視します。

割込み期間は、メモリと I/O の両方の操作に適用されます。単一ブロックの操作の割込み期間の定義は、複数ブロックのデータ転送の定義とは異なります。

SD I/O のサスペンドとレジューム

マルチファンクション SD I/O または I/O とメモリの両方の機能を持つカードでは、MMC/SD バスへのアクセスを共有する複数のデバイス (I/O とメモリ) があります。複数デバイス間で MMC/SD モジュールへのアクセスを共有するため、SD I/O およびコンボカードはオプションでサスペンド/レジュームの概念を実装します。カードがサスペンド/レジュームをサポートするときには、MMC/SD モジュールは 1 つの機能またはメモリへのデータ転送を一時的に中止 (サスペンド) して、別の機能またはメモリへの、より優先順位の高い転送用にバスを解放できます。この優先順位の高い転送が完了すると、本来の転送がアボートした所から再開 (リスタート) されます。サスペンド/レジュームのサポートは、カードごとのオプションです。MMC/SD バス上でサスペンド/レジューム操作を行うには、MMC/SD モジュールは以下の手順を実行します。

1. SDIO_D [3:0] ラインを現在使用している機能を特定します。
2. 優先順位が低いトランザクションや時間のかかるトランザクションをサスペンドするように要求します。
3. トランザクションのサスペンドの完了を待ちます。
4. 優先順位の高いトランザクションを開始します。
5. 優先順位の高いトランザクションの完了を待ちます。
6. サスペンドされたトランザクションをリストアします。

SD I/O ReadWait

オプションの ReadWait (RW) 操作は、SD の 1 ビットおよび 4 ビットモードについてのみ定義されます。ReadWait 操作によって、MMC/SD モジュールは複数のレジスタを読み出していることをカードに知らせ (IO_RW_EXTENDED、CMD53)、データ転送を一時的にストールし、その間に MMC/SD モジュールは SD I/O デバイス内の機能にコマンドを送信できます。カードが ReadWait プロトコルをサポートしているかどうかを判断するには、MMC/SD モジュールが内部カードレジスタの機能ビットをテストする必要があります。ReadWait のタイミングは、割込み期間に基づいています。

27.4.14 コマンドおよびレスポンス

アプリケーション固有コマンドと汎用コマンド

SDIO カードホストモジュールシステムは、さまざまな種類のアプリケーションに標準インタフェースを提供するように設計されています。この環境では、特定の顧客やアプリケーション機能へのニーズがあります。これらの機能を実装するために、この標準ではアプリケーション固有コマンド (ACMD) と汎用コマンド (GEN_CMD) の 2 種類のコマンドが定義されています。

APP_CMD (CMD55) コマンドを受信すると、カードは次のコマンドがアプリケーション固有コマンドであると予想します。ACMD は、通常の MultiMediaCard コマンドと同じ構造を持っており、同じ CMD 番号を持つことができます。カードは、APP_CMD (CMD55) の後に現れるコマンドを ACMD として認識します。APP_CMD (CMD55) のすぐ後に続くコマンドが定義済みのアプリケーション固有コマンドでないときには、標準コマンドが使用されます。たとえば、カードで SD_STATUS (ACMD13) が定義されているときに APP_CMD (CMD55) の直後に CMD13 を受信すると、これは SD_STATUS (ACMD13) であると解釈されます。ただし、カードが APP_CMD (CMD55) の直後に CMD7 を受信し、カードで ACMD7 が定義されていないときには、これは標準 (SELECT/DESELECT_CARD) CMD7 であると解釈されます。

製造業者固有の ACMD の 1 つを使用するには、SD カードホストが以下の手順を実行する必要があります。

1. APP_CMD (CMD55) を送信します。
カードは MultiMediaCard/SD モジュールに応答して、APP_CMD ビットがセットされていることと、ACMD が予期されていることを示します。
2. 必要な ACMD を送信します。
カードは MultiMediaCard/SD モジュールに応答して、APP_CMD ビットがセットされていることと、受け取ったコマンドが ACMD として解釈されたことを示します。ACMD でないコマンドが送信されたときには、カードはそのコマンドを通常の MultiMediaCard コマンドとして扱い、カードステータスレジスタの APP_CMD ビットはクリアされたままとなります。

無効なコマンド (ACMD でも CMD でもない) が送信されたときには、標準の MultiMediaCard 不正コマンドエラーとして処理されます。

GEN_CMD のバストランザクションは、単一ブロックの読み出しまたは書き込みコマンド (WRITE_BLOCK、CMD24、または READ_SINGLE_BLOCK、CMD17) と同じです。この場合、引数はアドレスではなくデータ転送の方向を示し、データブロックはベンダ固有のフォーマットと意味を持ちます。

カードは、GEN_CMD (CMD56) を送信する前に選択されなければなりません (転送状態でなければなりません)。データブロックサイズは、SET_BLOCKLEN (CMD16) によって定義されます。GEN_CMD (CMD56) へのレスポンスは、R1b フォーマットです。

コマンドタイプ

アプリケーション固有コマンドと汎用コマンドは、どちらも、次の 4 つのタイプに分類されます。

- **ブロードキャストコマンド (BC)** : すべてのカードに送信されます。レスポンスは返されません。
- **レスポンス付きブロードキャストコマンド (BCR)** : すべてのカードに送信されます。レスポンスは、すべてのカードから同時に受信します。
- **アドレス指定 (ポイントツーポイント) コマンド (AC)** : 選択されたカードに送信されます。SDIO_D ライン上のデータ転送は含みません。
- **アドレス指定 (ポイントツーポイント) データ転送コマンド (ADTC)** : 選択されたカードに送信されます。SDIO_D ライン上のデータ転送は含みます。

コマンドフォーマット

コマンドフォーマットについては、表 151 (854 ページ) を参照してください。

MultiMediaCard/SD モジュール用コマンド

表 168. ブロック指向書き込みコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD23	ac	[31:16] 0 にセット [15:0] ブロック数	R1	SET_BLOCK_COUNT	複数ブロック読み出しまたは書き込み コマンドで転送されるブロック数を定 義します。
CMD24	adtc	[31:0] データアド レス	R1	WRITE_BLOCK	SET_BLOCKLEN コマンドによって選 択されたサイズのブロックを書き込み ます。
CMD25	adtc	[31:0] データアド レス	R1	WRITE_MULTIPLE_BLOCK	STOP_TRANSMISSION まで、または、 要求されたブロック数を受信するま で、データのブロックを書き込み続け ます。
CMD26	adtc	[31:0] スタッ ビット	R1	PROGRAM_CID	カード識別レジスタのプログラミング です。このコマンドは、1 枚のカードに つき 1 度だけ発行する必要があります。 カードには、最初のプログラミン グの後にこの操作を防ぐためのハード ウェアが含まれています。通常、この コマンドは製造業者用に予約されてい ます。
CMD27	adtc	[31:0] スタッ ビット	R1	PROGRAM_CSD	CSD のプログラマブルビットのプログ ラミング用です。

表 169. ブロック指向書き込み保護コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD28	ac	[31:0] データアド レス	R1b	SET_WRITE_PROT	カードに書き込み保護機能がある場合、こ のコマンドはアドレス指定されたグルー プの書き込み保護ビットをセットします。 書き込み保護のプロパティは、カード固有 データ (WP_GRP_SZIE) でコード化され ます。
CMD29	ac	[31:0] データアド レス	R1b	CLR_WRITE_PROT	カードが書き込み保護機能を備えている 場合、このコマンドはアドレス指定された グループの書き込み保護ビットをクリア します。
CMD30	adtc	[31:0] 書き込み保 護 データ アクセ ス	R1	SEND_WRITE_PROT	カードが書き込み保護機能を備えている 場合、このコマンドはカードに書き込み保 護ビットのステータスを送るようによ うに要求 します。
CMD31	予約済み				

表 170. 消去コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD32 ... CMD34		予約済み。古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。			
CMD35	ac	[31:0] データアドレス	R1	ERASE_GROUP_START	消去対象として選択される範囲の最初の消去グループのアドレスを設定します。
CMD36	ac	[31:0] データアドレス	R1	ERASE_GROUP_END	消去対象として選択される連続した範囲内の最後の消去グループのアドレスを設定します。
CMD37		予約済み。古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。			
CMD38	ac	[31:0] スタッフビット	R1	ERASE	以前に選択されたすべての書き込みブロックを消去します。

表 171. I/O モードコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD39	ac	[31:16] RCA [15:15] レジスタ書き込みフラグ [14:8] レジスタアドレス [7:0] レジスタデータ	R4	FAST_IO	8 ビット (レジスタ) データフィールドの読み出しおよび書き込みに使用されます。このコマンドは、カードとレジスタをアドレス指定し、書き込みフラグがセットされている場合は書き込み用データを提供します。R4 レスポンスは、アドレス指定されたレジスタから読み出されたデータを含みます。このコマンドは、MultiMediaCard 標準で定義されていないアプリケーション依存レジスタにアクセスします。
CMD40	bcr	[31:0] スタッフビット	R5	GO_IRQ_STATE	システムを割り込みモードにします。
CMD41		予約済み			

表 172. カードのロック

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD42	adtc	[31:0] スタッフビット	R1b	LOCK_UNLOCK	パスワードのセット／リセット、またはカードのロック／アンロックを行います。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD43 ... CMD54		予約済み			

表 173. アプリケーション固有コマンド

CMD インデックス	タイプ	引数	レスポンスフォーマット	略語	説明
CMD55	ac	[31:16] RCA [15:0] スタッ ビット	R1	APP_CMD	次のコマンドビットが標準コマンドでなく、アプリケーション固有コマンドであることをカードに示します。
CMD56	adtc	[31:1] スタッ ビット [0] : RD/WR	-	-	汎用／アプリケーション固有コマンドに対し、データブロックをカードに転送するため、またはカードからデータを取得するために使用されます。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD57 ... CMD59	予約済み。				
CMD60 ... CMD63	製造業者用に予約されています。				

27.5 レスポンスフォーマット

すべてのレスポンスは SDIO コマンドライン SDIO_CMD 経由で送信されます。レスポンスの送信は、常に、レスポンスコードワードに対応するビット文字列の左のビットから始まります。コード長は、レスポンスのタイプに依存します。

レスポンスは、常にスタートビット（常に 0）から始まり、送信方向を示すビットが続きます (card=0)。下の表で x で示されている値は、可変エントリを示します。R3 レスポンスを除くすべてのレスポンスは、CRC によって保護されます。すべてのコマンドコードワードは、終了ビット（常に 1）で終わります。

レスポンスには 5 つのタイプがあります。フォーマットは、次のとおりです。

27.5.1 R1（ノーマルレスポンスコマンド）

コード長 = 48 ビット。45:40 ビットはレスポンスされるコマンドのインデックスを示します。この値は 2 進コード番号（0～63）として解釈されます。カードのステータスは 32 ビットにコード化されます。

表 174. R1 レスポンス

ビット位置	幅（ビット）	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	X	コマンドインデックス
[39:8]	32	X	カードステータス
[7:1]	7	X	CRC7
0	1	1	終了ビット

27.5.2 R1b

R1 と同じですが、オプションとしてデータライン上にビジー信号を出力します。カードは、コマンドを受信する前の状態に基づいて、これらのコマンドを受信した後、ビジーになることがあります。

27.5.3 R2 (CID、CSD レジスタ)

コード長 = 136 ビット。CID レジスタの内容は、CMD2 および CMD10 コマンドのレスポンスとして送信されます。CSD レジスタの内容は、CMD9 のレスポンスとして送信されます。CID および CSD のビット [127... 1] だけが送信され、これらのレジスタの予約ビット [0] は、レスポンスの終了ビットで置き換えられます。カードは、SDIO_D0 をローに保持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることがあり、ホストは CMD7 を発行して、カードを選択解除できます。

表 175. R2 レスポンス

ビット位置	幅 (ビット)	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	"111111"	コマンドインデックス
[127:1]	127	X	カードステータス
0	1	1	終了ビット

27.5.4 R3 (OCR レジスタ)

コード長 : 48 ビット。OCR レジスタの内容は、CMD1 へのレスポンスとして送信されます。レベルのコーディングは、次のようになります。制限された電圧ウィンドウ=ロー、カードビジー=ロー。

表 176. R3 レスポンス

ビット位置	幅 (ビット)	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	"111111"	予約済み
[39:8]	32	X	OCR レジスタ
[7:1]	7	"1111111"	予約済み
0	1	1	終了ビット

27.5.5 R4 (高速 I/O)

コード長 : 48 ビット。引数フィールドは、アドレス指定されたカードの RCA、読み出しまたは書き込みレジスタアドレス、およびその内容を含みます。

表 177. R4 レスポンス

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	"100111"	CMD39
[39:8] 引数フィールド	[31:16]	16	X	RCA
	[15:8]	8	X	レジスタアドレス
	[7:0]	8	X	読み出しレジスタの内容
[7:1]		7	X	CRC7
0		1	1	終了ビット

27.5.6 R4b

SD I/O のみ : CMD5 を受信した SDIO カードは、一意の SDIO レスポンス R4 で応答します。フォーマットは、次のとおりです。

表 178. R4b レスポンス

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	X	予約済み
[39:8] 引数フィールド	39	16	X	カードはレディ状態
	[38:36]	3	X	I/O 機能数
	35	1	X	現在のメモリ
	[34:32]	3	X	スタッフビット
	[31:8]	24	X	I/O ORC
[7:1]		7	X	予約済み
0		1	1	終了ビット

SD I/O カードが CMD5 を受信すると、カードの I/O 部分は、以降のすべてのコマンドに対して通常どおりに応答できるようになります。I/O カード内の機能のこの I/O 有効は、カードがリセット、パワーサイクル、または I/O リセットに書き込む CMD52 を受信するまで、セットされたままです。SD メモリ専用カードは、CMD5 に応答することがありますので注意してください。メモリ専用カードの正しいレスポンスは、*現在のメモリ* = 1 と *I/O 機能数* = 0 です。SD メモリカード仕様 version 1.0 に準拠するように作られたメモリ専用カードは、CMD5 を不正なコマンドとして検出して、応答しません。I/O を認識できるホストは、CMD5 を送信します。カードがレスポンス R4 で応答した場合、ホストはレスポンス R4 に含まれるデータから、カードの構成を判断します。

27.5.7 R5（割り込みリクエスト）

MultiMediaCard 専用です。コード長：48 ビット。ホストによってレスポンスが生成された場合、引数の RCA フィールドは 0x0 になります。

表 179. R5 レスポンス

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	"101000"	CMD40
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]		7	X	CRC7
0		1	1	終了ビット

27.5.8 R6

SD I/O 専用です。メモリデバイスによる CMD3 への通常のレスポンスです。表 180 に示されています。

表 180. R6 レスポンス

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	"101000"	CMD40
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]		7	X	CRC7
0		1	1	終了ビット

カード [23:8] ステータスビットは、CMD3 が I/O 専用カードに送信されると変更されます。この場合、レスポンスの 16 ビットは、SD I/O 専用の値です。

- ビット [15] COM_CRC_ERROR
- ビット [14] ILLEGAL_COMMAND
- ビット [13] ERROR
- ビット [12:0] 予約済み

27.6 SDIO I/O カード固有の操作

以下の機能は、SD I/O 固有の操作です。

- SDIO_D2 信号による SDIO 読み出しウェイト操作
- クロックの停止による SDIO 読み出しウェイト操作
- SDIO サスペンド／レジューム操作（書き込みおよび読み出しサスペンド）
- SDIO 割込み

SDIO は、SDIO_DCTRL[11] ビットがセットされている場合のみ、これらの機能をサポートします。ただし、特定のハードウェア実装を必要としない読み出しサスペンドを除きます。

27.6.1 SDIO_D2 信号による SDIO I/O 読み出しウェイト操作

最初のブロックが受信される前に読み出しウェイトインターバルを開始することができます。データパスが有効 (SDIO_DCTRL[0] ビットがセット)、SDIO 固有操作が有効 (SDIO_DCTRL[11] ビットがセット)、読み出しウェイト開始 (SDIO_DCTRL[10] = 0 および SDIO_DCTRL[8] = 1)、データ方向がカードから SDIO (SDIO_DCTRL[1] = 1) のとき、DPSM はアイドルから ReadWait へ直接移行します。ReadWait では、DPSM は 2 SDIO_CLK クロックサイクル後に、SDIO_D2 を 0 に駆動します。この状態では、RWSTOP ビット (SDIO_DCTRL[9]) をセットすると、DPSM は、さらに 2 SDIO_CLK クロックサイクルの間ウェイト状態にとどまり、1 クロックサイクルの間 SDIO_D2 を 1 に駆動します (SDIO 仕様に従う)。その後、DPSM はカードからデータを受信するまで再びウェイトを開始します。DPSM は、読み出しウェイト開始がセットされている場合でも、ブロックを受信中は読み出しウェイトインターバルを開始しません。読み出しウェイトインターバルは、CRC の受信後に開始されます。新しい読み出しウェイト操作を開始するには、RWSTOP ビットをクリアする必要があります。読み出しウェイトインターバル中は、SDIO は SDIO_D1 上で SDIO 割込みを検出することができます。

27.6.2 SDIO_CLK の停止による SDIO 読み出しウェイト操作

SDIO カードが先の読み出しウェイト方法をサポートしない場合、SDIO は SDIO_CLK を停止することで読み出しウェイトを実行することができます (SDIO_DCTRL は [セクション 27.6.1](#) に記載される方法と同じようにセットされますが、SDIO_DCTRL[10] = 1 です)。DPSM は、現在受信しているブロックの終了ビットの 2 SDIO_CLK サイクル後にクロックを停止し、読み出しウェイトスタートビットがセットされた後に再びクロックを開始します。

SDIO_CLK が停止されると、任意のコマンドをカードに発行できます。読み出しウェイトインターバル中、SDIO は SDIO_D1 上の SDIO 割込みを検出できます。

27.6.3 SDIO サスペンド／レジューム操作

カードにデータを送信している間、SDIO は書き込み操作をサスペンドできます。SDIO_CMD[11] ビットがセットされ、現在のコマンドがサスペンドコマンドであることを CPSM に示します。CPSM はレスポンスを解析し、カードから ACK を受信すると (サスペンドの受入れ)、現在のブロックの CRC トークンの受信後にアイドル状態となることを DPSM に知らせます。

ハードウェアは、サスペンド操作 (レジューム) を完了させるために送信しなければならない残りの送信ブロック数を保存しません。

書き込み操作は、カードからサスペンドコマンドの ACK を受信したときに、DPSM を無効にするだけで (SDIO_DCTRL[0] = 0)、ソフトウェアによってサスペンドできます。その場合、DPSM はアイドル状態に入ります。

読み出しをサスペンドするには、DPSM は Wait_r 状態でサスペンドされる機能が完全なパケットを送信するのを待ってから、データ処理を停止します。アプリケーションは、FIFO が空になって、DPSM が自動的にアイドル状態になるまで、RxFIFO の読み出しを続けます。

27.6.4 SDIO 割込み

SDIO_DCTRL[11] ビットがセットされると、SDIO_D1 ライン上で SDIO 割込みが検出されます。

SDIO 割込みが検出されると、SDIO_STA[22] (SDIOIT) ビットがセットされます。この静的ビットは、クリアビット SDIO_ICR[22] (SDIOITC) によってクリアできます。SDIOIT ステータスビットをセットすると、割り込みを生成できます。個別の割り込みイネーブル SDIO_MASK[22] ビット (SDIOITE) は、割り込みリクエストを有効化および無効化するために使用できます。

SD カード割り込みの発生 (SDIO_STA[22] ビットがセットされている) に対応するために、ホストソフトウェアは次のステップに従います。

1. SDIOIT ビットをクリアして、SDIOITE 割り込み信号を無効にします (SDIO_MASK[22] = '0')。
2. カードの割り込みリクエストを処理し、SD カードの割り込みのソースをクリアします。
3. SDIOITC ビットに '1' を書き込んで、SDIOIT ビットをクリアします (SDIO_ICR[22] = '1')。
4. SDIOITE ビットに '1' を書き込んで、SDIOIT 割り込み信号を有効にします (SDIO_MASK[22] = '1')。

ステップ 2 から 4 は、SDIO の割り込みサービスルーチンから実行できます。

27.7 HW フロー制御

HW フロー制御機能は、FIFO アンダーラン (TX モード) およびオーバーラン (RX モード) エラーを回避するために使用します。

この機能は、SDIO_CK を停止し、SDIO ステートマシンをフリーズします。FIFO を使用してデータを送受信できない間、データ転送がストールされます。SDIOCLK によってクロック供給されるステートマシンだけが停止し、APB2 インタフェースは動作を続けます。したがって、フロー制御がアクティブであっても、FIFO がいっぱいになったり、空になることがあります。

HW フロー制御を有効にするには、SDIO_CLKCR[14] レジスタビットを 1 にセットする必要があります。リセット後、フロー制御は無効になります。

27.8 SDIO レジスタ

デバイスは、APB2 経由でアクセス可能な 32 ビット幅の制御レジスタによってシステムと通信します。

27.8.1 SDIO電源制御レジスタ (SDIO_POWER)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

[1:0] **PWRCTRL** : 電源制御ビット

これらのビットは、カードクロックの現在の機能状態を定義するために使用されます。

00 : パワーオフ : カードへのクロック供給は停止されます。

01 : 予約済み

10 : 予約済み、パワーアップ

11 : パワーオン : カードにクロックが供給されます。

注 : このレジスタへの連続書き込みアクセスをする場合には、7 PCLK2 クロック周期以上が必要です。

注 : データ書き込み後、3 SDIOCLK クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。

27.8.2 SDIOクロック制御レジスタ (SDIO_CLKCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

SDIO_CLKCR レジスタは、SDIO_CK 出カクロックを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HWFC _EN	NEGE DGE	WID BUS		BYPAS S	PWRS AV	CLKEN	CLKDIV							
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **HWFC_EN** : HW フロー制御有効

0b : HW フロー制御は無効です。

1b : HW フロー制御は有効です。

HW フロー制御が有効なときの TXFIFOE および RXFIFOE 割込み信号の意味については、[セクション 27.8.11](#) の SDIO ステータスレジスタの定義を参照してください。

ビット 13 **NEGEDGE** : SDIO_CLK 反位相選択ビット

0b : コマンドとデータはSDMMCCLK立ち下がりエッジに続くSDMMC_CLK立ち上がりエッジで変更されます。(SDIO_CLK 立ち上がりエッジは SDIOCLK 立ち上がりエッジで発生します)。

1b : コマンドとデータはSDMMC_CLK立ち下がりエッジで変更されます。

BYPASS がアクティブな場合、NEGEDGE 値にかかわらず、SDIOCLK 立ち下がりエッジでデータとコマンドが変更されます。

ビット 12:11 **WIDBUS** : ワイドバスモード有効ビット

00 : デフォルトバスモード : SDIO_D0 を使用します。

01 : 4 ワイドバスモード : SDIO_D[3:0] を使用します。

10 : 8 ワイドバスモード : SDIO_D[7:0] を使用します。

ビット 10 **BYPASS** : クロック分周器バイパス有効ビット

0 : バイパス無効 : SDIOCLK は、SDIO_CLK 出力信号を駆動する前に、CLKDIV の値に従って分周されます。

1 : バイパス有効 : SDIOCLK は、SDIO_CLK 出力信号を直接駆動します。

ビット 9 **PWRSAPV** : 節電構成ビット

節電のため、バスがアイドル状態のときには、PWRSAPV をセットして SDIO_CLK クロック出力を無効にできます。

0 : SDIO_CLK クロックは常に有効です。

1 : SDIO_CLK はバスがアクティブなときに限り有効です。

ビット 8 **CLKEN** : クロック有効ビット

0 : SDIO_CLK は無効です。

1 : SDIO_CLK は有効です。

ビット 7:0 **CLKDIV** : クロック分周ファクタ

このフィールドは、入力クロック (SDIOCLK) と出力クロック (SDIO_CLK) の間の分周ファクタを定義します (SDIO_CLK 周波数 = SDIOCLK / [CLKDIV + 2])。

- 注 :
- 1 **SD/SDIO カードまたは MultiMediaCard が識別モードのときには、SDIO_CLK 周波数は 400 kHz 未満である必要があります。**
 - 2 **クロック周波数は、相対カードアドレスがすべてのカードに割り当てられているときには、最大カードバス周波数に変更できます。**
 - 3 **データ書き込み後、3 SDIOCLK クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。SDIO_CLK は、SD I/O カードの読み出しウェイトインターバルの間、停止することができます。この場合、SDIO_CLKCR レジスタは SDIO_CLK を制御しません。**

27.8.3 SDIO 引数レジスタ（SDIO_ARG）

アドレスオフセット：0x08

リセット値：0x0000 0000

SDIO_ARG レジスタは、コマンドメッセージの一部としてカードに送信される 32 ビットのコマンド引数を含みます。

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

ビット 31:0 **CMDARG** : コマンド引数

コマンドメッセージの一部としてカードに送られるコマンド引数です。コマンドが引数を含む場合、コマンドをコマンドレジスタに書き込む前に、このレジスタにロードする必要があります。

27.8.4 SDIO コマンドレジスタ（SDIO_CMD）

アドレスオフセット：0x0C

リセット値：0x0000 0000

SDIO_CMD レジスタは、コマンドインデックスおよびコマンドタイプビットを含みます。コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。コマンドタイプビットは、コマンドパスステートマシン（CPSM）を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SDIO サスペンド	CPSM EN	WAIT PEND	WAIT INT	WAITRESP		CMDINDEX					
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SDIOSUSPEND** : SD I/O サスペンドコマンド

このビットがセットされると、送信されるコマンドはサスペンドコマンドです（SDIO カードでのみ使用されます）。

ビット 10 **CPSMEN** : コマンドパスステートマシン（CPSM）有効ビット

このビットがセットされると、CPSM が有効になります。

ビット 9 **WAITPEND** : CPSM のデータ転送終了待ち（CmdPend 内部信号）。

このビットがセットされると、CPSM は、データ転送の終了を待ってから、コマンドの送信を開始します。この機能は、ストリームデータの転送モードが SDIO_DCTRL[2] = 1 である場合にのみ使用できます。

ビット 8 **WAITINT** : CPSM の割込みリクエスト待ち
このビットがセットされると、CPSM は、コマンドタイムアウトを無効にして、割込みリクエストを待ちます。

ビット 7:6 **WAITRESP** : レスponsビット待ち
これらは、CPSM がレスポンスを待つかどうか、および、待つ場合にはレスポンスの種類を設定するために使用されます。
00 : レスponsなし。CMDSENT フラグを待ちます。
01 : ショートレスポンス。CMDREND または CCRCFAIL フラグを待ちます。
10 : レスponsなし。CMDSENT フラグを待ちます。
11 : ロングレスポンス。CMDREND または CCRCFAIL フラグを待ちます。

ビット 5:0 **CMDINDEX** : コマンドインデックス
コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。

注 : 1 データ書き込み後、3 SDIOCLK クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。
2 MultiMediaCards はショートレスポンス (48 ビット長) またはロングレスポンス (136 ビット長) の 2 種類のレスポンスを送信できます。SD カードと SD I/O カードはショートレスポンスのみ送信でき、引数はレスポンスのタイプによって異なることがあります。ソフトウェアは、送信されたコマンドに応じてレスポンスのタイプを区別します。

27.8.5 SDIO コマンドレスポンスレジスタ (SDIO_RESPCMD)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

SDIO_RESPCMD レジスタは、最後に受信したコマンドレスポンスのコマンドインデックスフィールドを含みます。コマンドレスポンス送信にコマンドインデックスフィールドが含まれていない場合 (ロングまたは OCR レスpons)、RESPCMD フィールドは未知ですが、11111b (レスポンスの予約フィールドの値) が含まれている必要があります。

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

ビット 31:6 予約済みであり、リセット値に保持する必要があります。
ビット 5:0 **RESPCMD** : レスponsコマンドインデックス
読み出し専用ビットフィールドです。最後に受信したコマンドレスポンスのコマンドインデックスを含みます。

27.8.6 SDIO レスポンス 1~4 レジスタ (SDIO_RESPx)

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

リセット値 : 0x0000 0000

SDIO_RESP1/2/3/4 レジスタは、受信したレスポンスの一部であるカードのステータスを含みます。

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

ビット 31:0 **CARDSTATUSx** : 表 181 を参照してください。

カードステータスサイズは、レスポンスのタイプによって 32 または 127 ビットです。

表 181. レスポンスタイプと SDIO_RESPx レジスタ

レジスタ	ショートレスポンス	ロングレスポンス
SDIO_RESP1	カードステータス [31:0]	カードステータス [127:96]
SDIO_RESP2	使用されない	カードステータス [95:64]
SDIO_RESP3	使用されない	カードステータス [63:32]
SDIO_RESP4	使用されない	カードステータス [31:1]0b

カードステータスの最上位ビットから先に受信します。SDIO_RESP4 レジスタ LSB は常に 0b です。

27.8.7 SDIO データタイマレジスタ (SDIO_DTIMER)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

SDIO_DTIMER レジスタは、データタイムアウト期間（カードバスクロック周期単位）を含みます。

カウンタは SDIO_DTIMER レジスタから値をロードして、データパスステートマシン (DPSM) が Wait_R または ビジー状態に入ったときに、デクリメントを開始します。DPSM がこのどちらかの状態のときにタイマが 0 に達した場合、タイムアウトステータスフラグがセットされます。

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

ビット 31:0 **DATATIME** : データタイムアウト期間

カードバスクロック周期で表されたデータタイムアウト期間です。

注 : データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。

27.8.8 SDIO データ長レジスタ (SDIO_DLEN)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

SDIO_DLEN レジスタは、転送されるデータバイト数を含みます。値は、データ転送が開始されたときに、データカウンタにロードされます。

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATALENGTH** : データ長の値
転送されるデータバイト数です。

注 : **ブロックデータ転送では、データ長レジスタの値はブロックサイズの倍数である必要があります (SDIO_DCTRL を参照してください)。データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。**
SDIO マルチバイト転送の場合、データ長レジスタ内の値は 1 から 512 まででなければなりません。

27.8.9 SDIO データ制御レジスタ (SDIO_DCTRL)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

SDIO_DCTRL レジスタは、データパスステートマシン (DPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE				DMA EN	DT MODE	DTD DIR	DTEN
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SDIOEN** : SD I/O 有効機能
このビットがセットされると、DPSM は SD I/O カード固有の操作を実行します。

ビット 10 **RWMOD** : 読み出しウェイトモード
0 : SDIO_D2 を停止する読み出しウェイト制御です。
1 : SDIO_CK を使用した読み出しウェイト制御です。

ビット 9 **RWSTOP** : 読み出しウェイト停止
0 : RWSTART ビットがセットされると、読み出しウェイトが進行中となります。
1 : RWSTART ビットがセットされると、読み出しウェイト停止が有効になります。



ビット 8 **RWSTART** : 読み出しウェイト開始

このビットがセットされると、読み出しウェイト操作が開始されます。

ビット 7:4 **DBLOCKSIZE** : データブロックサイズ

ブロックデータ転送モードが選択されているときのデータブロック長を定義します。

0000 : (0 10 進) ロック長 = 2^0 = 1 バイト
0001 : (1 1 進) ロック長 = 2^{10} = 2 バイト
0010 : (2 10 進) ロック長 = 2^2 = 4 バイト
0011 : (3 3 進) ロック長 = 2^{10} = 8 バイト
0100 : (4 4 進) ロック長 = 2^{10} = 16 バイト
0101 : (5 5 進) ロック長 = 2^{10} = 32 バイト
0110 : (6 6 進) ロック長 = 2^{10} = 64 バイト
0111 : (7 7 進) ロック長 = 2^{10} = 128 バイト
1000 : (8 8 進) ロック長 = 2^{10} = 256 バイト
1001 : (9 9 進) ロック長 = 2^{10} = 512 バイト
1010 : (10 10 進) ロック長 = 2^{10} = 1024 バイト
1011 : (11 11 進) ロック長 = 2^{10} = 2048 バイト
1100 : (12 12 進) ロック長 = 2^{10} = 4096 バイト
1101 : (13 13 進) ロック長 = 2^{10} = 8192 バイト
1110 : (14 14 進) ロック長 = 2^{10} = 16384 バイト
1111 : (15 10 進) 予約済み

ビット 3 **DMAEN** : DMA 有効

0 : DMA は無効です。
1 : DMA は有効です。

ビット 2 **DTMODE** : データ転送モード選択 1 : ストリームまたは SDIO マルチバイトデータ転送

0 : ブロックデータ転送です。
1 : ストリームまたは SDIO マルチバイトデータ転送です。

ビット 1 **DTDIR** : データ転送方向選択

0 : コントローラからカードへ
1 : カードからコントローラへ

[0] **DTEN** : データ転送有効ビット

データ転送は、1b が DTEN ビットに書き込まれると開始します。方向ビット DTDIR に応じて、DPSM は Wait_S、Wait_R、または転送の始めにすぐに RW Start がセットされた場合には ReadWait へ移行します。データ転送の終わりに有効ビットをクリアする必要はありませんが、新しいデータ転送を有効にするには、SDIO_DCTRL を更新する必要があります。

注 : データ書き込み後、3 SDIOCLK クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。

DTMODE ビットの意味は、SDIOEN ビットの値によって変化します。SDIOEN=0 およびDTMODE=1 の場合、MultiMediaCard ストリームモードが有効になり、SDIOEN=1 およびDTMODE=1 の場合、ペリフェラルがSDIO マルチバイト転送を有効にします。

27.8.10 SDIO データカウンタレジスタ (SDIO_DCOUNT)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

SDIO_DCOUNT レジスタは、DPSM がアイドル状態から Wait_R または Wait_S 状態へ移行すると、データ長レジスタから値をロードします (SDIO_DLEN を参照してください)。データが転送されると、カウンタは 0 に達するまで値をデクリメントします。その後、DPSM はアイドル状態へ移行し、データステータス終了フラグ DATAEND がセットされます。

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATACOUNT** : データカウント値

このビットを読み出すと、残りの転送バイト数が返されます。書き込み操作は何の影響も与えません。

注 : このレジスタは、データ転送が完了したときだけ読み出してください。

27.8.11 SDIO ステータスレジスタ (SDIO_STA)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

SDIO_STA レジスタは、読み出し専用レジスタです。2 種類のフラグを含みます。

- **スタティックフラグ (ビット [23:22,10:0])** : これらのビットは、SDIO 割り込みクリアレジスタ (SDIO_ICR を参照) に書き込んでクリアされるまで、アサートされたままです。
- **ダイナミックフラグ (ビット [21:11])** : これらのビットは、基礎回路の状態に応じて状態が変わります (たとえば、FIFO フルおよびエンプティフラグは、データが FIFO に書き込まれるとアサートおよびネゲートされます)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOIT	RXD AVL	TXD AVL	RX FIFOE	TX FIFOE	RX FIFO	TX FIFO
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HF	TX FIFO HE	RXACT	TXACT	CMD ACT	DBCK END	Res.	DATA END	CMD5 ENT	CMDR END	RX OVERR	TXUND ERR	DTIME OUT	CTIME OUT	DCRC FAIL	CCRC FAIL
r	r	r	r	r	r		r	r	r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SDIOIT** : SDIO 割り込みが受信されました。

ビット 21 **RXDAVL** : 受信 FIFO にデータがあります。

ビット 20 **TXDAVL** : 送信 FIFO にデータがあります。



- ビット 19 **RXFIFOE** : 受信 FIFO エンプティ
- ビット 18 **TXFIFOE** : 送信 FIFO エンプティ
HW フロー制御が有効なとき、FIFO に 2 ワードが含まれると、TXFIFOE 信号がアクティブになります。
- ビット 17 **RXFIFOF** : 受信 FIFO がいっぱいです。
HW フロー制御が有効なとき、FIFO がいっぱいになる 2 ワード前に RXFIFOF 信号がアクティブになります。
- ビット 16 **TXFIFOF** : 送信 FIFO がいっぱいです。
- ビット 15 **RXFIFOHF** : 受信 FIFO がハーフフル状態です。FIFO には少なくとも 8 ワード入っています。
- ビット 14 **TXFIFOHE** : 送信 FIFO がハーフエンプティ状態です。FIFO には少なくとも 8 ワード書き込み可能です。
- ビット 13 **RXACT** : データ受信中です。
- ビット 12 **TXACT** : データ送信中です。
- ビット 11 **CMDACT** : コマンド転送中です。
- ビット 10 **DBCKEND** : データブロック送受信（CRC 確認成功）しました。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **DATAEND** : データの終わり（データカウンタ SDIDCOUNT がゼロ）です。
- ビット 7 **CMDSENT** : コマンドが送信されました（レスポンス不要）。
- ビット 6 **CMDREND** : コマンドレスポンスが受信されました（CRC 確認成功）。
- ビット 5 **RXOVERR** : FIFO オーバーランエラーを受信しました。
注 : *SDIO FIFO を読み出すために DMA が使用される場合（SDIO_DCTRL レジスタの DMAEN ビットがセットされている場合）、ユーザソフトウェアでは DMA ストリームを無効にし、'0' を書き込む必要があります（DMA リクエストの生成を無効にするため）。*
- ビット 4 **TXUNDERR** : 送信 FIFO アンダーランエラーです。
注 : *SDIO FIFO に書き込むために DMA が使用される場合（SDIO_DCTRL レジスタの DMAEN ビットがセットされている場合）、ユーザソフトウェアでは DMA ストリームを無効にし、DMAEN に '0' を書き込む必要があります（DMA リクエストの生成を無効にするため）。*
- ビット 3 **DTIMEOUT** : データタイムアウトです。
- ビット 2 **CTIMEOUT** : コマンドレスポンスタイムアウトです。
コマンドタイムアウト期間は、64 SDIO_CK クロック周期固定です。
- ビット 1 **DCRCFAIL** : データブロック送受信（CRC 確認失敗）しました。
- ビット 0 **CCRCFAIL** : コマンドレスポンス受信（CRC 確認失敗）しました。

27.8.12 SDIO 割り込みクリアレジスタ（SDIO_ICR）

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

SDIO_ICR レジスタは、書き込み専用レジスタです。ビットに 1b を書き込むと、SDIO_STA ステータスレジスタの対応するビットがクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIO ITC	Res.	Res.	Res.	Res.	Res.	Res.
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DBCK ENDC	Res.	DATA ENDC	CMD SENTC	CMD REND C	RX OVERR C	TX UNDERR C	DTIME OUTC	CTIME OUTC	DCRC FAILC	CCRC FAILC
					rW		rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SDIOITC** : SDIOIT フラグクリアビット

SDIOIT フラグをクリアするために、ソフトウェアによってセットされます。

0 : SDIOIT はクリアされません。

1 : SDIOIT はクリアされます。

ビット 21:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DBCKENDC** : DBCKEND フラグクリアビット

DBCKEND フラグをクリアするために、ソフトウェアによってセットされます。

0 : DBCKEND はクリアされません。

1 : DBCKEND はクリアされます。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DATAENDC** : DATAEND フラグクリアビット

DATAEND フラグをクリアするために、ソフトウェアによってセットされます。

0 : DATAEND はクリアされません。

1 : DATAEND はクリアされます。

ビット 7 **CMDSENTC** : CMDSENT フラグクリアビット

CMDSENT フラグをクリアするために、ソフトウェアによってセットされます。

0 : CMDSENT はクリアされません。

1 : CMDSENT はクリアされます。

ビット 6 **CMDREND C** : CMDREND フラグクリアビット

CMDREND フラグをクリアするために、ソフトウェアによってセットされます。

0 : CMDREND はクリアされません。

1 : CMDREND はクリアされます。

ビット 5 **RXOVERRC** : RXOVERR フラグクリアビット

RXOVERR フラグをクリアするために、ソフトウェアによってセットされます。

0 : RXOVERR はクリアされません。

1 : RXOVERR はクリアされます。

ビット 4 **TXUNDERRC** : TXUNDERR フラグクリアビット

TXUNDERR フラグをクリアするために、ソフトウェアによってセットされます。

0 : TXUNDERR はクリアされません。

1 : TXUNDERR はクリアされます。

ビット 3 **DTIMEOUTC** : DTIMEOUT フラグクリアビット

DTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。

0 : DTIMEOUT はクリアされません。

1 : DTIMEOUT はクリアされます。

- ビット 2 **CTIMEOUTC** : CTIMEOUT フラグクリアビット
CTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。
0 : CTIMEOUT はクリアされません。
1 : CTIMEOUT はクリアされます。
- ビット 1 **DCRCFAILC** : DCRCFAIL フラグクリアビット
DCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。
0 : DCRCFAIL はクリアされません。
1 : DCRCFAIL はクリアされます。
- ビット 0 **CCRCFAILC** : CCRCFAIL フラグクリアビット
CCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。
0 : CCRCFAIL はクリアされません。
1 : CCRCFAIL はクリアされます。

27.8.13 SDIO マスクレジスタ (SDIO_MASK)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

割込みマスクレジスタは、対応するビットを 1b にセットすることによって、割込みリクエストを生成するステータスフラグを決めます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIO ITIE	RXD AVLIE	TXD AVLIE	RX FIFO EIE	TX FIFO EIE	RX FIFO FIE	TX FIFO FIE
									r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HFIE	TX FIFO HEIE	RX ACTIE	TX ACTIE	CMD ACTIE	DBCK ENDIE	Res.	DATA ENDIE	CMD SENT IE	CMD REND IE	RX OVERR IE	TX UNDERR IE	DTIME OUTIE	CTIME OUTIE	DCRC FAILIE	CCRC FAILIE
r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31:23 予約済みであり、リセット値に保持する必要があります。
- ビット 22 **SDIOITIE** : SDIO モード割込み受信割込み有効
SDIO モード割込みを受信したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : SDIO モード割込み受信割込みは無効です。
1 : SDIO モード割込み受信割込みは有効です。
- ビット 21 **RXDAVLIE** : Rx FIFO データ有無割込み有効
Rx FIFO のデータの有無によって生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Rx FIFO のデータの有無による割込みは無効です。
1 : Rx FIFO のデータの有無による割込みは有効です。
- ビット 20 **TXDAVLIE** : Tx FIFO データ有無割込み有効
Tx FIFO のデータの有無によって生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : Tx FIFO のデータの有無による割込みは無効です。
1 : Tx FIFO のデータの有無による割込みは有効です。



ビット 19 RXFIFOEIE : Rx FIFO エンプティ割込み有効

Rx FIFO エンプティによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO エンプティ割込みは無効です。

1 : Rx FIFO エンプティ割込みは有効です。

ビット 18 TXFIFOEIE : Tx FIFO エンプティ割込み有効

Tx FIFO エンプティによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO エンプティ割込みは無効です。

1 : Tx FIFO エンプティ割込みは有効です。

ビット 17 RXFIFOIE : Rx FIFO フル割込み有効

Rx FIFO が満杯になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO フル割込みは無効です。

1 : Rx FIFO フル割込みは有効です。

ビット 16 TXFIFOIE : Tx FIFO フル割込み有効

Tx FIFO が満杯になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO フル割込みは無効です。

1 : Tx FIFO フル割込みは有効です。

ビット 15 RXFIFOHFIE : Rx FIFO ハーフフル割込み有効

Rx FIFO がハーフフル状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO ハーフフル割込みは無効です。

1 : Rx FIFO ハーフフル割込みは有効です。

ビット 14 TXFIFOHEIE : Tx FIFO ハーフエンプティ割込み有効

Tx FIFO がハーフエンプティ状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO ハーフエンプティ割込みは無効です。

1 : Tx FIFO ハーフエンプティ割込みは有効です。

ビット 13 RXACTIE : データ受信割込み有効

データ受信中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ受信割込みは無効です。

1 : データ受信割込みは有効です。

ビット 12 TXACTIE : データ送信割込み有効

データ送信中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ送信割込みは無効です。

1 : データ送信割込みは有効です。

ビット 11 CMDACTIE : コマンド動作中割込み有効

コマンド転送中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド動作中割込みは無効です。

1 : コマンド動作中割込みは有効です。

ビット 10 DBCKENDIE : データブロックエンド割込み有効

データブロックエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データブロックエンド割込みは無効です。

1 : データブロックエンド割込みは有効です。

ビット 9 予約済みであり、リセット値に保持する必要があります。



ビット 8 DATAENDIE : データエンド割込み有効

データエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データエンド割込みは無効です。

1 : データエンド割込みは有効です。

ビット 7 CMDSENTIE : コマンド送信割込み有効

コマンド送信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド送信割込みは無効です。

1 : コマンド送信割込みは有効です。

ビット 6 CMDRENDIE : コマンドレスポンス受信割込み有効

コマンドレスポンスの受信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドレスポンス受信割込みは無効です。

1 : コマンドレスポンス受信割込みは有効です。

ビット 5 RXOVERRIE : Rx FIFO オーバーランエラー割込み有効

Rx FIFO オーバーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO オーバーランエラー割込みは無効です。

1 : Rx FIFO オーバーランエラー割込みは有効です。

ビット 4 TXUNDERRIE : Tx FIFO アンダーランエラー割込み有効

Tx FIFO アンダーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO アンダーランエラー割込みは無効です。

1 : Tx FIFO アンダーランエラー割込みは有効です。

ビット 3 DTIMEOUTIE : データタイムアウト割込み有効

データタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データタイムアウト割込みは無効です。

1 : データタイムアウト割込みは有効です。

ビット 2 CTIMEOUTIE : コマンドタイムアウト割込み有効

コマンドタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドタイムアウト割込みは無効です。

1 : コマンドタイムアウト割込みは有効です。

ビット 1 DCRCFAILIE : データ CRC 失敗割込み有効

データ CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ CRC 失敗割込みは無効です。

1 : データ CRC 失敗割込みは有効です。

ビット 0 CCRCFAILIE : コマンド CRC 失敗割込み有効

コマンド CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド CRC 失敗割込みは無効です。

1 : コマンド CRC 失敗割込みは有効です。

27.8.14 SDIO FIFO カウンタレジスタ (SDIO_FIFOCNT)

アドレスオフセット : 0x48

リセット値 : 0x0000 0000

SDIO_FIFOCNT レジスタは、FIFO に書き込まれるか、FIFO から読み出される残りのワード数を含みます。FIFO カウンタは、データ制御レジスタ (SDIO_DCTRL レジスタ) のデータ転送有効ビット DTEN がセットされていて、DPSM がアイドル状態のときに、データ長レジスタ (SDIO_DLEN を参照) から値をロードします。データ長がワード境界 (4 の倍数) に合っていない場合、残りの 1~3 バイトは 1 ワードとみなされます。

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

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **FIFOCOUNT** : FIFO に書き込まれるか、FIFO から読み出される残りのワード数

27.8.15 SDIO データ FIFO レジスタ (SDIO_FIFO)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

受信および送信 FIFO は、32 ビット幅のレジスタとして読み書きできます。FIFO は、32 の連続したアドレスに 32 個のエントリを含みます。このため、CPUはソフトウェアによるFIFOへの読み書きを行うことができます。

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

ビット 31:0 **FIFOData** : 受信および送信 FIFO データ

FIFO データは、次のアドレスから 32 ビットワードの 32 個のエントリを占有します。

SDIO ベース+ 0x080~SDIO ベース+ 0xFC

次の表に SDIO レジスタの一覧を示します。

表 182. SDIO レジスタマップ

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などに付きましては如何なる責任にも負いません。

表 182. SDIO レジスタマップ (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	SDIO_DCOUNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT																								
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	SDIO_STA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOIT	RXDAVL	TXDAVL	RXFIOE	TXFIOE	RXFIOF	TXFIOF	RXFIOHF	TXFIOHF	RXACT	TXACT	CMDACT	DBCKEND	Res.	DATAEND	CMDSNT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	Res.	0	0	0	0	0	0	0	0	0
0x38	SDIO_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOITC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBCKENDC	Res.	DATAENDC	CMDSNTC	CMDREND	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
	リセット値										0												0		0	0	0	0	0	0	0	0	0
0x3C	SDIO_MASK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIOEIE	TXFIOEIE	RXFIOFIE	TXFIOFIE	RXFIOHFIE	TXFIOHFIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	Res.	DATAENDIE	CMDSNTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0x48	SDIO_FIFOCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFOCOUNT																							
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	SDIO_FIFO	FIFOData																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

28 コントローラエリアネットワーク (bxCAN)

28.1 概要

Basic Extended CAN ペリフェラル (略称 **bxCAN**) は CAN ネットワークのインタフェースです。CAN プロトコルバージョン 2.0A および B をサポートしています。小さな CPU 負荷で多くの受信メッセージを効率よく取り扱えるように設計されています。また、送信メッセージの優先順位に関する要件も満たしています。

安全性が重視されるアプリケーションの場合、CAN コントローラは、CAN タイムトリガ通信オプションをサポートするすべてのハードウェア機能を提供します。

28.2 bxCAN の主な機能

- CAN プロトコルバージョン 2.0 A、B Active のサポート
- ビットレート：最大 1 Mbit/s
- タイムトリガ通信オプションのサポート

送信

- 3 つの送信メールボックス
- 設定可能な送信優先順位
- SOF 送信時のタイムスタンプ

受信

- 2 つの 3 ステージ受信 FIFO
- スケーラブルフィルタバンク：
 - 28 個のフィルタバンクを CAN1 と CAN2 で共有
- 識別子リスト機能
- 設定可能な FIFO オーバーラン
- SOF 受信時のタイムスタンプ

タイムトリガ通信オプション

- 自動再送信モードの無効化
- 16 ビットのフリーランニングタイマ
- タイムスタンプを最後の 2 データバイトで送信

管理

- マスク可能な割り込み
- 固有のアドレス空間でのソフトウェア効率の高いメールボックス配置

デュアル CAN

- CAN1：スレーブ bxCAN と 512 バイトの SRAM メモリの間の通信を管理するマスタ bxCAN
- CAN2：スレーブ bxCAN。SRAM メモリへの直接アクセスなし。
- この 2 つの bxCAN セルは、512 バイトの SRAM メモ리를 共有します (図 319：デュアル CAN ブロック図を参照)。

28.3 bxCAN の概要

今日の CAN アプリケーションでは、ネットワークのノード数が増加し、複数のネットワークがゲートウェイを介してリンクされることもあります。一般に、システム内の（そして各ノードが扱う）メッセージ数は大幅に増加しています。アプリケーションメッセージに加えて、ネットワーク管理や診断のメッセージも導入されています。

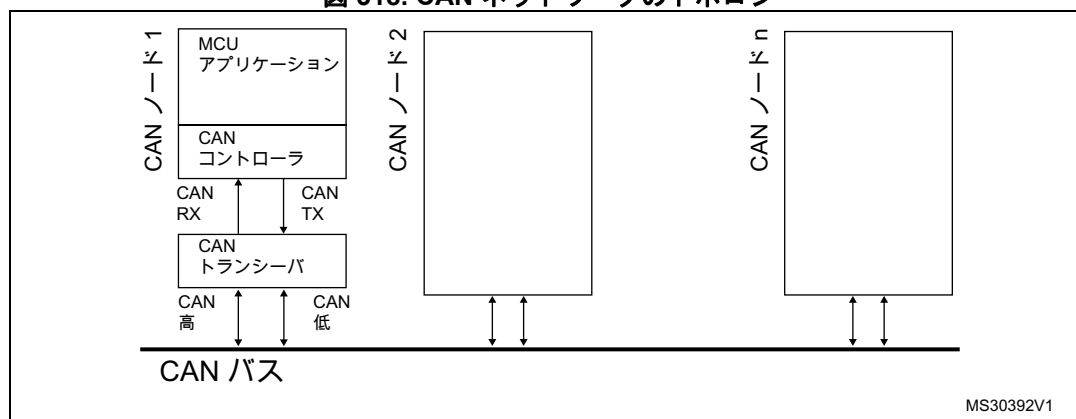
- 各種のメッセージを扱うには、高度なフィルタ機構が要求されます。

さらに、アプリケーションタスクはより多くの CPU 時間を必要とするため、メッセージ受信に起因するリアルタイム制約を減らす必要があります。

- 受信 FIFO 方式では、CPU は、メッセージを失うことなく長時間にわたってアプリケーションタスクに専念できます。

標準の CAN ドライバに基づく標準 HLP (Higher Layer Protocol : 上位層プロトコル) は、CAN コントローラとの効率的なインタフェースを必要とします。

図 318. CAN ネットワークのトポロジ



28.3.1 CAN 2.0B アクティブコア

bxCAN モジュールは、CAN メッセージの送受信を完全に自律的に処理します。標準識別子 (11 ビット) と拡張識別子 (29 ビット) は、ハードウェアによって完全にサポートされています。

28.3.2 制御レジスタ、ステータスレジスタ、設定レジスタ

アプリケーションは、これらのレジスタを以下の目的で使用します。

- CAN パラメータ (ボーレートなど) の設定
- 送信リクエスト
- 受信処理
- 割り込み管理
- 診断情報の入手

28.3.3 送信メールボックス

ソフトウェアがメッセージをセットアップするために、3 つの送信メールボックスが用意されています。最初に送信されるメールボックスは、送信スケジューラによって決定されます。

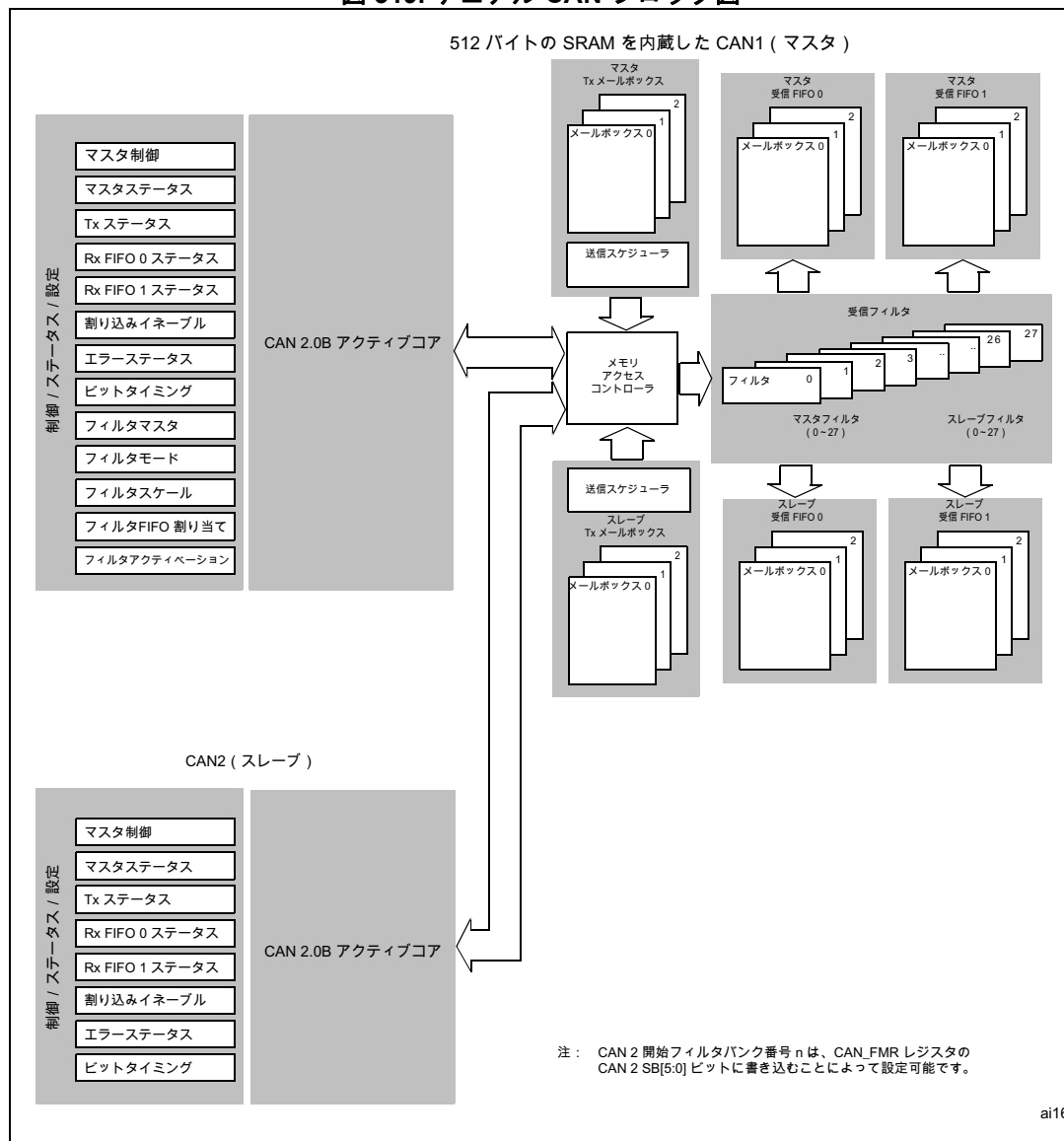
28.3.4 受信フィルタ

bxCAN は、受信メッセージからソフトウェアが必要なものを選び出し、他を破棄するために、28 個のスケラブルで設定可能な識別子フィルタバンクを備えています。

受信 FIFO

ハードウェアは、受信メッセージを格納するために 2 つの受信 FIFO を使用します。各 FIFO には、3 つの完全なメッセージを格納できます。FIFO の管理はすべてハードウェアによって行われます。

図 319. デュアル CAN ブロック図



28.4 bxCAN 動作モード

bxCAN には、**初期化**、**通常**、および**スリープ**という 3 つの主な動作モードがあります。ハードウェアリセットの後、bxCAN は消費電力を低減するためにスリープモードに入り、CANTX では内部プルアップがアクティブとなります。ソフトウェアは、CAN_MCR レジスタの INRQ または SLEEP ビットをセットすることによって、bxCAN に**初期化**または**スリープモード**に入るように要求します。いずれかのモードに入ると、bxCAN は、CAN_MSR レジスタの INAK または SLAK ビットをセットすることによってそれを確定し、内部プルアップは無効になります。INAK も SLAK もセットされていない場合、bxCAN は**通常モード**です。**通常モード**に入る前に、bxCAN は必ず CAN バスで**同期**する必要があります。同期のために、bxCAN は、CAN バスがアイドル状態になる（つまり、CANRX 上で 11 個の連続したリセッシブビットが観測される）のを待ちます。

28.4.1 初期化モード

ハードウェアが初期化モードにある間に、ソフトウェアの初期化も行うことができます。このモードに入るには、ソフトウェアは CAN_MCR レジスタの INRQ ビットをセットし、ハードウェアが CAN_MSR レジスタの INAK ビットをセットしてこのリクエストを確認するのを待ちます。

初期化モードから出るには、ソフトウェアで INRQ ビットをクリアします。INAK ビットがハードウェアによってクリアされると、bxCAN は初期化モードを終了します。

初期化モードでは、CAN バスとの間のすべてのメッセージ転送は停止し、CAN バス出力 CANTX のステータスはリセッシブ（ハイ）になります。

初期化モードに入っても、設定レジスタの内容は一切変化しません。

CAN コントローラを初期化するには、ソフトウェアでビットタイミング (CAN_BTR) および CAN オプション (CAN_MCR) レジスタの設定を行う必要があります。

CAN フィルタバンクに関連付けられたレジスタ (モード、スケール、FIFO 割当て、アクティベーション、およびフィルタ値) を初期化するには、ソフトウェアが FINIT ビット (CAN_FMR) をセットする必要があります。フィルタの初期化は、初期化モード以外でも行うことができます。

注： **FINIT=1 のとき、CAN 受信は無効化されます。**

フィルタの値は、CAN_FA1R レジスタの関連するフィルタアクティベーションビットを無効にすることによっても変更できます。

フィルタバンクを使用しない場合は、これを非アクティブのままにしておく (対応する FACT ビットをクリアしておく) ことを推奨します。

28.4.2 通常モード

初期化が終了すると、ソフトウェアはハードウェアに対し、CAN バスで同期して送受信を開始できるように通常モードに入るように要求する必要があります。

この通常モードに入るリクエストは、CAN_MCR レジスタの INRQ ビットをクリアすることによって行われます。bxCAN は通常モードに入り、11 個の連続したリセッシブビットの発生 (バスアイドル状態) を待って CAN バス上のデータ転送と同期すると、バス動作への参加準備完了となります。通常モードへの切り替えは、CAN_MSR レジスタの INAK ビットをクリアすることによって、ハードウェアによって確認されます。

フィルタ値の初期化は初期化モードとは別に行いますが、フィルタがアクティブでない (対応する FACTx ビットがクリアされている) ときに行う必要があります。フィルタのスケールとモードの設定は、通常モードに入る前に行う必要があります。

28.4.3 スリープモード (低電力)

消費電力を低減するため、bxCAN にはスリープモードと呼ばれる低電力モードがあります。ソフトウェアのリクエストで CAN_MCR レジスタの SLEEP ビットをセットすると、このモードに入ります。このモードでは bxCAN のクロックは停止しますが、ソフトウェアから bxCAN のメールボックスにアクセスすることは可能です。

bxCAN がスリープモードのときにソフトウェアが INRQ ビットをセットして初期化モードに入るように要求する場合には、SLEEP ビットもクリアする必要があります。

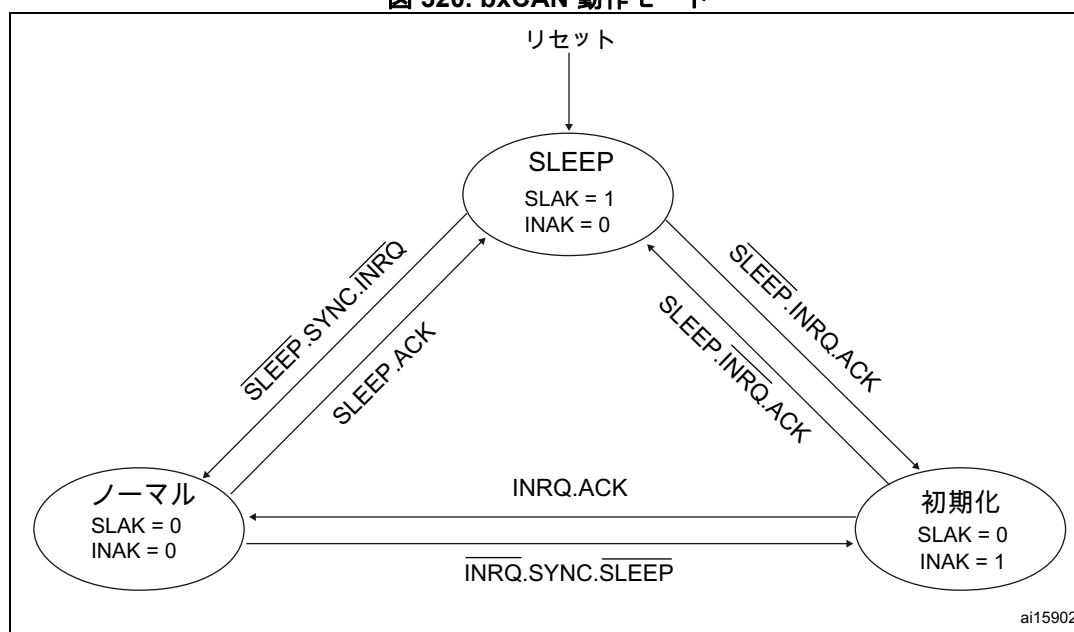
bxCAN がスリープモードからウェイクアップするには、ソフトウェアで SLEEP ビットをクリアするか、CAN バス動作を検出する必要があります。

CAN バス動作を検出すると、ハードウェアは、SLEEP ビットをクリアしてウェイクアップシーケンスを自動的に実行します (CAN_MCR レジスタの AWUM ビットがセットされている場合)。AWUM ビットがクリアされている場合、スリープモードから出るにはウェイクアップ割込み発生時にソフトウェアが SLEEP ビットをクリアする必要があります。

注： ウェイクアップ割込みが有効である (CAN_IER レジスタの WKUIE ビットがセットされている) 場合には、bxCAN がウェイクアップシーケンスを自動的に実行する場合でも、CAN バス動作の検出時にウェイクアップ割込みが生成されます。

SLEEP ビットがクリアされた後、bxCAN が CAN バスに同期するとスリープモードを終了します (図 320: bxCAN 動作モードを参照)。SLAK ビットがハードウェアによってクリアされると、スリープモードを終了します。

図 320. bxCAN 動作モード



1. ACK = ハードウェアが CAN_MSR レジスタの INAK または SLAK ビットをセットすることによってリクエストを確認する待ち状態。
2. SYNC = CAN バスがアイドル状態になる (つまり、11 個の連続したリセッピットが CANRX で観測される) までの bxCAN の待ち状態。

28.5 テストモード

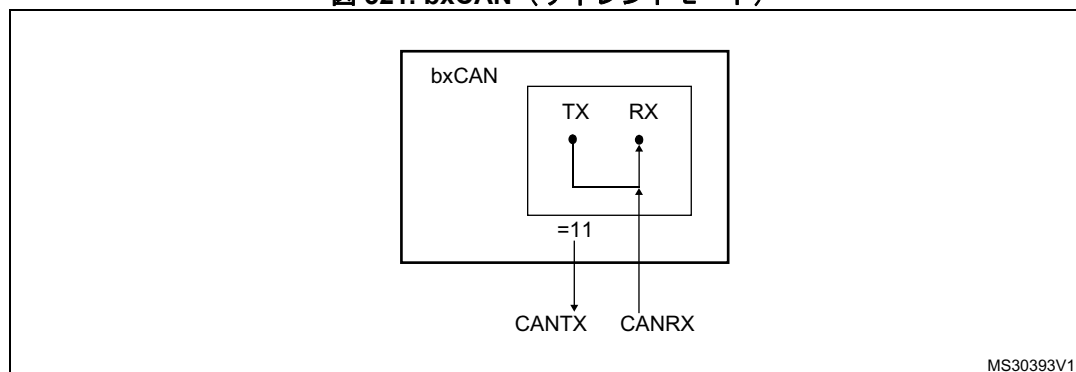
テストモードは、CAN_BTR レジスタの SILM ビットと LBKM ビットによって選択できます。これらのビットは、bxCAN が初期化モードにある間に設定する必要があります。テストモードが選択された後で通常モードに戻るには、CAN_MCR レジスタの INRQ ビットをリセットする必要があります。

28.5.1 サイレントモード

bxCAN をサイレントモードにするには、CAN_BTR レジスタの SILM ビットをセットします。

サイレントモードでは、bxCAN は有効なデータフレームやリモートフレームを受信できますが、CAN バスではリセシブビットのみを送信するため、送信を開始することはできません。bxCAN がドミナントビット (ACK ビット、オーバーロードフラグ、アクティブエラーフラグ) を送信する必要がある場合、CAN バスがリセシブ状態にとどまることがあっても CAN コアがこのドミナントビットを監視できるように、このビットは内部でルート変更されます。サイレントモードを使用すると、ドミナントビット (ACK ビット、エラーフレーム) の送信による影響を受けずに、CAN バス上のトラフィックを解析できます。

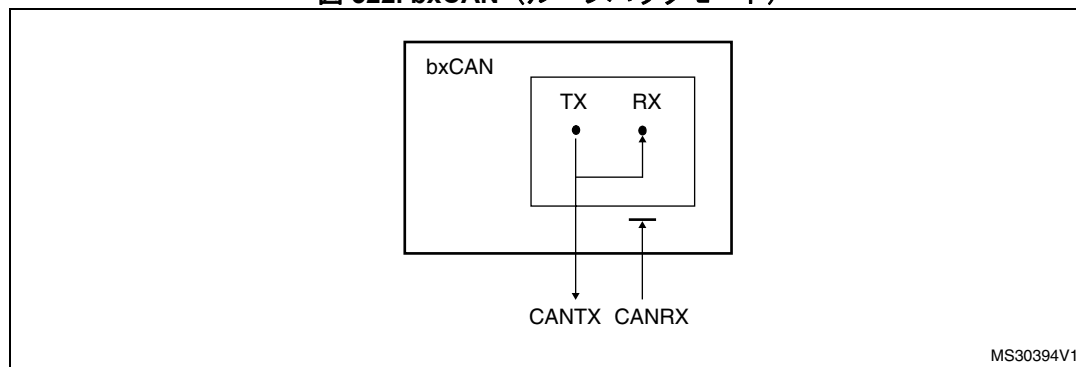
図 321. bxCAN (サイレントモード)



28.5.2 ループバックモード

bxCAN をループバックモードに設定するには、CAN_BTR レジスタの LBKM ビットをセットします。ループバックモードでは、bxCAN は自分が送信したメッセージを受信メッセージとして扱い、それらを受信メールボックスに格納します (受信フィルタを通過した場合)。

図 322. bxCAN (ループバックモード)



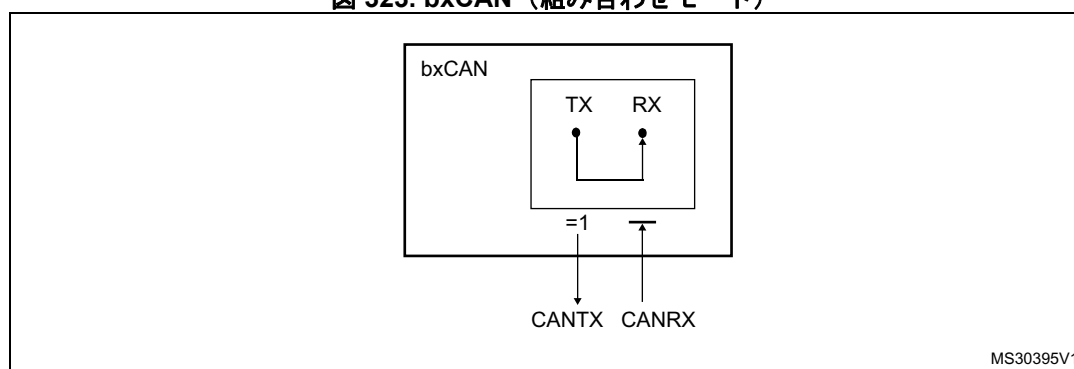
このモードは、セルフテスト機能のために用意されています。外部イベントから独立させるために、ループバックモードでは、CAN コアは ACK エラーを無視します (データ/リモートフレームの ACK スロットではドミナントビットはサンプリングされません)。このモードでは、bxCAN は Tx 出力が

ら Rx 入力へ内部フィードバックを行います。bxCAN は、CANRX 入力ピンの実際の値を無視します。送信メッセージは CANTX ピンで監視できます。

28.5.3 サイレントモードとループバックモードの組み合わせ

CAN_BTR レジスタの LBKM ビットと SILM ビットをセットすることによって、ループバックモードとサイレントモードを組み合わせることもできます。このモードは「ホットセルフテスト」に使用できます。つまり、CANTX および CANRX ピンに接続されている実行中の CAN システムに影響を与えずに、ループバックモードと同様に bxCAN をテストできます。このモードでは、CANRX ピンは bxCAN から切断され、CANTX ピンはリセッショに保持されます。

図 323. bxCAN (組み合わせモード)



28.6 デバッグモードでの挙動

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コア停止)、次の状態に応じて、bxCAN は通常どおりに動作を続けるか、または停止します。

- DBG モジュールで CAN1 の場合は DBG_CAN1_STOP ビット、CAN2 の場合は DBG_CAN2_STOP ビット。
- CAN_MCR の DBF ビット。詳細については、[セクション 28.9.2 : CAN 制御/ステータスレジスタ](#)を参照してください。

28.7 bxCAN 機能の説明

28.7.1 送信処理

メッセージを送信するには、アプリケーションは、**EMPTY** 状態の送信メールボックスを 1 つ選択し、識別子、データ長コード (DLC)、およびデータを設定してから、CAN_TxR レジスタの対応する TXRQ ビットをセットすることによって、送信を要求する必要があります。メールボックスが **EMPTY** 状態でなくなると、ソフトウェアはこのメールボックスレジスタへの書き込みができなくなります。TXRQ ビットがセットされると、メールボックスはただちに **PENDING** 状態になり、最高優先順位のメールボックスになるまで待機します。[送信優先順位](#)を参照してください。メールボックスの優先順位が最高になると、メールボックスは送信の **SCHEDULED** 状態になります。スケジュール済みメールボックスのメッセージの送信は、CAN バスがアイドル状態になると開始されます (**TRANSMIT** 状態になります)。メールボックスの内容が正常に送信されると、そのメールボックスは再び **EMPTY** 状態になります。ハードウェアは、CAN_TSR レジスタの RQCP ビットと TXOK ビットをセットして、送信が正常に終了したことを示します。

送信が失敗した場合、その原因がアービトレーションロストである場合は CAN_TSR レジスタの ALST ビットによって、または、送信エラー検出である場合は TERR ビットによって示されます。

送信優先順位

識別子による優先順位

複数の送信メールボックスが保留中のときには、送信順序は、メールボックスに格納されているメッセージの識別子によって決まります。CAN プロトコルのアービトレーションに従って、識別子の値が最小のメッセージの優先順位が最高になります。識別子の値が同じ場合には、メールボックス番号の小さい方が優先的にスケジュールされます。

送信リクエストの順序による優先順位

送信メールボックスは、CAN_MCR レジスタの TXFP ビットをセットすることによって、送信 FIFO として設定できます。このモードでは、優先順位は送信リクエストの順序によって決まります。

このモードは分割送信の場合に非常に役立ちます。

アボート

ユーザは、CAN_TSR レジスタの ABRQ ビットをセットすることによって、送信リクエストをアボートできます。**PENDING** または **SCHEDULED** 状態では、メールボックスはただちにアボートされます。メールボックスが **TRANSMIT** 状態のときにアボートリクエストが出されると、次の 2 つの結果につながります。メールボックスの送信が正常に行われた場合、CAN_TSR レジスタの TXOK ビットがセットされ、メールボックスは **EMPTY** 状態になります。送信に失敗した場合、メールボックスは **SCHEDULED** 状態になり、送信がアボートされ、TXOK ビットがクリアされて、メールボックスは **EMPTY** 状態になります。いずれの場合にも、現在の送信が終了するとメールボックスは最低でも再び **EMPTY** 状態になります。

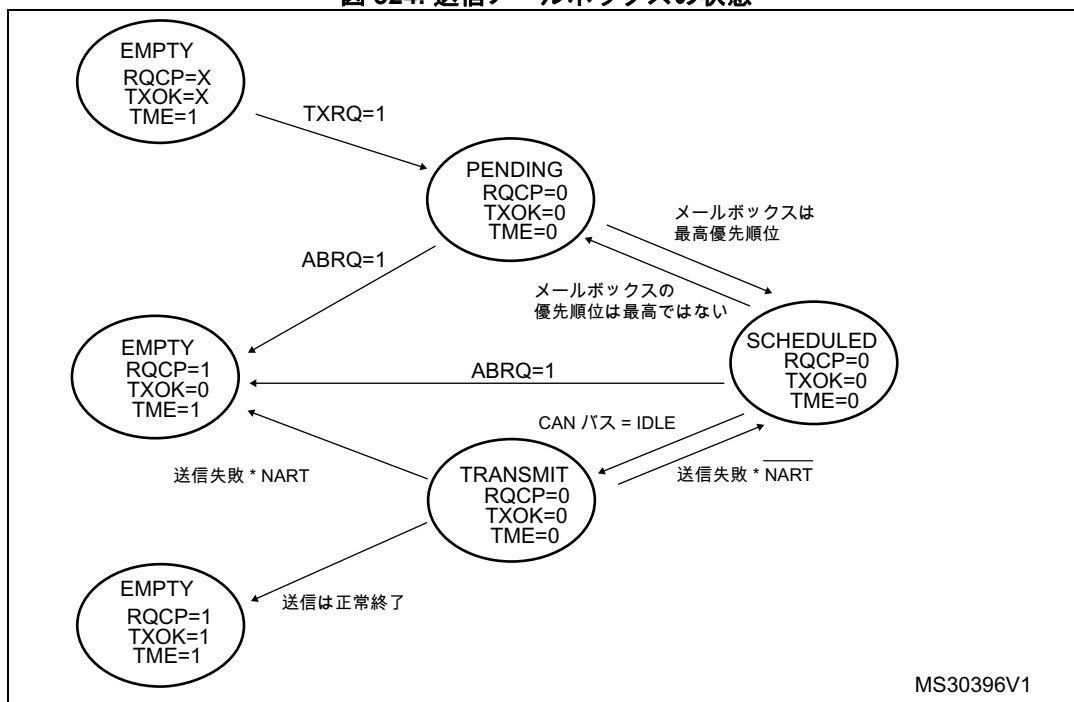
自動再送信禁止モード

このモードは、CAN 規格のタイムトリガ通信オプションの要件を満たすために実装されています。ハードウェアをこのモードに設定するには、CAN_MCR レジスタの NART ビットをセットする必要があります。

このモードでは、個々の送信は一度だけ開始されます。最初の試みがアービトレーションロスやエラーのために失敗した場合、ハードウェアはメッセージ送信を自動的にリスタートしません。

最初の送信試行の終わりに、ハードウェアはそのリクエストが完了したものとみなし、CAN_TSR レジスタの RQCP ビットをセットします。送信の結果は、CAN_TSR レジスタの TXOK、ALST、および TERR ビットによって示されます。

図 324. 送信メールボックスの状態



28.7.2 タイムトリガ通信モード

このモードでは、CAN ハードウェアの内部カウンタがアクティブになり、このカウンタを使用して受信／送信メールボックスの CAN_RDTxR／CAN_TDTxR レジスタに格納されるタイムスタンプ値を生成します。内部カウンタは、CAN ビット時間ごとにインクリメントされます（[セクション 28.7.7: ビットタイミング](#)を参照）。内部カウンタは、送受信ともに、フレーム開始（SOF）ビットのサンプルポイントでキャプチャされます。

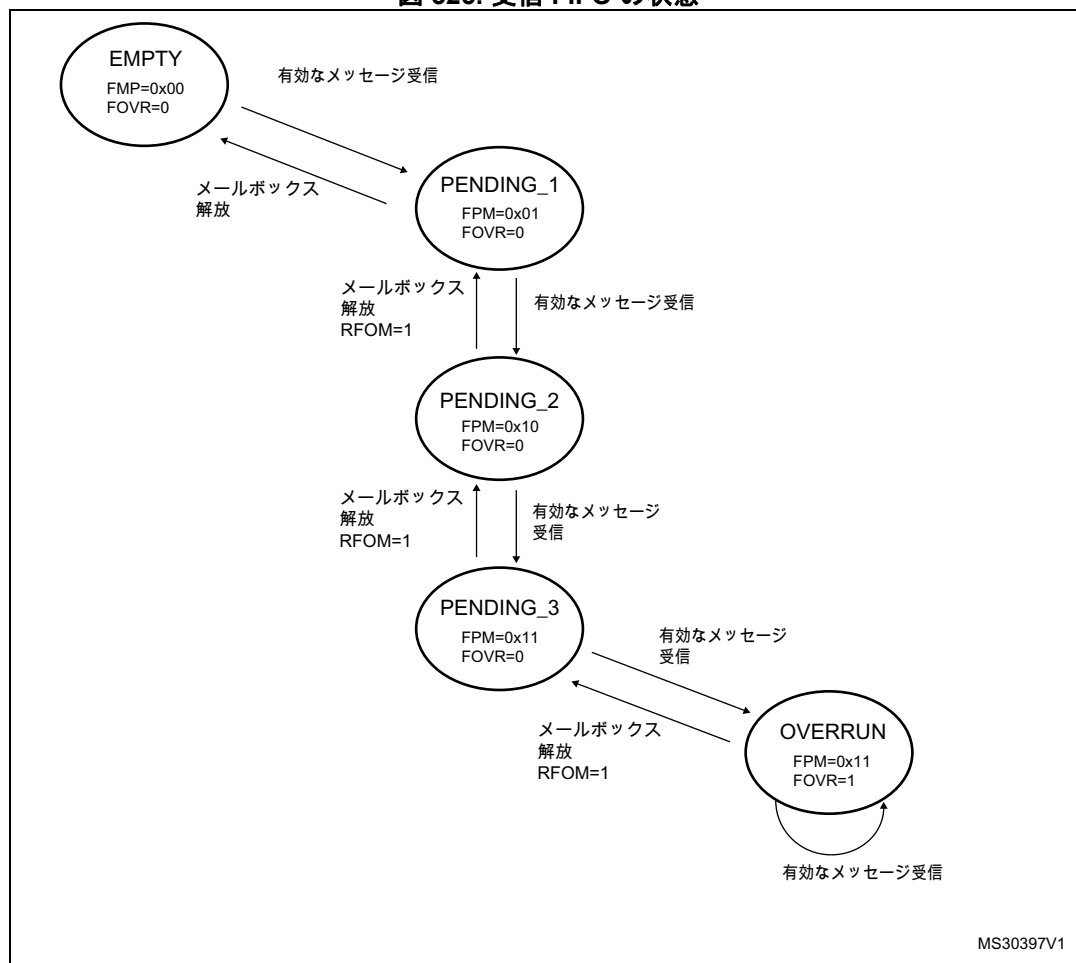
28.7.3 受信処理

CAN メッセージの受信には、3 つのメールボックスが FIFO として用意されています。CPU 負荷を軽減し、ソフトウェアを単純化し、データの一貫性を保証するために、FIFO の管理はすべてハードウェアによって行われます。アプリケーションは、FIFO 出力メールボックスを通じて、FIFO に格納されたメッセージにアクセスします。

有効なメッセージ

受信メッセージは、CAN プロトコルに従って正しく受信され（EOF フィールドの最後から 2 番目のビットまでエラーなし）、かつ、識別子フィルタリングを正常に通過した場合に、有効であるとみなされます（[セクション 28.7.4: 識別子フィルタリング](#)を参照）。

図 325. 受信 FIFO の状態



FIFO 管理

EMPTY 状態から始まり、受信した最初の有効なメッセージが FIFO に格納されると、FIFO は **PENDING_1** 状態になります。ハードウェアは、CAN_RFR レジスタの FMP[1:0] ビットを値 01b に設定して、このイベントを通知します。このメッセージは FIFO 出力メールボックスからアクセスできます。ソフトウェアは、このメールボックスの内容を読み出し、CAN_RFR レジスタの RFOM ビットをセットすることによってメールボックスを解放します。FIFO は再び **EMPTY** 状態になります。その間に新しい有効なメッセージを受信すると、FIFO は **PENDING_1** 状態にとどまり、出力メールボックスから新しいメッセージにアクセスできます。

アプリケーションがメールボックスを解放しない場合には、次の有効なメッセージが FIFO に格納され、FIFO は **PENDING_2** 状態 (FMP[1:0] = 10b) になります。この格納プロセスは、次の有効なメッセージに対しても繰り返され、FIFO は **PENDING_3** 状態 (FMP[1:0] = 11b) になります。この時点で、メールボックスが次の有効なメッセージを格納できるように、ソフトウェアは、RFOM ビットをセットして、出力メールボックスを解放する必要があります。解放しないと、受信した次の有効なメッセージが失われます。

[セクション 28.7.5 : メッセージストレージ](#)も参照してください。

オーバーラン

FIFO が **PENDING_3** 状態 (すなわち、3 個のメールボックスが満杯) になると、次の有効なメッセージの受信で **OVERRUN** 状態となり、メッセージが失われます。ハードウェアは、CAN_RFR レジスタの FOVR ビットをセットすることによって、オーバーラン状態を通知します。どのメッセージが失われるかは、FIFO の設定に依存します。

- FIFO ロック機能が無効 (CAN_MCR レジスタの RFLM ビットがクリア) である場合、最後に FIFO に格納されたメッセージが新しい受信メッセージによって上書きされます。この場合、アプリケーションは常に最新のメッセージにアクセスできます。
- FIFO ロック機能が有効 (CAN_MCR レジスタの RFLM ビットがセット) である場合、最新のメッセージが破棄され、ソフトウェアは FIFO 内にある 3 個の最も古いメッセージにアクセスできます。

受信関連の割込み

メッセージが FIFO に格納されると、FMP[1:0] ビットが更新され、割込みリクエストが生成されます (CAN_IER レジスタの FMPIE ビットがセットされている場合)。

FIFO が満杯になると (3 番目のメッセージが格納されると)、CAN_RFR レジスタの FULL ビットがセットされ、割込みが生成されます (CAN_IER レジスタの FFIE ビットがセットされている場合)。

オーバーラン状態になると、FOVR ビットがセットされ、割込みが生成されます (CAN_IER レジスタの FOVIE ビットがセットされている場合)。

28.7.4 識別子フィルタリング

CAN プロトコルでは、メッセージの識別子は、ノードのアドレスには関連付けられておらず、メッセージの内容に関連したものとなっています。したがって、トランスミッタは、そのメッセージをすべてのレシーバにブロードキャスト (一斉送信) します。メッセージを受信すると、受信ノードは、識別子の値に基づいてこのメッセージがソフトウェアに必要なものかどうかを判断します。このメッセージが必要な場合は、SRAM にコピーされます。そうでない場合、このメッセージは、ソフトウェアによる介入なしに破棄される必要があります。

この要件を満たすため、bxCAN コントローラは 28 個の設定可能でスケーラブルなフィルタバンク (27-0) をアプリケーションに提供します。それ以外のデバイスでは、ソフトウェアに必要なメッセージのみを受信できるように、bxCAN コントローラは 14 個の設定可能でスケーラブルなフィルタバンク (13-0) をアプリケーションに提供します。このハードウェアフィルタリングによって、ソフトウェアでフィルタリングを行う場合に必要となる CPU リソースを節約できます。各フィルタバンク x は、2 個の 32 ビットレジスタ (CAN_FxR0 と CAN_FxR1) で構成されています。

スケーラブルな幅

フィルタをアプリケーションニーズに合わせて最適化/適応するため、各フィルタバンクは個別に拡大/縮小できます。フィルタのスケールに応じて、フィルタバンクは以下のフィルタを提供します。

- 1 個の 32 ビットフィルタ : STDID[10:0]、EXTID[17:0]、IDE、および RTR ビット用。
- 2 個の 16 ビットフィルタ : STDID[10:0]、RTR、IDE、および EXTID[17:15] ビット用。

 326を参照してください。

さらに、これらのフィルタはマスクモードまたは識別子リストモードに設定できます。

マスクモード

マスクモードでは、識別子レジスタは、識別子のどのビットを「一致が必要」として扱い、どのビットを「無視」として扱うかを指定するマスクレジスタに関連付けられます。

識別子リストモード

識別子リストモードでは、マスクレジスタが識別子レジスタとして使用されます。このように、識別子とマスクを定義する代わりに、2つの識別子を指定することで、単一識別子の数を倍にします。受信識別子のすべてのビットが、フィルタレジスタで指定されたビットと一致する必要があります。

フィルタバンクのスケールとモードの設定

フィルタバンクは、対応する CAN_FMR レジスタによって設定されます。フィルタバンクを設定するには、CAN_FAR レジスタの FACT ビットをクリアしてフィルタバンクを無効にする必要があります。フィルタのスケールは、CAN_FS1R レジスタの対応する FSCx ビットによって設定されます (図 326 を参照)。対応するマスク/識別子レジスタの識別子リストまたは識別子マスクモードは、CAN_FMR レジスタの FBMx ビットによって設定されます。

識別子のグループをフィルタするには、マスクモードでマスク/識別子レジスタを設定します。

単一識別子を選択するには、識別子リストモードでマスク/識別子レジスタを設定します。

アプリケーションで使用されないフィルタは、非アクティブのままにしておきます。

フィルタバンク内の各フィルタには、各フィルタバンクのモードとスケールに応じて 0 から最大値までの番号 (フィルタ番号) が付けられます。

フィルタの設定については図 326 を参照してください。

図 326. フィルタバンクのスケール設定 - レジスタ構造



該当フィルタインデックス

メッセージは、FIFO で受信するとアプリケーションからアクセス可能になります。通常、アプリケーションのデータは SRAM のいずれかの位置にコピーされます。データを正しい位置にコピーするには、アプリケーションが識別子によってデータを識別する必要があります。この操作を省き、SRAM へのアクセスを容易にするために、CAN コントローラは該当フィルタインデックス（Filter MatchIndex）を搭載しています。

このインデックスは、フィルタの優先順位規則に従って、メッセージとともにメールボックスに格納されます。したがって、受信した各メッセージには関連付けられた該当フィルタインデックスがあります。

該当フィルタインデックスは、次の 2 つの方法で使用できます。

- 該当フィルタインデックスと予期される値のリストを比較します。
- 該当フィルタインデックスを配列のインデックスとして使用し、データの格納先にアクセスします。

マスクされていないフィルタについては、ソフトウェアが識別子を比較する必要はありません。

フィルタがマスクされている場合、ソフトウェアは、マスクされたビットのみを比較します。

フィルタ番号のインデックス値では、フィルタバンクのアクティブ状態を考慮に入れません。さらに、FIFO ごとに 1 つずつ、2 つの独立した番号付け方式を使用します。図 327 の例を参照してください。



図 327. フィルタ番号の例

フィルタ バンク	FIFO0	フィルタ 番号	フィルタ バンク	FIFO1	フィルタ 番号
0	ID リスト (32 ビット)	0 1	2	ID マスク (16 ビット)	0 1
1	ID マスク (32 ビット)	2	4	ID リスト (32 ビット)	2 3
3	ID リスト (16 ビット)	3 4 5 6	7	非アクティブ ID リスト (16 ビット)	4 5
5	非アクティブ ID リスト (32 ビット)	7 8	8	ID マスク (16 ビット)	6 7
6	ID マスク (16 ビット)	9 10	10	非アクティブ ID リスト (16 ビット)	8 9 10 11
9	ID リスト (32 ビット)	11 12	11	ID リスト (32 ビット)	12 13
13	ID マスク (32 ビット)	13	12	ID マスク (32 ビット)	14

ID = 識別子

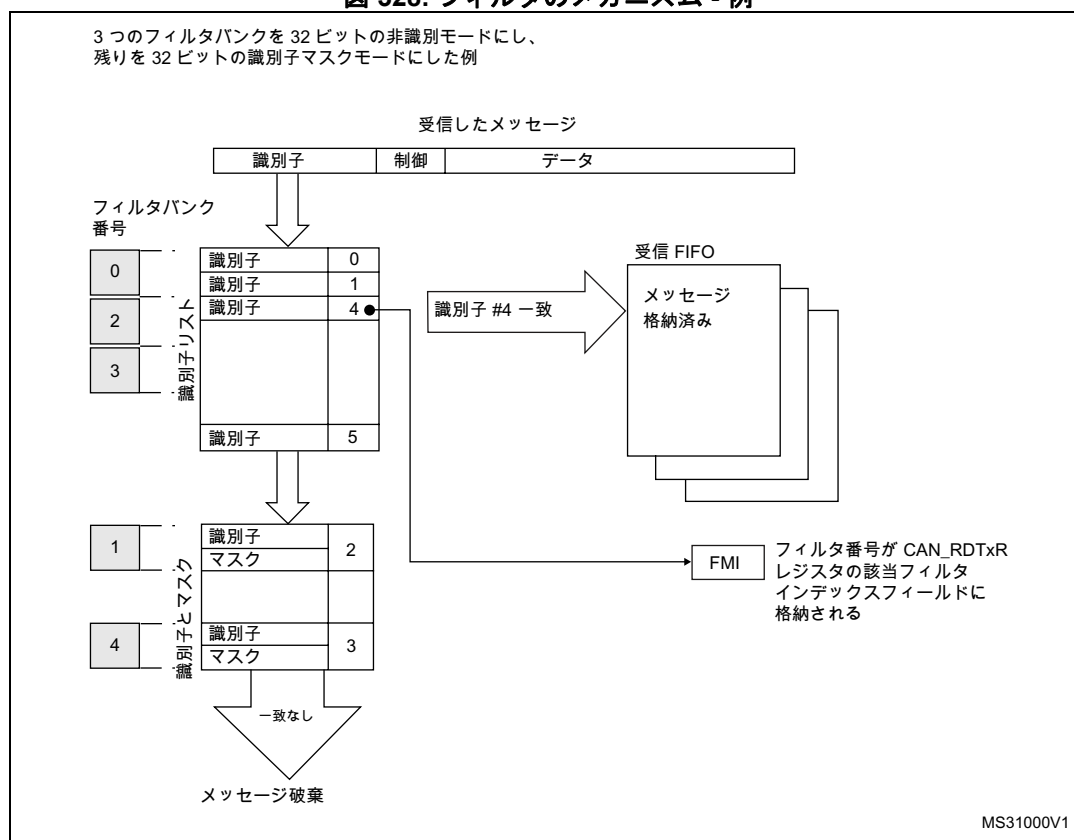
MS30399V1

フィルタの優先順位規則

フィルタの組み合わせによっては、1 つの識別子が複数のフィルタを正常に通過することがあります。この場合、受信メールボックスに格納される該当フィルタ値は、次の優先順位規則に基づいて選択されます。

- 32 ビットフィルタは 16 ビットフィルタよりも優先されます。
- 同じスケールのフィルタでは、識別子マスクモードよりも識別子リストモードが優先されます。
- スケールとモードが同じフィルタでは、フィルタ番号によって優先順位が与えられます（番号が小さいほど優先順位が高くなります）。

図 328. フィルタのメカニズム - 例



上の例は bxCAN のフィルタリングの原理を示しています。メッセージを受信すると、その識別子は、まず識別子リストモードで設定されたフィルタと比較されます。一致する項目があれば、メッセージは関連する FIFO に格納され、一致したフィルタのインデックスが該当フィルタインデックスに格納されます。この例では、識別子は識別子 #4 と一致するため、メッセージの内容と FMI 2 が FIFO に格納されます。

一致する項目がなければ、受信メッセージの識別子は、次にマスクモードで設定されたフィルタと比較されます。

この識別子がフィルタ内のどの識別子とも一致しない場合、メッセージはハードウェアによって破棄され、ソフトウェアが介入することはありません。

28.7.5 メッセージストレージ

CAN メッセージに関するソフトウェアとハードウェアのインタフェースは、メールボックスによって実装されます。メールボックスには、識別子、データ、制御、ステータス、タイムスタンプ情報など、メッセージに関するすべての情報が格納されます。

送信メールボックス

ソフトウェアは、送信しようとするメッセージを空の送信メールボックスにセットアップします。送信のステータスは、ハードウェアによって CAN_TSR レジスタに示されます。

表 183. 送信メールボックスの配置

送信メールボックスのベースアドレスのオフセット	レジスタ名
0	CAN_TlRxR
4	CAN_TDTxR
8	CAN_TDLxR
12	CAN_TDHxR

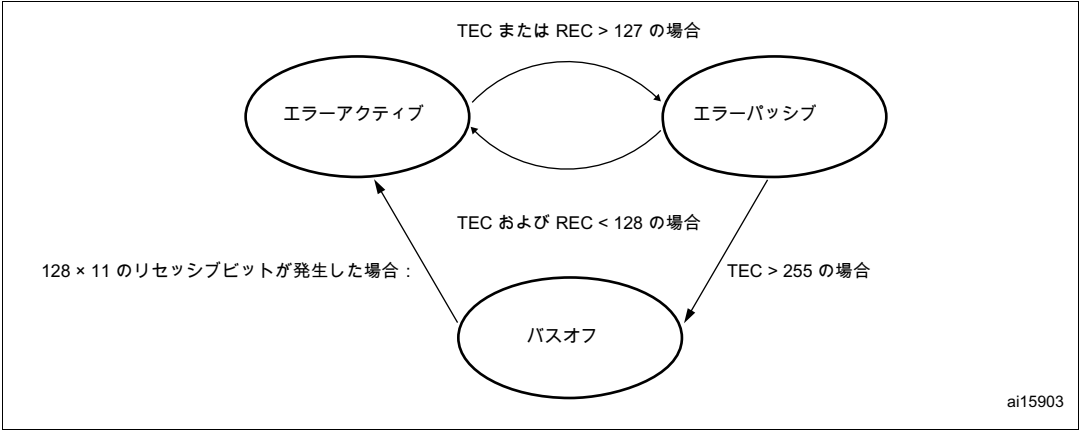
受信メールボックス

メッセージを受信すると、ソフトウェアは FIFO 出力メールボックスを通してそのメッセージにアクセスできます。ソフトウェアが読み出しなどのメッセージの処理を終えると、ソフトウェアは、CAN_RFR レジスタの RFOM ビットによって FIFO 出力メールボックスを解放し、次の受信メッセージにアクセスできるようにする必要があります。該当フィルタインデックスは、CAN_RDTxR レジスタの MFMI フィールドに格納されます。16 ビットタイムスタンプの値は、CAN_RDTxR レジスタの TIME[15:0] フィールドに格納されます。

表 184. 受信メールボックスの配置

受信メールボックスのベースアドレスのオフセット (バイト)	レジスタ名
0	CAN_RlRxR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

図 329. CAN エラー状態図



28.7.6 エラー管理

CAN プロトコルに記述されているエラー管理は、エラー条件によってインクリメント／デクリメントされる送信エラーカウンタ (CAN_ESR レジスタの TEC 値) と受信エラーカウンタ (CAN_ESR レジスタの REC 値) を使用して、すべてハードウェアによって処理されます。TEC および REC の管理の詳細については、CAN 規格を参照してください。

ソフトウェアは、この 2 つのカウンタを読み出してネットワークの安定性を判断できます。さらに、CAN ハードウェアは、現在のエラーステータスに関する詳細な情報を CAN_ESR レジスタで提供します。CAN_IER レジスタ (ERRIE ビットなど) によって、ソフトウェアは、エラー検出時の割込み生成をきわめて柔軟に設定できます。

バスオフからの復帰

TEC が 255 を超えるとバスオフ状態になります。この状態は CAN_ESR レジスタの BOFF ビットによって示されます。バスオフ状態では、bxCAN はメッセージの送受信を行うことができません。

CAN_MCR レジスタの ABOM ビットに応じて、bxCAN は、自動的に、またはソフトウェアのリクエストに基づいて、バスオフ状態から復帰します (再びエラーアクティブになります)。しかし、いずれの場合も、bxCAN は、少なくとも CAN 規格で定められた復帰シーケンス (CANRX で観測される 11 個の連続したリセッスビットの発生 128 回) の間は待機する必要があります。

ABOM がセットされると、bxCAN は、バスオフ状態に入った後で自動的に復帰シーケンスを開始します。

ABOM がクリアされている場合、ソフトウェアは、bxCAN に対して初期化モードに入って出るようにリクエストして、復帰シーケンスを開始する必要があります。

注： *初期化モードでは、bxCAN は CANRX 信号を監視しないため、復帰シーケンスを完了できません。復帰するには、bxCAN は通常モードでなければなりません。*

28.7.7 ビットタイミング

ビットタイミング回路は、シリアルバスラインを監視し、スタートビットエッジでの同期と後続エッジでの再同期によって、サンプリングとサンプリングポイントの調整を行います。

この動作は、公称ビット時間を次のように 3 つのセグメントに分割するとわかりやすく説明できます。

- **同期セグメント (SYNC_SEG) :** この時間セグメント内にビット変化が生じることが想定されます。この時間の長さは 1 タイムクオンタム ($1 \times t_q$) に固定されています。
- **ビットセグメント 1 (BS1) :** サンプルポイントの位置を定義します。これには、CAN 規格の PROP_SEG と PHASE_SEG1 が含まれます。セグメントの時間は 1~16 タイムクオンタムの間でプログラミング可能ですが、ネットワークのさまざまなノード間の周波数の差に起因する正の位相ドリフトを補正するために、自動的に延長されることがあります。
- **ビットセグメント 2 (BS2) :** 送信ポイントの位置を定義します。これは、CAN 規格の PHASE_SEG2 に相当します。セグメントの時間は 1~8 タイムクオンタムの間でプログラミング可能ですが、負の位相ドリフトを補正するために、自動的に短縮されることもあります。

再同期ジャンプ幅 (SJW) は、ビットセグメントの延長／短縮量の上限を定め、1~4 タイムクオンタムの間でプログラミング可能です。

有効なエッジとは、コントローラ自体はリセッスビットを送信しないという条件で、あるビット時間において、ドミナントからリセッスへのバスレベルの最初の遷移と定義されます。

有効なエッジが SYNC_SEG ではなく BS1 で検出された場合、サンプルポイントが遅くなるように、BS1 は最大で SJW 延長されます。

逆に、有効なエッジが SYNC_SEG ではなく BS2 で検出された場合、送信ポイントが早くなるように、BS2 は最大で SJW 短縮されます。

プログラミングエラーに対する安全対策として、ビットタイミングレジスタ (CAN_BTR) は、デバイスが STANDBY モードの場合にのみ設定可能です。

注： CAN ビットタイミングと再同期メカニズムの詳細については、ISO11898 規格を参照してください。

図 330. ビットタイミング

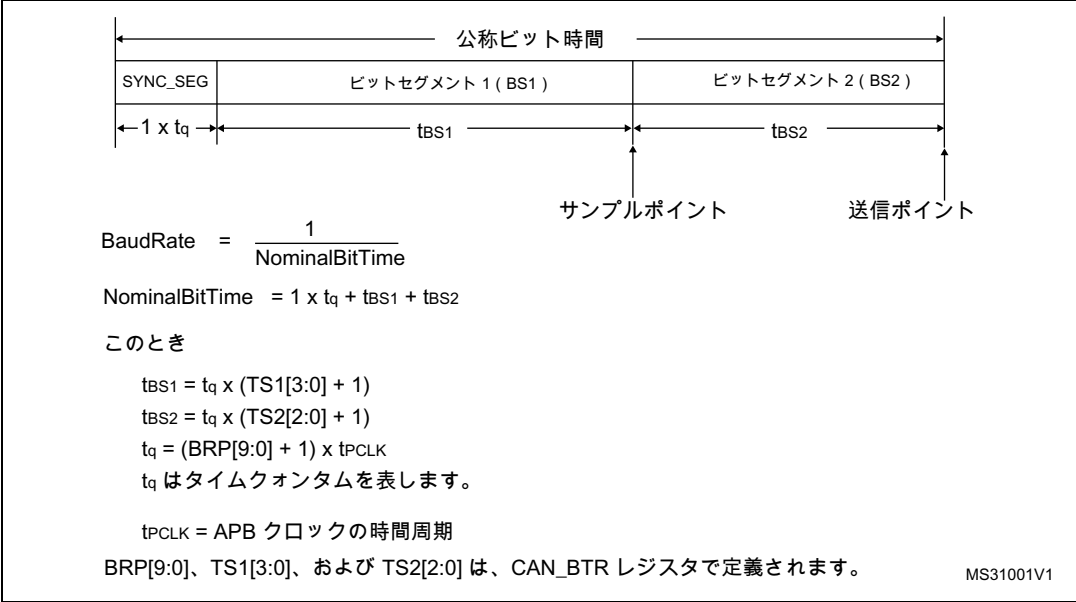
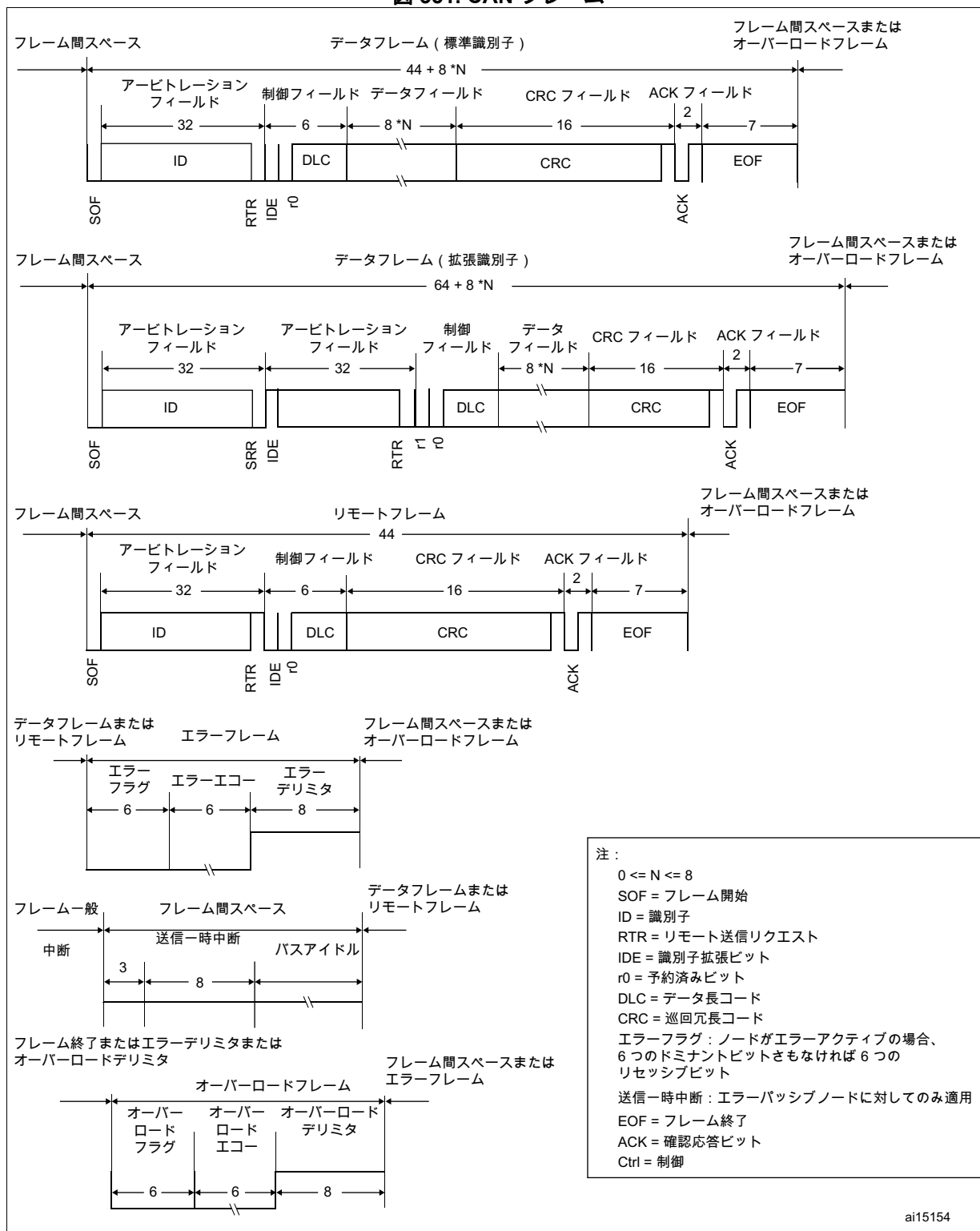


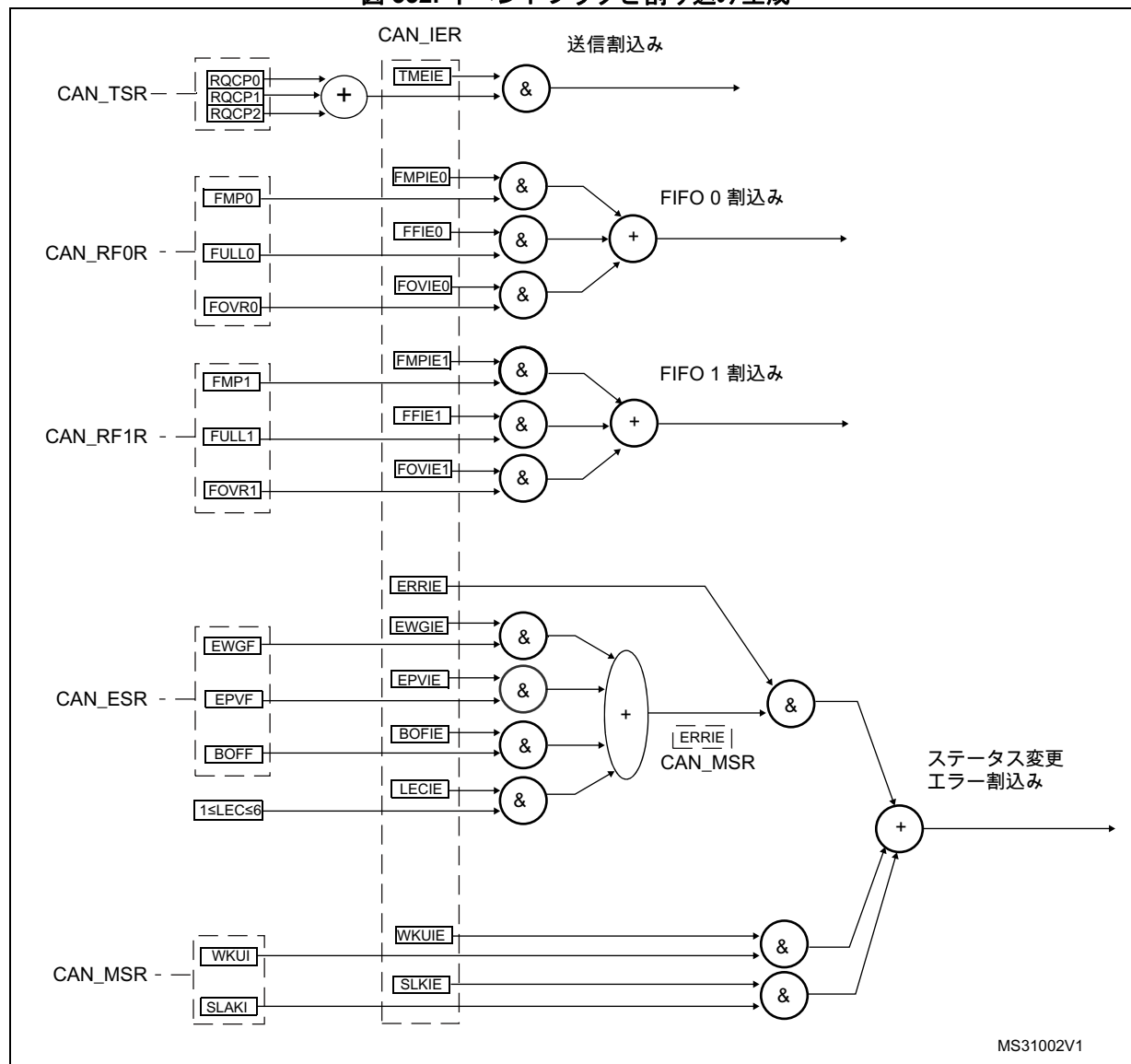
図 331. CAN フレーム



28.8 bxCAN の割込み

bxCAN には 4 つの割込みベクタが確保されています。各割込みソースは、CAN 割込み有効レジスタ (CAN_IER) によって個別に有効/無効にできます。

図 332. イベントフラグと割り込み生成



- **送信割り込み**は次のイベントによって生成できます。
 - 送信メールボックス 0 が空になり、CAN_TSR レジスタの RQCP0 ビットがセットされます。
 - 送信メールボックス 1 が空になり、CAN_TSR レジスタの RQCP1 ビットがセットされます。
 - 送信メールボックス 2 が空になり、CAN_TSR レジスタの RQCP2 ビットがセットされます。
- **FIFO 0 割り込み**は次のイベントによって生成できます。
 - 新しいメッセージの受信、CAN_RF0R レジスタの FMP0 ビットが“00”ではない場合。
 - FIFO0 の満杯状態、CAN_RF0R レジスタの FULL0 ビットがセットされる場合。
 - FIFO0 のオーバーラン状態、CAN_RF0R レジスタの FOVR0 ビットがセットされる場合。

- **FIFO 1 割り込み**は次のイベントによって生成できます。
 - 新しいメッセージの受信、CAN_RF1R レジスタの FMP1 ビットが“00”ではない場合。
 - FIFO1 の満杯状態、CAN_RF1R レジスタの FULL1 ビットがセットされる場合。
 - FIFO1 のオーバーラン状態、CAN_RF1R レジスタの FOVR1 ビットがセットされる場合。
- **エラー／ステータス変更割り込み**は次のイベントによって生成できます。
 - エラー条件：エラー条件の詳細については、CAN エラーステータスレジスタ (CAN_ESR) を参照してください。
 - ウェイクアップ条件：CAN Rx 信号で SOF が観測されたとき。
 - スリープモードへの遷移。

28.9 CAN レジスタ

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

28.9.1 レジスタのアクセス保護

特定の設定レジスタに誤ったアクセスを行うと、ハードウェアが CAN ネットワーク全体の動作を一時的に妨害することがあります。このため、ソフトウェアが CAN_BTR レジスタを変更できるのは、CAN ハードウェアが初期化モードのときに限られます。

誤ったデータを送信しても CAN ネットワークレベルでは問題は生じませんが、アプリケーションには重大な影響を与える可能性があります。ソフトウェアが送信メールボックスを変更できるのは、メールボックスが EMPTY 状態のときに限られます。図 324: 送信メールボックスの状態を参照してください。

フィルタ値を変更するには、関連するフィルタバンクを非アクティブにするか、FINIT ビットをセットします。さらに、CAN_FMR、CAN_FSR、および CAN_FFR レジスタにあるフィルタ設定 (スケール、モード、および FIFO 割当て) の変更は、CAN_FMR レジスタでフィルタ初期化モードがセットされている (FINIT = 1) 場合に限られます。

28.9.2 CAN 制御／ステータスレジスタ

レジスタの説明で使用されている略語のリストについては、セクション 1.1 を参照してください。

CAN マスタ制御レジスタ (CAN_MCR)

アドレスオフセット : 0x00
リセット値 : 0x0001 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ
rs								rw	rw	rw	rw	rw	rw	rw	rw



ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DBF** : デバッグ凍結

- 0 : デバッグ時に CAN が機能します。
- 1 : デバッグ時に CAN の送受信機能が凍結されます。受信 FIFO は、まだ正常にアクセス／制御できます。

ビット 15 **RESET** : bxCAN ソフトウェアマスタリセット

- 0 : 通常動作。
- 1 : bxCAN のマスタリセットを強制します。リセット後は、スリープモードがアクティブになります (FMP ビットと CAN_MCR レジスタはリセット値に初期化されます)。このビットは自動的に 0 にリセットされます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TTCM** : タイムトリガ通信モード

- 0 : タイムトリガ通信モードは無効です。
- 1 : タイムトリガ通信モードは有効です。

注 : [タイムトリガ通信モードの詳細については、セクション 28.7.2 : タイムトリガ通信モードを参照してください。](#)

ビット 6 **ABOM** : 自動バスオフ管理

- このビットは、バスオフ状態から出るときの CAN ハードウェアの動作を制御します。
 - 0 : 11 個のリセシブビットが 128 回観測され、ソフトウェアが CAN_MCR レジスタの INRQ ビットをまずセットし、クリアすると、ソフトウェアリクエストでバスオフ状態から出ます。
 - 1 : 11 個のリセシブビットが 128 回観測されると、ハードウェアによって自動的にバスオフ状態から出ます。
- バスオフ状態の詳細については、[セクション 28.7.6 : エラー管理](#)を参照してください。

ビット 5 **AWUM** : 自動ウェイクアップモード

- このビットは、スリープモードでのメッセージ受信時の CAN ハードウェアの動作を制御します。
 - 0 : ソフトウェアリクエストで CAN_MCR レジスタの SLEEP ビットをクリアしてスリープモードから出ます。
 - 1 : CAN メッセージの検出時にハードウェアによって自動的にスリープモードから出ます。
- CAN_MCR レジスタの SLEEP ビットと CAN_MSR レジスタの SLAK ビットはハードウェアによってクリアされます。

ビット 4 **NART** : 自動再送信禁止

- 0 : CAN ハードウェアは、CAN 規格に基づいてメッセージを正常に送信するまで、メッセージを自動的に再送信します。
- 1 : 送信結果 (正常、エラー、アービトレーションロスト) にかかわらず、メッセージは 1 回だけ送信されます。

ビット 3 **RFLM** : 受信 FIFO ロックモード

- 0 : 受信 FIFO はオーバーラン時にロックされません。受信 FIFO が満杯になると、次の受信メッセージは前のメッセージに上書きされます。
- 1 : 受信 FIFO はオーバーランに対してロックされます。受信 FIFO が満杯になると、次の受信メッセージは破棄されます。

ビット 2 **TXFP** : 送信 FIFO 優先順位

このビットは、複数のメールボックスが同時にペンディング状態のときの送信順序を制御します。
0 : 優先順位はメッセージの識別子によって決まります。
1 : 優先順位はリクエスト順に (時系列に) 決まります。

ビット 1 **SLEEP** : スリープモードリクエスト

このビットは、CAN ハードウェアがスリープモードに入るように要求するために、ソフトウェアによってセットされます。現在の CAN 動作 (CAN フレームの送受信) が完了すると、すぐにスリープモードに入ります。
ソフトウェアによってこのビットがクリアされると、スリープモードを終了します。
AWUM ビットがセットされていて、CAN Rx 信号で SOF ビットが検出されると、このビットはハードウェアによってクリアされます。
このビットはリセット後にセットされます。CAN はスリープモードで開始します。

ビット 0 **INRQ** : 初期化リクエスト

ソフトウェアによってこのビットがクリアされると、ハードウェアが通常モードに切り替わります。Rx 信号で 11 個の連続したリセシブビットが観測されると、CAN ハードウェアが同期し、送受信の準備ができます。ハードウェアは、CAN_MSR レジスタの INAK ビットをクリアすることによって、このイベントを通知します。
このビットは、CAN ハードウェアに初期化モードに入るようにリクエストするために、ソフトウェアによってセットされます。ソフトウェアが INRQ ビットをセットすると、CAN ハードウェアは現在の CAN 動作 (送受信) が完了するのを待ってから、初期化モードに入ります。ハードウェアは、CAN_MSR レジスタの INAK ビットをセットすることによって、このイベントを通知します。

CAN マスタステータスレジスタ (CAN_MSR)

アドレスオフセット : 0x04
リセット値 : 0x0000 0C02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **RX** : CAN Rx 信号

CAN_RX ピンの実際の値を監視します。

ビット 10 **SAMP** : 最終サンプルポイント

最後のサンプルポイントにおける RX の値 (受信した現在のビット値) です。

ビット 9 **RXM** : 受信モード

CAN ハードウェアは、現在はレシーバです。

ビット 8 **TXM** : 送信モード

CAN ハードウェアは、現在はトランスミッタです。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **SLAKI** : スリープ確認割込み

SLKIE=1 の場合、このビットは、bxCAN がスリープモードに入ったことを通知するために、ハードウェアによってセットされます。CAN_IER レジスタの SLKIE ビットがセットされている場合にこのビットがセットされると、ステータス変更割込みが生成されます。
SLAK がクリアされると、このビットはソフトウェアまたはハードウェアによってクリアされます。

注: *SLKIE=0 の場合、SLAKI ビットのポーリングはできません。この場合、SLAK ビットはポーリングできません。*

ビット 3 **WKUI** : ウェイクアップ割り込み

このビットは、CAN ハードウェアがスリープモードのときに SOF ビットが検出されたことを通知するために、ハードウェアによってセットされます。CAN_IER レジスタの WKUIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。
このビットはソフトウェアによってクリアされます。

ビット 2 **ERRI** : エラー割込み

エラー検出時に CAN_ESR のビットがセットされており、CAN_IER で対応する割込みが有効になっている場合、このビットはハードウェアによってセットされます。CAN_IER レジスタの ERRIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。
このビットはソフトウェアによってクリアされます。

ビット 1 **SLAK** : スリープ確認応答

このビットは、ハードウェアによってセットされ、CAN ハードウェアがスリープモードにあることをソフトウェアに知らせます。このビットは、ソフトウェアからのスリープモードリクエスト (CAN_MCR レジスタの SLEEP ビットをセット) に対する確認応答になります。
このビットは、CAN ハードウェアがスリープモードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセシブビットを観測する必要があります。

注: *CAN_MCR レジスタの SLEEP ビットがクリアされると、スリープモードから出るプロセスがトリガされます。SLEEP ビットのクリアについては、CAN_MCR レジスタの AWUM ビットの説明を参照してください。*

ビット 0 **INAK** : 初期化確認応答

このビットは、CAN ハードウェアが初期化モードにあることをソフトウェアに知らせるために、ハードウェアによってセットされます。このビットは、ソフトウェアからの初期化リクエスト (CAN_MCR レジスタの INRQ ビットをセット) に対する確認応答になります。
このビットは、CAN ハードウェアが初期化モードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセシブビットを観測する必要があります。

CAN 送信ステータスレジスタ (CAN_TSR)

アドレスオフセット : 0x08
リセット値 : 0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

ビット 31 LOW2 : メールボックス 2 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 2 が最低優先順位の時、このビットがハードウェアによってセットされます。

ビット 30 LOW1 : メールボックス 1 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 1 が最低優先順位の時、このビットがハードウェアによってセットされます。

ビット 29 LOW0 : メールボックス 0 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 0 が最低優先順位の時、このビットがハードウェアによってセットされます。

注: ペンディング状態のメールボックスが1 つしかないとき、LOW[2:0] ビットは0 にセットされません。

ビット 28 TME2 : 送信メールボックス 2 エンプティ

このビットは、メールボックス 2 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 27 TME1 : 送信メールボックス 1 エンプティ

このビットは、メールボックス 1 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 26 TME0 : 送信メールボックス 0 エンプティ

このビットは、メールボックス 0 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 25:24 CODE[1:0] : メールボックスコード

少なくとも 1 個の送信メールボックスが空き状態の場合、このコード値は、次の空き状態の送信メールボックスの番号になります。
すべての送信メールボックスが待ち状態の場合、このコード値は、最低優先順位の送信メールボックスの番号になります。

ビット 23 ABRQ2 : メールボックス 2 アポートリクエスト

対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 22:20 予約済みであり、リセット値に保持する必要があります。**ビット 19 TERR2** : メールボックス 2 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 18 ALST2 : メールボックス 2 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 17 TXOK2 : メールボックス 2 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0: 前回の送信は失敗しました。

1: 前回の送信は成功しました。

このビットは、メールボックス 2 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。 [図 324](#) を参照してください。

ビット 16 RQCP2 : メールボックス 2 リクエスト完了

最後のリクエスト (送信またはアポート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN_TMD2R レジスタの TXRQ2 ビットをセット)。

このビットをクリアするとメールボックス 2 のすべてのステータスビット (TXOK2、ALST2、および TERR2) がクリアされます。

ビット 15 **ABRQ1** : メールボックス 1 アポートリクエスト

対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **TERR1** : メールボックス 1 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 10 **ALST1** : メールボックス 1 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 9 **TXOK1** : メールボックス 1 送信OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 324 を参照してください。

ビット 8 **RQCP1** : メールボックス 1 リクエスト完了

最後のリクエスト (送信またはアポート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN_TI1R レジスタの TXRQ1 ビットをセット)。

このビットをクリアするとメールボックス 1 のすべてのステータスビット (TXOK1、ALST1、および TERR1) がクリアされます。

ビット 7 **ABRQ0** : メールボックス 0 アポートリクエスト

対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TERR0** : メールボックス 0 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 2 **ALST0** : メールボックス 0 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 1 **TXOK0** : メールボックス 0 送信OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 324 を参照してください。

ビット 0 **RQCP0** : メールボックス 0 リクエスト完了

最後のリクエスト (送信またはアポート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN_TI0R レジスタの TXRQ0 ビットをセット)。

このビットをクリアするとメールボックス 0 のすべてのステータスビット (TXOK0、ALST0、および TERR0) がクリアされます。

CAN 受信 FIFO 0 レジスタ (CAN_RF0R)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]	
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RFOM0** : FIFO 0 出力メールボックスの解放

FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。

このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

ビット 4 **FOVR0** : FIFO 0 オーバーラン

このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。

ビット 3 **FULL0** : FIFO 0 フル

FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **FMP0[1:0]** : FIFO 0 メッセージペンディング

これらのビットは、受信 FIFO 内でペンディング状態になっているメッセージの数を示します。
FMP は、ハードウェアが FIFO に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM0 ビットをセットして出力メールボックスを解放するたびに減少します。

CAN 受信 FIFO 1 レジスタ (CAN_RF1R)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]	
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RFOM1** : FIFO 1 出力メールボックスの解放

FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。

このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

ビット 4 **FOVR1** : FIFO 1 オーバーラン

このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 3 **FULL1** : FIFO 1 フル

FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **FMP1[1:0]** : FIFO 1 メッセージペンディング

これらのビットは、受信 FIFO1 内でペンディング状態になっているメッセージの数を示します。

FMP1 は、ハードウェアが FIFO1 に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM1 ビットをセットして出力メールボックスを解放するたびに減少します。

CAN 割り込み有効レジスタ (CAN_IER)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Res.	Res.	Res.	LEC IE	BOF IE	EPV IE	EWG IE	Res.	FOV IE1	FF IE1	FMP IE1	FOV IE0	FF IE0	FMP IE0	TME IE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **SLKIE** : スリープ割り込み有効

0 : SLAKI ビットがセットされると、割り込みは生成されません。

1 : SLAKI ビットがセットされると、割り込みが生成されます。

ビット 16 **WKUIE** : ウェイクアップ割り込み有効

0 : WKUI ビットがセットされると、割り込みは生成されません。

1 : WKUI ビットがセットされると、割り込みが生成されます。

ビット 15 **ERRIE** : エラー割り込みイネーブル

0 : CAN_ESR レジスタでエラー条件がペンディング状態のとき、割り込みは生成されません。

1 : CAN_ESR レジスタでエラー条件がペンディング状態のとき、割り込みが生成されます。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。



ビット 11 **LECIE** : 最終エラーコード割込み有効

0 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされるとき、ERRI ビットはセットされません。

1 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされると、ERRI ビットがセットされます。

ビット 10 **BOFIE** : バスオフ割込み有効

0 : BOFF がセットされるとき、ERRI ビットはセットされません。

1 : BOFF がセットされると、ERRI ビットがセットされます。

ビット 9 **EPVIE** : エラーパッシブ割込み有効

0 : EPVF がセットされるとき、ERRI ビットはセットされません。

1 : EPVF がセットされると、ERRI ビットがセットされます。

ビット 8 **EWGIE** : エラー警告割込み有効

0 : EWGF がセットされるとき、ERRI ビットはセットされません。

1 : EWGF がセットされると、ERRI ビットがセットされます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FOVIE1** : FIFO オーバーラン割込み有効

0 : FOVR がセットされるとき、割込みは生成されません。

1 : FOVR がセットされると、割込みが生成されます。

ビット 5 **FFIE1** : FIFO フル割込み有効

0 : FULL ビットがセットされるとき、割込みは生成されません。

1 : FULL ビットがセットされると、割込みが生成されます。

ビット 4 **FMPIE1** : FIFO メッセージペンディング割込み有効

0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。

1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。

ビット 3 **FOVIE0** : FIFO オーバーラン割込み有効

0 : FOVR ビットがセットされるとき、割込みは生成されません。

1 : FOVR ビットがセットされると、割込みが生成されます。

ビット 2 **FFIE0** : FIFO フル割込み有効

0 : FULL ビットがセットされるとき、割込みは生成されません。

1 : FULL ビットがセットされると、割込みが生成されます。

ビット 1 **FMPIE0** : FIFO メッセージペンディング割込み有効

0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。

1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。

ビット 0 **TMEIE** : 送信メールボックスエンプティ割込み有効

0 : RQCPx ビットがセットされるとき、割込みは生成されません。

1 : RQCPx ビットがセットされると、割込みが生成されます。

注 : [セクション 28.8 : bxCAN の割込み](#)を参照してください。

CAN エラーステータスレジスタ (CAN_ESR)

アドレスオフセット : 0x18
リセット値 : 0x0000 0000

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

ビット 31:24 **REC[7:0]** : 受信エラーカウンタ

CAN プロトコルの障害隔離メカニズムの実装部です。受信時にエラーがあった場合、このカウンタは、CAN 規格に定められたエラー条件に応じて、1 または 8 だけインクリメントされます。受信が正常に行われるたびに、このカウンタは 1 だけデクリメントされます (カウンタの値が 128 を超えた場合は、120 にリセットされます)。カウンタ値が 127 を超えると、CAN コントローラはエラーパッシブ状態に入ります。

ビット 23:16 **TEC[7:0]** : 9 ビット送信エラーカウンタの最下位バイト

CAN プロトコルの障害隔離メカニズムの実装部です。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **LEC[2:0]** : 最終エラーコード

このフィールドはハードウェアによってセットされ、最後に CAN バスで検出されたエラーのエラー条件を示すコードを保持しています。メッセージがエラーなしで転送 (受信または送信) されている場合には、このフィールドは“0”にクリアされます。
LEC[2:0] ビットは、ソフトウェアによって値 0b111 にセットできます。ビットはハードウェアによって更新され、現在の通信ステータスを示します。

- 000 : エラーなし
- 001 : スタッエラー
- 010 : フォームエラー
- 011 : ACK エラー
- 100 : ビットリセッショエラー
- 101 : ビットドミナントエラー
- 110 : CRC エラー
- 111 : ソフトウェアによってセットされます

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **BOFF** : バスオフフラグ

このビットは、バスオフ状態に入るとハードウェアによってセットされます。バスオフ状態に入るのは、TEC オーバーフローのとき (255 を超えたとき) です。[セクション 28.7.6 \(920 ページ\)](#) を参照してください。

ビット 1 **EPVF** : エラーパッシブフラグ

このビットは、エラーパッシブ限界に達したときに (受信エラーカウンタまたは送信エラーカウンタ > 127)、ハードウェアによってセットされます。

ビット 0 **EWGF** : エラー警告フラグ

このビットは、警告限界に達したときに、ハードウェアによってセットされます (受信エラーカウンタまたは送信エラーカウンタ ≥ 96)。



CAN ビットタイミングレジスタ (CAN_BTR)

アドレスオフセット : 0x1C

リセット値 : 0x0123 0000

このレジスタは、CAN ハードウェアが初期化モードのときに、ソフトウェアによってのみアクセスできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]			
r/w	r/w					r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]									
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **SILM** : サイレントモード (デバッグ用)

0 : 通常動作

1 : サイレントモード

ビット 30 **LBKM** : ループバックモード (デバッグ用)

0 : ループバックモードは無効です。

1 : ループバックモードは有効です。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。

ビット 25:24 **SJW[1:0]** : 再同期ジャンプ幅

これらのビットは、再同期を行うために CAN ハードウェアがビットを伸縮できる最大タイムクオンタム数を定義します。

$$t_{RJW} = t_q \times (SJW[1:0] + 1)$$

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **TS2[2:0]** : 時間セグメント 2

これらのビットは、時間セグメント 2 に含まれるタイムクオンタム数を定義します。

$$t_{BS2} = t_q \times (TS2[2:0] + 1)$$

ビット 19:16 **TS1[3:0]** : 時間セグメント 1

これらのビットは、時間セグメント 1 に含まれるタイムクオンタム数を定義します。

$$t_{BS1} = t_q \times (TS1[3:0] + 1)$$

ビットタイミングの詳細については、[セクション 28.7.7 : ビットタイミング \(920 ページ\)](#) を参照してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **BRP[9:0]** : ボーレートプリスケアラ

これらのビットは、タイムクオンタムの長さを定義します。

$$t_q = (BRP[9:0] + 1) \times t_{PCLK}$$

28.9.3 CAN メールボックスレジスタ

ここでは、送受信メールボックスのレジスタについて説明します。詳細なレジスタ配置については、[セクション 28.7.5 : メッセージストレージ \(918 ページ\)](#) を参照してください。

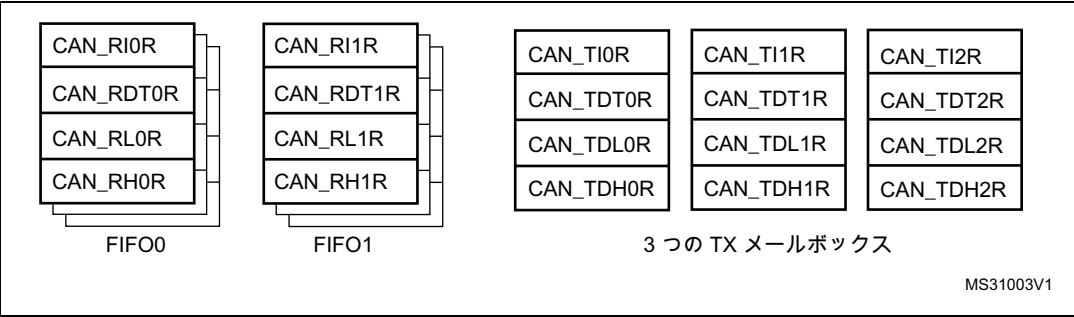
送信メールボックスと受信メールボックスのレジスタは同じですが、次の点が異なります。

- CAN_RDTxR レジスタの FMI フィールド。
- 受信メールボックスは常に書き込み保護されます。
- 送信メールボックスは空のとき (CAN_TSR レジスタの対応する TME ビットがセットされているとき) のみ、書き込み可能です。

送信メールボックスは 3 つ、受信メールボックスは 2 つあります。各受信メールボックスでは、3 種類の深さのレベルの FIFO にアクセスでき、FIFO 内の最も古い受信メッセージにのみアクセスできます。

各メールボックスは 4 つのレジスタから構成されています。

図 333. CAN メールボックスレジスタ



CAN 送信メールボックス識別子レジスタ (CAN_TIxR) (x=0..2)

アドレスオフセット : 0x180、0x190、0x1A0

リセット値 : 0xFFFF XXXX (ただし、ビット 0、TXRQ = 0 を除く)

メールボックスが送信待ち (TMEx リセット) のとき、すべての送信レジスタは書き込み保護されます。

このレジスタは、送信リクエスト制御 (ビット 0、リセット値 0) も実装します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子

IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。

ビット 20:3 **EXID[17:0]** : 拡張識別子

拡張識別子の最下位ビットです。

ビット 2 **IDE** : 識別子拡張

このビットは、メールボックス内のメッセージの識別子のタイプを定義します。

0 : 標準識別子。

1 : 拡張識別子。

ビット 1 **RTR** : リモート送信リクエスト

0 : データフレーム

1 : リモートフレーム

ビット 0 **TXRQ** : 送信メールボックスリクエスト

対応するメールボックスに送信を要求するために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

CAN メールボックスデータ長制御/タイムスタンプレジスタ (CAN_TDTxR) (x = 0~2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x184、0x194、0x1A4

リセット値 : 0xFFFF XXXX

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

ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ

このフィールドは、SOF 送信時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TGT** : 送信グローバルタイム

このビットは、ハードウェアがタイムトリガ通信モードにある (CAN_MCR レジスタの TTCM ビットがセットされている) ときのみにアクティブとなります。

0 : タイムスタンプ TIME[15:0] は送信されません。

1 : タイムスタンプ TIME[15:0] の値は、8 バイトメッセージの最後の 2 データバイトで送信されます。このとき、データバイト 7 の TIME[7:0] とデータバイト 6 の TIME[15:8] は、CAN_TDHxR[31:16] レジスタ (DATA6[7:0]、DATA7[7:0]) に書き込まれたデータを置換します。これらの 2 バイトを CAN バス経由で送信するには、DLC を 8 としてプログラミングする必要があります。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **DLC[3:0]** : データ長コード

このフィールドは、データフレーム内のデータバイト数、またはリモートフレームリクエストのデータバイト数を定義します。

メッセージは、DLC フィールドの値に応じて、0~8 個のデータバイトを含むことができます。

CAN メールボックスデータ下位レジスタ (CAN_TDLxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x188、0x198、0x1A8

リセット値 : 0XXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **DATA3[7:0]** : データバイト 3
メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2
メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1
メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0
メッセージのデータバイト 0 です。
メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

CAN メールボックスデータ上位レジスタ (CAN_TDHxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x18C、0x19C、0x1AC

リセット値 : 0XXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:24 **DATA7[7:0]** : データバイト 7
メッセージのデータバイト 7 です。
注: このメッセージの TGT と TTCM がアクティブな場合、DATA7 と DATA6 はタイムスタンプの値で置換されます。
- ビット 23:16 **DATA6[7:0]** : データバイト 6
メッセージのデータバイト 6 です。
- ビット 15:8 **DATA5[7:0]** : データバイト 5
メッセージのデータバイト 5 です。
- ビット 7:0 **DATA4[7:0]** : データバイト 4
メッセージのデータバイト 4 です。

CAN 受信 FIFO メールボックス識別子レジスタ (CAN_RlRxR) (x=0..1)

アドレスオフセット : 0x1B0、0x1C0
リセット値 : 0xFFFF XXXX
すべての受信レジスタは書き込み保護されます。

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

- ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子
IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。
- ビット 20:3 **EXID[17:0]** : 拡張識別子
拡張識別子の最下位ビットです。
- ビット 2 **IDE** : 識別子拡張
このビットは、メールボックス内のメッセージの識別子のタイプを定義します。
0 : 標準識別子。
1 : 拡張識別子。
- ビット 1 **RTR** : リモート送信リクエスト
0 : データフレーム
1 : リモートフレーム
- ビット 0 予約済みであり、リセット値に保持する必要があります。



CAN 受信 FIFO メールボックスデータ長制御/タイムスタンプレジスタ (CAN_RDTxR) (x = 0..1)

アドレスオフセット : 0x1B4、0x1C4
リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

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

- ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ
このフィールドは、SOF 検出時にキャプチャされた 16 ビットタイマの値を含みます。
- ビット 15:8 **FMI[7:0]** : 該当フィルタインデックス
このレジスタは、メールボックスに格納されたメッセージが通過したフィルタの索引を含みます。識別子フィルタリングの詳細については、[セクション 28.7.4 : 識別子フィルタリング \(914 ページ\)](#) の**該当フィルタインデックス**項を参照してください。
- ビット 7:4 予約済みであり、リセット値に保持する必要があります。
- ビット 3:0 **DLC[3:0]** : データ長コード
このフィールドは、データフレームに格納されるデータバイト数 (0~8) を定義します。リモートフレームリクエストの場合、このフィールドは 0 です。

CAN 受信 FIFO メールボックスデータ下位レジスタ (CAN_RDLxR) (x=0..1)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x1B8、0x1C8

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA3[7:0]** : データバイト 3
メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2
メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1
メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0
メッセージのデータバイト 0 です。
メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

CAN 受信 FIFO メールボックスデータ上位レジスタ (CAN_RDHxR) (x=0..1)

アドレスオフセット : 0x1BC、0x1CC

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA7[7:0]** : データバイト 7
メッセージのデータバイト 3 です。

ビット 23:16 **DATA6[7:0]** : データバイト 6
メッセージのデータバイト 2 です。

ビット 15:8 **DATA5[7:0]** : データバイト 5
メッセージのデータバイト 1 です。

ビット 7:0 **DATA4[7:0]** : データバイト 4
メッセージのデータバイト 0 です。

28.9.4 CAN フィルタレジスタ

CAN フィルタマスタレジスタ (CAN_FMR)

アドレスオフセット : 0x200
リセット値 : 0x2A1C 0E01

このレジスタのすべてのビットは、ソフトウェアによってセット/クリアされます。

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

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **CANSB[5:0]** : CAN 開始バンク
これらのビットは、ソフトウェアによってセット/クリアされます。CAN インタフェース (スレーブ) の開始バンクを 1 から 27 の範囲で定義します。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **FINIT** : フィルタ初期化モード
フィルタバンクの初期化モード
0 : アクティブフィルタモード。
1 : フィルタの初期化モード。

CAN フィルタモードレジスタ (CAN_FM1R)

アドレスオフセット : 0x204

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 注: **FBMx**: フィルタモード [図 326: フィルタバンクのスケール設定 - レジスタ構造 \(916 ページ\)](#) を参照してください。

フィルタ x のレジスタのモードです。
0: フィルタバンク x の 2 つの 32 ビットレジスタは識別子マスクモードになっています。
1: フィルタバンク x の 2 つの 32 ビットレジスタは識別子リストモードになっています。
注: ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

CAN フィルタスケールレジスタ (CAN_FS1R)

アドレスオフセット : 0x20C

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FSCx**: フィルタスケール設定
これらのビットは、フィルタ 13-0 のスケール設定を定義します。
0: デュアル 16 ビットスケール設定
1: シングル 32 ビットスケール設定
注: ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

注: [図 326: フィルタバンクのスケール設定 - レジスタ構造 \(916 ページ\)](#) を参照してください。

CAN フィルタ FIFO 割当てレジスタ (CAN_FFA1R)

アドレスオフセット : 0x214
リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FFAx** : フィルタ x のフィルタ FIFO 割り当て
このフィルタを通過するメッセージは、特定の FIFO に保存されます。
0 : フィルタが FIFO 0 に割り当てられます。
1 : フィルタが FIFO 1 に割り当てられます。

注 : ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

CAN フィルタアクティベーションレジスタ (CAN_FA1R)

アドレスオフセット : 0x21C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FACT27	FACT26	FACT25	FACT24	FACT23	FACT22	FACT21	FACT20	FACT19	FACT18	FACT17	FACT16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FACT15	FACT14	FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FACTx** : アクティブなフィルタ
ソフトウェアがこのビットをセットしてフィルタ x をアクティブにします。フィルタ x レジスタ (CAN_FxR[0:7]) を変更するには、FACTx ビットをクリアするか、CAN_FMR レジスタの FINIT ビットをセットする必要があります。
0 : フィルタ x はアクティブではありません。
1 : フィルタ x はアクティブです。

注 : ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

CAN フィルタバンク i レジスタ x (CAN_FiRx) (i = 0..27、x = 1、2)

アドレスオフセット : 0x240 から 0x31C
リセット値 : 0xFFFF XXXX

28 個のフィルタバンク (i= 0 ~27) が存在します。各フィルタバンク i は、2 個の 32 ビットレジスタ (CAN_FiR[2:1]) から構成されています。

このレジスタを変更できるのは、CAN_FAxR レジスタの FACTx ビットがクリアされている場合か、CAN_FMR レジスタの FINIT ビットがセットされている場合に限られます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

すべての設定において :

ビット 31:0 **FB[31:0]** : フィルタビット

識別子

レジスタの各ビットは、予期される識別子の対応するビットのレベルを指定します。
0 : ドミナントビットが予期されます。
1 : リセシブビットが予期されます。

マスク

レジスタの各ビットは、関連する識別子レジスタのビットが、予期される識別子の対応するビットに一致する必要があるかどうかを指定します。
0 : 無視 (このビットは比較に使用されません)
1 : 一致が必要 (受信識別子のビットのレベルは、フィルタの対応する識別子レジスタで指定されたレベルと同じでなければなりません)

注 : フィルタのスケールとモードの設定に応じて、各レジスタの機能は異なることがあります。フィルタ配置、機能の説明、およびマスクレジスタの関連については、[セクション 28.7.4 : 識別子フィルタリング \(914 ページ\)](#) を参照してください。

マスクモードにおけるマスク／識別子レジスタのビット配置は、識別子リストモードの場合と同じです。

フィルタバンクのレジスタ配置／アドレスについては、[表 185 \(945 ページ\)](#) を参照してください。



28.9.5 bxCAN レジスタマップ

レジスタ境界アドレスについては、[セクション 2.2.2 \(49 ページ\)](#) を参照してください。オフセット 0x200 から 31C までのレジスタは、CAN1 にのみ存在します。

表 185. bxCAN レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x000	CAN_MCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBF	RESET	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																1	0								0	0	0	0	0	0	1	0					
0x004	CAN_MSR	Res		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RX	SAMP	RXM	TXM	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																					1	1	0	0													
0x008	CAN_TSR	LOW[2:0]			TME[2:0]			CODE[1:0]			ABRQ2			Res	Res	Res	TERR2	ALST2	TXOK2	RQCP2	ABRQ1	Res	Res	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res	Res	Res	Res						
	リセット値	0	0	0	1	1	1	0	0	0					0	0	0	0	0				0	0	0	0	0											
0x00C	CAN_RF0R	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																																					
0x010	CAN_RF1R	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																																					
0x014	CAN_IER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																0	0	0																			
0x018	CAN_ESR	REC[7:0]								TEC[7:0]								Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																					
0x01C	CAN_BTR	SILM	LBKM	Res	Res	Res	Res	SJW[1:0]			Res	TS2[2:0]			TS1[3:0]			Res	Res	Res	Res	Res	Res	BRP[9:0]														
	リセット値	0	0					0	0		0	1	0	0	0	1	1							0	0	0	0	0	0	0	0	0	0	0	0			
0x020-0x17F		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
0x180	CAN_TI0R	STID[10:0]/EXID[28:18]												EXID[17:0]																IDE	RTR	TXR0						
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0					
0x184	CAN_TDT0R	TIME[15:0]																Res	Res	Res	Res	Res	Res	Res	Res	Res	TGT	Res	Res	Res	Res	DLC[3:0]						
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									x					x	x	x	x					

表 185. bxCAN レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x188	CAN_TDL0R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x18C	CAN_TDH0R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x190	CAN_TI1R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	TXRQ			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x194	CAN_TDT1R	TIME[15:0]															Res	Res	Res	Res	Res	Res	Res	TGT	Res	Res	Res	Res	DLC[3:0]				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									x					x	x	x	x
0x198	CAN_TDL1R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x19C	CAN_TDH1R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1A0	CAN_TI2R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	TXRQ			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x1A4	CAN_TDT2R	TIME[15:0]															Res	Res	Res	Res	Res	Res	Res	TGT	Res	Res	Res	Res	DLC[3:0]				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									x					x	x	x	x
0x1A8	CAN_TDL2R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1AC	CAN_TDH2R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1B0	CAN_RI0R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	Res			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x1B4	CAN_RDT0R	TIME[15:0]															FMI[7:0]								Res	Res	Res	Res	DLC[3:0]				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					x	x	x	x	x
0x1B8	CAN_RDL0R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

表 185. bxCAN レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
0x1BC	CAN_RDH0R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0x1C0	CAN_RI1R	STID[10:0]/EXID[28:18]											EXID[17:0]															IDE	RTR	Res																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
0x1C4	CAN_RDT1R	TIME[15:0]																FMI[7:0]								Res	Res	Res	Res	DLC[3:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					x	x	x	x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0x1C8	CAN_RDL1R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0x1CC	CAN_RDH1R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0x1D0- 0x1FF		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
0x200	CAN_FMR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CANSB[5:0]								Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

表 185. bxCAN レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x240	CAN_F0R1	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x244	CAN_F0R2	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x248	CAN_F1R1	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x24C	CAN_F1R2	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
⋮ ⋮ ⋮	⋮ ⋮ ⋮	⋮ ⋮ ⋮																															
0x318	CAN_F27R1	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x31C	CAN_F27R2	FB[31:0]																															
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

29 USB On-The-Go フルスピード (OTG_FS)

29.1 概要

Portions Copyright (C) 2004, 2005 Synopsys, Inc.掲載許可取得済み。

このセクションでは、OTG_FS コントローラのアーキテクチャとプログラミングモデルについて説明します。

このセクションでは、次の略記が使用されます。

FS	フルスピード
LS	ロースピード
MAC	メディアアクセスコントローラ
OTG	On-The-Go
PFC	パケット FIFO コントローラ
PHY	物理層
USB	ユニバーサルシリアルバス
UTMI	USB 2.0 トランシーバマクロセルインタフェース (UTMI)
UTMI	USB トランシーバマクロセルインタフェース
ULPI	UTMI + ローピンインタフェース
LPM	リンク電源管理
BCD	バッテリー充電検出回路
HNP	ホストネゴシエーションプロトコル
SRP	セッションリクエストプロトコル

このセクションでは、参照文献として以下の文書が使用されています。

- USB On-The-Go Supplement, Revision 1.3
- USB On-The-Go Supplement, Revision 2.0
- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- USB 2.0 ECN 用の正誤表 : Link Power Management (LPM) - 7/2007
- バッテリー充電 (BC) 仕様 Revision 1.2

USB OTG は、デバイスとホストの両方の機能をサポートするデュアルロールデバイス (DRD) コントローラであり、*On-The-Go Supplement to the USB 2.0 Specification* に完全に準拠しています。USB OTG は、*USB 2.0 Specification* に完全に準拠したホスト専用またはデバイス専用のコントローラとして設定することも可能です。OTG_FS は以下の表 186: サポートされる OTG_FS 速度に定義される速度をサポートします。USB OTG は HNP と SRP の両方をサポートします。必要な外部デバイスは、OTG モードで使用される V_{BUS} 用のチャージポンプのみです。

表 186. サポートされる OTG_FS 速度

	HS (480 Mb/s)	FS (12 Mb/s)	LS (1.5 Mb/s)
ホストモード	-	X	X
デバイスモード	-	X	-

29.2 USB_OTG の主な機能

主な機能は、標準機能、ホストモード機能、およびデバイスモード機能の 3 つのカテゴリに分けて考えることができます。

29.2.1 標準機能

OTG_FS インタフェースの機能概要を次に示します：

- Universal Serial Bus Specification Rev 2.0 に準拠した USB-IF です。
- On-The-Go Supplement Rev 1.3 Specification に詳細が規定されているオプションの On-The-Go (OTG) プロトコルの完全サポート (PHY) を含みます。
 - A/B デバイス識別 (ID ライン) を統合的にサポートしています。
 - ホストネゴシェーションプロトコル (HNP) およびセッションリクエストプロトコル (SRP) を統合的にサポートしています。
 - OTG アプリケーションでバッテリーを節約するために、ホストで V_{BUS} をオフすることができます。
 - 内部コンパレータによって OTG に V_{BUS} のレベルを監視させることができます。
 - ホスト-ペリフェラルのダイナミックな役割切り替えをサポートします。
- 次の動作を行うようにソフトウェアでの設定が可能です。
 - SRP 対応の USB FS ペリフェラル (B デバイス)
 - SRP 対応の USB FS/LS ホスト (A デバイス)
 - USB On-The-Go フルスピードデュアルロールデバイス
- 以下の要素により FS SOF および LS Keep-alive をサポートします。
 - SOF パルス PAD 接続性
 - タイマ (TIMx) に対する SOF パルス内部接続
 - 設定可能なフレーム周期
 - 設定可能なフレームの終わりの割り込み
- USB サスペンド中のシステム停止や、デジタルコア、PHY、および DFIFO 電源管理回路など内部のクロックドメインの電源オフのような、節電機能を内蔵しています。
- 以下のような高度な FIFO 制御を備えた専用の 1.25 KB RAM を備えています。
 - RAM 空間を複数の FIFO に分割して、RAM を柔軟かつ効率的に使用できるように設定できます。
 - 各 FIFO は複数のパケットを保持できます。
 - 動的なメモリ割り当てが可能です。
 - FIFO を 2 のべき乗でないサイズに設定して、連続的なメモリ位置の使用が可能です。
- システムの介入なしで、最大 1 フレーム (1 ms) の間、最大 USB バンド幅を保証します。
- バッテリ充電仕様 Revision 1.2 で説明されているとおり、充電ポート検出をサポートしています。

29.2.2 ホストモード機能

ホストモードでの OTG_FS インタフェースの主な機能と条件は次の通りです:

- V_{BUS} 電圧発生用の外部チャージポンプを持っています。
- 最大 12 個のホストチャネル (パイプ) を持っています。各チャネルはどんなタイプの USB 転送も割り当てられるように動的に設定可能です。
- 次を保持するハードウェアスケジューラを内蔵しています。
 - 最大 12 個の割り込みに加えて、周期的ハードウェアキュー内でのアイソクロナス転送リクエスト
 - 最大 12 つの制御に加えて、非周期的ハードウェアキュー内でのバルク転送リクエスト
- USB データ RAM の効率的な使用のために、共有 Rx FIFO、周期的 Tx FIFO、および非周期的 Tx FIFO を管理します。

29.2.3 ペリフェラルモード機能

ペリフェラルモードでの OTG_FS インタフェースの主要機能を次に示します:

- 1 つの双方向制御エンドポイント 0 を持っています。
- バルク、割り込み、またはアイソクロナス転送をサポートするために、設定可能な 5 つの IN エンドポイント (EP) を持っています。
- バルク、割り込み、またはアイソクロナス転送をサポートするために、設定可能な 5 つの OUT エンドポイント (EP) を持っています。
- USB データ RAM の効率的な使用のために、共有 RX FIFO と Tx-OUT FIFO を管理します。
- アプリケーションの負荷を軽減するために、最大 6 つの専用 Tx-IN FIFO (有効な IN EP に対して 1 つずつ) を管理します。
- ソフト切断機能をサポートしています。

29.2.4 USB の分割レール

一部のパッケージオプションでは、 V_{DD} より制限された電圧仕様を持つ分割レール V_{DDUSB} があり、より低い範囲で動作できます。

この場合、 V_{DD} は V_{DDUSB} の最低機能レベル以下になるため、事前に次のことに注意してください。

ホスト専用アプリケーション:

- 事前に注意すべきことは特にありません。

V_{BUS} で電源が供給される USB2.0 ペリフェラルデバイス:

- 事前に注意すべきことは特にありません。 V_{BUS} 検出は、 V_{BUS} 状態が MCU の電源によって直接反映されるため、不要です。

その他すべてのケース (V_{BUS} 検出が必要):

- データシート (電気的特性、およびアプリケーションブロック図の付録) を参照してください。異なる範囲の V_{DD} 供給は、このケースが適用されます。

29.4.2 OTG コア

USB OTG は、外部水晶発振器を使用して、リセットおよびクロックコントローラ (RCC) から 48 MHz $\pm 0.25\%$ のクロックを受け取ります。USB クロックは、フルスピード (12 M ビット/秒) で 48 MHz ドメインを駆動するために使用され、OTG コアを設定する前に有効にされなければなりません。

CPU は、AHB ペリフェラルバスを使用して OTG コアレジスタに対する読み書きを実行します。USB イベントは 1 本の USB OTG 割り込みラインによって通知されます ([セクション 29.13: OTG_FS 割り込み](#)を参照)。

CPU は、専用の OTG 位置 (プッシュレジスタ) に 32 ビットのワードを書き込むことによって、USB を介してデータを提供します。データは USB データ RAM 内に設定された Tx データ FIFO に自動的に格納されます。各 IN エンドポイント (ペリフェラルモードの場合) または OUT チャンネル (ホストモードの場合) に対して 1 つの Tx FIFO プッシュレジスタがあります。

CPU は、専用の OTG アドレス (ポップレジスタ) から 32 ビットのワードを読み出すことによって、USB からデータを受け取ります。データは、1.25 KB の USB データ RAM 内に設定された共有 Rx FIFO から自動的に取得されます。各 OUT エンドポイントまたは IN チャンネルに対して 1 つの Rx FIFO ポップレジスタがあります。

USB プロトコル層は、シリアルインタフェースエンジン (SIE) によって駆動され、オンチップ物理層 (PHY) 内のトランシーバモジュールによって USB 上で直列化 (シリアル化) されます。

29.4.3 フルスピード OTG PHY

組み込みのフルスピード OTG PHY は、OTG FS コアによって制御され、UTMI+ バス (UTMIFS) のフルスピードサブセットを通じて USB 制御信号およびデータ信号を運びます。USB 接続の物理サポートを提供します。

フルスピード OTG PHY には、以下のコンポーネントが含まれます。

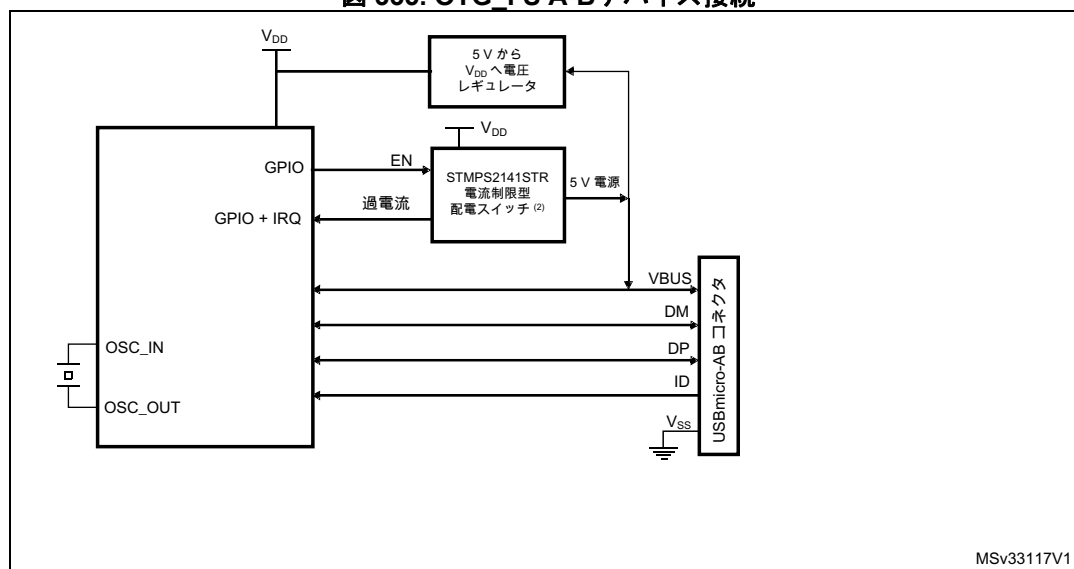
- ホストとデバイスの両方で使用される FS/LS トランシーバモジュール。シングルエンド USB ライン上で送受信を直接駆動します。
- A/B デバイス識別のための ID ラインをサンプリングするために使用される組み込みの ID プルアップ抵抗。
- デバイスの現在の役割に依存して OTG_FS コアによって制御される 組み込みの DP/DM プルアップおよびプルダウン抵抗。ペリフェラルとして、DP プルアップ抵抗を有効にし、フルスピードペリフェラル接続を知らせます。信号を送るタイミングは、 V_{BUS} が有効なレベルであると判断されてすぐです (B セッションが有効な場合)。ホストモードでは DP と DM の両方でプルダウン抵抗が有効にされます。プルアップ抵抗とプルダウン抵抗は、デバイスの役割がホストネゴシエーションプロトコル (HNP) を通じて変更されたとき、動的に切り替えられます。
- プルアップ/プルダウン抵抗 ECN 回路。DP プルアップ回路は、USB Rev2.0 に適用される抵抗の技術変更通知 (Engineering Change Notice) に従って、OTG_FS から個別に制御される 2 つの抵抗で構成されます。DP プルアップの動的なトリミングは、ノイズの低減と Tx/Rx 信号品質の向上を可能にします。
- V_{BUS} レベル検出用コンパレータ。 V_{BUS} の有効性、A-B セッションの有効性、およびセッション終了電圧閾値を検出するために使用されるヒステリシスを有します。これらは、セッションリンクエストプロトコル (SRP) の駆動、有効な起動およびセッション終了条件の検出、および USB 操作中の V_{BUS} 電源の定常的な監視のために使用されます。
- V_{BUS} パルス駆動回路。SRP 中に抵抗により V_{BUS} を充電/放電するために使用します (弱駆動)。

注意： **USB OTG FS ペリフェラルの正常動作を保証するためには、AHB 周波数を 14.2 MHz より高くする必要があります。**



29.5 OTG デュアルロールデバイス (DRD)

図 335. OTG_FS A-Bデバイス接続



1. VBUS で電源が供給されるデバイスを設定するときだけ必要な外部電圧レギュレータ
2. アプリケーションが VBUS で電源が供給されるデバイスをサポートしなければならない場合のみ必要な STPS2141STR。アプリケーションボード上で 5V が使用可能な場合、基本的なパワースイッチを使用することができます。

29.5.1 ID ラインの検出

ホストかペリフェラル（デフォルト）かの役割は、ID 入力ピンに応じて割り当てられます。ID ラインのステータスは、USB 端子への挿入時に、USB ケーブルのどちら側がマイクロ AB レセプタクルに接続されたかによって決まります。

- USB ケーブルの B 側がフローティング ID ワイヤに接続された場合、組み込みのプルアップ抵抗が ID ラインのハイレベルを検出して、デフォルトのペリフェラルの役割が確立されます。この設定では、OTG_FS はセクション 6.8.2 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go B デバイス）。
- USB ケーブルの A 側がアース付き ID ラインに接続された場合、OTG_FS は、ホストのソフトウェア初期化のために ID ラインステータス変化割り込み（OTG_GINTSTS の CIDSCHG ビット）を発行して、自動的にホストの役割に切り替わります。この設定では、OTG_FS はセクション 6.8.1 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go A デバイス）。

29.5.2 HNP デュアルロールデバイス

グローバル USB 設定レジスタの HNP 対応ビット（OTG_GUSBCFG の HNPCAP ビット）は、OTG_FS コアがホストネゴシエーションプロトコル（HNP）に従って、役割を A ホストから A ペリフェラルに（またその逆）、または B ペリフェラルから B ホスト（またその逆）に動的に変更できるようにします。現在のデバイスのステータスは、グローバル OTG 制御およびステータスレジスタのコネクタ ID ステータスビット（OTG_GOTGCTL の CIDSTS ビット）とグローバル割り込みおよびステータスレジスタの現在の動作モードビット（OTG_GINTSTS の CMOD ビット）の値の組み合わせによって読み出すことができます。

HNP プログラミングモデルの詳細は、[セクション 29.16: OTG_FS プログラミングモデル](#)に記載されています。

29.5.3 SRP デュアルロールデバイス

グローバル USB 設定レジスタの SRP 対応ビット (OTG_GUSBCFG の SRPCAP ビット) は、OTG_FS コアが A デバイスの節電のために V_{BUS} の生成を停止できるようにします。A デバイスは OTG_FS の役割がホストかペリフェラルかに関係なく、常に、 V_{BUS} を駆動する役割を持っていることに注意してください。

SRP A/B デバイスのプログラミングモデルの詳細は、[セクション 29.16: OTG_FS プログラミングモデル](#)に記載されています。

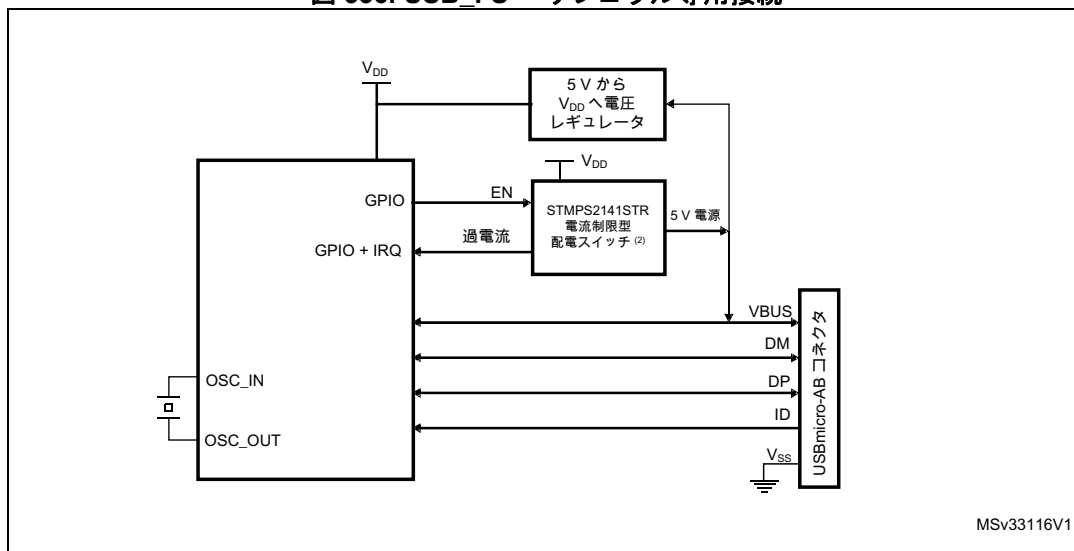
29.6 USB ペリフェラル

このセクションでは、USB ペリフェラルモードでの OTG_FS の機能について説明します。OTG_FS は、以下の環境で USB ペリフェラルとして機能します。

- OTG B ペリフェラル
 - USB ケーブルの B 側が挿入された場合の OTG B デバイスのデフォルトの状態
- OTG A ペリフェラル
 - HNP が OTG_FS の役割をペリフェラルに切り替えた後の OTG A デバイスの状態
- B デバイス
 - ID ラインが存在し、機能していて、USB ケーブルの B 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG_GUSBCFG の HNPCAP ビット) がクリアされている場合 On-The-Go Rev1.3 の 6.8.3 項を参照)。
- ペリフェラルのみ ([図 336 : USB_FS ペリフェラル専用接続](#)を参照)
 - [セクション 29.15.4 : OTG USB 設定レジスタ \(OTG_GUSBCFG\)](#) の強制デバイスモードビット (FDMOD) が 1 にセットされると、OTG_FS コアは USB ペリフェラル専用として機能するよう強制されます (On-The-Go Rev1.3 の 6.8.3 項を参照)。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。

注 : B デバイスまたはペリフェラル専用設定の場合に電源がバスから供給されるデバイスを実装するには、 V_{DD} チップ供給 (V_{BUS} からの) を生成する外部レギュレータを追加する必要があります。

図 336. USB_FS ペリフェラル専用接続



1. バスから電源が供給されるデバイスを構築するには、レギュレータを使用します。

29.6.1 SRP 対応ペリフェラル

グローバル USB 設定レジスタの SRP 対応ビット (OTG_GUSBCFG の SRPCAP ビット) は、OTG_FS コアがセッションリクエストプロトコル (SRP) をサポートできるようにします。この方法により、USB セッションがサスペンド状態にある間、 V_{BUS} の供給を停止して、リモート A デバイスの消費電力を節約できます。

SRP ペリフェラルモードのプログラミングモデルの詳細は、[B デバイスセッションリクエストプロトコル](#)のセクションに記載されています。

29.6.2 ペリフェラルの状態

電源が供給された状態

V_{BUS} 入力は、USB ペリフェラルが電源供給状態に入ることが許される B セッション有効電圧を検出します (USB 2.0 のセクション 9.1 を参照)。その場合、OTG_FS は、DP プルアップ抵抗を自動的に接続して、フルスピードデバイス接続をホストに知らせ、セッションリクエスト割り込みを生成して (OTG_GINTSTS の SRQINT ビット)、電源供給状態を通知します。

V_{BUS} 入力は、また、USB 動作中にホストによって適切な V_{BUS} レベルが供給されることを保証します。 V_{BUS} 電源が B セッションの有効値以下に低下したことが検出された場合 (たとえば、電源の変動があった場合や、ホストポートの電源がオフとなった場合など)、OTG_FS は自動的に切断され、セッション終了検出割り込み (OTG_GOTGINT の SEDET ビット) が生成されて、OTG_FS が電源供給状態でなくなったことを通知します。

電源供給状態では、OTG_FS は、ホストからリセットシグナリングを受け取ることを期待しています。他の USB 操作を実行することはできません。リセットシグナリングを受け取ると、リセット検出割り込み (OTG_GINTSTS の USBRST ビット) が生成されます。リセットシグナリングが完了すると、エミュレーション終了割り込み (OTG_GINTSTS の ENUMDNE ビット) が生成され、OTG_FS はデフォルト状態に入ります。

ソフト切断

電源供給状態は、ソフト切断機能によってソフトウェアで終了させることができます。DP プルアップ抵抗は、デバイス制御レジスタのソフト切断ビット (OTG_DCTL の SDIS ビット) をセットすることによって取り除かれ、USB ケーブルが実際にはホストポートから取り除かれていなくても、ホスト側にデバイス切断検出割り込みを生成します。

デフォルト状態

デフォルト状態では、OTG_FS はホストから SET_ADDRESS コマンドを受け取ることを期待しています。他の USB 操作を実行することはできません。有効な SET_ADDRESS コマンドが USB 上でデコードされると、アプリケーションは対応する番号をデバイス設定レジスタのデバイスアドレスフィールドに書き込みます (OTG_DCFG の DAD ビット)。その場合、OTG_FS アドレス状態に入り、設定された USB アドレスでホストトランザクションに応答する準備が整います。

サスペンド状態

OTG_FS ペリフェラルは、USB の活動状態を定期的に監視します。USB のアイドル状態が 3 ms 続くと、アーリーサスペンド割り込み (OTG_GINTSTS の ESUSP ビット) が発行され、適切な場合は 3 ms 後にサスペンド割り込み (OTG_GINTSTS の USBSUSP ビット) によって確認されます。その場合、デバイスステータスレジスタのデバイスサスペンドビット (OTG_DSTS の SUSPSTS ビット) が自動的にセットされ、OTG_FS はサスペンド状態に入ります。

サスペンド状態は、デバイス自身によって終了することもできます。この場合、アプリケーションはデバイス制御レジスタのリモートウェイクアップシグナリングビット (OTG_DCTL の RWUSIG ビット) をセットし、1~15 ms 後にクリアします。

ホストからレジャームシグナリングが検出されると、レジャーム割り込み (OTG_GINTSTS の WKUPINT ビット) が生成され、デバイスサスペンドビットは自動的にクリアされます。

29.6.3 ペリフェラルエンドポイント

OTG_FS コアは、以下の USB エンドポイントをインスタンス化します。

- コントロールエンドポイント 0 :
 - 双方向であり、制御メッセージのみを取り扱います。
 - 入力および出力トランザクションを処理する個別のレジスタセットを備えています。
 - 固有の制御レジスタ (OTG_DIEPCTL0/OTG_DOEPCTL0)、転送設定レジスタ (OTG_DIEPTSIZ0/OTG_DOEPSIZ0)、およびステータス割り込みレジスタ (OTG_DIEPINT0/OTG_DOEPINT0) を備えています。制御および転送サイズレジスタ内で使用可能なビットのセットは、他のエンドポイントとは少し違います。
- 5IN エンドポイント
 - それぞれ、アイソクロナス転送、バルク転送、またはインタラプト転送タイプをサポートするように設定できます。
 - また、それぞれに固有の制御レジスタ (OTG_DIEPCTLx)、転送設定レジスタ (OTG_DIEPTSIZx)、およびステータス割り込みレジスタ (OTG_DIEPINTx) を備えています。
 - デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK) を使用して、すべての IN エンドポイント (EP0 も含む) 上で 1 種類のエンドポイント割り込みソースを有効/無効にできます。
 - 現在のフレーム内で転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあるときにアサートされる、不完全アイソクロナス IN 転送割り込み (OTG_GINTSTS の ISOIXFR ビット) をサポートします。この割り込みは、周期的フレーム割り込み (OTG_GINTSTS の EOPF) の終わりとともにアサートされます。
- 5OUT エンドポイント
 - それぞれ、アイソクロナス転送、バルク転送、またはインタラプト転送タイプをサポートするように設定できます。
 - また、それぞれに固有の制御レジスタ (OTG_DOEPCTLx)、転送設定レジスタ (OTG_DOEPSIZx)、およびステータス割り込みレジスタ (OTG_DOEPINTx) を備えています。
 - デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK) を使用して、すべての OUT エンドポイント (EP0 も含む) 上で 1 種類のエンドポイント割り込みソースを有効/無効にすることができます。
 - 現在のフレーム内で転送が完了していないアイソクロナス OUT エンドポイントが少なくとも 1 つあるときにアサートされる、不完全アイソクロナス OUT 転送割り込み (OTG_GINTSTS の INCOMPISOOUT ビット) をサポートします。この割り込みは、周期的フレーム割り込み (OTG_GINTSTS の EOPF) の終わりとともにアサートされます。

エンドポイント制御

- 以下のエンドポイント制御は、デバイスエンドポイント-x IN/OUT 制御レジスタ (OTG_DIEPCTLx/OTG_DOEPCTLx) を通じて、アプリケーションから使用できます。
 - － エンドポイント有効/無効
 - － 現在の設定でのエンドポイントのアクティブ化
 - － USB 転送タイプ (アイソクロナス、バルク、インタラプト) のプログラミング
 - － サポートされるパケットサイズのプログラミング
 - － IN エンドポイントに関連する Tx FIFO 番号のプログラミング
 - － 期待される、または送信されるデータ 0/データ 1 PID (バルク/インタラプトのみ) のプログラミング
 - － トランザクションが送受信される偶数/奇数フレームのプログラミング (アイソクロナス転送のみ)
 - － オプションで、FIFO のステータスに関係なく、常にホストに否定応答する NAK ビットのプログラミング
 - － オプションで、エンドポイントに対するホストトークンを常に停止する STALL ビットのプログラミング
 - － オプションで、OUT エンドポイントが受信データの CRC フィールドをチェックしないようにする SNOOP モードのプログラミング

エンドポイント転送

アプリケーションは、デバイスエンドポイント-x 転送サイズレジスタ (OTG_DIEPTSIZx/OTG_DOEPSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。エンドポイント制御レジスタのエンドポイントイネーブルビットをセットする前にプログラミングを行う必要があります。エンドポイントが有効になると、これらのフィールドは、OTG_FS コアが現在の転送ステータスで更新すると、読み出し専用になります。

以下の転送パラメータがプログラム可能です。

- 転送サイズ (バイト単位)
- 全体の転送サイズを構成するパケットの数

エンドポイントのステータス/割り込み

デバイスエンドポイント-x 割り込みレジスタ (OTG_DIEPINTx/OTG_DOEPINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割り込みレジスタの OUT エンドポイント割り込みビットまたは IN エンドポイント割り込みビット (OTG_GINTSTS の OEPINT ビット、または OTG_GINTSTS の IEPINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、デバイスオールエンドポイント割り込み (OTG_DAINTE) レジスタを読み出して、デバイスエンドポイント-x 割り込みレジスタの正確なエンドポイント番号を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_DAINTE および OTG_GINTSTS レジスタの対応するビットをクリアする必要があります。

ペリフェラルコアは、以下のステータスチェックおよび割り込み生成を行います。

- データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割り込み
- セットアップステージが終了しているかどうか (コントロール OUT のみ)
- 関連する送信 FIFO が半エンプティか完全エンプティか (IN エンドポイント)
- NAK 応答がホストに送信されたかどうか (アイソクロナス IN のみ)
- Tx FIFO がエンプティのとき IN トークンが受け取られたかどうか (バルク IN/インタラプト IN のみ)
- エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったかどうか
- バブルエラー条件が検出されたかどうか
- アプリケーションによるエンドポイントの無効化が可能かどうか
- アプリケーションによるエンドポイント NAK が可能かどうか (アイソクロナス IN のみ)
- 3 つを超える連続したセットアップパケットが受信されたかどうか (コントロール OUT のみ)
- タイムアウト条件が検出されたかどうか (コントロール IN のみ)
- アイソクロナス OUT パケットが割り込みを生成せずにドロップされたかどうか

29.7 USB ホスト

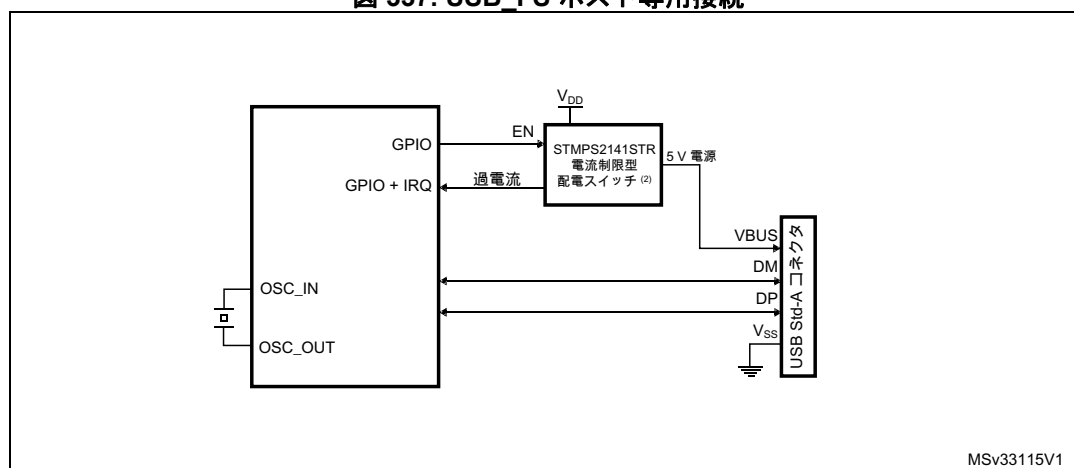
このセクションでは、USB ホストモードでの OTG_FS の機能について説明します。OTG_FS は以下の環境で USB ホストとして機能します。

- OTG A ホスト
 - USB ケーブルの A 側が挿入されたときの OTG A デバイスのデフォルト状態
- OTG B ホスト
 - HNP がホストの役割に切り替わった後の OTG B デバイス
- A デバイス
 - ID ラインが存在し、機能していて、USB ケーブルの A 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG_GUSBCFG の HNPCAP ビット) がクリアされている場合。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。
- ホストのみ
 - [29.15.4](#) グローバル USB 設定レジスタの強制ホストモードビット (OTG_GUSBCFG の FHMOD ビット) は、OTG_FS コアを強制的に USB ホスト専用として機能させます。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。

注： オンチップの 5 V V_{BUS} の発生はサポートされていません。このため、チャージポンプ、または 5 V がアプリケーションボード上で使用可能な場合は基本的な電源スイッチを外部で追加して、5 V の V_{BUS} ラインを駆動しなければなりません。外付けチャージポンプは、任意の GPIO 出力からでも駆動できます。これは、OTG A ホスト、A デバイス、およびホスト専用設定の場合に必要です。



図 337. USB_FS ホスト専用接続



1. アプリケーションが V_{BUS} で電源が供給されるデバイスをサポートしなければならない場合のみ STMPS2141STR が必要です。アプリケーションボード上で 5 V が使用可能な場合、基本的なパワースイッチを使用することができます。
2. V_{DD} 範囲は 2.0~3.6 V です。

29.7.1 SRP 対応ホスト

SRP のサポートはグローバル USB 設定レジスタの SRP 対応ビット (OTG_GUSBCFG の SRPCAP ビット) を通じて使用可能です。SRP 機能が有効なときには、ホストは、USB セッションがサスペンド状態にある間、 V_{BUS} の電力供給を停止することによって、消費電力を節約できます。

SRP ホストモードのプログラミングモデルの詳細は、[A デバイスセッションリクエストプロトコル](#) のセクションに記載されています。

29.7.2 USB ホストの状態

ホストポートパワー

オンチップの 5 V V_{BUS} の発生はサポートされていません。このため、チャージポンプ、または 5 V がアプリケーションボード上で使用可能な場合は基本的な電源スイッチを外部で追加して、5 V の V_{BUS} ラインを駆動しなければなりません。外付けチャージポンプは、任意の GPIO 出力からでも駆動できます。アプリケーションが選んだ GPIO を使用して V_{BUS} に電源を供給することにした場合は、ホストポート制御およびステータスレジスタのポートパワービット (OTG_HPRT の PPWR ビット) もセットする必要があります。

有効な V_{BUS}

HNP または SRP が有効化された場合、VBUS センシングピンは V_{BUS} に接続する必要があります。 V_{BUS} 入力は、USB 動作中にチャージポンプによって適切な V_{BUS} レベルが供給されることを保証します。予期せぬ V_{BUS} 電圧の低下 (V_{BUS} の有効閾値 4.25 V 以下へ) が発生すると、セッション終了検出ビット (OTG_GOTGINT の SEDET ビット) によって OTG 割り込みがトリガされます。その場合、アプリケーションは V_{BUS} 電源供給を停止し、ポートパワービットをクリアする必要があります。

HNP と SRP が共に無効化された場合、VBUS センシングピンを V_{BUS} に接続する必要はなく、GPIO として使用できます。

チャージポンプの過電流フラグは、電氣的ダメージを防止するためにも使用できます。チャージポンプからの過電流フラグ出力を任意の GPIO 入力に接続して、有効レベルになったときにポート割り込みを生成するように設定します。過電流 ISR は迅速に V_{BUS} の発生を無効にし、ポートパワービットをクリアする必要があります。

ペリフェラル接続のホスト検出

SRP または HNP を有効化し、かつ USB ペリフェラルまたは B デバイスを何時でも接続できる場合でも、 V_{BUS} が有効レベル(5 V) で検出できなくなるまで、OTG_FS はバス接続を検出しません。 V_{BUS} が有効なレベルにあり、リモート B デバイスが接続されると、OTG_FS コアは、ホストポート制御およびステータスレジスタのデバイス接続ビット (OTG_HPRT の PCDET ビット) によってトリガされるホストポート割り込みを発行します。

HNP と SRP が共に切り離されたとき、USB ペリフェラルまたは B デバイスが接続されると直ちに検出されます。OTG_FS コアは、ホストポート制御およびステータス内のデバイス接続ビット (OTG_HPRT の PCDET ビット) から発生するホストポート割り込みを発行します。

ペリフェラル切断のホスト検出

ペリフェラル切断イベントは、切断検出割り込み (OTG_GINTSTS の DISCINT ビット) をトリガします。

ホストのエニュメレーション

ペリフェラル接続を検出した後、ホストは、USB リセットおよび設定コマンドを新しいペリフェラルに送信することによって、エニュメレーションプロセスを開始しなければなりません。

USB リセットの駆動を開始する前に、アプリケーションは、デバウンス終了ビット (OTG_GOTGINT の DBCDNE ビット) によってトリガされる OTG 割り込みを待ちます。これは、DP (FS) または DM (LS) 上のプルアップ抵抗の接続によって生じた電氣的デバウンスの後、バスが再び安定状態に戻ったことを示します。

アプリケーションは、ホストポート制御およびステータスレジスタのポートリセットビット (OTG_HPRT の PRST ビット) を最短 10 ms、最長 20 ms の間セット状態に保つことによって、USB を介した USB リセットシグナリング (シングルエンド、ゼロ) を駆動します。アプリケーションは、タイミングカウントをチェックした後、ポートリセットビットがクリアされているかどうかを確認します。

USB リセットシーケンスが完了すると、ポートイネーブル/ディセーブル変更ビット (OTG_HPRT の PENCHNG ビット) によってホストポート割り込みがトリガされます。これにより、エニュメレーションされたペリフェラルの速度をホストポート制御およびステータスレジスタのポート速度フィールド (OTG_HPRT の PSPD ビット) から読み出せることと、ホストが SOF (FS) またはキープアライブ (LS) の駆動を開始しようとしていることがアプリケーションに通知されます。これで、ホストは、ペリフェラル設定コマンドを送信することによってペリフェラルエニュメレーションを完了する準備ができたことになります。

ホストのサスペンド

アプリケーションは、ホストポート制御およびステータスレジスタのポートサスペンドビット (OTG_HPRT の PSUSP ビット) をセットすることによって、USB の活動をサスペンド (一時停止) します。OTG_FS コアは、SOF の送信を停止し、サスペンド状態に入ります。

サスペンド状態は、リモートデバイスが主導して (リモートウェイクアップ)、終了することもできます。この場合、ホストポート制御およびステータスレジスタのポートレジュームビット (OTG_HPRT の PRES ビット) セルフセットが検出されると、リモートウェイクアップ割り込み (OTG_GINTSTS の WKUPINT ビット) が生成され、USB によってレジュームシグナリングが自動的に駆動されます。アプリケーションは、レジュームウィンドウの時間を決めてから、ポートレジュームビットをクリアしてサスペンド状態を終了し、SOF をリスタートする必要があります。

サスペンド状態をホスト主導で終了した場合、アプリケーションは、ポートレジュームビットをセットして、ホストポートでのシグナリングのレジュームを開始し、レジュームウィンドウの時間を決め、最後に、ポートレジュームビットをクリアする必要があります。

29.7.3 ホストチャネル

OTG_FS コアは、12 つのホストチャネルをインスタンス化します。各ホストチャネルは USB ホスト転送をサポートします (USB パイプ)。ホストは、同時に 12 つを超える転送リクエストをサポートすることはできません。アプリケーションからの 12 つを超える転送リクエストがペンディング中の場合、ホストコントローラドライバ (HCD) は、チャネルが以前のデューティから使用可能になったとき、すなわち、転送完了およびチャネルホールド割り込みの受信後、チャネルの再割り当てを行う必要があります。

各ホストチャネルは、入力／出力および任意のタイプの周期的/非周期的トランザクションをサポートするように設定できます。各ホストチャネルは、適切な制御レジスタ (OTG_HCCHARx)、転送設定レジスタ (OTG_HCTSIZx)、およびステータス／割り込みレジスタ (OTG_HCINTx) と関連するマスキングレジスタ (OTG_HCINTMSKx) を使用します。

ホストチャネル制御

- アプリケーションは、ホストチャネル x 特性レジスタ (OTG_HCCHARx) を通じて、以下のホストチャネル制御を使用できます。
 - チャネルの有効化／無効化
 - ターゲット USB ペリフェラルの FS/LS 速度のプログラミング
 - ターゲット USB ペリフェラルのアドレスのプログラミング
 - ターゲット USB ペリフェラルのエンドポイント数のプログラミング
 - IN/OUT 転送方向のプログラミング
 - USB 転送タイプ (コントロール、バルク、インタラプト、アイソクロナス) のプログラミング
 - 最大パケットサイズ (MPS) のプログラミング
 - 奇数／偶数フレームで実行される周期的転送のプログラミング

ホストチャネル転送

アプリケーションは、ホストチャネル転送サイズレジスタ (OTG_HCTSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。プログラミングは、ホストチャネル特性レジスタのチャネルイネーブルビットをセットする前に行う必要があります。エンドポイントが有効になると OTG_FS コアが現在の転送ステータスに従って更新するので、パケットカウンタフィールドは読み出し専用になります。

- 以下の転送パラメータがプログラム可能です。
 - 転送サイズ (バイト単位)
 - 全体の転送サイズを構成するパケットの数
 - 初期データ PID

ホストチャネルのステータス／割り込み

ホストチャネル x 割り込みレジスタ (OTG_HCINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割り込みレジスタのホストチャネル割り込みビット (OTG_GINTSTS の HCINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、ホストオールチャネル割り込み (OTG_HAINT) レジスタを読み出して、ホストチャネル x 割り込みレジスタの正確なチャネル数を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_HAINT および OTG_GINTSTS レジスタの対応するビットをクリアする必要があります。各チャネルの各割り込みソースのマスキングビットも OTG_HCINTMSKx レジスタで使用可能です。

- ホストコアは、以下のステータスチェックおよび割り込み生成を行います。
 - データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割り込み
 - 転送完了、USB トランザクションエラー、またはアプリケーションからの無効化コマンドのためにチャンネルが停止したかどうか
 - 関連する送信 FIFO が半エンプティか完全エンプティか (IN エンドポイント)
 - ACK 応答を受け取った時
 - NAK 応答を受け取った時
 - STALL 応答を受け取った時
 - CRC エラー、タイムアウト、ビットスタッフエラー、偽 EOP による USB トランザクションエラー
 - バブルエラー
 - フレームオーバーラン
 - データトグルエラー

29.7.4 ホストスケジューラ

ホストコアは、アプリケーションによってポストされたUSBトランザクションリクエストを自律的に並べ替えて、管理する組み込みハードウェアスケジューラを備えています。各フレームの開始時に、ホストは、周期的 (アイソクロナスおよびインタラプト) トランザクションを最初に実行し、次に、非周期的 (コントロールおよびバルク) トランザクションを行うことによって USB の仕様に従って、アイソクロナスおよびインタラプト転送タイプを優先します。

ホストは、リクエストキュー (1 つは周期的転送のキュー、もう 1 つは非周期的転送のキュー) を通じて USB トランザクションを処理します。各リクエストキューは、最大 8 個のエントリを保持できます。各エントリはアプリケーションからのペンディング中のトランザクションリクエストを表し、IN または OUT チャンネル番号と USB 上でトランザクションを実行するためのその他の情報を保持しています。リクエストがキューに書き込まれた順序によって USB インタフェース上のトランザクションの順序が決まります。

各フレームの先頭で、ホストは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。現在のフレームに対してスケジュールされたアイソクロナスまたはインタラプトトランザクションが現在のフレームの終了時点でまだペンディング中である場合、ホストは、不完全周期転送割り込み (OTG_GINTSTS の IPXFR ビット) を発行します。OTG_FS コアは周期的と非周期的リクエストキューの管理を行います。周期的送信 FIFO およびキューステータスレジスタ (OTG_HPTXSTS) と非周期的送信 FIFO およびキューステータスレジスタ (OTG_HNPTXSTS) は読み出し専用レジスタであり、アプリケーションから各リクエストキューのステータスを読み出すときに使うことができます。これらのレジスタには次が含まれます。

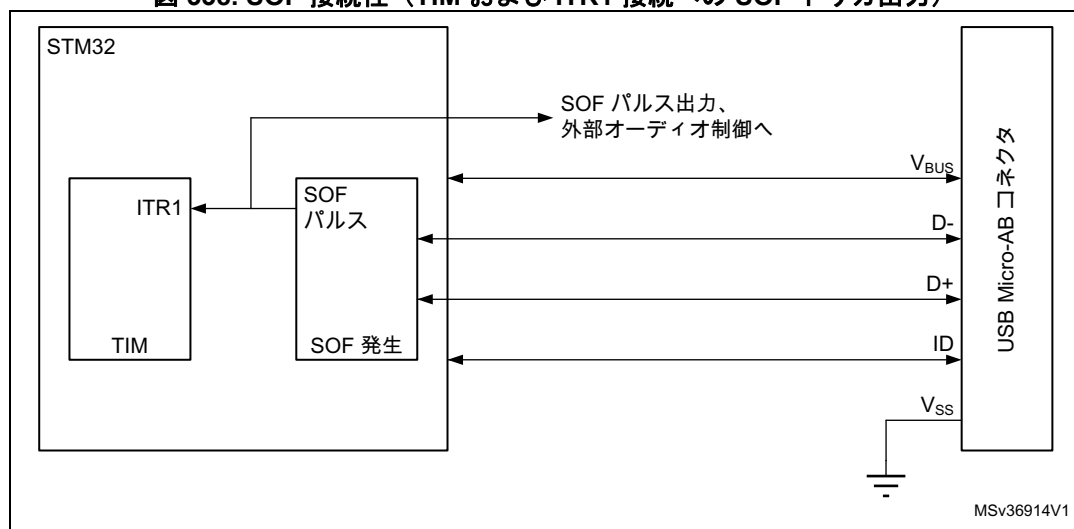
- 周期 (非周期) リクエストキューで現在使用可能なフリーエントリの数 (最大 8 エントリ)。
- 周期 (非周期) Tx FIFO (OUT トランザクション) で現在使用可能な空きスペース
- IN/OUT トークン、ホストチャンネル番号、およびその他のステータス情報

リクエストキューはそれぞれ最大 8 つのエントリを保持できるので、アプリケーションは、最大 8 つのペンディング中の周期トランザクションに加えて、8 つのペンディング中の非周期トランザクションについて、SB に物理的に到達する前に、ホストトランザクションを事前にスケジュールできます。

ホストスケジューラ (キュー) へトランザクションリクエストをポストするときは、アプリケーションは OTG_HNPTXSTS レジスタの PTXQSAV ビット、または OTG_HNPTXSTS レジスタの NPTQSAV ビットを読み出して、周期的 (非周期的) リクエストキュー内に少なくとも 1 つのエントリが存在することを確認する必要があります。

29.8 SOF トリガ

図 338. SOF 接続性 (TIM および ITR1 接続への SOF トリガ出力)



OTG_FS コアは、ホスト、ペリフェラル、SOF パルス出力接続機能内で SOF フレーミングを監視、追跡、設定する手段を提供します。

これらのユーティリティは、オーディオペリフェラルが PC によって提供されるアイソクロナスストリームに同期する必要がある場合や、ホストがオーディオペリフェラルの要件に従ってフレーミング速度をトリム（微調整）する必要がある場合など、適応型オーディオクロック生成技術にとって特に役立ちます。

29.8.1 ホストの SOF

ホストモードでは、2 つの連続した SOF (FS) またはキープアライブ (LS) トークンの生成の間に発生する PHY クロックの数は、ホストフレームインターバルレジスタ (HFIR) でプログラム可能であり、SOF フレーミング周期に対する制御をアプリケーションに与えます。割り込みは、どのフレームの開始時点でも生成されます (OTG_GINTSTS の SOF ビット)。現在のフレーム番号と次の SOF までの残り時間が、ホストフレーム番号レジスタ (HFNUM) で追跡されます。

SOF パルス信号は、任意の SOF 開始トークンで生成され 12 システムクロックサイクルの幅です。SOF パルスは、タイマの入カトリガにも内部接続されるので、入力キャプチャ機能、出力比較機能、およびタイマを SOF パルスによってトリガできます。

29.8.2 ペリフェラルの SOF

デバイスモードでは、フレーム割り込みの開始は、SOF トークンが USB 上で受信されるたびに生成されます (OTG_GINTSTS の SOF ビット)。対応するフレーム番号は、デバイスステータスレジスタ (OTG_DSTS の FNSOF ビット) から読み出すことができます。12 システムクロックサイクルの幅を持つ SOF パルス信号も生成されます。SOF パルス信号は、TIM 入力トリガにも内部接続されるので、入力キャプチャ機能、出力比較機能、およびタイマを SOF パルスによってトリガできます。

周期的フレーム割り込みの終わり (OTG_GINTSTS の EOPF) は、デバイス設定レジスタの周期的フレームインターバルフィールド (OTG_DCFG の PFIVL ビット) に応じて、タイムフレームインターバルの 80 %、85 %、90 % または 95 % が経過した時点を実アプリケーションに通知するために使用されます。この機能を使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

29.9 電力オプション

OTG PHY の消費電力は、サポートされる OTG リビジョンに応じて、一般コア設定レジスタの 2 つまたは 3 つのビットによって制御されます。

- PHY パワーダウン (OTG_GCCFG の PWRDWN)
PHY のフルスピードトランシーバモジュールのオン/オフを切り替えます。事前にセットして、すべての USB 動作を可能にする必要があります。
- V_{BUS} 検出イネーブル (OTG_GCCFG の VBDEEN)
OTG の動作に関連する V_{BUS} レベル検出用コンパレータのオン/オフを切り替えます。

USB セッションがまだ有効でないか、デバイスが切断されているときには、USB サスペンド状態で節電テクニックを使用できます。

- PHY クロックを停止します (OTG_PCGCCTL の STPPCLK ビット)。
クロックゲーティング制御レジスタのストップ PHY クロックビットをセットすると、OTG フルスピードコア内部の 48 MHz クロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。USB クロック切り替え機能による動的な電力消費は、48 MHz クロック入力アプリケーションによって動作状態を維持された場合でも節減されます。
ほとんどのトランシーバも無効になり、非同期レジュームやリモートウェイクアップイベントの検出を担当する部分だけが動作状態を維持します。
- ゲート HCLK (OTG_PCGCCTL の GATEHCLK ビット)
クロックゲーティング制御レジスタのゲート HCLK ビットをセットすると、OTG_FS コア内部のシステムクロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。レジスタ読み出しおよび書き込みインタフェースのみが動作状態を維持します。USB クロック切り替え機能による動的な電力消費は、システムクロックが他の目的のためにアプリケーションによって動作状態を維持されている場合でも節減されます。
- USB システムの停止
OTG_FS が USB サスペンド状態にある場合、アプリケーションはシステム内のすべてのクロックソースを完全にシャットダウンすることによって、電力消費全体を大幅に削減する方法を取ることができます。USB システムの停止は、まず、ストップ PHY クロックビットをセットし、次に、パワー制御システムモジュール (PWR) でシステムディープスリープモードを設定することによって有効になります。
OTG_FS コアは、USB 上でリモートウェイクアップ (ホストとして) またはリモートレジューム (デバイスとして) 信号の非同期検出によって、システムと USB クロックの両方を自動的に再起動します。

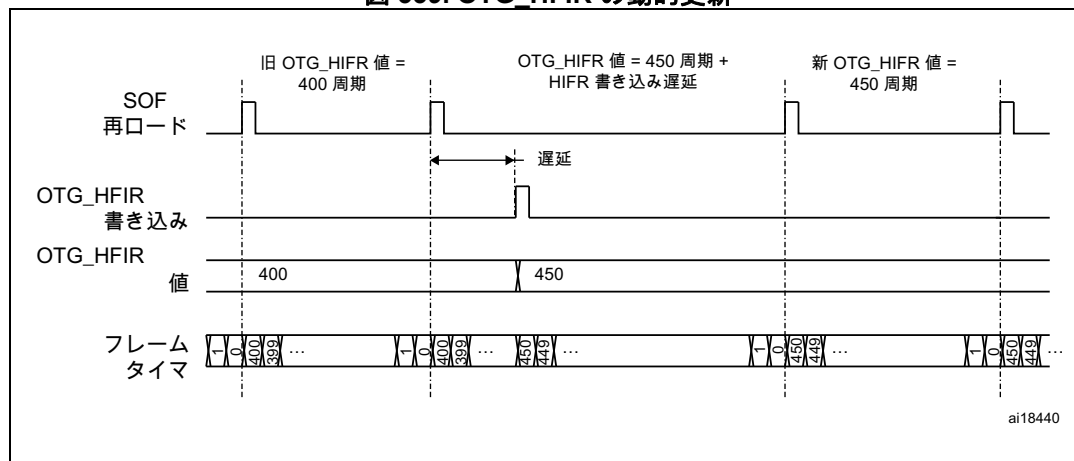
動的消費電力を節減するために、USB データ FIFO は、OTG_FS コアによってアクセスされたときだけクロック駆動されます。

29.10 OTG_HFIR レジスタの動的更新

USB コアはホストモードで SOF フレーミング周期を動的にトリミングする機能を内蔵しています。この機能を使うと、外部デバイスをマイクロ SOF フレームと同期させることができます。

現在の SOF フレーム内で OTG_HFIR レジスタを変更すると、[図 339](#) で説明するように次のフレーム SOF 周期補正が適用されます。

図 339. OTG_HFIR の動的更新

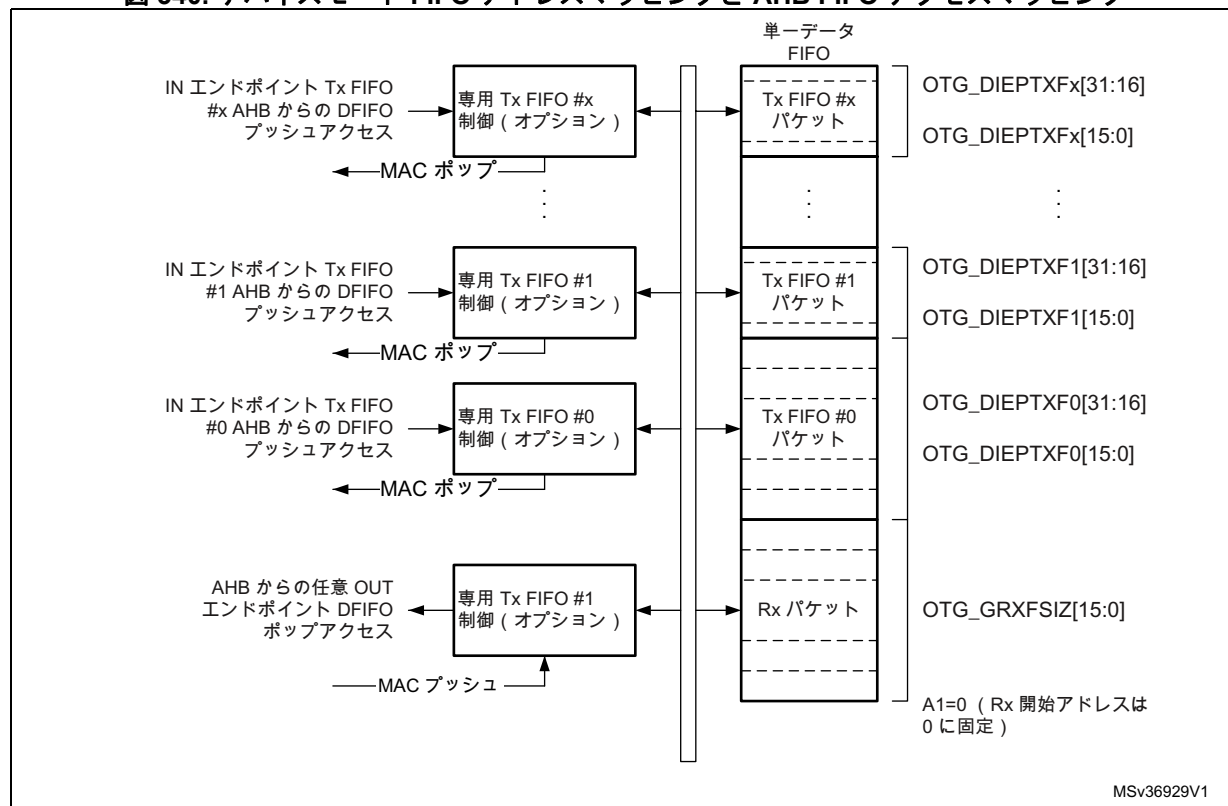


29.11 USB データ FIFO

USB システムは、高度な FIFO 制御メカニズムを持つ 1.25 KB の専用 RAM を備えています。OTG_FS コアの packets FIFO コントローラモジュールは、RAM 空間を、アプリケーションが USB 転送の前に一時的にデータを格納する（プッシュする）Tx FIFO と、USB から受信したデータをアプリケーションが取得する（ポップする）前に一時的に格納される Rx FIFO に編成します。割り当てられる FIFO の数と、これらが RAM の内部でどのように構成されるかは、デバイスの役割に依存します。ペリフェラルモードでは、追加の Tx FIFO がアクティブな IN エンドポイントのそれぞれに対して割り当てられます。FIFO のサイズは、アプリケーションの要件に合うように、ソフトウェアで設定できます。

29.11.1 ペリフェラル FIFO アーキテクチャ

図 340. デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



ペリフェラル Rx FIFO

OTG ペリフェラルは、すべての OUT エンドポイントに向けたデータを受信する単一の受信 FIFO を使用します。受信バケットは、Rx FIFO の空きスペースが使用可能になるまで、連続的にスタックされます。受信バケットのステータス (OUT エンドポイントの宛先番号、バイト数、データ PID、および受信データの有効性を含みます) も、コアによってデータペイロードの最上部に格納されます。使用可能なスペースがなくなると、ホストトランザクションは NACK され、アドレス指定されたエンドポイントに割り込みが受信されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (OTG_GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ペリフェラルによる受信 RAM バッファへの書き込みをより効率的にします。

- すべての OUT エンドポイントは同じ RAM バッファを共有します (共有 FIFO)。
- OTG_FS コアは、OUT トークンのいかなるホストシーケンスについても、限度まで受信 FIFO に書き込むことができます。

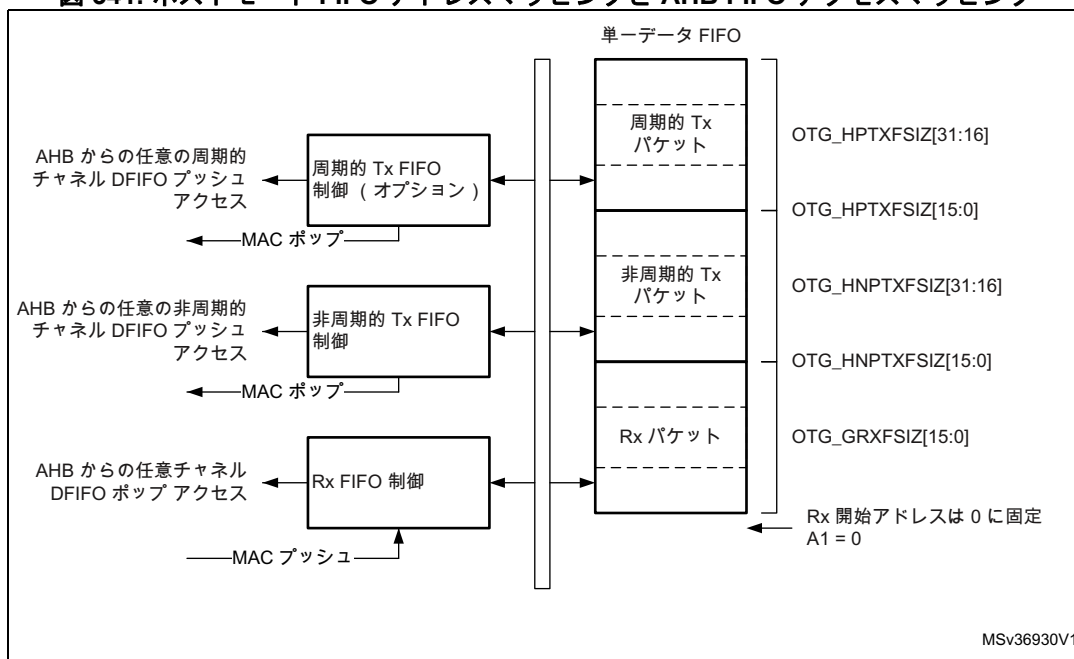
アプリケーションは Rx FIFO 非エンプティ割り込み (OTG_GINTSTS の RXFLVL ビット) の受信を、少なくとも 1 バケットがダウンロード可能である限り継続します。アプリケーションは、受信ステータス読み出しおよびポップレジスタ (OTG_GRXSTSP) からバケット情報を読み出し、最後に、エンドポイントに関するポップアドレスから読み取ることによって受信 FIFO からデータを取得します。

ペリフェラル Tx FIFO

コアは、各 IN エンドポイント用の専用 FIFO を備えています。アプリケーションは、IN エンドポイント 0 についてはエンドポイント 0 送信 FIFO サイズレジスタ (OTG_DIEPTXF0) に書き込み、IN エンドポイント x についてはデバイス IN エンドポイント送信 FIFOx レジスタ (OTG_DIEPTFXx) に書き込むことによって、FIFO のサイズを設定します。

29.11.2 ホスト FIFO アーキテクチャ

図 341. ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



ホスト Rx FIFO

ホストは、すべての周期的および非周期的のランザクションについて 1 つの受信 FIFO を使用します。この FIFO は、USB から受信したデータ (受信パケットのペイロード) をシステムメモリに転送されるまで保持する受信バッファとして使用されます。任意のリモート IN エンドポイントから受信されたパケットは、空きスペースがある限り連続的にスタックされます。各受信パケット (ホストチャンネルデスティネーション、バイト数、データ PID、および受信データの有効性を含みます) のステータスも、FIFO に格納されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (OTG_GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ホストによる受信データバッファへの書き込みをより効率的にします。

- IN に設定されたすべてのホストチャンネルは、同じ RAM バッファを共有します (共有 FIFO)。
- OTG_FS コアは、ホストのソフトウェアによって駆動された IN トークンのいかなるシーケンスについても、限度まで受信 FIFO に書き込むことができます。

アプリケーションは、少なくとも 1 つのパケットがダウンロード可能である限り、Rx FIFO 非エンピティ割り込みを受け取ります。アプリケーションは、受信ステータス読み出しおよびポップレジスタからパケット情報を読み出し、最後に、受信 FIFO からデータを吸い上げます。

ホスト Tx FIFO

ホストは、すべての非周期的（コントロールおよびバルク）OUT トランザクションについて 1 つの送信 FIFO を、すべての周期的（アイソクロナスおよびインタラプト）OUT トランザクションについて 1 つの送信 FIFO を使用します。FIFO は、USB 経由で送信されるデータ（送信パケットのペイロード）を保持する送信バッファとして使用されます。周期的（非周期的）Tx FIFO のサイズは、ホストの周期的（非周期的）送信 FIFO サイズレジスタ（OTG_HPTXFSIZ/OTG_HNPTXFSIZ）で設定されます。

Tx FIFO が 2 つ実装されているのは、USB フレームよりも周期的トラフィックタイプに対して高い優先度が与えられていることによります。各フレームの先頭で、組み込みのホストスケジューラは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。

2 つの送信 FIFO アーキテクチャにより、USB ホストは周期的送信データバッファと非周期的送信データバッファを個別に最適化できます。

- OUT 方向の周期的（非周期的）トランザクションをサポートするように設定されたすべてのホストチャネルは、同じ RAM バッファを共有します（共有 FIFO）。
- OTG_FS コアは、ホストのソフトウェアによって駆動された OUT トークンのいかなるシーケンスについても、限度まで周期的（非周期的）送信 FIFO に書き込むことができます。

OTG_FS コアは、AHB 設定レジスタの周期的 Tx FIFO エンプティレベルビット（OTG_GAHBCFG の PTXFELVL ビット）の値に応じて、周期的 Tx FIFO が半分または完全にエンプティである限り、周期的 Tx FIFO エンプティ割り込み（OTG_GINTSTS の PTXFE ビット）を発行します。アプリケーションは、周期的 Tx FIFO と周期的リクエストキューの両方に空きスペースが存在する限り、前もって送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの周期的送信 FIFO およびキューステータスレジスタ（OTG_HPTXSTS）を読み出すことで知ることができます。

OTG_FS コアは、AHB 設定レジスタの非周期的 Tx FIFO エンプティレベルビット（OTG_GAHBCFG の TXFELVL ビット）の値に応じて、非周期的 Tx FIFO が半分または完全にエンプティである限り、非周期的 Tx FIFO エンプティ割り込み（OTG_GINTSTS の NPTXFE ビット）を発行します。アプリケーションは、非周期的 Tx FIFO と非周期的リクエストキューの両方に空きスペースが存在する限り、送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの非周期的送信 FIFO およびキューステータスレジスタ（OTG_HNPTXSTS）を読み出すことで知ることができます。

29.11.3 FIFO RAM の割り当て

デバイスモード

受信 FIFO RAM 割り当て：アプリケーションは SETUP パケット用に RAM を割り当てる必要があります。

- 制御エンドポイントで SETUP パケットを受信するため受信 FIFO に 10 ロケーションを確保する必要があります。コアは、SETUP パケット用に予約されている位置に他のデータを書き込むことはありません。
- 1 つの位置は、グローバル OUT NAK のために割り当てられます。
- 各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、(最大パケットサイズ / 4) + 1 のスペースを割り当てる必要があります。複数のアイソクロナスエンドポイントが有効な場合、連続パケットを受信するには、少なくとも、(最大パケットサイズ / 4) + 1 の 2 倍のスペースが割り当てられなければなりません。一般に、(最大パケットサイズ / 4) + 1 の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

- 各エンドポイントの最後のパケットとともに、送信完了ステータス情報が FIFO にプッシュされます。各 OUT エンドポイントに対して 1 つのロケーションを予約することが推奨されます。

デバイス RxFIFO =

$(4 * \text{制御エンドポイント数} + 6) + ((\text{最大 USB 使用パケット} / 4) + \text{ステータス情報につき } 1) + (2 * \text{OUT エンドポイント数}) + \text{グローバル NAK につき } 1$

例：MPS は周期的 USB パケットでは 1,024 バイト、非周期的 USB パケットでは 512 バイトです。3 つの OUT エンドポイント、3 つの IN エンドポイント、1 つの制御エンドポイント、3 つのホストチャンネルがあります。

デバイス RxFIFO = $(4 * 1 + 6) + ((1,024 / 4) + 1) + (2 * 4) + 1 = 276$

送信 FIFO RAM の割り当て： 各 IN エンドポイント送信 FIFO に必要な最小 RAM スペースは、その特定の IN エンドポイントに対する最大パケットサイズを収容できるスペースとなります。

注： *送信 IN エンドポイント FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。*

ホストモード

受信 FIFO RAM 割り当て：

各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、 $(\text{最大パケットサイズ} / 4) + 1$ のスペースを割り当てる必要があります。複数のアイソクロナスチャンネルが有効な場合、連続パケットを受信するには、少なくとも、 $(\text{最大パケットサイズ} / 4) + 1$ の 2 倍のスペースが割り当てられなければなりません。一般に、 $(\text{最大パケットサイズ} / 4) + 1$ の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

ホストチャンネルの最後のパケットとともに、転送完了ステータス情報が FIFO にプッシュされます。したがって、このために 1 つの位置が割り当てられなければなりません。

ホスト RxFIFO = $(\text{最大 USB 使用パケット} / 4) + \text{ステータス情報につき } 1 + 1 \text{ 転送完了}$

例：ホスト RxFIFO = $((1,024 / 4) + 1) + 1 = 258$

送信 FIFO RAM の割り当て：

ホストの非周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての非周期的 OUT チャンネルの中で最大のパケットサイズを収容できるスペースとなります。

一般に、最大パケットサイズの 2 倍のスペースが推奨されます。現在のパケットが USB に転送されているときに、CPU が次のパケットを取得できるからです。

非周期的 TxFIFO = $\text{最大非周期 USB 使用パケット} / 4$

例：非周期的 TxFIFO = $(512 / 4) = 128$

ホストの周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての周期的 OUT チャンネルの中で最大のパケットサイズを収容できるスペースとなります。少なくとも 1 つのアイソクロナス OUT エンドポイントがある場合には、スペースはそのチャンネルの最大パケットサイズの少なくとも 2 倍でなければなりません。

ホスト周期的 TxFIFO = $\text{最大周期 USB 使用パケット} / 4$

例：ホスト周期的 TxFIFO = $(1,024 / 4) = 256$

注： *非周期的送信 FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。*

29.12 OTG_FS システムのパフォーマンス

USB とシステムの最高のパフォーマンスを達成するには、大きな RAM バッファ、柔軟に設定可能な FIFO サイズ、AHB プッシュ／ポップレジスタを通した迅速な 32 ビット FIFO アクセス、そして、特に、高度な FIFO 制御メカニズムが必要です。このメカニズムにより、OTG_FS は、現在の USB シーケンスに関係なく、使用可能な RAM スペースに効率的に書き込むことができます。これらの機能によって、以下のことが可能になります。

- アプリケーションは、CPU のバンド幅使用率を最適化するための介入を調整する十分な余裕ができます。
 - データを効率的に USB に送信できるときまで、大量の送信データを事前に蓄積することができます。
 - 単一の受信 FIFO からデータをダウンロードするための時間の余裕が増えるという利点があります。
- USB コアは、最高動作速度を維持できるので、アプリケーションの介入がある場合に比べて、自律的に動作できる幅が広がり、最大のフルスピードバンド幅を実現できます。
 - 大量の送信データを自由に蓄積して、USB 経由のデータ送信を自律的に管理できます。
 - 受信バッファの使用可能な空きスペースが大量にあるので、USB から入ってくるデータを自律的に書き込むことができます。

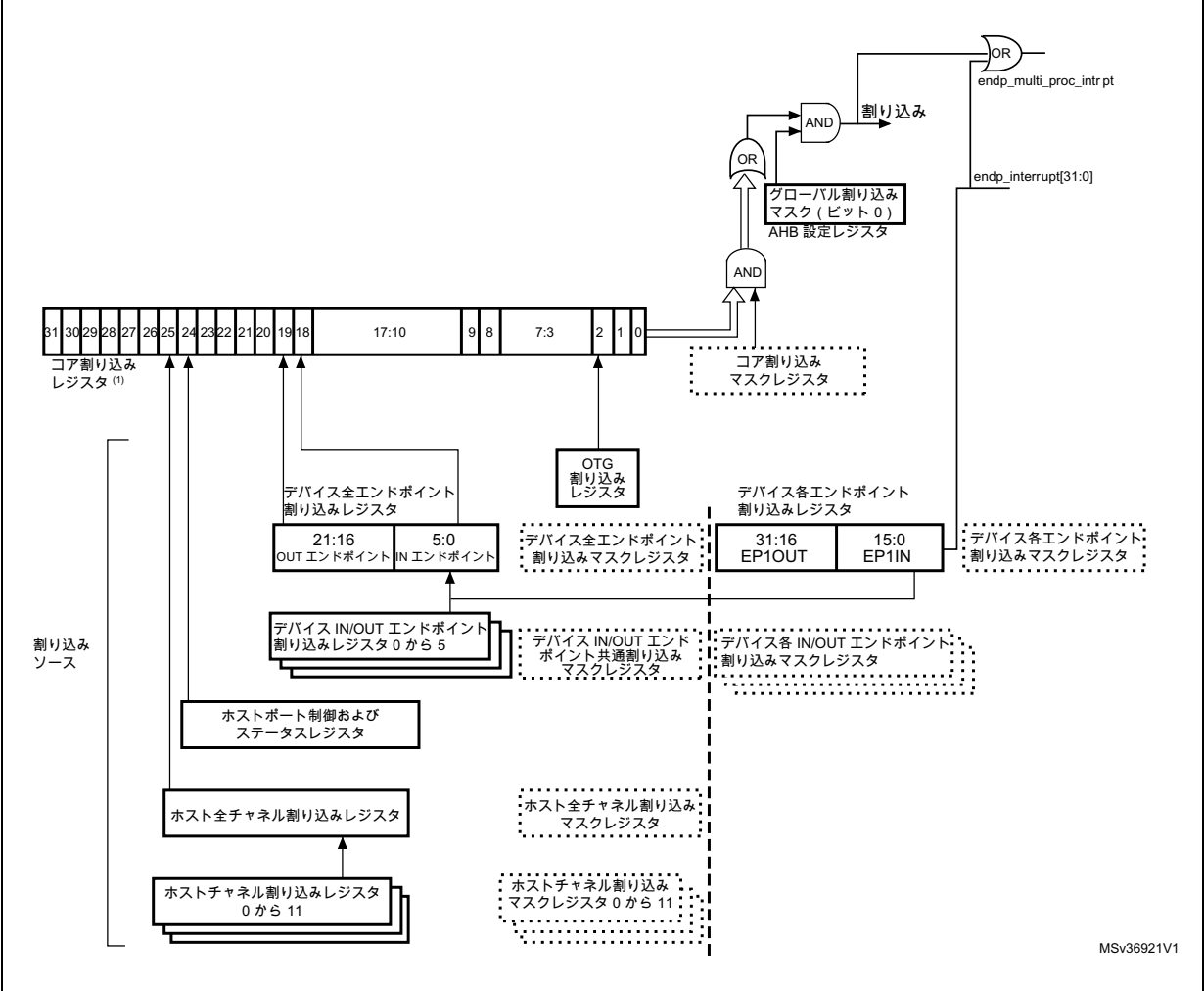
OTG_FS コアは 1.25 KB の RAM バッファを非常に効率的に使用でき、また、1.25 KB の送信／受信データはフルスピードフレームをカバーするには十分以上なので、USB システムは CPU の介入がなくても、最大 1 USB フレーム (1 ms) に対応する最高のフルスピードデータ速度に耐えることができます。

29.13 OTG_FS 割り込み

OTG_FS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割り込みが生成され、コア割り込みレジスタ (OTG_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

[図 342](#) に、割り込みの階層を示します。

図 342. 割り込みの階層



1. コア割り込みレジスタビットは、[OTG コア割り込みレジスタ \(OTG_GINTSTS\) \(987 ページ\)](#) に示されています。

29.14 OTG_FS 制御／ステータスレジスタ

アプリケーションは、AHB スレーブインタフェースを通じて制御／ステータスレジスタ (CSR) を読み書きすることによって、OTG_FS コントローラを制御します。これらのレジスタは 32 ビット幅であり、アドレスは 32 ビットブロックに配置されています。OTG_FS レジスタは、ワード (32 ビット) 単位でアクセスする必要があります。

CSR は、次のように分類されます。

- コアグローバルレジスタ
- ホストモードレジスタ
- ホストグローバルレジスタ
- ホストポート CSR
- ホストチャネル固有レジスタ
- デバイスモードレジスタ
- デバイスグローバルレジスタ
- デバイスエンドポイント固有レジスタ
- パワーおよびクロックゲーティングレジスタ
- データ FIFO (DFIFO) アクセスレジスタ

コアグローバル、パワーおよびクロックゲーティング、データ FIFO (DFIFO) アクセス、およびホストポート CSR レジスタのみが、ホストモードおよびデバイスモードの両方でのアクセスが可能です。OTG_FS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割り込みが生成され、コア割り込みレジスタ (OTG_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

29.14.1 CSR メモリマップ

ホストモードおよびデバイスモードレジスタは、それぞれ異なるアドレスを占めます。すべてのレジスタは AHB クロックドメインで実装されます。

グローバル CSR マップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能です。

表 188. コアグローバル制御およびステータスレジスタ (CSR)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_GOTGCTL	0x000	OTG 制御およびステータスレジスタ (OTG_GOTGCTL) (978 ページ)
OTG_GOTGINT	0x004	OTG 割り込みレジスタ (OTG_GOTGINT) (981 ページ)
OTG_GAHBCFG	0x008	OTG AHB 設定レジスタ (OTG_GAHBCFG) (982 ページ)
OTG_GUSBCFG	0x00C	OTG USB 設定レジスタ (OTG_GUSBCFG) (983 ページ)
OTG_GRSTCTL	0x010	OTG リセットレジスタ (OTG_GRSTCTL) (985 ページ)
OTG_GINTSTS	0x014	OTG コア割り込みレジスタ (OTG_GINTSTS) (987 ページ)
OTG_GINTMSK	0x018	OTG 割り込みマスクレジスタ (OTG_GINTMSK) (991 ページ)



表 188. コアグローバル制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_GRXSTSR	0x01C	OTG_FS 受信ステータスデバッグ読み出し/OTGステータス読み出しおよびポップ レジスタ (OTG_GRXSTSR/OTG_GRXSTSP) (994 ページ)
OTG_GRXSTSP	0x020	
OTG_GRXFSIZ	0x024	OTG 受信 FIFO サイズレジスタ (OTG_GRXFSIZ) (995 ページ)
OTG_HNPTXFSIZ/OTG_DIEPTXF0 ⁽¹⁾	0x028	OTG ホスト非周期的送信 FIFO サイズレジスタ (OTG_HNPTXFSIZ) / エンドポ イント 0 送信 FIFO サイズ (OTG_DIEPTXF0)
OTG_HNPTXSTS	0x02C	OTG 非周期的送信 FIFO/キューステータスレジスタ (OTG_HNPTXSTS) (997 ページ)
OTG_GCCFG	0x038	OTG 一般コア設定レジスタ (OTG_GCCFG) (998 ページ)
OTG_CID	0x03C	OTG コア ID レジスタ (OTG_CID) (999 ページ)
OTG_GLPMCFG	0x54	OTG コア LPM 設定レジスタ (OTG_GLPMCFG) (1000 ページ)
OTG_HPTXFSIZ	0x100	OTG ホスト周期的送信 FIFO サイズレジスタ (OTG_HPTXFSIZ) (1004 ページ)
OTG_DIEPTFXx	0x104	OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTFXx) (x = 1~5、ここで x は FIFO_number) (1004 ページ)
	0x124	
	...	
	0x184	

1. 一般的ルールとして、OTG_HNPTXFSIZ はホストモードで、OTG_DIEPTXF0 はデバイスモードで使用されます。

ホストモード CSR マップ

これらのレジスタは、コアがホストモードに変わるたびにプログラムされなければなりません。

表 189. ホストモード制御およびステータスレジスタ (CSR)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_HCFG	0x400	OTG ホスト設定レジスタ (OTG_HCFG) (1005 ページ)
OTG_HFIR	0x404	OTG ホストフレームインターバルレジスタ (OTG_HFIR) (1006 ページ)
OTG_HFNUM	0x408	OTG ホストフレーム番号/残りフレーム時間 レジスタ (OTG_HFNUM) (1007 ページ)
OTG_HPTXSTS	0x410	OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG_HPTXSTS) (1008 ページ)
OTG_HAINT	0x414	OTG ホスト全チャネル割り込みレジスタ (OTG_HAINT) (1009 ページ)
OTG_HAINTMSK	0x418	OTG ホスト全チャネル割り込みマスクレジスタ (OTG_HAINTMSK) (1009 ペ ージ)
OTG_HPRT	0x440	OTG ホストポート制御およびステータスレジスタ (OTG_HPRT) (1010 ページ)
OTG_HCCHARx	0x500	OTG ホストチャネル x 特性レジスタ (OTG_HCCHARx) (x = 0..11、ここで x = Channel_number) (1012 ページ)
	0x520	
	...	
	0x660	

表 189. ホストモード制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_HCINTx	0x508 0x528 0x668	OTG ホストチャネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..11, ここで x = Channel_number) (1014 ページ)
OTG_HCINTMSKx	0x50C 0x52C 0x66C	OTG ホストチャネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..11, ここで x = Channel_number) (1015 ページ)
OTG_HCTSIZx	0x510 0x530 0x670	OTG ホストチャネル x 転送サイズレジスタ (OTG_HCTSIZx) (x = 0..11, ここで x = Channel_number) (1016 ページ)

デバイスモード CSR マップ

これらのレジスタは、コアがデバイスモードに変わるたびにプログラムする必要があります。

表 190. デバイスモード制御およびステータスレジスタ

項目 (略称)	オフセット アドレス	レジスタ名
OTG_DCFG	0x800	OTG デバイス設定レジスタ (OTG_DCFG) (1017 ページ)
OTG_DCTL	0x804	OTG デバイス制御レジスタ (OTG_DCTL) (1018 ページ)
OTG_DSTS	0x808	OTG デバイスステータスレジスタ (OTG_DSTS) (1020 ページ)
OTG_DIEPMSK	0x810	OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK) (1021 ページ)
OTG_DOEPMSK	0x814	OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK) (1022 ページ)
OTG_DAIN	0x818	OTG デバイス全エンドポイント割り込みレジスタ (OTG_DAIN) (1022 ページ)
OTG_DAINMSK	0x81C	OTG 全エンドポイント割り込みマスクレジスタ (OTG_DAINMSK) (1023 ページ)
OTG_DVBUSDIS	0x828	OTG デバイス V _{BUS} 放電時間レジスタ (OTG_DVBUSDIS) (1024 ページ)
OTG_DVBUSPULSE	0x82C	OTG デバイス V _{BUS} パルシング時間レジスタ (OTG_DVBUSPULSE) (1024 ページ)
OTG_DIEPEMPMSK	0x834	OTG デバイス IN エンドポイント FIFO エンプティ割り込みマスクレジスタ (OTG_DIEPEMPMSK) (1025 ページ)
OTG_DIEPCTL0	0x900	OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_DIEPCTL0) (1025 ページ)

表 190. デバイスモード制御およびステータスレジスタ (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_DIEPCTLx	0x920 0x940 ... 0x9A0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1~5、 ここで x = Endpoint_number) (1027 ページ)
OTG_DIEPINTx	0x908 0x928 0x9A8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx) (x = 0 ~5、ここで x = Endpoint_number) (1033 ページ)
OTG_DIEPTSIZ0	0x910	OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_DIEPTSIZ0) (1035 ページ)
OTG_DTXFSTSx	0x918 0x938 0x9B8	OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) (x = 0~5、ここで x = Endpoint_number) (1038 ページ)
OTG_DIEPTSIZx	0x930 0x950 ... 0x9B0	OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSIZx) (x = 1~5、ここで x = Endpoint_number) (1037 ページ)
OTG_DOEPTCTL0	0xB00	OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG_DOEPTCTL0) (1029 ページ)
OTG_DOEPTCTLx	0xB20 0xB40 ... 0xBA0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPTCTLx) (x = 1~5、 ここで x = Endpoint_number) (1031 ページ)
OTG_DOEPTINTx	0xB08 0xB28 ... 0xBA8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPTINTx) (x = 0 ~5、ここで x = Endpoint_number) (1034 ページ)
OTG_DOEPTSIZ0	0xB10	OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_DOEPTSIZ0) (1036 ページ)
OTG_DOEPTSIZx	0xB30 0xB50 ... 0xBB0	OTG デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_DOEPTSIZx) (x = 1~5、ここで x = Endpoint_number) (1038 ページ)

データ FIFO (DFIFO) アクセスレジスタマップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能であり、特定のエンドポイントまたはチャンネルの FIFO スペースを指定された方向で読み書きするために使用されます。ホストチャンネルが IN タイプの場合 FIFO はそのチャンネル上では読み出しのみが可能です。同様に、ホストチャンネルが OUT タイプの場合 FIFO はそのチャンネル上では書き込みのみが可能です。

表 191. データ FIFO (DFIFO) アクセスレジスタマップ

FIFO アクセスレジスタセクション	アドレス範囲	アクセス
デバイス IN エンドポイント 0 / ホスト OUT チャンネル 0 : DFIFO 書き込みアクセス デバイス OUT エンドポイント 0 / ホスト IN チャンネル 0 : DFIFO 読み出しアクセス	0x1000~0x1FFC	w r
デバイス IN エンドポイント 1 / ホスト OUT チャンネル 1 : DFIFO 書き込みアクセス デバイス OUT エンドポイント 1 / ホスト IN チャンネル 1 : DFIFO 読み出しアクセス	0x2000~0x2FFC	w r
...
デバイス IN エンドポイント x ⁽¹⁾ / ホスト OUT チャンネル x ⁽¹⁾ : DFIFO 書き込みアクセス デバイス OUT エンドポイント x ⁽¹⁾ / ホスト IN チャンネル x ⁽¹⁾ : DFIFO 読み出しアクセス	0xX000~0xXFFC	w r

1. デバイスモードでは x = 5、ホストモードでは x = 11 です。

パワーおよびクロックゲーティング CSR マップ

パワーおよびクロックゲーティングの場合、シングルレジスタです。このレジスタは、ホストモードとデバイスモードの両方で使用できます。

表 192. パワーおよびクロックゲーティング制御およびステータスレジスタ

レジスタ名	項目 (略称)	オフセットアドレス : 0xE00~0xFFFF
パワーおよびクロックゲーティング制御レジスタ	PCGCCTL	0xE00~0xE04
予約済み	-	0xE05~0xFFFF

29.15 OTG_FS レジスタ

これらのレジスタは、ホストモードとデバイスモードの両方で使用でき、これらのモード間で切り替わっても再プログラムの必要はありません。

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

29.15.1 OTG 制御およびステータスレジスタ (OTG_GOTGCTL)

アドレスオフセット : 0x000

リセット値 : 0x0001 0000

OTG_GOTGCTL レジスタは、コアの OTG 機能の動作を制御し、そのステータスを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTG VER	BSVLD	ASVLD	DBCT	CID STS
											rw	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	BVALO VAL	BVALO EN	AVALO VAL	AVALO EN	VBVAL OVAL	VBVAL OEN	SRQ	SRQ SCS
			rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	r

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OTGVER** : OTG バージョン

OTG バージョンを選択します。

0 : OTG バージョン 1.3 です。このバージョンでは、コアは SRP のデータラインパルシングおよび VBUS パルシングをサポートしています。

1 : OTG バージョン 2.0 です。このバージョンでは、コアは SRP のデータラインパルシングのみをサポートしています。

ビット 19 **BSVLD** : B セッション有効

デバイスモードでのトランシーバのステータスを示します。

0 : Bセッションは無効。

1 : Bセッションは有効。

OTG モードでは、このビットを使用して、デバイスが接続されているか、切断されているかを判断できます。

注 : デバイスモードでのみアクセス可能です。

ビット 18 **ASVLD** : A セッション有効

ホストモードでのトランシーバのステータスを示します。

0 : Aセッションは無効。

1 : Aセッションは有効。

注 : ホストモードでのみアクセス可能です。

ビット 17 **DBCT** : ロング/ショートデバウンス時間

検出された接続のデバウンス時間を示します。

0 : ロングデバウンス時間 (100 ms + 2.5 μs)。物理的接続に使用されます。

1 : ショートデバウンス時間 (2.5 μs)。ソフト接続に使用されます。

注 : ホストモードでのみアクセス可能です。

ビット 16 CIDSTS : コネクタ ID ステータス

接続イベント時のコネクタ ID のステータスを示します。

0 : OTG_FS コントローラは A デバイスモードです。

1 : OTG_FS コントローラは B デバイスモードです。

注： デバイスモードとホストモードの両方でアクセス可能です。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 EHEN : 組み込みホスト有効化

OTG A デバイスのステートマシンと組み込みホストのステートマシンから選択するために使用します。

0 : OTG A デバイスのステートマシンを選択します。

1 : 組み込みホストのステートマシンを選択します。

ビット 11 DHNPEN : デバイス HNP 有効化

アプリケーションは、接続された USB ホストから SetFeature.SetHNPEnable コマンドを正常に受信したときに、このビットをセットします。

0 : HNP は、アプリケーションでは無効です。

1 : HNP は、アプリケーションで有効です。

注： デバイスモードでのみアクセス可能です。

ビット 10 HSHNPEN : ホストセット HNP 有効化

アプリケーションは、接続されたデバイスで HNP を正常に有効できたとき (SetFeature.SetHNPEnable コマンドを使用して)、このビットをセットします。

0 : ホストセット HNP は無効です。

1 : ホストセット HNP は有効です。

注： ホストモードでのみアクセス可能です。

ビット 9 HNPRQ : HNP リクエストあり

アプリケーションは、接続された USB ホストへの HNP リクエストを開始するために、このビットをセットします。アプリケーションは、OTG_GOTGINT レジスタのホストネゴシエーション成功ステータス変更ビット (OTG_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。

0 : HNP リクエストなし

1 : HNP リクエストあり

注： デバイスモードでのみアクセス可能です。

ビット 8 HNGSCS : ホストネゴシエーション成功

コアは、ホストネゴシエーションに成功したとき、このビットをセットします。コアは、このレジスタの HNP リクエストビット (HNPRQ) がセットされたとき、このビットをクリアします。

0 : ホストネゴシエーション失敗

1 : ホストネゴシエーション成功

注： デバイスモードでのみアクセス可能です。

ビット 7 BVALOVAL : B ペリフェラルセッション有効上書き値

このビットは、BVALOEN ビットがセットされている場合に、Bvalid 信号に上書き値をセットするために使用されます。

0 : BVALOEN = 0 のとき、Bvalid 値は“1”です。

1 : BVALOEN = 1 のとき、Bvalid 値は“1”です。

注： デバイスモードでのみアクセス可能です。

ビット 6 BVALOEN : B ペリフェラルセッション有効値の上書き有効化

このビットは、BVALOVAL ビットを使用して Bvalid 信号を上書きするソフトウェアを有効化/無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの Bvalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った Bvalid 信号は BVALOVAL ビット値で上書きされます。

注： デバイスモードでのみアクセス可能です。



ビット 5 AVALOVAL : A ペリフェラルセッション有効上書き値

このビットは、AVALOEN ビットがセットされている場合に、Avalid 信号に上書き値をセットするために使用されます。

0 : AVALOEN = 0 のとき、Avalid 値は“1”です。

1 : AVALOEN = 1 のとき、Avalid 値は“1”です。

注： ホストモードでのみアクセス可能です。

ビット 4 AVALOEN : A ペリフェラルセッション有効値の上書き有効化

このビットは、AVALOVAL ビットを使用して Avalid 信号を上書きするソフトウェアを有効化／無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの Avalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った Avalid 信号は AVALOVAL ビット値で上書きされます。

注： ホストモードでのみアクセス可能です。

ビット 3 VBVALOVAL : V_{BUS} 有効上書き値

このビットは、VBVALOEN ビットがセットされている場合に、vbusvalid 信号に上書き値をセットするために使用されます。

0 : VBVALOEN = 0 のとき、vbusvalid 値は“1”です。

1 : VBVALOEN = 1 のとき、vbusvalid 値は“1”です。

注： ホストモードでのみアクセス可能です。

ビット 2 VBVALOEN : V_{BUS} 有効値上書き有効化

このビットは、VBVALOVAL ビットを使用して vbusvalid 信号を上書きするソフトウェアを有効化／無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの vbusvalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った vbusvalid 信号は VBVALOVAL ビット値で上書きされます。

注： ホストモードでのみアクセス可能です。

ビット 1 SRQ : セッションリクエスト

アプリケーションは USB 上でセッションリクエストを開始するために、このビットをセットします。アプリケーションは、OTG_GOTGINT レジスタのホストネゴシエーション成功ステータス変更ビット (OTG_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。

USB 1.1 フルスピードシリアルトランシーバインタフェースを使用してセッションリクエストを開始する場合、アプリケーションは、このレジスタの B セッション有効ビット (OTG_GOTGCTL の BSVLD ビット) がクリアされた後 V_{BUS} が 0.2 V へ放電するまで待つ必要があります。この放電時間は PHY ごとに異なり PHY ベンダーから得ることができます。

0 : セッションリクエストなし

1 : セッションリクエスト

注： デバイスモードでのみアクセス可能です。

ビット 0 SRQSCS : セッションリクエスト成功

コアは、セッションリクエストの開始に成功したとき、このビットをセットします。

0 : セッションリクエスト失敗

1 : セッションリクエスト成功

注： デバイスモードでのみアクセス可能です。

29.15.2 OTG 割り込みレジスタ (OTG_GOTGINT)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アプリケーションは OTG 割り込みがあると、このレジスタを読み出し、このレジスタのビットをクリアすることによって OTG 割り込みをクリアします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID CHNG	DBC DNE	ADTO CHG	HNG DET	Res.
											rc_w1	rc_w1	rc_w1	rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HNSS CHG	SRSS CHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
						rc_w1	rc_w1						rc_w1		

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 IDCHNG :

このビットがセットされている場合、ID 入力ピンの値に変更があることを示します。

ビット 19 DBCDNE : デバウンス終了

コアは、デバイスが接続された後、デバウンスが完了したときに、このビットをセットします。アプリケーションは、この割り込みを確認した後に USB リセットの駆動を開始できます。このビットは、OTG_GUSBCFG レジスタの HNP 対応ビットまたは SRP 対応ビット (OTG_GUSBCFG の HNPCAP ビットと SRPCAP ビット) がセットされたときのみ有効です。

注: ホストモードでのみアクセス可能です。

ビット 18 ADTOCHG : A デバイスタイムアウト変更

コアは B デバイスの接続を待っている間に A デバイスがタイムアウトしたことを示すために、このビットをセットします。

注: デバイスモードとホストモードの両方でアクセス可能です。

ビット 17 HNGDET : ホストネゴシエーションの検出

コアは USB 上でホストネゴシエーションリクエストを検出したときに、このビットをセットします。

注: デバイスモードとホストモードの両方でアクセス可能です。

ビット 16:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 HNSSCHG : ホストネゴシエーション成功ステータス変化

コアは USB ホストネゴシエーションリクエストの成功または失敗時に、このビットをセットします。アプリケーションは、OTG_GOTGCTL レジスタのホストネゴシエーション成功ビット (OTG_GOTGCTL の HNGSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

注: デバイスモードとホストモードの両方でアクセス可能です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 8 SRSSCHG : セッションリクエスト成功ステータス変化

コアは、セッションリクエストの成功または失敗のときに、このビットをセットします。アプリケーションは、OTG_GOTGCTL レジスタのセッションリクエスト成功ビット (OTG_GOTGCTL の SRQSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

注: デバイスモードとホストモードの両方でアクセス可能です。

ビット 2 SEDET : セッション終了検出

コアは、このビットをセットして、V_{BUS} の電圧レベルが B ペリフェラルセッションにとって有効でなくなったことを示します (V_{BUS} < 0.8 V のとき)。

注: デバイスモードとホストモードの両方でアクセス可能です。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。



29.15.3 OTG AHB 設定レジスタ (OTG_GAHBCFG)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

このレジスタを使用して、電源投入後、またはモード変化後のコアを設定できます。このレジスタは、主に AHB システム関連の設定パラメータを含んでいます。初期プログラミングの後は、このレジスタを変更しないでください。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。

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

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PTXFELVL** : 周期的 Tx FIFO エンプティレベル

OTG_GINTSTS レジスタの周期的 Tx FIFO エンプティ割り込みビット (OTG_GINTSTS の PTXFE ビット) がトリガされるタイミングを示します。

0 : OTG_GINTSTS の PTXFE 割り込みビットは、周期的 Tx FIFO が半エンプティであることを示します。

1 : OTG_GINTSTS の PTXFE 割り込みビットは、周期的 Tx FIFO が完全エンプティであることを示します。

注 : ホストモードでのみアクセス可能です。

ビット 7 **TXFELVL** : Tx FIFO エンプティレベル

デバイスモードでは、このビットは、IN エンドポイント送信 FIFO エンプティ割り込み (OTG_DIEPINTx の TXFE) がトリガされるタイミングを示します。

0 : OTG_DIEPINTx の TXFE 割り込みは、IN エンドポイント Tx FIFO が半エンプティであることを示します。

1 : OTG_DIEPINTx の TXFE 割り込みは、IN エンドポイント Tx FIFO が完全エンプティであることを示します。

ホストモードでは、このビットは、非周期的 Tx FIFO エンプティ割り込み (OTG_GINTSTS の NPTXFE ビット) がトリガされるタイミングを示します。

0 : OTG_GINTSTS の NPTXFE 割り込みは、非周期的 Tx FIFO が半エンプティであることを示します。

1 : OTG_GINTSTS の NPTXFE 割り込みは、非周期的 Tx FIFO が完全エンプティであることを示します。

ビット 6:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **GINTMSK** : グローバル割り込みマスク

アプリケーションは、このビットを使用して、アプリケーション自身に対する割り込みラインのアサーションをマスクまたはマスク解除します。このビットの設定に関係なく、割り込みステータスレジスタはコアによって更新されます。

0 : アプリケーションへの割り込みアサーションをマスクします。

1 : アプリケーションへの割り込みアサーションをマスク解除します。

注 : デバイスモードとホストモードの両方でアクセス可能です。

29.15.4 OTG USB 設定レジスタ (OTG_GUSBCFG)

アドレスオフセット : 0x00C

リセット値 : 0x0000 1440

このレジスタを使用して、電源投入後、またはホストモードあるいはデバイスモードへの変化後に、コアを設定することができます。このレジスタは USB および USB-PHY 関連の設定パラメータを含んでいます。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。初期プログラミングの後には、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRDT				HNP CAP	SRP CAP	Res.	PHY SEL	Res.	Res.	Res.	TOTAL		
		rw				rw	rw		r				rw		

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **FDMOD** : 強制デバイスモード

このビットに 1 を書き込むと、コアは OTG_ID 入力ピンに関係なく、強制的にデバイスモードになります。

0 : 通常モード

1 : 強制デバイスモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 29 **FHMOD** : 強制ホストモード

このビットに 1 を書き込むと、コアは OTG_ID 入力ピンに関係なく、強制的にホストモードになります。

0 : 通常モード

1 : 強制ホストモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 28:26 予約済みであり、USB OTG HS および FS のリセット値に保持する必要があります。

ビット 25:15 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **TRDT** : USB ターンアラウンド時間

ターンアラウンド時間を PHY クロックで設定します。これらは、表 193 : TRDT 値 (FS) によってアプリケーション AHB 周波数により設定する必要があります。TRDT 値が高ければ、IN トークンへの USB 応答時間を伸ばして、データ FIFO へのより長い AHB 読み出しアクセス遅延を補正できます。

注 : デバイスモードでのみアクセス可能です。

ビット 9 **HNPCAP** : HNP 機能

アプリケーションは、このビットを使用して、OTG_FS コントローラの HNP 機能を制御します。

0 : HNP 機能は無効です。

1 : HNP 機能は有効です。

注 : デバイスモードとホストモードの両方でアクセス可能です。



ビット 8 **SRPCAP** : SRP 機能

アプリケーションは、このビットを使用して、OTG_FS コントローラの SRP 機能を制御します。コアが非 SRP 対応 B デバイスとして動作する場合、デバイスは、接続された A デバイス (ホスト) に V_{BUS} を起動し、セッションを開始させるように要求することはできません。
0 : SRP 機能は無効です。
1 : SRP 機能は有効です。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **PHYSEL** : フルスピードシリアルトランシーバの選択

このビットは常に 1 であり、アクセスは読み出し専用です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TOTAL** : FS タイムアウトの較正

アプリケーションがこのフィールドでプログラムする PHY クロックの数は、PHY によって導入される追加遅延を考慮して、コアのフルスピードパケット間タイムアウト時間に追加されます。ライン状態条件の生成時に PHY によって導入される遅延は PHY ごとに異なる場合があるので、較正が必要なことがあります。
フルスピード動作での USB の標準タイムアウト値は 16~18 ビット時間です。アプリケーションは、エニュメレーションのスピードに基づいて、このフィールドをプログラムする必要があります。PHY クロックごとに追加されるビット時間の数は 0.25 ビット時間です。

表 193. TRDT 値 (FS)

AHB 周波数範囲 (MHz)		TRDT 最小値
最小値	最大値	
14.2	15	0xF
15	16	0xE
16	17.2	0xD
17.2	18.5	0xC
18.5	20	0xB
20	21.8	0xA
21.8	24	0x9
24	27.5	0x8
27.5	32	0x7
32	-	0x6

29.15.5 OTG リセットレジスタ (OTG_GRSTCTL)

アドレスオフセット : 0x10

リセット値 : 0x8000 0000

アプリケーションは、このレジスタを使用して、コア内のさまざまなハードウェア機能をリセットします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM					TXF FLSH	RXF FLSH	Res.	FCRST	PSRST	CSRST
					rw					rs	rs		rs	rs	r

ビット 31 **AHBIDL** : AHB マスタアイドル
AHB マスタステートマシンがアイドル状態にあることを示します。
注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 30:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **TXFNUM** : Tx FIFO 番号
これは Tx FIFO フラッシュビットを使用して一掃しなければならない FIFO 番号です。コアが Tx FIFO フラッシュビットをクリアするまでは、このフィールドを変更しないでください。
00000 :
- ホストモードでの非周期的 Tx FIFO の一掃
- デバイスモードでの Tx FIFO 0 の一掃
00001 :
- ホストモードでの周期的 Tx FIFO の一掃
- デバイスモードでの Tx FIFO 1 の一掃
00010 : デバイスモードでの Tx FIFO 2 の一掃
...
01111 : デバイスモードでの Tx FIFO 15 の一掃
10000 : デバイスモードまたはホストモードのすべての送信 FIFO をフラッシュします。
注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 5 **TXFFLSH** : Tx FIFO の一掃
このビットは、単一またはすべての送信 FIFO を選択的にフラッシュしますが、コアがトランザクション中の場合はフラッシュできません。
アプリケーションは、コアが Tx FIFO への書き込み中か Tx FIFO からの読み出し中のどちらでもないことを確認した後でなければ、このビットに書き込むことはできません。以下のレジスタを使用して確認してください。
読み出し — NAK 有効割り込みを使用すると、コアが FIFO から読み出しを行っていないことを確認できます。
書き込み — OTG_GRSTCTL の AHBIDL ビットで、コアが FIFO に書き込みを行っていないことを保証します。
FIFO が再設定される場合は通常一掃を実施することをお勧めします。デバイスエンドポイントが無効の間も、FIFO 一掃の実施をお勧めします。アプリケーションは、コアがこのビットをクリアするまで待ってから、他の操作を実行しなければなりません。phy_clk または hclk のより低速なクロックを使用してこのビットをクリアするには、8 クロックかかります。
注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 4 RXFFLSH : Rx FIFO の一掃

アプリケーションは、このビットを使用して Rx FIFO 全体を一掃できますが、まず、コアがトランザクション中でないことを確認する必要があります。

アプリケーションは、コアが Rx FIFO からの読み出し中、または Rx FIFO への書き込み中のどちらでもないことを確認した後でなければ、このビットに書き込んでなりません。

アプリケーションは、他の操作を実行する前に、このビットがクリアされるまで待つ必要があります。このビットがクリアされるまで、8 クロック (PHY または AHB クロックの中で最も低速なクロック) かかります。

注： デバイスモードとホストモードの両方でアクセス可能です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 FCRST : ホストフレームカウンタリセット

アプリケーションは、このビットに書き込むことによって、コア内のフレーム数カウンタをリセットします。フレームカウンタがリセットされると、コアによって次に送出される SOF のフレーム番号は 0 になります。

アプリケーションでこのビットに 1 が書き込まれた場合、このビットは数クロックサイクル内にコアによってクリアされますので、その値を読み戻すことはできません。

注： ホストモードでのみアクセス可能です。

ビット 1 PSRST : 部分的なソフトリセット

内部ステートマシンをリセットしますが、エニユメレーション情報は保持されます。特定の PHY エラーからの復帰に使用できます。

注： デバイスモードとホストモードの両方でアクセス可能です。

ビット 0 CSRST : コアソフトリセット

以下のように HCLK および PHY クロックドメインをリセットします。

割り込みと、以下のビットを除くすべての CSR レジスタビットをクリアします。

- OTG_PCGCCTL レジスタ の GATEHCLK ビット
- OTG_PCGCCTL レジスタ の STPPCLK ビット
- OTG_HCFG レジスタ の FSLSPCS ビット
- OTG_DCFG レジスタ の DSPD ビット
- OTG_DCTL レジスタ の SDIS ビット
- OTG_GCCFG レジスタ

すべてのモジュールのステートマシン (AHB スレーブユニット用を除く) がアイドル状態にリセットされ、すべての送信 FIFO および受信 FIFO が一掃されます。

AHB 転送の最後のデータフェーズの完了後 AHB マスタ上のすべてのトランザクションは速やかに終了されます。USB 上のすべてのトランザクションをただちに終了します。

アプリケーションは、いつでもこのビットに書き込むことによって、コアをリセットできます。これはセルフクリアビットであり、コア内のすべての必要なロジックがリセットされた後、コアがこのビットをクリアします。コアの現在の状態によって、クリアには数クロックかかります。このビットがクリアされると、ソフトウェアは PHY ドメインにアクセスできるようになるまで少なくとも 3 PHY クロック待つ必要があります (同期遅延)。また、ソフトウェアは、操作を開始する前に、このレジスタのビット 31 が 1 にセットされていること (AHB マスタがアイドル状態) を確認する必要があります。

一般に、ソフトウェアリセットは、ソフトウェア開発時に使用され、上記の USB 設定レジスタの PHY 選択ビットを動的に変更するときにも使用されます。PHY を変更するときには PHY の対応クロックが選択され PHY ドメインで使用されます。新しいクロックが選択されたら、適切な動作のためには PHY ドメインをリセットする必要があります。

注： デバイスモードとホストモードの両方でアクセス可能です。

29.15.6 OTG コア割り込みレジスタ (OTG_GINTSTS)

アドレスオフセット : 0x014

リセット値 : 0x1400 0020

このレジスタは、現在のモード（デバイスモードまたはホストモード）において、システムレベルのイベントのためにアプリケーションに割り込みをかけます。

このレジスタには、ホストモードでのみ有効なビットと、デバイスモードでのみ有効なビットがあります。このレジスタは、現在のモードも示します。rc_w1 タイプの割り込みステータスビットをクリアするには、アプリケーションは、そのビットに 1 を書き込む必要があります。

FIFO ステータス割り込みは読み出し専用です。これらの割り込みの処理中に、ソフトウェアが FIFO への書き込みまたは FIFO からの読み出しを行うと、FIFO 割り込み条件は自動的にクリアされます。

初期化の前に割り込みが発生するのを避けるためには、アプリケーションは、割り込みビットのマスクを解除する前に OTG_GINTSTS レジスタをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	LPM INT	PTXFE	HCINT	HPRT INT	RST DET	Res.	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	rc_w1		rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUM DNE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXF E	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

ビット 31 **WKUPINT** : レジューム／リモートウェイクアップ検出割り込み

サスペンド (L2) または LPM (L1) 状態中のウェイクアップ割り込み

– サスペンド (L2) 中

デバイスモードでは、この割り込みは、USB 上でレジュームが検出されたときにアサートされます。ホストモードでは、この割り込みは USB 上でリモートウェイクアップが検出されたときにアサートされます。

– LPM (L1) 中

この割り込みは USB 上でホストから開始されたレジュームまたはデバイスから開始されたリモートウェイクアップのいずれかに対してアサートされます。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 30 **SRQINT** : セッションリクエスト／新規セッション検出割り込み

ホストモードでは、この割り込みは、デバイスからのセッションリクエストが検出されたときにアサートされます。デバイスモードでは、この割り込みは、V_{BUS} が B ペリフェラルデバイスに対して有効な範囲にあるときにアサートされます。デバイスモードとホストモードの両方でアクセス可能です。

ビット 29 **DISCINT** : 切断検出割り込み

デバイスの切断が検出されたときにアサートされます。

注 : ホストモードでのみアクセス可能です。

ビット 28 **CIDSCHG** : コネクタ ID ステータス変化

コアはコネクタ ID ステータスに変化があったとき、このビットをセットします。

注 : デバイスモードとホストモードの両方でアクセス可能です。



ビット 27 LPMINT : LPM 割り込み

デバイスモードでは、この割り込みは、デバイスが LPM トランザクションを受信し、非エラー応答で応答したときにアサートされます。

ホストモードでは、この割り込みは、デバイスが LPM トランザクションに非エラー応答で応答したとき、またはホストのコアがプログラムした回数 (OTG_GLPMCFG の RETRYCNT) の LPM トランザクションを完了したときにアサートされます。

このフィールドは、OTG_GLPMCFG の LPMEN ビットが 1 にセットされたときにのみ有効です。

ビット 26 PTXFE : 周期的 TxFIFO エンプティ

周期的送信 FIFO が半エンプティか完全エンプティであり、周期的リクエストキューに少なくとも 1 つのエントリを書き込むためのスペースがあるときにアサートされます。半エンプティか完全エンプティであるかは、OTG_GAHBCFG レジスタの周期的 TxFIFO エンプティレベルビット (OTG_GAHBCFG の PTXFELVL ビット) によって判断されます。

注: ホストモードでのみアクセス可能です。

ビット 25 HCINT : ホストチャンネル割り込み

コアは、コアのチャンネルの 1 つで割り込みがペンディング中であることを示すために、このビットをセットします (ホストモード)。アプリケーションは、OTG_HAINT レジスタを読み出して、割り込みが発生しているチャンネルの正確な数を確認した後、対応する OTG_HCINTx レジスタを読み出して、割り込みの正確な原因を確認する必要があります。アプリケーションは OTG_HCINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

注: ホストモードでのみアクセス可能です。

ビット 24 HPRTINT : ホストポート割り込み

コアは、ホストモードの OTG_FS コントローラポートの 1 つのポートステータスに変化があったことを示すために、このビットをセットします。アプリケーションは、OTG_HPRT レジスタを読み出して、この割り込みの原因となった正確なイベントを特定する必要があります。アプリケーションは OTG_HPRT レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

注: ホストモードでのみアクセス可能です。

ビット 23 RSTDET : リセット検出割り込み

デバイスモードでは、この割り込みは、デバイスがサスペンド状態にあるときに部分パワーダウンモードで USB 上にリセットが検出されたときにアサートされます。

注: デバイスモードでのみアクセス可能です。

ビット 22 予約済みであり、リセット値に保持する必要があります。**ビット 21 IPXFR : 不完全な周期的転送**

ホストモードでは、コアは、現在のフレームでスケジュールされていて、まだペンディング中の不完全な周期的トランザクションが存在するとき、この割り込みビットをセットします。

INCOMPISOOUT : 不完全アイソクロナス OUT 転送

デバイスモードでは、コアは、現在のフレームで転送が完了していないアイソクロナス OUT エンドポイントが少なくとも 1 つあることを示すために、この割り込みをセットします。この割り込みは、このレジスタの周期的フレーム割り込み終了 (EOPF) ビットとともにアサートされます。

ビット 20 ISOIXFR : 不完全アイソクロナス IN 転送

コアは、現在のフレームで転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあることを示すために、この割り込みをセットします。この割り込みは、このレジスタの周期的フレーム割り込み終了 (EOPF) ビットとともにアサートされます。

注: デバイスモードでのみアクセス可能です。

ビット 19 OEPINT : OUT エンドポイント割り込み

コアは、コアの OUT エンドポイントの 1 つで割り込みがペンディングされていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG_DAINレジスタを読み出して、割り込みが発生した OUT エンドポイントの正確な数を確認した後、対応する OTG_DOEPINTx レジスタを読み出して、割り込みの正確な原因を特定する必要があります。アプリケーションは、対応する OTG_DOEPINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

注: デバイスモードでのみアクセス可能です。

ビット 18 IEPINT : IN エンドポイント割り込み

コアは、コアの IN エンドポイントの 1 つで割り込みが保留されていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG_DAINレジスタを読み出して、割り込みが発生した IN エンドポイントの正確な数を確認した後、対応する OTG_DIEPINTx レジスタを読み出して、割り込みの正確な原因を特定する必要があります。アプリケーションは、対応する OTG_DIEPINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

注: デバイスモードでのみアクセス可能です。

ビット 17:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 EOPF : 周期的フレーム終了割り込み

現在のフレームで、OTG_DCFG レジスタの周期的フレームインターバルフィールド (OTG_DCFG の PFIVL ビット) で指定された時間に達したことを示します。

注: デバイスモードでのみアクセス可能です。

ビット 14 ISOODRP : アイソクロナス OUT パケット欠落割り込み

コアは、アイソクロナス OUT エンドポイントの最大サイズのバケットに対応できるだけの十分なスペースが Rx FIFO になかったために Rx FIFO にアイソクロナス OUT パケットを書き込めなかったとき、このビットをセットします。

注: デバイスモードでのみアクセス可能です。

ビット 13 ENUMDNE : エニユメレーション終了

コアは、スピードのエニユメレーションが完了したことを示すために、このビットをセットします。アプリケーションは、OTG_DSTS レジスタを読み出して、エニユメレーションされた速度を取得する必要があります。

注: デバイスモードでのみアクセス可能です。

ビット 12 USBRST : USB リセット

コアは USB 上でリセットが検出されたことを示すために、このビットをセットします。

注: デバイスモードでのみアクセス可能です。

ビット 11 USBSUSP : USB サスペンド

コアは USB 上でサスペンドが検出されたことを示すために、このビットをセットします。コアは、データライン上で長時間アクティビティが 1 つもなかった場合、サスペンド状態に入ります。

注: デバイスモードでのみアクセス可能です。

ビット 10 ESUSP : アーリーサスペンド

コアは USB 上で 3 ms 間、アイドル状態が検出されたことを示すために、このビットをセットします。

注: デバイスモードでのみアクセス可能です。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 GONAKEFF : グローバル OUT NAK 有効

アプリケーションによってセットされた OTG_DCTL レジスタのセットグローバル OUT NAK ビット (OTG_DCTL の SGONAK ビット) がコアで有効になったことを示します。このビットは、OTG_DCTL レジスタのクリアグローバル OUT NAK ビット (OTG_DCTL の CGONAK ビット) に書き込むことによってクリアできます。

注: デバイスモードでのみアクセス可能です。



ビット 6 GINAKEFF : グローバルIN非周期的NAK 有効

アプリケーションによってセットされた OTG_DCTL レジスタのセットグローバル 非周期的 IN NAK ビット (OTG_DCTL の SGINAK ビット) がコアで有効になったことを示します。すなわち、コアは、アプリケーションによってセットされたグローバル IN NAK ビットをサンプリングしました。このビットは、OTG_DCTL レジスタのクリアグローバル非周期的 IN NAK ビット (OTG_DCTL の CGINAK ビット) をクリアすることによってクリアできます。

この割り込みは、必ずしも NAK ハンドシェイクが USB に送出されることを意味しません。STALL ビットが NAK ビットに優先します。

注: デバイスモードでのみアクセス可能です。

ビット 5 NPTXFE : 非周期的 Tx FIFO エンプティ

この割り込みは、非周期的 Tx FIFO が半エンプティか完全エンプティであり、非周期的送信リクエストキューに、少なくとも 1 つのエントリを書き込むスペースがあるときにアサートされます。半エンプティか完全エンプティであるかは、OTG_GAHBCFG レジスタの非周期的 Tx FIFO エンプティレベル ビット (OTG_GAHBCFG の TXFELVL ビット) によって判断されます。

注: ホストモードでのみアクセス可能です。

ビット 4 RXFLVL : Rx FIFO 非エンプティ

Rx FIFO から読み出されるべきペンディング中のパケットが少なくとも 1 つあることを示します。

注: ホストモードとデバイスモードの両方でアクセス可能です。

ビット 3 SOF : フレーム開始

ホストモードでは、コアは SOF (FS) またはキープアライブ (LS) が USB で送信されることを示すために、このビットをセットします。アプリケーションは、このビットに 1 を書き込むことによって、割り込みをクリアする必要があります。

デバイスモードでは、コアは、USB 上で SOF トークンが受信されたことを示すために、このビットをセットします。アプリケーションは、OTG_DSTS レジスタを読み出すことによって、現在のフレーム番号を知ることができます。この割り込みは、コアが FS で動作しているときのみ確認できます。

注: このレジスタはパワーオンリセット後すぐに読み出されると“1”を返すことがあります。パワーオンリセット後すぐに“1”が読み出された場合、そのレジスタのビットが、SOF が送信された (ホストモードの場合) または SOF が受信された (デバイスモードの場合) ことを示すわけではなく、この割り込みの読み出し値は、ホストとデバイス間の有効な接続が確立されて初めて有効になります。パワーオンリセット後にビットがセットされた場合、アプリケーションはそのビットをクリアすることができます。

注: ホストモードとデバイスモードの両方でアクセス可能です。

ビット 2 OTGINT : OTG 割り込み

コアは、OTG プロトコルイベントが完了したことを示すために、このビットをセットします。アプリケーションは OTG 割り込みステータスレジスタ (OTG_GOTGINT) を読み出して、この割り込みの原因となった正確なイベントを確認する必要があります。アプリケーションは OTG_GOTGIN レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

注: ホストモードとデバイスモードの両方でアクセス可能です。

ビット 1 MMIS : モード不一致割り込み

コアは、アプリケーションが以下へのアクセスを試みているときに、このビットをセットします。

- ホストモードレジスタ。コアがデバイスモードで動作しているとき。
- デバイスモードレジスタ。コアがホストモードで動作しているとき。

レジスタへのアクセスは、OKAY 応答によって AHB 上で完了しますが、内部的にはコアによって無視され、コアの動作には影響を与えません。

注: ホストモードとデバイスモードの両方でアクセス可能です。

ビット 0 CMOD : 現在の動作モード

現在の動作モードを示します。

- 0 : デバイスモード
- 1 : ホストモード

注: ホストモードとデバイスモードの両方でアクセス可能です。

29.15.7 OTG 割り込みマスクレジスタ (OTG_GINTMSK)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタは、コア割り込みレジスタとともに働き、アプリケーションに割り込みをかけます。割り込みビットがマスクされると、そのビットに関連する割り込みは生成されません。ただし、その割り込みに対応するコア割り込みレジスタ (OTG_GINTSTS) のビットは、セットされたままです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSC HGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDE TM	Res.	IPXFR M/IISO OXFR M	IISOIX FRM	OEPINT	IEPINT	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	r	rw		rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFM	ISOOD RPM	ENUM DNEM	USBRST	USBSU SPM	ESUSPM	Res.	Res.	GONA KEFFM	GINAK EFFM	NPTXFE M	RXFLV LM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

ビット 31 **WUIM** : レジューム／リモートウェイクアップ検出割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードとデバイスモードの両方でアクセス可能です。

ビット 30 **SRQIM** : セッションリクエスト／新規セッション検出割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードとデバイスモードの両方でアクセス可能です。

ビット 29 **DISCINT** : 切断検出割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードでのみアクセス可能です。

ビット 28 **CIDSC HGM** : コネクタ ID ステータス変化マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードとデバイスモードの両方でアクセス可能です。

ビット 27 **LPMINTM** : LPM 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードとデバイスモードの両方でアクセス可能です。

ビット 26 **PTXFEM** : 周期的 Tx FIFO エンプティマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードでのみアクセス可能です。

ビット 25 **HCIM** : ホストチャンネル割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードでのみアクセス可能です。



ビット 24 **PRTIM** : ホストポート割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **ホストモードでのみアクセス可能です。**

ビット 23 **RSTDETM** : リセット検出割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **IPXFRM** : 不完全周期的転送マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **ホストモードでのみアクセス可能です。**

IISOXFRM : 不完全アイソクロナス OUT 転送マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 20 **IISOXFRM** : 不完全アイソクロナス IN 転送マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 19 **OEPINT** : OUT エンドポイント割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 18 **IEPINT** : IN エンドポイント割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 17:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EOPFM** : 周期的フレーム終了割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 14 **ISOODRPM** : アイソクロナス OUT パケット欠落割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 13 **ENUMDNEM** : エニユメレーション終了マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 12 **USBRS** : USB リセットマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : **デバイスモードでのみアクセス可能です。**

ビット 11 **USBSUSPM** : USB サスペンドマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードでのみアクセス可能です。

ビット 10 **ESUSPM** : アーリーサスペンドマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードでのみアクセス可能です。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GONAKEFFM** : グローバル OUT NAK 有効マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードでのみアクセス可能です。

ビット 6 **GINAKEFFM** : グローバル非周期的IN NAK 有効マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードでのみアクセス可能です。

ビット 5 **NPTXFEM** : 非周期的 Tx FIFO エンプティマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : ホストモードでのみアクセス可能です。

ビット 4 **RXFLVLM** : 受信 FIFO非エンプティマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 3 **SOFM** : フレーム開始マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 2 **OTGINT** : OTG 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 1 **MMISM** : モード不一致割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

29.15.8 OTG_FS 受信ステータスデバッグ読み出し／OTGステータス読み出しおよびポップレジスタ (OTG_GRXSTSR/OTG_GRXSTSP)

読み出しアドレスオフセット : 0x01C

ポップアドレスオフセット : 0x020

リセット値 : 0x0000 0000

受信ステータスデバッグ読み出しレジスタを読み出すと、受信 FIFO の最上部の内容が返されます。受信ステータス読み出しおよびポップレジスタを読み出すと Rx FIFO の最上部のデータエントリも取り出され（ポップされ）ます。

受信ステータスの内容は、ホストモードとデバイスモードで異なった解釈をする必要があります。コアは、受信 FIFO がエンプティのときには、受信ステータスのポップ／読み出しを無視して、0x0000 0000 の値を返します。アプリケーションは、コア割り込みレジスタの受信 FIFO 非エンプティビット (OTG_GINTSTS の RXFLVL ビット) がアサートされたときにのみ、受信ステータス FIFO をポップしなければなりません。

ホストモード :

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

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:17 **PKTSTS** : パケットステータス

- 受信パケットのステータスを示します。
- 0010 : IN データパケットが受信されました。
- 0011 : IN 転送が完了しました（割り込みをトリガします）。
- 0101 : データトグルエラー（割り込みをトリガします）。
- 0111 : チャネル停止（割り込みをトリガします）。
- その他 : 予約済み

ビット 16:15 **DPID** : データ PID

- 受信パケットのデータ PID を示します。
- 00 : データ 0
- 10 : データ 1

ビット 14:4 **BCNT** : バイトカウント

受信した IN データパケットのバイト数を示します。

ビット 3:0 **CHNUM** : チャネル番号

現在の受信パケットが属するチャネル番号を示します。

デバイスモード :

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:21 **FRMNUM** : フレーム番号

これは USB 上でパケットが受信されるフレーム番号の下位 4 ビットです。このフィールドは、アイソクロナス OUT エンドポイントがサポートされる場合にのみサポートされます。

ビット 20:17 **PKTSTS** : パケットステータス

受信パケットのステータスを示します。

0001 : グローバル OUT NAK (割り込みをトリガします)。

0010 : OUT データパケットが受信されました。

0011 : OUT 転送が完了しました (割り込みをトリガします)。

0100 : SETUP トランザクションが完了しました (割り込みをトリガします)。

0110 : SETUP データパケットが受信されました。

その他 : 予約済み

ビット 16:15 **DPID** : データ PID

受信した OUT データパケットのデータ PID を示します。

00 : データ 0

10 : データ 1

ビット 14:4 **BCNT** : バイトカウント

受信したデータパケットのバイト数を示します。

ビット 3:0 **EPNUM** : エンドポイント番号

現在の受信パケットが属するエンドポイント番号を示します。

29.15.9 OTG 受信 FIFO サイズレジスタ (OTG_GRXFSIZ)

アドレスオフセット : 0x024

リセット値 : 0x0000 0200

アプリケーションは RxFIFO に割り当てられなければならない RAM サイズをプログラムできます。

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



ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RXFD** : Rx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

29.15.10 OTG ホスト非周期的送信 FIFOサイズレジスタ (OTG_HNPTXFSIZ) /
エンドポイント 0 送信 FIFO サイズ (OTG_DIEPTXF0)

アドレスオフセット : 0x028

リセット値 : 0x0200 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTXFD/TX0FD															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSA/TX0FSA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ホストモード

ビット 31:16 **NPTXFD** : 非周期的 Tx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

ビット 15:0 **NPTXFSA** : 非周期的送信 RAM 開始アドレス

このフィールドは、非周期的送信 FIFO RAM のメモリ開始アドレスを設定します。

デバイスモード

ビット 31:16 **TX0FD** : エンドポイント 0 TxFIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

ビット 15:0 **TX0FSA** : エンドポイント 0 送信 RAM 開始アドレス

このフィールドは、エンドポイント 0 送信 FIFO RAM のメモリ開始アドレスを設定します。

29.15.11 OTG 非周期的送信 FIFO／キューステータスレジスタ (OTG_HNPTXSTS)

アドレスオフセット : 0x02C

リセット値 : 0x0008 0200

注 : デバイスモードでは、このレジスタは無効です。

この読み出し専用レジスタは、非周期的 Tx FIFO および非周期的送信リクエストキューの空きスペース情報を含みます。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **NPTXQTOP** : 非周期的送信リクエストキューの先頭

MAC によって処理中の非周期的 Tx リクエストキューのエントリ。

ビット 30:27チャンネル／エンドポイント番号

ビット 26:25

00 : IN/OUT トークン

01 : ゼロ長送信パケット (デバイス IN／ホスト OUT)

11 : チャンネル停止コマンド

ビット 24 : 終了 (選択されたチャネル／エンドポイントの最後のエントリ)

ビット 23:16 **NPTQXSAV** : 非周期的送信リクエストキューの使用可能スペース

非周期的送信リクエストキューの使用可能な空きスペースの量を示します。このキューは IN および OUT リクエストの両方を保持します。

0 : 非周期的送信リクエストキューは満杯です。

1 : 1 位置が使用可能です。

2 : 2 位置が使用可能です。

n : n 位置が使用可能です (0 ≤ n ≤ 8)。

その他 : 予約済み

ビット 15:0 **NPTXFSAV** : 使用可能な非周期的 Tx FIFO スペース

非周期的 Tx FIFO 内で使用できる空きスペースの量を示します。

値は 32 ビットワード単位です。

0 : 非周期的 Tx FIFO は満杯です。

1 : 1 ワードが使用可能です。

2 : 2 ワードが使用可能です。

n : n ワードが使用可能です (0 ≤ n ≤ 512)

その他 : 予約済み

29.15.12 OTG 一般コア設定レジスタ (OTG_GCCFG)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBDEN	SDEN	PDEN	DCD EN	BCDEN	PWR DWN
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DCDET
												rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **VBDEN** : USB V_{BUS} 検出有効化

V_{BUS} レベル検出用コンパレータを有効にして、V_{BUS} の有効レベルを検出します (USB ホストおよびデバイス動作用の V_{BUS} パッド上で)。HNP または SRP サポートが有効であれば、V_{BUS} コンパレータは VBDEN 値とは関係なく、自動的に有効化されます。

0 : V_{BUS} 検出無効

1 : V_{BUS} 検出有効

ビット 20 **SDEN** : 2 次検出 (SD) モードイネーブル

このビットは、BCD を SD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 19 **PDEN** : 1 次検出 (PD) モードイネーブル

このビットは、BCD を PD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 18 **DCDEN** : データ接触検出 (DCD) モードイネーブル

このビットは、BCD を DCD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 17 **BCDEN** : バッテリ充電検出 (BCD) イネーブル

このビットは、USB デバイス内の BCD サポートを有効にするために、ソフトウェアによってセットされます。有効にすると、USB PHY は BCD によって完全に制御され、通常の通信では使用できなくなります。BCD の検出完了後は、通常の USB 動作を可能にするために、このビットを "0" にクリアして BCD を OFF モードにします。

ビット 16 **PWRDWN** : パワーダウン制御

送信／受信のためにトランシーバを起動するために使用されます。リセットされている場合、トランシーバはパワーダウン状態に保たれます。セットする場合、BCD 機能をオフにする必要があります (BCDEN=0)。

0 : USB FS トランシーバ無効

1 : USB FS トランシーバ有効

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

- ビット 3 **PS2DET** : DM プルアップ検出ステータス
- このビットは、PD 中のみアクティブになって DM 電圧レベルと VLGC 閾値の比較の結果を示します。通常の状況では、DM レベルは閾値を下回る必要があります。閾値を上回る場合は、DM が外部的にハイレベルにプルアップされることを意味します。これは、PS2 ポート (DP および DM ラインの両方をプルアップする) への接続、または BCD 仕様に準拠しない一部の独自の充電器への接続によって発生します。
- 0 : 通常のポートが検出されました (SDP、CDP、または DCP に接続)。
1 : PS2 ポートまたは独自の充電器が検出されました。
- ビット 2 **SDET** : 2 次検出 (SD) ステータス
- このビットは、SD の結果を示します。
- 0 : CDP が検出されました。
1 : DCP が検出されました。
- ビット 1 **PDET** : 1 次検出 (PD) ステータス
- このビットは、PD の結果を示します。
- 0 : BCD サポートは検出されませんでした (SDP または独自デバイスに接続)。
1 : BCD のサポートが検出されました (CDP、または DCP に接続)。
- ビット 0 **DCDET** : データ接触検出 (DCD) ステータス
- このビットは、DCD の結果を示します。
- 0 : データラインの接触は検出されませんでした。
1 : データラインの接触が検出されました。

29.15.13 OTG コア ID レジスタ (OTG_CID)

アドレスオフセット : 0x03C

リセット値 : 0x0000 2000

これは、製品 ID を含む読み出し専用レジスタです。

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

ビット 31:0 **PRODUCT_ID** : 製品 ID フィールド

アプリケーションによってプログラム可能な ID フィールド。

29.15.14 OTG コア LPM 設定レジスタ (OTG_GLPMCFG)

アドレスオフセット : 0x54

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EN BESL	LPMRCNTSTS			SND LPM	LPMRCNT			LPMCHIDX				L1RSM OK
			rw	r	r	r	rs	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP STS	LPMRSP		L1DS EN	BESLTHRS				L1SS EN	REM WAKE	BESL				LPM ACK	LPM EN
r	r	r	rw	rw	rw	rw	rw	rw	rw/r	rw/r	rw/r	rw/r	rw/r	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ENBESL** : ベストエフォート型サービス遅延の有効化

このビットは LPM 正誤表で定義されているように BESL 機能を有効にします。

0 : コアは以下の文書に記載されているように機能します。

USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007

1 : コアは LPM 正誤表に記載されているように機能します。

USB 2.0 ECN 用の正誤表 : Link Power Management (LPM) - 7/2007

注 : この文書では更新された動作 (LPM 正誤表に記載) のみが考慮されているため、アプリケーション切り替えで ENBESL ビットを 1 にセットする必要があります。

ビット 27:25 **LPMRCNTSTS** : LPM 再試行カウンタステータス

現在の LPM シーケンスでは、送信すべき LPM ホストの再試行回数がまだ残っています。

注 : ホストモードでのみアクセス可能です。

ビット 24 **SNDLPM** : 送信 LPM トランザクション

アプリケーションソフトウェアがこのビットをセットすると、2 つのトークン EXT および LPM を含む LPM トランザクションが送信されます。デバイスから有効な応答 (STALL、NYET、または ACK) を受信した時点、またはコアがプログラムされた回数の LPM 再試行を送信し終えた時点で、このビットはハードウェアによってクリアされます。

注 : このビットは、ホストがローカルポートに接続されている場合のみセットする必要があります。

注 : ホストモードでのみアクセス可能です。

ビット 23:21 **LPMRCNT** : LPM 再試行カウント

デバイスが ERROR 応答を返すと、このビットは追加の LPM 再試行回数となります。追加の再試行は、有効なデバイスからの応答 (STALL、NYET、または ACK) が受信されるまで、ホストによって実行されます。

注 : ホストモードでのみアクセス可能です。

ビット 20:17 **LPMCHIDX** : LPM チャネルインデックス

ローカルデバイスへの LPM トランザクションの送信中に LPM トランザクションを適用すべきチャネルの番号を示します。LPM チャネルインデックスに基づき、コアは自動的に対応するチャネルでプログラムされたデバイスアドレスとエンドポイント番号を LPM トランザクションに挿入します。

注 : ホストモードでのみアクセス可能です。

ビット 16 L1RSMOK : スリープ状態レジューム OK

デバイスまたはホストがスリープ状態からレジュームを開始できることを示します。このビットは LPM スリープ状態 (L1) のときに有効です。このビットは 50 μ s ($T_{L1Residency}$) の遅延後に SLEEP モードでセットされます。

このビットは、SLPSTS = 0 のときにリセットされます。

1 : アプリケーションまたはホストはスリープ状態からレジュームを開始できます。

0 : アプリケーションまたはホストはスリープ状態からレジュームを開始できません。

ビット 15 SLPSTS : ポートスリープステータス**デバイスモード :**

このビットは、USB バスにスリープ状態が存在する限りセットされます。コアは、ACK 応答が LPM トランザクションに送信され、 $T_{L1TokenRetry}$ タイマが時間切れになった時点で、スリープ状態に入ります。PHY クロックを停止させるには、アプリケーションで OTG_PCGCCTL の STPPCLK ビットをセットすることにより、PHY サスペンド入力信号をアサートする必要があります。

アプリケーションはスリープへの遷移を確認する際に、LPMRSP の ACK ではなく、SLPSTS を信頼する必要があります。

コアは、以下の場合にスリープ状態から抜け出します。

- USB ライン上で何らかのアクティビティがあったとき。
- アプリケーションが OTG_DCTL の RWUSIG ビットへ書き込むときやデバイスをリセットまたはソフト切断するとき。

ホストモード :

デバイスから ACK 応答を受け取ったローカルポートへのコアによる LPM トランザクションの成功の副作用として、ホストはスリープ状態 (L1) へ遷移します。このビットの読み出し値は、ポートの現在のスリープ状態を反映しています。

コアは、次の動作後にこのビットをクリアします。

- コアがリモート L1 ウェイクアップ信号を検出します、
- アプリケーションで、OTG_HPRT レジスタの PRST ビットまたは PRES ビットをセットします、または
- アプリケーションで、コア割り込みレジスタの L1 レジューム/リモートウェイクアップ検出割り込みビットまたは切断検出割り込みビット (OTG_GINTSTS の WKUPINT または DISCINT ビット) をセットします。

0 : コアは L1 状態ではありません。

1 : コアは L1 状態です。

ビット 14:13 LPMRST : LPM 応答**デバイスモード :**

受信した LPM トランザクションへのコアの応答は、これら 2 つのビットに反映されます。

ホストモード :

ローカルデバイスから受け取った LPM トランザクションへのハンドシェイク応答

- 11 : ACK
- 10 : NYET
- 01 : STALL
- 00 : ERROR (ハンドシェイク応答なし)

ビット 12 L1DSEN : L1 ディープスリープ有効化

L1 SLEEP モードにおける PHY のサスペンドを有効にします。L1 SLEEP モード時に最大限の節電を行うために、このビットはいかなる場合もアプリケーション切り替えによって 1 にセットする必要があります。

ビット11:8 **BESLTHRS** : BESL 閾値

デバイスモード :

BESL 値がこの BESL_Thres[3:0] フィールドの規定値以上である場合、コアは PHY を L1 状態でディープ低電力モードに移行させます。

ホストモード :

コアは PHY を L1 状態でディープ低電力モードに移行させます。BESLTHRS[3:0] は、デバイスから開始されたレジュームを検出した場合に、レジュームシグナリングに USB バス上のホスト (T_{L1HubDrvResume2}) が反映されるまでの時間を指定します。

ホストモードでは、BESLTHRS には 1100b より大きい値を設定してはなりません。この値が T_{L1HubDrvResume2}の最大値を超えてしまうからです。

Thres[3:0]ホストモードレジュームシグナリング時間 (μs)

0000 : 75

0001 : 100

0010 : 150

0011 : 250

0100 : 350

0101 : 450

0110 : 950

その他の値 : 予約済み

ビット7 **L1SSEN** : L1 シャロースリープ有効化

L1 SLEEP モードにおける PHY のサスペンドを有効にします。L1 SLEEP モード時に最大限の節電を行うために、このビットはいかなる場合もアプリケーション切り替えによって 1 にセットする必要があります。

ビット6 **REMWAKE** : bRemoteWake 値

ホストモード :

LPM トランザクションの wIndex フィールドの送信されるリモートウェイクアップの値

デバイスモード (読み出し専用) :

このフィールドは、ACK、NYET、または STALL 応答が LPM トランザクションに送信される場合に、受信した LPM トークン bRemoteWake bmAttribute で更新されます。

ビット 5:2 BESL : ベストエフォート型サービス遅延**ホストモード :**

LPM トランザクションの送信される BESL の値この値は、ホストから開始されたレジュームの処理時間 ($T_{L1HubDrvResume1}$) の間にレジュームを開始するためにも使用されます。

デバイスモード (読み出し専用) :

このフィールドは、ACK、NYET、または STALL 応答が LPM トランザクションに送信される場合に、受信した LPM トークン BESL bmAttribute で更新されます。

BESL[3:0] T_{BESL} (μ s)

0000 : 125
0001 : 150
0010 : 200
0011 : 300
0100 : 400
0101 : 500
0110 : 1000
0111 : 2000
1000 : 3000
1001 : 4000
1010 : 5000
1011 : 6000
1100 : 7000
1101 : 8000
1110 : 9000
1111 : 10000

ビット 1 LPMACK : LPM トークン確認応答有効化

デバイスのアプリケーションソフトウェアによって事前にプログラムされた LPM トークンへのハンドシェイク応答

1 : ACK

ACK は事前にプログラムされているものの、コア (デバイス) は成功した LPM トランザクションに対してのみ ACK 応答を返します。LPM トランザクションは次の場合に成功します。

- EXT トークンまたは LPM トークンのどちらにも PID/CRC5 エラーがない (それ以外の場合は ERROR)。
- LPM トランザクションで受信した有効な bLinkState = 0001B (L1) (それ以外の場合は STALL)。
- 送信シーケンスにペンディング中のデータがない (それ以外の場合は NYET)。

0 : NYET

事前にプログラムされたソフトウェアビットは、次の場合に、LPM トークンへの応答に上書きされます。

- 受信した bLinkState が L1 でない場合 (STALL 応答)、または
- いずれかの LPM トークンパケットで、破損によるエラーが検出された場合 (エラー応答)

注 : デバイスモードでのみアクセス可能です。

ビット 0 LPMEN : LPM サポートイネーブル

アプリケーションは、このビットを使用して、OTG_FS コア LPM 機能を制御します。

コアが非 LPM 対応ホストとして動作する場合、ホストは接続されたデバイスまたはハブに LPM モードを有効にするように要求することはできません。

コアが非 LPM 対応デバイスとして動作する場合、デバイスはいかなる LPM トランザクションにも応答できません。

0 : LPM 機能は無効です。

1 : LPM 機能は有効です。

29.15.15 OTG ホスト周期的送信 FIFO サイズレジスタ (OTG_HPTXFISIZ)

アドレスオフセット : 0x100

リセット値 : 0x0200 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXFISIZ															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXSA															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **PTXFD** : ホスト周期的 Tx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

ビット 15:0 **PTXSA** : ホスト周期的 Tx FIFO の開始アドレス

このフィールドは、周期的送信 FIFO RAM のメモリ開始アドレスを設定します。

29.15.16 OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTXFx) (x = 1~5、ここで x は FIFO_number)

アドレスオフセット : $0x104 + (\text{FIFO_number} - 1) * 0x04$

リセット値 :

$\text{FIFO_number} = 5 : 0x0200\ 0200 + (5 * 0x200)$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INEPTXFD															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXSA															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **INEPTXFD** : IN エンドポイント Tx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

ビット 15:0 **INEPTXSA** : IN エンドポイント FIFOx 送信 RAM 開始アドレス

このフィールドは IN エンドポイント送信 FIFOx のメモリ開始アドレスを含みます。アドレスは 32 ビットメモリロケーションに整列されている必要があります。

29.15.17 ホストモードレジスタ

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

ホストモードレジスタは、ホストモードでのコアの動作に影響を与えます。ホストモードレジスタは、結果が定義されていないため、デバイスモードではアクセスしないでください。ホストモードレジスタは、次のように分類することができます。

29.15.18 OTG ホスト設定レジスタ (OTG_HCFG)

アドレスオフセット : 0x400

リセット値 : 0x0000 0000

このレジスタは、電源投入後のコアを設定します。ホストを初期化した後は、このレジスタを変更しないでください。

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FSLSS** : FS および LS のみのサポート

アプリケーションは、このビットを使用して、コアのエnumレーション速度を制御します。このビットを使用して、アプリケーションは、接続されたデバイスが HS トラフィックをサポートする場合でも、FS ホストとしてコアにエnumレーションさせることができます。初期プログラミングの後は、このフィールドを変更しないでください。

1: 接続されたデバイスが HS トラフィックをサポートしている場合でも、FS/LS のみ (読み出し専用)。

ビット 1:0 **FSLSPCS** : FS/LS PHY クロック選択

コアが FS ホストモードのとき、
01 : PHY クロックは 48 MHz で動作しています。

その他 : 予約済み

コアが LS ホストモードのとき、
00 : 予約済み
01 : PHY クロック周波数として 48 MHz を選択
10 : PHY クロック周波数として 6 MHz を選択
11 : 予約済み

注 : **FSLSPCS** は、接続されたデバイスの速度に応じて、接続イベント時にセットする必要があります (このビットを変更した後、ソフトウェアリセットを実行する必要があります)。

29.15.19 OTG ホストフレームインターバルレジスタ (OTG_HFIR)

アドレスオフセット : 0x404

リセット値 : 0x0000 EA60

このレジスタは、OTG_FS コントローラがエニュメレーションした現在の速度でのフレームインターバル情報を格納します。

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **RLDCTRL** : 再ロード制御

このビットを使用して、ランタイム時に HFIR レジスタの動的再ロードが可能です。

0 : HFIR は動的に再ロードすることができません。

1 : HFIR はランタイム時に動的に再ロードすることができます。

このビットは、初期設定時にプログラムする必要があり、ランタイム時にそのを変更してはなりません。

ビット 15:0 **FRIVL** : フレームインターバル

アプリケーションがこのフィールドにプログラムする値は、2 つの連続した SOF (FS) またはキープアライブトークン (LS) の間のインターバルを指定します。このフィールドは、必要なフレームインターバルを構成する PHY クロック数を含みます。アプリケーションは、ホストポート制御およびステータスレジスタのポートイネーブルビット (OTG_HPRT の PENA ビット) がセットされた後のみ、このレジスタに値を書き込むことができます。値がプログラムされていない場合、コアは、ホスト設定レジスタの FS/LS PHY クロック選択フィールド (OTG_HCFG の FSLSPCS) で指定された PHY クロックに基づいて値を計算します。初期設定の後には、RLDCTRL ビットがセットされている場合を除いて、このフィールドの値を変更しないでください。そのような場合、FRIVL には各 SOF イベントが再ロードされます。

29.15.20 OTG ホストフレーム番号／残りフレーム時間 レジスタ (OTG_HFNUM)

アドレスオフセット : 0x408

リセット値 : 0x0000 3FFF

このレジスタは、現在のフレーム番号を示します。また、現在のフレームの残り時間 (PHY クロック数) も示します。

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

ビット 31:16 **FTREM** : 残りフレーム時間

現在のフレームの残り時間を PHY クロック数で示します。このフィールドは PHY クロックごとにデクリメントされます。値がゼロになると、このフィールドにはフレームインターバルレジスタの値が再ロードされ、新しい SOF が USB に送信されます。

ビット 15:0 **FRNUM** : フレーム番号

このフィールドは USB 上で新しい SOF が送信されるたびにインクリメントされ、0x3FFF に達すると 0 にクリアされます。

29.15.21 OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG_HPTXSTS)

アドレスオフセット : 0x410

リセット値 : 0x0008 0100

この読み出し専用レジスタは、周期的 Tx FIFO および周期的送信リクエストキューの空きスペース情報を含みます。

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

- ビット 31:24 **PTXQTOP** : 周期的送信リクエストキューの先頭
- これは MAC によって処理中の周期的 Tx リクエストキューのエントリを示します。
- このレジスタは、デバッグに使用されます。
- ビット 31 : 奇数/偶数フレーム
- 0 : 偶数フレームで送信
- 1 : 奇数フレームで送信
- ビット 30:27チャンネル/エンドポイント番号
- ビット 26:25タイプ
- 00 : IN/OUT
- 01 : 長さゼロの packets
- 11 : チャンネルコマンドディセーブル
- ビット 24 : 終了 (選択されたチャンネル/エンドポイントの最後のエントリ)
- ビット 23:16 **PTXQSAV** : 周期的送信リクエストキューの使用可能スペース
- 周期的送信リクエストキューで書き込みに使用できる空きロケーションの数を示します。このキューは IN および OUT リクエストの両方を保持します。
- 00 : 周期的送信リクエストキューは満杯です。
- 01 : 1 位置が使用可能です。
- 10 : 2 位置が使用可能です。
- bxn : n 位置が使用可能です (0 ≤ n ≤ 8)。
- その他 : 予約済み
- ビット 15:0 **PTXFSAVL** : 周期的送信データ FIFO の使用可能スペース
- 周期的 Tx FIFO で書き込みに使用できる空きロケーションの数を示します。
- 値は 32 ビットワード単位です。
- 0000 : 周期的 Tx FIFO は満杯です。
- 0001 : 1 ワードが使用可能です。
- 0010 : 2 ワードが使用可能です。
- bxn : n ワードが使用可能です (0 ≤ n ≤ PTXFD)
- その他 : 予約済み

29.15.22 OTG ホスト全チャネル割り込みレジスタ (OTG_HAINT)

アドレスオフセット : 0x414

リセット値 : 0x0000 000

チャネル上で重大なイベントが発生すると、ホスト全チャネル割り込みレジスタは、コア割り込みレジスタのホストチャネル割り込みビット (OTG_GINTSTS の HCINT ビット) を使用して、アプリケーションに割り込みをかけます。これを [図 342](#) に示します。チャネルごとに 1 つの割り込みビットがあります (最大 16 ビット)。このレジスタのビットは、アプリケーションが対応するホストチャネル x 割り込みレジスタのビットをセット/クリアしたときにセット/クリアされます。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **HAINT** : チャネル割り込み

チャネルあたり 1 ビット: チャネル 0 に対してビット 0、チャネル 15 に対してビット 15

29.15.23 OTG ホスト全チャネル割り込みマスクレジスタ (OTG_HAINTMSK)

アドレスオフセット : 0x418

リセット値 : 0x0000 0000

ホスト全チャネル割り込みマスクレジスタは、ホスト全チャネル割り込みレジスタとともに働き、チャネル上でイベントが発生したときに、アプリケーションに割り込みをかけます。チャネルごとに 1 つの割り込みマスクビットがあります (最大 16 ビット)。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **HAINTM** : チャネル割り込みマスク

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

チャネルあたり 1 ビット: チャネル 0 に対してビット 0、チャネル 15 に対してビット 15



29.15.24 OTG ホストポート制御およびステータスレジスタ (OTG_HPRT)

アドレスオフセット : 0x440

リセット値 : 0x0000 0000

このレジスタは、ホストモードでのみ使用可能です。現時点では OTG ホストは 1 つのポートのみをサポートしています。

1 つのレジスタは、ポートごとの USB リセット、イネーブル、サスペンド、レジューム、接続ステータス、テストモードなどの USB ポート関連情報を保持します。図 342 に示されています。このレジスタの rc_w1 ビットは、コア割り込みレジスタのホストポート割り込みビット (OTG_GINTSTS の HPRTINT ビット) を使用して、アプリケーションへの割り込みをトリガできます。ポート割り込みの場合、アプリケーションは、このレジスタを読み出して、割り込みの原因となったビットをクリアする必要があります。rc_w1 ビットの場合、アプリケーションは、このビットに 1 を書き込んで、割り込みをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD		PTCTL
													r	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTL			PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POC CHNG	POCA	PEN CHNG	PENA	PCDET	PCSTS
rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:17 **PSPD** : ポートスピード

このポートに接続されたデバイスのスピードを示します。

01 : フルスピード

10 : ローススピード

11 : 予約済み

ビット 16:13 **PTCTL** : ポートテスト制御

アプリケーションがこのフィールドにゼロ以外の値を書き込んで、ポートをテストモードにすると、対応するパターンがポートにシグナリングされます。

0000 : テストモードは無効です

0001 : Test_J モード

0010 : Test_K モード

0011 : Test_SE0_NAK モード

0100 : Test_Packet モード

0101 : Test_Force_Enable

その他 : 予約済み

ビット 12 **PPWR** : ポートパワー

アプリケーションは、このフィールドを使用して、このポートへの電力供給を制御し、コアは、過電流条件が発生すると、このビットをクリアします。

0 : パワーオフ

1 : パワーオン

ビット 11:10 **PLSTS** : ポートラインステータス

USB データラインの現在のロジックレベルを示します。

ビット 10 : OTG_DP のロジックレベル

ビット 11 : OTG_DM のロジックレベル

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PRST** : ポートリセット

アプリケーションがこのビットをセットすると、このポートでリセットシーケンスが開始されます。アプリケーションは、リセットシーケンスの完了後、リセット時間を決めて、このビットをクリアする必要があります。

0 : ポートは非リセット中

1 : ポートはリセット中

アプリケーションは、ポートのリセットが開始されるまで、少なくとも 10 ms の間、このビットをセット状態に保つ必要があります。USB 規格では最大限度が規定されていませんが、アプリケーションは、このビットをクリアする前に、必要最小時間に加えて、さらに 10 ms の間、このビットをセット状態に保つことができます。

ハイスピード : 50 ms

フルスピード/ロースピード : 10 ms

ビット 7 **PSUSP** : ポートサスペンド

アプリケーションは、このビットをセットして、このポートをサスペンドモードにします。コアは、このビットがセットされると SOF の送信のみを停止します。PHY クロックを停止するには、アプリケーションは PHY のサスペンド入力ピンをアサートするポートクロック停止ビットをセットする必要があります。

このビットの読み出し値は、ポートの現在のサスペンドステータスを反映しています。このビットは、リモートウェイクアップ信号の検出後、またはアプリケーションがこのレジスタのポートリセットビットまたはポートレジャームビット、またはコア割り込みレジスタのレジャーム/リモートウェイクアップ検出割り込みビットまたは切断検出割り込みビット（それぞれ OTG_GINTSTS の WKUINT または DISCINT ビット）をセットしたときに、コアによってクリアされます。

0 : ポートは非サスペンドモード

1 : ポートはサスペンドモード

ビット 6 **PRES** : ポートレジャーム

アプリケーションは、このビットをセットして、ポート上でレジャーム信号を駆動します。コアは、アプリケーションがこのビットをクリアするまで、レジャーム信号の駆動を続行します。

コアが、コア割り込みレジスタのポートレジャーム/リモートウェイクアップ検出割り込みビット (OTG_GINTSTS の WKUINT ビット) によって示される USB リモートウェイクアップシーケンスを検出した場合、コアは、アプリケーションの介入なしでレジャームシグナリングの駆動を開始し、切断条件を検出すると、このビットをクリアします。このビットの読み出し値は、コアが現在レジャーム信号を駆動中であるかどうかを示します。

0 : レジャーム信号を駆動していません。

1 : レジャーム信号を駆動しています。

LPM が有効で、コアが L1 状態にあるとき、このビットは次のように動作します。

1. アプリケーションは、このビットをセットして、ポート上でレジャーム信号を駆動します。

2. コアは OTG_GLPMPCFG レジスタの BESLTHRS[3:0] フィールドに指定された所定時間まで、レジャーム信号の駆動を続行します。

3. コアが、コア割り込みレジスタのポート L1 レジャーム/リモート L1 ウェイクアップ検出割り込みビット (OTG_GINTSTS の WKUPINT ビット) によって示される USB リモートウェイクアップシーケンスを検出した場合、コアは、アプリケーションの介入なしでレジャームシグナリングの駆動を開始し、レジャームの最後にビットをクリアします。このビットは、コアおよびアプリケーション両方でセット/クリアできます。このビットは、ホストにデバイスが 1 つも接続されていない場合でもコアによってクリアされます。

ビット 5 **POCCHNG** : ポート過電流の変化

コアは、このレジスタのポート過電流アクティブビット (ビット 4) に変化があったとき、このビットをセットします。



- ビット 4 **POCA** : ポート過電流アクティブ
ポートの過電流条件を示します。
0 : 非過電流状態
1 : 過電流状態
- ビット 3 **PENCHNG** : ポートイネーブル／ディセーブルの変化
コアは、このレジスタのポートイネーブルビット 2 のステータスに変化があったとき、このビットをセットします。
- ビット 2 **PENA** : ポートイネーブル
ポートは、リセットシーケンス後、コアによってのみ有効にされ、過電流条件、切断条件、またはアプリケーションがこのビットをクリアすることによって無効にされます。アプリケーションは、レジスタへの書き込みによってこのビットをクリアすることはできません。クリアして、ポートを無効にすることだけができます。このビットは、アプリケーションへの割り込みをトリガしません。
0 : ポートを無効化
1 : ポートを有効化
- ビット 1 **PCDET** : ポート接続の検出
コアは、デバイス接続が検出されると、このビットをセットして、コア割り込みレジスタのホストポート割り込みビット (OTG_GINTSTS の HPRTINT ビット) を使用して、アプリケーションへの割り込みをトリガします。アプリケーションは、このビットに 1 を書き込むことによって、割り込みをクリアする必要があります。
- ビット 0 **PCSTS** : ポート接続ステータス
0 : ポートにはデバイスが接続されていません。
1 : ポートにデバイスが接続されています。

29.15.25 OTG ホストチャネル x 特性レジスタ (OTG_HCCHARx) (x = 0..11, ここで x = Channel_number)

アドレスオフセット : 0x500 + (Channel_number * 0x20)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHENA	CHDIS	ODD FRM	DAD							MCNT		EPTYP		LSDEV	Res.
rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPDIR	EPNUM					MPSIZ									
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 CHENA : チャネルイネーブル

このフィールドは、アプリケーションによってセットされ OTG ホストによってクリアされます。

0 : チャネルは無効です。

1 : チャネルは有効です。

ビット 30 CHDIS : チャネル無効化

アプリケーションは、チャネルへの転送が完了する前でも、このビットをセットして、そのチャネル上での送信／受信を停止することができます。アプリケーションは、チャネルを無効として扱う前に、チャネル無効割り込みを待つ必要があります。

ビット 29 ODDFRM : 奇数フレーム

このフィールドは OTG ホストが奇数フレームで転送を実行しなければならないことを示すために、アプリケーションによってセット（リセット）されます。このフィールドは、周期的（アイソクロナスおよび割り込み）トランザクションに対してのみ適用されます。

0 : 偶数フレーム

1 : 奇数フレーム

ビット 28:22 DAD : デバイスアドレス

このフィールドは、データソースまたはデータシンクとして機能する特定のデバイスを選択します。

ビット 21:20 MCNT : マルチカウント

このフィールドは、この周期的エンドポイントについてフレームあたりで実行されなければならないトランザクションの数をホストに示します。非周期的転送の場合、このフィールドは使用されません。

00 : 予約済み。このフィールドは定義されていない結果をもたらします。

01 : 1 トランザクション

10 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 2 つです。

11 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 3 つです。

注 : このフィールドは、01 以上にセットする必要があります。

ビット 19:18 EPTYP : エンドポイントタイプ

選択された転送タイプを示します。

00 : コントロール

01 : アイソクロナス

10 : バルク

11 : インタラプト

ビット 17 LSDEV : ロースピードデバイス

このフィールドは、このチャネルがロースピードデバイスと通信中であることを示すために、アプリケーションによってセットされます。

ビット 16 予約済みであり、リセット値に保持する必要があります。**ビット 15 EPDIR : エンドポイントの方向**

トランザクションが IN か OUT かを示します。

0 : OUT

1 : IN

ビット 14:11 EPNUM : エンドポイント番号

データソースまたはデータシンクとして機能しているデバイスのエンドポイント番号を示します。

ビット 10:0 MPSIZ : 最大パケットサイズ

関連するエンドポイントの最大パケットサイズを示します。

29.15.26 OTG ホストチャネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..11, ここで x = Channel_number)

アドレスオフセット : 0x508 + (Channel_number * 0x20)

リセット値 : 0x0000 0000

このレジスタは USB および AHB 関連イベントに関するチャネルのステータスを示します。[図 342](#)に示されています。アプリケーションは、コア割り込みレジスタのホストチャネル割り込みビット (OTG_GINTSTS の HCINT ビット) がセットされたとき、このレジスタを読み出す必要があります。アプリケーションがこれらのレジスタを読み出すためには、まず、ホスト全チャネル割り込みレジスタ (OTG_HAINT) を読み出して、ホストチャネル x 割り込みレジスタの正確なチャネル数を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_HAINT および OTG_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR	FRM OR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
					rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DTERR** : データトグルエラー

ビット 9 **FRMOR** : フレームオーバーラン

ビット 8 **BBERR** : バブルエラー

ビット 7 **TXERR** : トランザクションエラー

USB 上で以下のエラーの 1 つが発生したことを示します。

- CRC チェック失敗
- タイムアウト
- ビットスタッフエラー
- 偽の EOP

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **ACK** : ACK レスポンス受信/送信割り込み

ビット 4 **NAK** : NAK レスポンス受信割り込み

ビット 3 **STALL** : STALL レスポンス受信割り込み

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CHH** : チャネル停止

USB トランザクションエラーまたはアプリケーションによるディセーブルリクエストへの応答のいずれかの理由により、転送が異常終了したことを示します。

ビット 0 **XFRC** : 転送完了

転送がエラーなく正常に完了しました。

29.15.27 OTG ホストチャネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..11, ここで x = Channel_number)

アドレスオフセット : 0x50C + (Channel_number * 0x20)

リセット値 : 0x0000 0000

このレジスタは、前のセクションで説明した各チャネルステータスのマスクを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR M	FRM ORM	BBERR M	TXERR M	Res.	ACKM	NAKM	STALL M	Res.	CHHM	XFRC M
					rw	rw	rw	rw		rw	rw	rw		rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DTERRM** : データトグルエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 9 **FRMORM** : フレームオーバーランマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 8 **BBERRM** : バブルエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 7 **TXERRM** : トランザクションエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **ACKM** : ACK レスポンス受信/送信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 4 **NAKM** : NAK レスポンス受信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 3 **STALLM** : STALL 応答受信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CHHM** : チャンネル停止マスク
0 : 割り込みはマスクされます。
1 : 割り込みはマスクされません。

ビット 0 **XFCRM** : 転送完了マスク
0 : 割り込みはマスクされます。
1 : 割り込みはマスクされません。

29.15.28 OTG ホストチャネル x 転送サイズレジスタ (OTG_HCTSIZx)
(x = 0..11, ここで x = Channel_number)

アドレスオフセット : 0x510 + (Channel_number * 0x20)
リセット値 : 0x0000 0000

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **DPID** : データ PID
アプリケーションは、このフィールドを最初のトランザクションで使用する PID のタイプでプログラムします。ホストは、残りの転送の間、このフィールドを維持します。
00 : データ 0
10 : データ 1
11 : SETUP (制御) / 予約済み (非制御)

ビット 28:19 **PKTCNT** : パケットカウント
このフィールドは、アプリケーションによって、送信 (OUT) または受信 (IN) されるパケットの予想数でプログラムされます。
ホストは OUT/IN パケットの送信または受信が成功するたびに、このカウントをデクリメントします。このカウントがゼロに達すると、アプリケーションに割り込んで、正常完了を示します。

ビット 18:0 **XFRSIZ** : 転送サイズ
OUT の場合、このフィールドは、ホストが転送時に送信するデータバイト数です。
IN の場合、このフィールドは、アプリケーションが転送のために確保したバッファサイズです。アプリケーションは、IN トランザクション (周期的および非周期的) の場合、このフィールドを最大パケットサイズの整数倍でプログラムすることが期待されます。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

29.15.29 デバイスモードレジスタ

29.15.30 OTG デバイス設定レジスタ (OTG_DCFG)

アドレスオフセット : 0x800

リセット値 : 0x0220 0000

このレジスタは、電源投入後、または特定の制御コマンドあるいは列挙の後、コアをデバイスモードに設定します。初期プログラミングの後には、このレジスタを変更しないでください。

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

ビット 31 : 予約済みであり、リセット値に保持する必要があります。

ビット 15 **ERRATIM** : 不規則エラー割り込みマスク

- 1 : 不規則エラー時のアーリーサスペンド割り込みをマスクします。
- 0 : 不規則エラー時にアーリーサスペンド割り込みが生成されます。

ビット 12:11 **PFIVL** : 周期的フレームインターバル

周期的フレーム終了割り込みを使用してアプリケーションに通知しなければならないフレーム内の時間を示します。これを使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

- 00 : フレームインターバルの 80%
- 01 : フレームインターバルの 85%
- 10 : フレームインターバルの 90%
- 11 : フレームインターバルの 95%

ビット 10:4 **DAD** : デバイスアドレス

アプリケーションは、各 SetAddress 制御コマンドの後、このフィールドをプログラムする必要があります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **NZLSOHSK** : 非ゼロ長ステータスOUTハンドシェイク

アプリケーションは、このフィールドを使用して、制御転送のステータスステージの OUT トランザクション中に、コアが非ゼロ長のデータパケットを受信したときに送信するハンドシェイクを選択できます。

1 : 非ゼロ長ステータスの OUT トランザクション時には STALL ハンドシェイクを送信し、受信した OUT パケットをアプリケーションに送信しません。

0 : 受信した OUT パケット (ゼロ長または非ゼロ長) をアプリケーションに送信し、デバイスエンドポイント制御レジスタの NAK および STALL ビットに基づいてハンドシェイクを送信します。

ビット 1:0 **DSPD** : デバイススピード

アプリケーションがコアにエニュメレーションを要求するスピード、またはアプリケーションがサポートできる最高スピードを示します。ただし、実際のバススピードは、チャープシーケンスの完了後に決まり、コアが接続される USB ホストのスピードに基づきます。

- 00 : 予約済み
- 01 : 予約済み
- 10 : 予約済み
- 11 : フルスピード (USB 1.1 トランシーバのクロックは 48 MHz です)



29.15.31 OTG デバイス制御レジスタ (OTG_DCTL)

アドレスオフセット : 0x804

リセット値 : 0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS BESL RJCT	Res.	Res.
													rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PO PRG DNE	CGO NAK	SGO NAK	CGI NAK	SGI NAK	TCTL			GON STS	GIN STS	SDIS	RWU SIG
				rw	w	w	w	w	rw	rw	rw	r	r	rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **DSBESLRJCT** : ディープスリープ BESL 拒否

コアは、プログラムされた BESL 閾値より大きい BESL 値を持つ LPM リクエストを拒否します。
NYET 応答は、BESL 閾値より大きい BESL 値を持つ LPM トークンに送信されます。デフォルトでは、
ディープスリープ BESL 拒否機能は無効です。

ビット 17:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **POPRGDNE** : パワーオンプログラミング終了

アプリケーションは、このビットを使用して、パワーダウンモードからのウェイクアップ後に、レジ
スタのプログラミングが終了したことを示します。

ビット 10 **CGONAK** : グローバル OUT NAK のクリア

このフィールドに書き込むと、グローバル OUT NAK がクリアされます。

ビット 9 **SGONAK** : グローバル OUT NAK のセット

このフィールドに書き込むと、グローバル OUT NAK がセットされます。
アプリケーションは、このビットを使用して、すべての OUT エンドポイントに NAK ハンドシェイク
を送信します。
アプリケーションがこのビットをセットするのは、コア割り込みレジスタのグローバル OUT NAK 有効
ビット (OTG_GINTSTS の GONAKEFF ビット) がクリアされていることを確認した後でなければ
なりません。

ビット 8 **CGINAK** : グローバル IN NAK のクリア

このフィールドに書き込むと、グローバル IN NAK がクリアされます。

ビット 7 **SGINAK** : グローバル IN NAK のセット

このフィールドに書き込むと、グローバル非周期的 IN NAK がセットされます。アプリケーションは、
このビットを使用して、すべての非周期的 IN エンドポイントに NAK ハンドシェイクを送信します。
アプリケーションがこのビットをセットするのは、コア割り込みレジスタのグローバル IN NAK 有効
ビット (OTG_GINTSTS の GINAKEFF ビット) がクリアされていることを確認した後でなければ
なりません。

ビット 6:4 **TCTL** : テスト制御

000 : テストモードは無効です
001 : Test_J モード
010 : Test_K モード
011 : Test_SE0_NAK モード
100 : Test_Packet モード
101 : Test_Force_Enable
その他 : 予約済み

- ビット 3 **GONSTS** : グローバル OUT NAK のステータス
- 0 : ハンドシェイクは、FIFO ステータスと NAK および STALL ビットの設定に基づいて送信されます。
- 1 : 使用可能なスペースに関係なく Rx FIFO にはデータが書き込まれません。SETUP トランザクションを除き、すべてのパケットに NAK ハンドシェイクを送信します。すべてのアイソクロナス OUT パケットがドロップされます。
- ビット 2 **GINSTS** : グローバル IN NAK ステータス
- 0 : ハンドシェイクは、送信 FIFO 内の使用可能なデータに基づいて送信されます。
- 1 : NAK ハンドシェイクは、送信 FIFO 内の使用可能なデータに関係なく、非周期的 IN エンドポイントに送出されます。
- ビット 1 **SDIS** : ソフト切断
- アプリケーションは、このビットを使用して USB OTG コアにソフト切断を実行することを伝えます。このビットがセットされている限り、ホストはデバイスが接続されているとは解釈せず、デバイスは USB 上で信号を受信しません。コアは、アプリケーションがこのビットをクリアするまで、切断状態にとどまります。
- 0 : 通常動作。ソフト切断後にこのビットがクリアされると、コアは USB ホストに対してデバイス接続イベントを生成します。デバイスが再接続されると、USB ホストはデバイスのエニュメレーションをリスタートします。
- 1 : コアは、USB ホストに対して、デバイス切断イベントを生成します。
- ビット 0 **RWUSIG** : リモートウェイクアップ信号
- アプリケーションがこのビットをセットすると、コアはリモートシグナリングを実行して USB ホストをウェイクアップします。コアをサスペンド状態から抜け出させるには、アプリケーションは、このビットをセットする必要があります。USB 2.0 仕様で規定されているように、アプリケーションは、このビットをセットしてから 1 ms から 15 ms 後にクリアする必要があります。
- LPM が有効で、コアが L1 (スリープ) 状態にあるとき、アプリケーションがこのビットをセットすると、コアは L1 リモートシグナリングを開始して、USB ホストをウェイクアップします。コアをスリープ状態から抜け出させるには、アプリケーションは、このビットをセットする必要があります。LPM 仕様で規定されているように、このビットはアプリケーションによってセットされてから 50 μ s ($T_{L1DevDrvResume}$) 後にハードウェアによって自動的にクリアされます。アプリケーションは、前の LPM トランザクションからの bRemoteWake がゼロのときこのビットをセットする必要があります (GLPMCFG レジスタの REMWAKE ビットを参照)。

表 194 は USB ホストがデバイスの切断を検出するためにソフト切断 (SDIS) ビットがセットされなければならない最短時間 (デバイスの状態による) を示します。クロックジッタに対応するには、アプリケーションは指定された最短時間まで遅延を追加することが推奨されます。

表 194. ソフト切断のための最短時間

動作速度	デバイスの状態	最短時間
フルスピード	サスペンド状態	1 ms + 2.5 μ s
フルスピード	アイドル	2.5 μ s
フルスピード	アイドル状態でもサスペンド状態でもない (トランザクション実行中)	2.5 μ s

29.15.32 OTG デバイスステータスレジスタ (OTG_DSTS)

アドレスオフセット : 0x808

リセット値 : 0x0000 0010

このレジスタは USB 関連イベントに関するコアのステータスを示します。デバイス全割り込みレジスタ (OTG_DAINTE) からの割り込み時に読み出されなければなりません。

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

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **DEVLNSTS** : デバイスラインステータス

USB データラインの現在のロジックレベルを示します。

ビット [23] : D+ のロジックレベル

ビット [22] : D- のロジックレベル

ビット 21:8 **FNSOF** : 受信 SOF のフレーム番号

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **EERR** : 不規則エラー

コアは、このビットをセットして、不規則エラーを報告します。

不規則エラーが発生すると、OTG_FS コントローラはサスペンド状態に移行し、OTG_GINTSTS レジスタのアーリーサスペンドビット (OTG_GINTSTS の ESUP ビット) でアプリケーションへの割り込みが生成されます。不規則エラーによるアーリーサスペンドがアサートされた場合、アプリケーションは、ソフト切断回復を実行するしかありません。

ビット 2:1 **ENUMSPD** : エニュメレーションされた速度

チャープシーケンスによる速度検出後の OTG_FS コントローラの速度を示します。

01 : 予約済み

10 : 予約済み

11 : フルスピード (PHY クロックは 48 MHz で動作しています)

その他 : 予約済み

ビット 0 **SUSPSTS** : サスペンドステータス

デバイスモードでは、このビットは、USB 上でサスペンド状態が検出されている限り、セットされます。USB データライン上で 3 ms の間アクティビティが 1 つもなかった場合、コアはサスペンド状態に入ります。コアは、以下の場合にサスペンド状態から抜け出します。

- USB データライン上で何らかのアクティビティがあったとき。
- アプリケーションが OTG_DCTL レジスタのリモートウェイクアップナリングビット (OTG_DCTL の RWUSIG ビット) に書き込みを行ったとき。

29.15.33 OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK)

アドレスオフセット : 0x810

リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG_DIEPINTx レジスタとともに機能して、IN エンドポイントごとに 1 つの割り込みを生成します。このレジスタの対応するビットに書き込むことによって OTG_DIEPINTx レジスタの特定のステータスに対する IN エンドポイント割り込みはマスクできます。ステータスビットはデフォルトでマスクされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAKM	Res.	Res.	Res.	Res.	Res.	Res.	INEPN EM	INEPN MM	ITTXFE MSK	TOM	Res.	EPDM	XFRC M
		rw							rw	rw	rw	rw		rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **NAKM** : NAK 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 12:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **INEPNEM** : IN エンドポイント NAK 有効マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 5 **INEPNMM** : EP 不一致付きで受信された IN トークンマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 4 **ITTXFEMSK** : TxFIFO エンプティ時に受信された IN トークンマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 3 **TOM** : タイムアウト状態マスク (非アイソクロナスエンドポイント)

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDM** : エンドポイントディセーブル割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 0 **XFRCM** : 転送完了割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

29.15.34 OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK)

アドレスオフセット : 0x814

リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG_DOEPINTx レジスタとともに機能して、OUT エンドポイントごとに 1 つの割り込みを生成します。このレジスタの対応するビットに書き込むことによって OTG_DOEPINTx レジスタの特定のステータスに対する OUT エンドポイント割り込みはマスクできます。ステータスビットはデフォルトでマスクされます。

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

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OTEPDM** : エンドポイント無効時 OUT トークン受信マスク制御 OUT エンドポイントにのみ適用されます。

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

ビット 3 **STUPM** : STUPM : SETUP フェーズ終了マスク制御エンドポイントにのみ適用されます。

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDM** : エンドポイントディセーブル割り込みマスク

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

ビット 0 **XFRCM** : 転送完了割り込みマスク

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

29.15.35 OTG デバイス全エンドポイント割り込みレジスタ (OTG_DAIN)

アドレスオフセット : 0x818

リセット値 : 0x0000 0000

エンドポイント上で重要なイベントが発生すると、OTG_DAIN レジスタは、OTG_GINTSTS レジスタのデバイス OUT エンドポイント割り込みビットまたはデバイス IN エンドポイント割り込みビット（それぞれ、OTG_GINTSTS の OEPINT または IEPINT ビット）を使用して、アプリケーションに割り込みをかけます。エンドポイントごとに 1 つの割り込みビットがあります（OUT エンドポイントおよび IN エンドポイントに対してそれぞれ最大 16 ビット）。双方向エンドポイントの場合、対応する IN および OUT 割り込みビットが使用されます。このレジスタのビットは、アプリケーションが

対応するデバイスエンドポイント x 割り込みレジスタのビット (OTG_DIEPINTx/OTG_DOEPINTx) をセット/クリアしたときに、セット/クリアされます。

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

ビット 31:16 **OEPINT** : OUT エンドポイント割り込みビット

OUT エンドポイントチャンネルあたり 1 ビット :

OUT エンドポイント 0 に対してビット 16、OUT エンドポイント 3 に対してビット 18

ビット 15:0 **IEPINT** : IN エンドポイント割り込みビット

IN エンドポイントあたり 1 ビット :

IN エンドポイント 0 に対してビット 0、IN エンドポイント 3 に対してビット 3

29.15.36 OTG 全エンドポイント割り込みマスクレジスタ (OTG_DAINTRMSK)

アドレスオフセット : 0x81C

リセット値 : 0x0000 0000

OTG_DAINTRMSK レジスタは、デバイスエンドポイント割り込みレジスタとともに機能して、デバイスエンドポイント上にイベントが発生すると、アプリケーションに割り込みます。ただし、その割り込みに対応する OTG_DAINTR レジスタのビットはセットされたままです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OEPMSK															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPMSK															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **OEPMSK** : OUT EP 割り込みマスクビット

OUT エンドポイント (EP) あたり 1 ビット :

OUT EP 0 に対してビット 16、OUT EP 3 に対してビット 18

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

ビット 15:0 **IEPMSK** : IN EP 割り込みマスクビット

IN エンドポイントあたり 1 ビット :

IN EP 0 に対してビット 0、IN EP 3 に対してビット 3

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

29.15.37 OTG デバイス V_{BUS} 放電時間レジスタ (OTG_DVBUSDIS)

アドレスオフセット : 0x0828

リセット値 : 0x0000 17D7

このレジスタは SRP 中の V_{BUS} 放電時間 (V_{BUS} パルシング後) を指定します。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **VBUSDT** : デバイス V_{BUS} 放電時間

SRP 中の V_{BUS} 放電時間 (V_{BUS} パルシング後) を指定します。この値は以下になります。

V_{BUS} 放電時間 (PHY クロック / 1024)

この値は V_{BUS} の負荷によって調整の必要があるかもしれません。

29.15.38 OTG デバイス V_{BUS} パルシング時間レジスタ (OTG_DVBUSPULSE)

アドレスオフセット : 0x082C

リセット値 : 0x0000 05B8

このレジスタは SRP 期間中の V_{BUS} パルシング時間を指定します。

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DVBUSP** : デバイス V_{BUS} パルシング時間

SRP 中の V_{BUS} パルシング時間を指定します。この値は以下になります。

V_{BUS} パルシング時間 (PHY クロック / 1024)

29.15.39 OTG デバイス IN エンドポイント FIFO エンプティ割り込みマスクレジスタ (OTG_DIEPEMPMSK)

アドレスオフセット : 0x834

リセット値 : 0x0000 0000

このレジスタは、IN エンドポイント FIFO エンプティ割り込みの生成 (TXFE_OTG_DIEPINTx) を制御するために使用されます。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **INEPTXFEM** : IN EP Tx FIFO エンプティ割り込みマスクビット

これらのビットは OTG_DIEPINTx に対するマスクビットとして機能します。

IN EP あたり TXFE 割り込み 1 ビット :

IN EP 0 に対してビット 0、IN EP 3 に対してビット 3

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

29.15.40 OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_DIEPCTL0)

アドレスオフセット : 0x900

リセット値 : 0x0000 0000

このセクションでは、USB_OTG_FS の OTG_DIEPCTL0 レジスタについて説明します。非ゼロ制御エンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	STNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAK STS	Res.
rs	rs			w	w	rw	rw	rw	rw	rs		r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
r														rw	rw

ビット 31 EPENA : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイント 0 でデータ送信を開始します。
コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- エンドポイント無効
- 転送完了

ビット 30 EPDIS : エンドポイントディセーブル

アプリケーションは、このビットをセットして、エンドポイントへの転送が完了する前でも、そのエンドポイントでのデータ送信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。
アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、エンドポイント上で SETUP パケットが受信された後、そのエンドポイントに対してこのビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 TXFNUM : Tx FIFO 番号

この値は、IN エンドポイント 0 に割り当てられる FIFO 番号に設定されます。

ビット 21 STALL : STALL のハンドシェイク

アプリケーションは、このビットのセットのみが可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル INNAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 EPTYP : エンドポイントタイプ

コントロールのために "00" にハードコード化されています。

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

このビットがアプリケーションまたはコアによってセットされると、コアは、Tx FIFO に使用可能なデータがある場合でも、データの送信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **USBAEP** : USB アクティブエンドポイント

このビットは常に 1 にセットされ、制御エンドポイント 0 がすべての設定およびインタフェースにおいて常に有効であることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

29.15.41 OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1~5、ここで x = Endpoint_number)

アドレスオフセット : $0x900 + (\text{Endpoint_number} * 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EO NUM/DPID
rs	rs	w	w	w	w	rw	rw	rw	rw	rw/rs		rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **EPENA** : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。

コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

ビット 30 **EPDIS** : エンドポイントディセーブル

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信/受信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29 **SODDFRM** : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。



ビット 28 SD0PID : DATA0 PID の設定

インタラプト／バルク IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

SEVNFRM : 偶数フレームの設定

アイソクロナス IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数／奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割り込み時、またはエンドポイントで SETUP パケットが受信された後、OUT エンドポイントに対してこのビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 TXFNUM : Tx FIFO 番号

これらのビットは、このエンドポイントに関連する FIFO 番号を指定します。各有効 IN エンドポイントは、別々の FIFO 番号にプログラムされなければなりません。

このフィールドは、IN エンドポイントについてのみ有効です。

ビット 21 STALL : STALL のハンドシェイク

非制御の非アイソクロナス IN エンドポイントにのみ適用されます (アクセスタイプは rw です)。

アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。

コントロールエンドポイントにのみ適用されます (アクセスタイプは rs です)。

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 20 予約済みであり、リセット値に保持する必要があります。**ビット 19:18 EPTYP : エンドポイントタイプ**

これは、この論理エンドポイントがサポートする転送タイプです。

00 : コントロール

01 : アイソクロナス

10 : バルク

11 : インタラプト

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、

非アイソクロナス IN エンドポイントの場合 : コアは、Tx FIFO に使用可能なデータがある場合でも、IN エンドポイントでのデータの送信を停止します。

アイソクロナス IN エンドポイントの場合 : コアは、Tx FIFO に使用可能なデータがある場合でも、ゼロ長のデータパケットを送出します。

このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 **EONUM** : 偶数／奇数フレーム

アイソクロナス IN エンドポイントにのみ適用されます。
コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信／受信する偶数／奇数番号をプログラムする必要があります。
0 : 偶数フレーム
1 : 奇数フレーム

DPID : エンドポイントデータPID

割り込み／バルク IN エンドポイントにのみ適用されます。
このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。
0 : データ 0
1 : データ 1

ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに応じてエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

29.15.42 OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG_DOEPCTL0)

アドレスオフセット : 0xB00

リセット値 : 0x0000 8000

このセクションでは、OTG_DOEPCTL0 レジスタについて説明します。非ゼロ制御エンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	Res.
w	r			w	w					rs	rw	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
r														r	r

ビット 31 EPENA : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイント 0 上でデータ送信を開始します。
コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

ビット 30 EPDIS : エンドポイントディセーブル

アプリケーションは、コントロール OUT エンドポイント 0 を無効にできません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。
アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割り込み時、または SETUP パケットを受信した後に、このビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 STALL : STALL のハンドシェイク

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。NAK ビット、すなわち、グローバル OUT NAK がこのビットとともにセットされた場合は、STALL ビットが優先します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 20 SNPM : スヌープモード

このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。

ビット 19:18 EPTYP : エンドポイントタイプ

制御のために 2'b00 にハードコード化されています。

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、Rx FIFO に受信パケットを収容できるスペースがある場合でも、コアはデータの受信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 USBAEP : USB アクティブエンドポイント

このビットは常に 1 にセットされ、制御エンドポイント 0 がすべての設定およびインタフェースにおいて常にアクティブであることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 MPSIZ : 最大パケットサイズ

制御 OUT エンドポイント 0 の最大パケットサイズは、制御 IN エンドポイント 0 でプログラムされるサイズと同じです。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

29.15.43 OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPTCTLx) (x = 1~5、ここで x = Endpoint_number)

OUT エンドポイントのアドレスオフセット : 0xB00 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SD1 PID/ SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w					rw/rs	rw	rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 EPENA : エンドポイントイネーブル

- IN および OUT エンドポイントに適用されます。
- アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。
- コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。
- SETUP フェーズ終了
 - エンドポイント無効
 - 転送完了

ビット 30 EPDIS : エンドポイントディセーブル

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信/受信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29 SD1PID : DATA1 PID の設定

インタラプト/バルク IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA1 に設定されます。

SODDFRM : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。

ビット 28 SD0PID : DATA0 PID の設定

インタラプト/バルク OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

SEVNFRM : 偶数フレームの設定

アイソクロナス OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにもかかわらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、転送完了割り込み時、または SETUP パケットを受信した後に、OUT エンドポイントに対してこのビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 STALL : STALL のハンドシェイク

非コントロールの非アイソクロナス OUT エンドポイントにのみ適用されます（アクセスタイプは rw です）。

アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット（グローバル IN NAK またはグローバル OUT NAK）がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。

コントロールエンドポイントにのみ適用されます（アクセスタイプは rs です）。

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット（グローバル IN NAK またはグローバル OUT NAK）がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 20 SNPM : スヌープモード

このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。

ビット 19:18 EPTYP : エンドポイントタイプ

これは、この論理エンドポイントがサポートする転送タイプです。

00 : コントロール

01 : アイソクロナス

10 : パルク

11 : インタラプト

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、

コアは、Rx FIFO に受信パケットを収容するためのスペースがある場合でも、OUT エンドポイントでのデータの受信を停止します。

このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 **EONUM** : 偶数／奇数フレーム

アイソクロナス IN および OUT エンドポイントにのみ適用されます。
コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信／受信する偶数／奇数番号をプログラムする必要があります。
0 : 偶数フレーム
1 : 奇数フレーム

DPID : エンドポイントデータPID

インタラプト／バルク OUT エンドポイントにのみ適用されます。
このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。
0 : データ 0
1 : データ 1

ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに応じてエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

29.15.44 OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx)
(x = 0~5、ここで x = Endpoint_number)

アドレスオフセット : 0x908 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。
[図 342](#) に示されています。アプリケーションは、コア割り込みレジスタの IN エンドポイント割り込みビット (OTG_GINTSTS の IEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、デバイス全エンドポイント割り込み (OTG_DAINTE) レジスタを読み出して、デバイスのエンドポイント x 割り込みレジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_DAINTE および OTG_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEP NE	Res.	ITTXFE	TOC	Res.	EP DISD	XFRC
								r	rc_w1/ rw		rc_w1	rc_w1		rc_w1	rc_w1



ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TXFE** : 送信 FIFO エンプティ

この割り込みは、このエンドポイントの Tx FIFO が半エンプティまたは完全エンプティのいずれかであるとき、アサートされます。半エンプティか完全エンプティであるかは、OTG_GAHBCFG レジスタの Tx FIFO エンプティレベルビット (OTG_GAHBCFG の TXFELVL ビット) によって判断されます。

ビット 6 **INEPNE** : IN エンドポイント NAK 有効

このビットは、アプリケーションが OTG_DIEPCTLx レジスタの CNAK ビットに書き込むことによって IN エンドポイント NAK をクリアするとクリアできます。
この割り込みは、コアが (アプリケーションまたはコアによって) 設定された NAK ビットをサンプリングしたことを示します。この割り込みは、アプリケーションによってセットされた IN エンドポイント NAK ビットがコアで有効になったことを示します。
この割り込みは、必ずしも NAK ハンドシェイクが USB に送出されることを保証しません。STALL ビットが NAK ビットに優先します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **ITTXFE** : Tx FIFO がエンプティ時に受信された IN トークン

非アイソクロナス IN エンドポイントにのみ適用されます。
関連する Tx FIFO (周期的/非周期的) が空のときに IN トークンが受信されたことを示します。この割り込みは、IN トークンが受信されたエンドポイント上でアサートされます。

ビット 3 **TOC** : タイムアウト条件

コントロール IN エンドポイントにのみ適用されます。
このエンドポイント上の最後の IN トークンに対して、コアが USB 上でタイムアウト条件を検出したことを示します。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDISD** : エンドポイントディセーブル割り込み

このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。

ビット 0 **XFRC** : 転送完了割り込み

このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

29.15.45 OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPINTx)
(x = 0~5、ここで x = Endpoint_number)

アドレスオフセット : 0xB08 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。
[図 342](#) に示されています。アプリケーションは、OTG_GINTSTS レジスタの OUT エンドポイント割り込みビット (OTG_GINTSTS の OEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、OTG_DAINTE レジスタを読み出して、OTG_DOEPINTx レジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_DAINTE および OTG_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2B STUP	Res.	OTEP DIS	STUP	Res.	EP DISD	XFRC
									rc_w1/ rw		rc_w1	rc_w1		rc_w1	rc_w1

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **B2BSTUP** : 連続 SETUP パケットの受信
コントロール OUT エンドポイントにのみ適用されます。
このビットは、コアがこの特定のエンドポイントで 3 つを超える連続 SETUP パケットを受信したことを示します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OTEPDIS** : エンドポイント無効時 OUT トークン受信
コントロール OUT エンドポイントにのみ適用されます。
エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったことを示します。この割り込みは、OUT トークンが受信されたエンドポイントでアサートされます。

ビット 3 **STUP** : SETUP フェーズ終了
コントロール OUT エンドポイントにのみ適用されます。
コントロールエンドポイントの SETUP フェーズが完了し、現在の制御転送のために、これ以上、連続 SETUP パケットは受信されなかったことを示します。この割り込みにより、アプリケーションは、受信した SETUP データパケットをデコードできます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDISD** : エンドポイントディセーブル割り込み
このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。

ビット 0 **XFRC** : 転送完了割り込み
このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

29.15.46 OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_DIEPTSIZ0)

アドレスオフセット : 0x910

リセット値 : 0x0000 0000

アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。デバイスコントロールエンドポイント 0 制御レジスタのエンドポイントイネーブルビット (OTG_DIEPTL0 の EPENA) を使用して、エンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことができず。

非ゼロエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT		Res.	Res.	Res.
											rw	rw			



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
									rw	rw	rw	rw	rw	rw	rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:19 **PKTCNT** : パケットカウント

エンドポイント 0 のデータ転送サイズを構成する USB パケットの総数を示します。
このフィールドは、パケット（最大サイズまたはショートパケット）が TxFIFO から読み出されるたびにデクリメントされます。

ビット 18:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **XFRSIZ** : 転送サイズ

エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使い切った後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。
コアは、外部メモリからのパケットが TxFIFO に書き込まれるたびに、このフィールドをデクリメントします。

29.15.47 OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_DOEPTSIZE0)

アドレスオフセット : 0xB10

リセット値 : 0x0000 0000

アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。OTG_DOEPTCTL0 レジスタのエンドポイントイネーブルビット (OTG_DOEPTCTL0 の EPENA ビット) を使用してエンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことができません。

非ゼロエンドポイントは、エンドポイント 1~5 のレジスタを使用します。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **STUPCNT** : SETUP パケットカウント

このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。
01 : 1 パケット
10 : 2 パケット
11 : 3 パケット

ビット 28:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **PKTCNT** : パケットカウント

このフィールドは、パケットが Rx FIFO に書き込まれた後、ゼロまでデクリメントされます。

ビット 18:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **XFRSIZ** : 転送サイズ

エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使いきった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、RxFIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

29.15.48 OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSLx) (x = 1~5、ここで x=Endpoint_number)

アドレスオフセット : $0x910 + (\text{Endpoint_number} * 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。OTG_DIEPTSLx レジスタのエンドポイントイネーブルビット (OTG_DIEPTSLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアはこのレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことしかできません。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **MCNT** : マルチカウント

周期的 IN エンドポイントの場合、このフィールドは、USB でフレームごとに送信されなければならないパケット数を示します。コアは、このフィールドを使用して、アイソクロナス IN エンドポイントのデータ PID を計算します。

01 : 1 パケット

10 : 2 パケット

11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。

このフィールドは、パケット (最大サイズまたはショートパケット) が Tx FIFO から読み出されるたびにデクリメントされます。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転送サイズを使いきった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、外部メモリからのパケットが Tx FIFO に書き込まれるたびに、このフィールドをデクリメントします。



29.15.49 OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) (x = 0~5、ここで x = Endpoint_number)

IN エンドポイントのアドレスオフセット : $0x918 + (\text{Endpoint_number} \times 0x20)$ この読み出し専用レジスタは、デバイス IN エンドポイント Tx FIFO の空きスペース情報を含みます。

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

31:16 予約済みであり、リセット値に保持する必要があります。

15:0 INEPTFSAV : IN エンドポイント Tx FIFO の使用可能スペース

エンドポイントの Tx FIFO で使用できる空きスペースの量を示します。

値は 32 ビットワード単位です。

0x0 : エンドポイント Tx FIFO は満杯です。

0x1 : 1 ワードが使用可能です。

0x2 : 2 ワードが使用可能です。

0xn : n ワードが使用可能です。

その他 : 予約済み

29.15.50 OTG デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_DOEPTSIZEx) (x = 1~5、ここで x = Endpoint_number)

アドレスオフセット : $0xB10 + (\text{Endpoint_number} \times 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。OTG_DOEPTCTLx レジスタのエンドポイントイネーブルビット (OTG_DOEPTCTLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアはこのレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことしかできません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ		
	r/rw	r/rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **RXDPID** : 受信データ PID

アイソクロナス OUT エンドポイントにのみ適用されます。
これは、このエンドポイントの最後のパケットで受信されたデータの PID です。
00 : データ 0
10 : データ 1

STUPCNT : SETUP パケットカウント

コントロール OUT エンドポイントにのみ適用されます。
このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。
01 : 1 パケット
10 : 2 パケット
11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。
このフィールドは、パケット（最大サイズまたはショートパケット）が Rx FIFO に書き込まれるたびにデクリメントされます。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転送サイズを使い果たした後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。
コアは、RxFIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

29.15.51 OTG パワーおよびクロックゲーティング制御レジスタ (OTG_PCGCCTL)

アドレスオフセット : 0xE00

リセット値 : 0x0000 0000

このレジスタは、ホストモードとデバイスモードで使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHY SLEEP	ENL1 GTG	PHY SUSP	Res.	Res.	GATE HCLK	STPP CLK
								r	r	rw	r			rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **SUSP** : ディープスリープ

このビットは、L1 状態時に PHY がディープスリープであることを示します。

ビット 6 **PHYSLEEP** : スリープ状態の PHY

このビットは、PHY がスリープ状態であることを示します。

ビット 5 **ENL1GTG** : スリープクロックゲーティング有効化

このビットがセットされている場合、コアが utmi_l1_suspend_n をアサートできなければ、スリープ状態でコアの内部クロックゲーティングが有効化されます。このビットがセットされていない場合、PHY クロックはスリープ状態ではゲートされません。



ビット 4 **PHYSUSP** : PHYをサスペンド

PHY がサスペンドされたことを示します。このビットは、アプリケーションが STPPCLK ビットをセットした後、PHY がサスペンドされると更新されます。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **GATEHCLK** : HCLK をゲート

USB がサスペンドされるか、セッションが有効でないとき、アプリケーションは、このビットをセットして、AHB スレーブとマスタおよびウェイクアップロジック以外のモジュールへの HCLK をゲーティングします。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。

ビット 0 **STPPCLK** : PHY クロックを停止

USB がサスペンドされるか、セッションが有効でないか、デバイスが切断されると、アプリケーションは、このビットをセットして、PHY クロックを停止します。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。

29.15.52 OTG_FS レジスタマップ

次の表に、USB OTG のレジスタマップとリセット値を示します。

表 195. OTG_FS レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	OTG_GOTGCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTGVER	BSVLD	ASVLD	DBCT	CIDSTS	Res.	Res.	Res.	EHEN	DHNPEN	HSHPEN	HNPRQ	HNGSCS	BVALOVAL	BVALOEN	AVALOVAL	AVALOEN	VBVALOVAL	VBVALOEN	SRQ	SRQSCS	
	リセット値												0	0	0	0	1				0	0	0	0	0	0	0	0	0	0	0	0	0	
0x004	OTG_GOTGINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDCHNG	DBCONE	ADTOCHG	HNGDET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.	
	リセット値												0	0	0	0								0	0						0			
0x008	OTG_GAHBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GINTMSK
	リセット値																								0	0								0
0x00C	OTG_GUSBCFG	Res.	FDMOD	FHMOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRDT				HNPCAP	SRPCAP	Res.	PHYSEL	Res.	Res.	Res.	TOTAL		
	リセット値		0	0																	0	1	0	1	0	0		1				0	0	0
0x010	OTG_GRSTCTL	AHBIDL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFNUM				TXFFLSH	RXFFLSH	Res.	FCRST	PSRST	CSRST		
	リセット値	1																					0	0	0	0	0	0	0		0	0	0	



表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x014	OTG_GINTSTS	WKUINT	SRQINT	DISCINT	CIDSCHG	LPIMINT	PTXFE	HCINT	HPRTINT	RSTDET	Res.	IPXFR/INCOMPSOOUT	ISOIXFR	OEPI	IEPI	Res.	Res.	EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	Res.	Res.	GONAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD
	リセット値	0	0	0	1	0	1	0	0	0		0	0	0	0			0	0	0	0	0	0			0	0	1	0	0	0	0	0
0x018	OTG_GINTMSK	WUIM	SRQIM	DISCINT	CIDSCHGM	LPIMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	Res.	IPXFRM/ISOOXFRM	ISOIXFRM	OEPI	IEPI	Res.	Res.	EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
	リセット値	0	0	0	0	0	0	0	0	0		0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0	0	
0x01C	OTG_GRXSTSR (ホストモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT										CHNUM				
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_GRXSTSR (デバイスモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT										EPNUM				
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	OTG_GRXSTSR (ホストモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT										CHNUM				
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_GRXSTSPR (デバイスモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT										EPNUM				
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	OTG_GRXFSIZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFD																
	リセット値																0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x028	OTG_HNPTXFSIZ/ OTG_DIEPTXF0	NPTXFD/TX0FD															NPTXFSA/TX0FSA																
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x02C	OTG_HNPTXSTS	Res.	NPTXQTOP							NPTQXSAV							NPTXFSAV																
	リセット値		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x038	OTG_GCCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBEN	SDEN	PDEN	DCDEN	BCDEN	PWRDWN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2DET	SDET	PDET	DCDET
	リセット値											0	0	0	0	0	0													0	0	0	0
0x03C	OTG_CID	PRODUCT_ID																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x054	OTG_ GLPMCFCG	Res.	Res.	Res.	ENBESL	LPMR CNTSTS			SNDLPM	LPM RCNT			LPMCHIDX				L1RSMOK	SLPSTS	LPM RSP			L1DSEN	BESLTHRS				L1SSEN	REMWAKE	BESL				LPMACK	LPMEN
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x100	OTG_ HPTXFSIZ	PTXFSIZ																PTXSA																
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
0x104	OTG_ DIEPTXF1	INEPTXFD																INEPTXSA																
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
0x108	OTG_ DIEPTXF2	INEPTXFD																INEPTXSA																
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	
...	...																																	
0x204	OTG_ DIEPTXF5	INEPTXFD																INEPTXSA																
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	
0x400	OTG_ HCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLSPCS		
	リセット値																													0	0	0		
0x404	OTG_ HFIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCTRL	FRIVL																
	リセット値																0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	
0x408	OTG_ HFNUM	FTREM																FRNUM																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x410	OTG_ HPTXSTS	PTXQTOP								PTXQSAV								PTXFSAVL																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
0x414	OTG_ HAINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINT																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x418	OTG_ HAINTMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINTM																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x440	OTG_ HPRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD	PTCTL			PPWR		PLSTS		Res.	PRST	PSUSP	PRES	POCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS	
	リセット値															0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	
0x500	OTG_ HCCHAR0	CHENA	CHDIS	ODDFRM	DAD						MCNT			EPTYP	LSDEV	Res.	EPDIR	EPNUM				MPSIZ												
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

[illegible]

表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x728	OTG_ HCINT11	Res.																					DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC	
	リセット値																						0	0	0	0		0	0		0	0		
0x800	OTG_ DCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRATIM				PFIVL								Res.	NZLSOHSK		DSPD	
	リセット値																	0				0	0	0	0	0	0	0	0	0	0	0	0	
0x804	OTG_ DCTL	Res.													DSBSLRJCT							POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK					GONSTS	GINSTS	SDIS	RWUSIG
	リセット値														0							0	0	0	0	0	0	0	0	0	0	1	0	
0x808	OTG_ DSTS	Res.										DEV LN STS														Res.	Res.	Res.	Res.	EERR	ENUMSPD		SUSPSTS	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	
0x810	OTG_ DIEPMSK	Res.																			NAKM	Res.	Res.	Res.	Res.	Res.	INEPNEM	INEPNMM			Res.	Res.	Res.	
	リセット値																					0					0	0	0	0	0	0	0	
0x814	OTG_ DOEPMSK	Res.																			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTEPDM	STUPDM	Res.	EPDM	XFRCM	
	リセット値																											0	0	0	0	0	0	
0x818	OTG_ DAINT																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x81C	OTG_ DAINTMSK																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x828	OTG_ DVBUSDIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
	リセット値																																	
0x82C	OTG_DVB USPULSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
	リセット値																																	
0x834	OTG_DIE PEMPMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
	リセット値																																	
0x900	OTG_ DIEPCTL0	EPENA	EPDIS			SNAK	CNAK						STALL	Res.		EPTYP	NAKSTS	Res.	USBAEP														MPSIZ	
	リセット値	0	0			0	0	0	0	0	0	0			0	0	0	1														0		

表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x908	OTG_ DIEPINT0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	8	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC			
	リセット値																									1	0	0	0	0		0	0				
0x910	OTG_ DIEPTSIZ0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ									
	リセット値												0	0														0	0	0	0	0	0	0			
0x918	OTG_ DTXFSTS0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV																			
	リセット値																	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0				
0x920	OTG_ DIEPCTL1	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	Res.	MPSIZ													
	リセット値	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0						0	0	0	0	0	0	0	0	0	0				
0x928	OTG_ DIEPINT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC				
	リセット値																									1	0	0	0	0		0	0				
0x930	OTG_ DIEPTSIZ1	Res.	MCNT	PKTCNT										XFRSIZ																							
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x938	OTG_ DTXFSTS1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV																			
	リセット値																	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0				
0x940	OTG_ DIEPCTL2	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	Res.	MPSIZ													
	リセット値	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0						0	0	0	0	0	0	0	0	0	0				
.																																			
0x9A0	OTG_ DIEPCTL5	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	Res.	MPSIZ													
	リセット値	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0						0	0	0	0	0	0	0	0	0	0				
.																																			

表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x9A8	OTG_ DIEPINT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC						
	リセット値																									1	0	0	0	0	0	0	0						
.																																						
0x9B8	OTG_ DTXFSTS5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV																					
	リセット値																	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0						
.																																						
0x9B0	OTG_ DIEPTSIZ5	Res.	MCNT		PKTCNT										XFRSIZ																								
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0xB00	OTG_ DOEPCTL0	EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ								
	リセット値	0	0			0	0					0	0	0	0		1														0	0							
0xB08	OTG_ DOEPINT0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2BSTUP	Res.	Res.	Res.	Res.	EPDISD	XFRC						
	リセット値																										0		0	0	0	0	0						
0xB10	OTG_ DOEPTSIZ0	Res.	STUPCNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ												
	リセット値		0	0										0													0	0	0	0	0	0	0	0					
0xB20	OTG_ DOEPCCTL1	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFIRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ																	
	リセット値	0	0	0	0	0	0					0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0						
0xB28	OTG_ DOEPINT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	Res.	Res.	Res.	EPDISD	XFRC						
	リセット値																										0		0	0	0	0	0	0					
0xB30	OTG_ DOEPTSIZ1	Res.	RXDPID/ STUPCNT	PKTCNT										XFRSIZ																									
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0xB50	OTG_ DOEPTSIZ2	Res.	RXDPID/ STUPCNT	PKTCNT										XFRSIZ																									
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

表 195. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...																															
0xBA0	OTG_ DOEPCCTL5	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFIRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ											
	リセット値	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
...																															
0xBA8	OTG_ DOEPINT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	OTEPDIS	STUP	Res.	EPDISD	XFRC
	リセット値																										0		0	0		0	0
...																															
0xBB0	OTG_ DOEPTSIZ5	Res.	RXDPID/ STUPCNT	PKTCNT												XFRSIZ																	
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE00	OTG_ PCGCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHYSLEEP	ENL1GTG	PHYSUSP	Res.	Res.	GATEHCLK	STPPCLK
	リセット値																								0	0	0	0			0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

29.16 OTG_FS プログラミングモデル

29.16.1 コアの初期化

アプリケーションはコア初期化シーケンスを実行する必要があります。パワーアップ中にケーブルが接続された場合、OTG_GINTSTS レジスタの現在の動作モードビット (OTG_GINTSTS の CMOD ビット) はモードを反映します。OTG_FS コントローラは、A プラグが接続されたときにはホストモードに、B プラグが接続されたときにはデバイスモードに入ります。

このセクションでは、電源投入後の OTG_FS コントローラの初期化について説明します。アプリケーションは、ホストモードかデバイスモードかに関係なく、初期化シーケンスに従う必要があります。コアのすべてのグローバルレジスタは、コアの設定に従って以下の手順で初期化されます。

1. OTG_GAHBCFG レジスタの以下のフィールドをプログラムします。
 - グローバル割り込みマスクビット GINTMSK = 1
 - Rx FIFO 非エンプティ (OTG_GINTSTS の RXFLVL ビット)
 - 周期的 Tx FIFO エンプティレベル
2. OTG_GUSBCFG レジスタの以下のフィールドをプログラムします。
 - HNP 対応ビット
 - SRP 対応ビット
 - OTG_FS タイムアウト較正フィールド
 - USB ターンアラウンド時間フィールド
3. ソフトウェアは、OTG_GINTMSK レジスタの以下のビットのマスクを解除する必要があります。
OTG 割り込みマスク
モード不一致割り込みマスク
4. ソフトウェアは、OTG_GINTSTS レジスタの CMOD ビットを読み出すことによって、OTG_FS コントローラがホストモードで動作中か、デバイスモードで動作中かを判別することができます。

29.16.2 ホストの初期化

コアをホストとして初期化するには、アプリケーションは、以下の手順を実行しなければなりません。

1. OTG_GINTMSK レジスタの HPRTINT ビットをプログラムして、マスクを解除します。
2. OTG_HCFCG レジスタをプログラムして、フルスピードホストを選択します。
3. OTG_HPRT レジスタの PPWR ビットを 1 にプログラムします。これによって USB 上で V_{BUS} が駆動されます。
4. OTG_HPRT0 の PCDET 割り込みを待ちます。これは、デバイスがポートに接続されていることを示します。
5. OTG_HPRT レジスタの PRST ビットを 1 にプログラムします。これによってリセットプロセスが開始されます。
6. リセットプロセスが完了するまで、少なくとも 10 ms 待ちます。
7. OTG_HPRT レジスタの PRST ビットを 0 にプログラムします。
8. OTG_HPRT レジスタの PENCHNG 割り込みを待ちます。
9. OTG_HPRT レジスタの PSPD ビットを読み出して、エニューメレーションされたスピードを取得します。
10. HFIR レジスタを、選択された PHY クロック 1 に対応する値でプログラムします。
11. OTG_HCFCG レジスタの FSLSPCS フィールドを、手順 9 で検出したデバイスの速度に従ってプログラムします。FSLSPCS が変更されている場合は、ポートのリセットを行う必要があります。
12. OTG_GRXFSIZ レジスタをプログラムして、受信 FIFO のサイズを選択します。
13. OTG_HNPTXFSIZ レジスタをプログラムして、非周期的トランザクションの非周期的送信 FIFO のサイズと開始アドレスを選択します。
14. OTG_HPTXFSIZ レジスタをプログラムして、周期的トランザクションの周期的送信 FIFO のサイズと開始アドレスを選択します。

デバイスと通信するには、システムソフトウェアは、少なくとも 1 つのチャンネルを初期化し、有効にする必要があります。

29.16.3 デバイスの初期化

アプリケーションは、パワーアップ時、またはホストモードからデバイスモードへの変更後、以下の手順を実行して、コアをデバイスとして初期化する必要があります。

1. OTG_DCFG レジスタの以下のフィールドをプログラムします。
 - デバイススピード
 - 非ゼロ長ステータスOUTハンドシェイク
2. OTG_GINTMSK レジスタをプログラムして、次の割り込みのマスクを解除します。
 - USB リセット
 - エニューメレーション終了
 - アーリーサスペンド
 - USB サスペンド
 - SOF
3. OTG_GINTSTS の USBRST 割り込みを待ちます。これは、この割り込みの受信時、約 10ms 続くリセットが USB で検出されたことを示します。

OTG_GINTSTS の ENUMDNE 割り込みを待ちます。この割り込みは、USB 上でリセットが終了したことを示します。この割り込みを受信したとき、アプリケーションは、OTG_DSTS レジスタを読み出して、エニューメレーション速度を判別し、[エニューメレーション完了時のエンドポイント初期化 \(1071 ページ\)](#) にリストされている手順を実行する必要があります。



この時点で、デバイスは、SOF パケットを受け入れて、コントロールエンドポイント 0 でコントロール転送を実行する準備ができたことになります。

29.16.4 ホストプログラミングモデル

チャネルの初期化

アプリケーションが接続されたデバイスと通信するためには、1 つまたは複数のチャネルを初期化する必要があります。チャネルを初期化して、有効にするには、アプリケーションは、以下の手順を実行する必要があります。

1. OTG_GINTMSK レジスタをプログラムして、以下の割り込みのマスクを解除します。
2. チャネル割り込み
 - OUT トランザクションの非周期的送信 FIFO エンプティ割り込み (パケットカウンタフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能)
 - OUT トランザクションの非周期的送信 FIFO 半エンプティ割り込み (パケットカウンタフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能)
3. OTG_HAINTMSK レジスタをプログラムして、選択されたチャネルの割り込みのマスクを解除します。
4. OTG_HCINTMSK レジスタをプログラムして、ホストチャネル割り込みレジスタで指定された、所定のトランザクション関連割り込みのマスクを解除します。
5. 選択されたチャネルの OTG_HCTSIZx レジスタを、合計転送サイズ (バイト数) およびショートパケットを含む予想されるパケット数でプログラムします。アプリケーションは、PID フィールドを初期データ PID (最初の OUT トランザクションで使用される PID、または最初の IN トランザクションから期待される PID) でプログラムする必要があります。
6. 選択されたチャネルの OTG_HCCHARx レジスタを、タイプ、スピード、方向など、デバイスのエンドポイントの特性でプログラムします。(チャネルは、アプリケーションがパケットを送信または受信する準備ができていない場合にのみ、チャネルイネーブルビットを 1 にセットすることで有効にできます)。

チャネルの停止

アプリケーションは、OTG_HCCHARx レジスタの CHDIS および CHENA ビットを 1 にセットすることによって、任意のチャネルを無効にできます。これにより、OTG_FS ホストは、ポストされたリクエスト (ある場合) をフラッシュして、チャネル停止割り込みを生成できます。アプリケーションは、チャネルを他のトランザクションに再割り当てする前に、OTG_HCINTx 内の CHH 割り込みを待つ必要があります。OTG_FS ホストは、USB 上ですでに転送が開始しているトランザクションに対しては割り込みを行いません。

チャネルを無効にする前に、アプリケーションは、非周期的リクエストキュー (非周期的チャネルを無効にするとき)、または周期的リクエストキュー (周期的チャネルを無効にするとき) に使用可能な空きスペースが少なくとも 1 つ以上存在することを確認する必要があります。アプリケーションは、リクエストキューがフル状態のときには (チャネルを無効化する前)、OTG_HCCHARx レジスタの CHDIS ビットを 1 にセットし CHENA ビットを 0 にクリアするだけで、ポストされたリクエストを一掃できます。

アプリケーションは、以下のいずれかの条件のとき、チャンネルを無効にすることが期待されます。

1. IN または OUT チャンネルで、OTG_HCINTx レジスタの STALL、TXERR、BBERR、または DTERR 割り込みが受信されたとき。アプリケーションは、停止割り込みを受信する前に、同じチャンネルについての他の割り込み (DTERR、Nak、Data、TXERR) を受信できなければなりません。
2. OTG_GINTSTS レジスタの DISCINT (デバイス切断) 割り込みを受信したとき。(アプリケーションは、すべての有効なチャンネルを無効にすることが期待されます)。
3. アプリケーションが正常完了の前に転送を中断したとき。

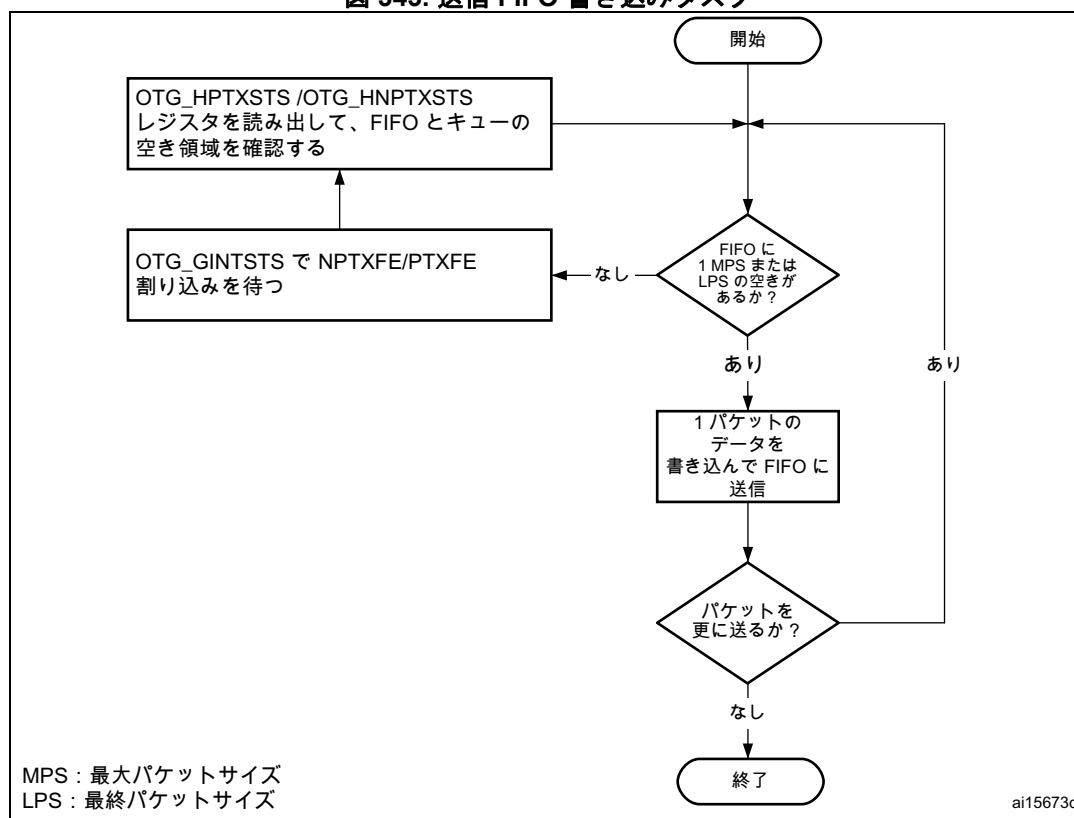
動作モデル

アプリケーションが接続されたデバイスと通信するためには、チャンネルを初期化する必要があります。このセクションでは、さまざまなタイプの USB トランザクションについて、実行しなければならない動作のシーケンスについて説明します。

● 送信 FIFO への書き込み

OTG_FS ホストは、パケットの最後のダブルワードの書き込みとともに、周期的/非周期的リクエストキューにエントリ (OUT リクエスト) を自動的に書き込みます。アプリケーションは、送信 FIFO への書き込みを開始する前に、周期的/非周期的リクエストキューに少なくとも 1 つ以上の空きスペースがあることを確認しなければなりません。アプリケーションは送信 FIFO に必ずダブルワード単位で書き込みを行う必要があります。パケットサイズが DWORD 単位で揃えられていない場合、アプリケーションはパディングを使用する必要があります。OTG_FS ホストは、プログラムされた最大パケットサイズと転送サイズに基づいて、実際のパケットサイズを決定します。

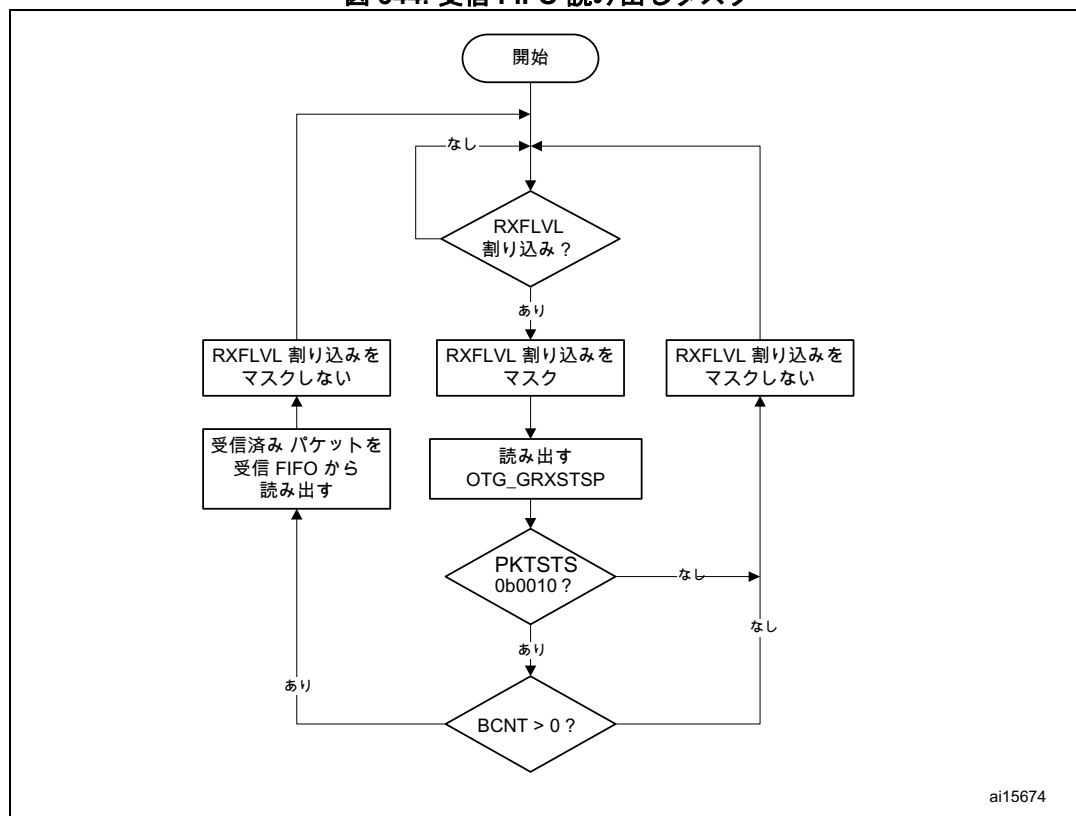
図 343. 送信 FIFO 書き込みタスク



● 受信 FIFO の読み出し

アプリケーションは、IN データパケット (bx0010) 以外のすべてのパケットステータスを無視しなければなりません。

図 344. 受信 FIFO 読み出しタスク



● バルクおよびコントロール OUT/SETUP トランザクション

典型的なバルクまたはコントロール OUT/SETUP パイプライントランザクションレベルの動作を 図 345 に示します。チャンネル 1 (ch_1) に注目してください。2 つのバルク OUT パケットが転送されます。コントロール SETUP トランザクションは同じように動作しますが、1 パケットのみです。ここでは、以下のように仮定されています。

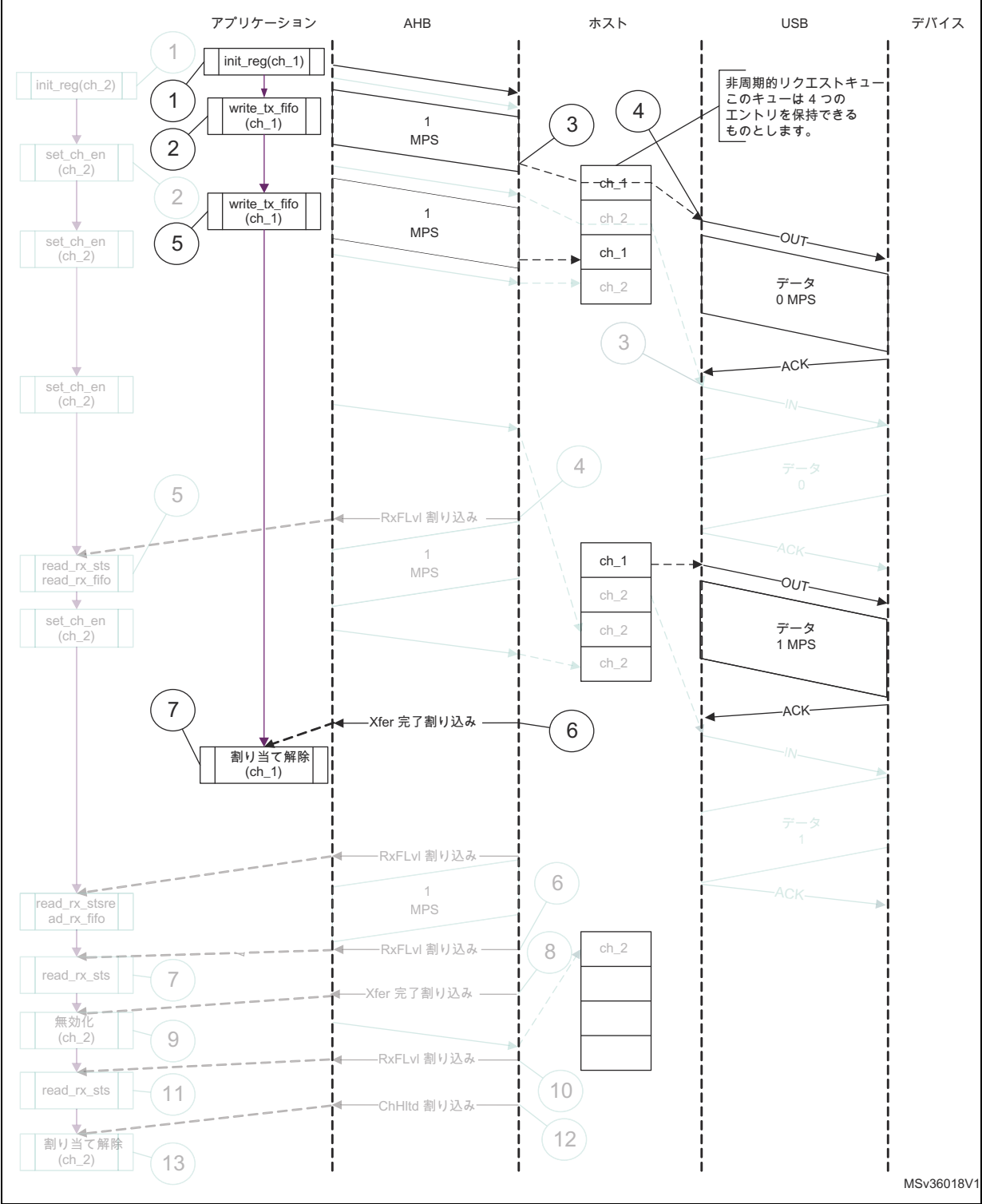
- アプリケーションは、最大パケットサイズの 2 つのパケットを送信しようとしています (転送サイズ = 1,024 バイト)。
- 非周期的送信 FIFO は 2 つのパケットを保持することができます (FS の場合 128 バイト)。
- 非周期的リクエストキューの深さ = 4。

● 通常のバルクおよびコントロール OUT/SETUP 動作

動作シーケンス (チャンネル 1) は、次のとおりです。

1. チャンネル 1 を初期化します。
2. チャンネル 1 の最初のパケットを書き込みます。
3. 最後のワードの書き込みと同時に、コアは、非周期的リクエストキューにエントリを書き込みます。
4. 非周期的キューがエンプティでなくなるとすぐに、コアは、現在のフレームで OUT トークンの送信を試みます。
5. チャンネル 1 の 2 番目 (最後) のパケットを書き込みます。
6. コアは、最後のトランザクションが正常に完了すると、XFRC 割り込みを生成します。
7. XFRC 割り込みに応答して、他の転送のためにチャンネル割り当てを解除します。
8. 非 ACK 応答の取り扱い

図 345. 通常のバルク／コントロール OUT/SETUP



バルクおよびコントロール OUT/SETUP トランザクションのためのチャンネル固有の割り込みサービスルーチンを、以下のコードサンプルで示します。

- バルク／コントロール OUT/SETUP とバルクまたはコントロール IN トランザクションの割り込みサービスルーチン

- a) バルク／コントロール OUT/SETUP

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else if (STALL)
{
    Transfer Done = 1
    Unmask CHH
    Disable Channel
}
else if (NAK or TXERR )
{
    Rewind Buffer Pointers
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
    else
    {
        Reset Error Count
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
```

```
}
```

アプリケーションは、送信 FIFO およびリクエストキューの中に使用可能なスペースがある場合、データパケットを送信 FIFO に書き込むことが期待されます。アプリケーションは、OTG_GINTSTS レジスタの NPTXFE 割り込みを使用して、送信 FIFO のスペースを見つけることができます。

b) バルク/コントロール IN

```
Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
{
    Reset Error Count
    Unmask CHH
    Disable Channel
    Reset Error Count
    Mask ACK
}
else if (TXERR or BBERR or STALL)
{
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
}
else if (DTERR)
{
    Reset Error Count
}
```

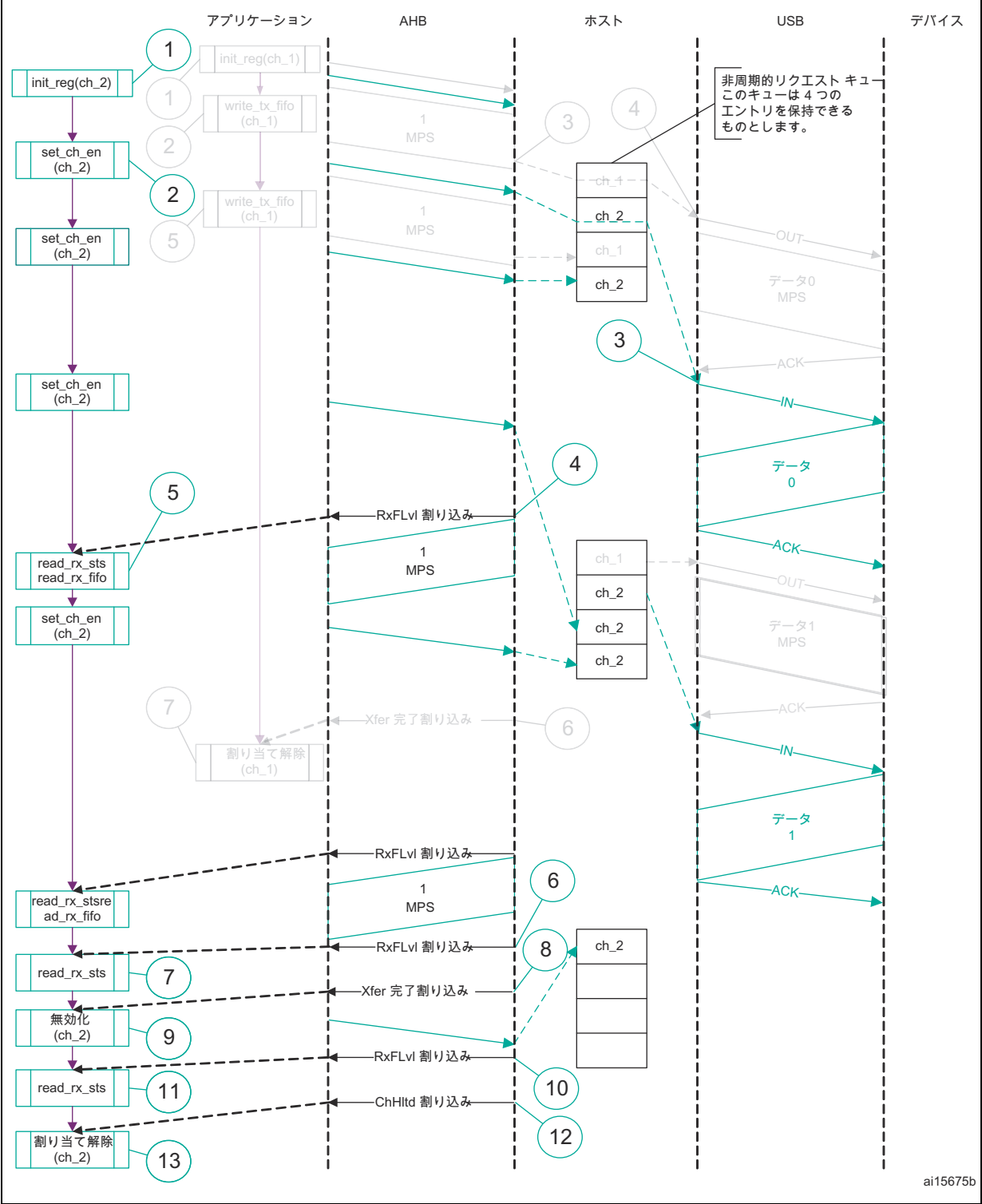
アプリケーションは、リクエストキューのスペースが使用可能なときには、XFRC 割り込みが受信されるまで、リクエストを書き込むことが期待されます。

● バルクおよびコントロール IN トランザクション

典型的なバルクまたはコントロール IN パイプライントランザクションレベルの動作を [図 346](#) に示します。チャンネル 2 (ch_2) に注目してください。ここでは、以下のように仮定されています。

- アプリケーションは、最大パケットサイズの 2 つのパケットを受信しようとしています (転送サイズ = 1024 バイト)。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットと、パケットあたり 2 つのステータスワード (FS 場合 72 バイト) を収容することができます。
- 非周期的リクエストキューの深さ = 4。

図 346. バルク／コントロール IN トランザクション



動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。
2. OTG_HCCHAR2 の CHENA ビットをセットして、IN リクエストを非周期的リクエストキューに書き込みます。
3. コアは、現在の OUT トランザクションの完了後、IN トークンの送信を試みます。
4. コアは、受信したパケットが受信 FIFO に書き込まれると、RXFLVL 割り込みを生成します。
5. RXFLVL 割り込みに応答して、RXFLVL 割り込みをマスクし、受信パケットステータスを読み出して受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。続いて、RXFLVL 割り込みのマスクを解除します。
6. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。
7. アプリケーションは、受信パケットステータスを読み出して、受信パケットステータスが IN データパケットでないとき (OTG_GRXSTSR ≠ 0b0010 の PKTSTS) には無視する必要があります。
8. コアは、受信パケットステータスを読み出されると、XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して、チャンネルを無効にし、それ以降のリクエストについて、OTG_HCCHAR2 レジスタへの書き込みを停止します。コアは、OTG_HCCHAR2 に書き込まれると、非周期的リクエストキューにチャンネルディセーブルリクエストを書き込みます。
10. コアは、停止ステータスが受信 FIFO に書き込まれると、RXFLVL 割り込みを生成します。
11. 受信パケットステータスを読み出して、それを無視します。
12. コアは、停止ステータスが受信 FIFO からポップされるとすぐに、CHH 割り込みを生成します。
13. CHH 割り込みに応答して、他の転送のためにチャンネル割り当てを解除します。
14. 非 ACK 応答の取り扱い

● コントロールトランザクション

コントロール転送のセットアップ、データ、およびステータスステージは 3 つの個別の転送として実行されなければなりません。セットアップ、データ、およびステータスステージの OUT トランザクションは、すでに説明したバルク OUT トランザクションと同様に実行されます。データまたはステータスステージの IN トランザクションは、すでに説明したバルク IN トランザクションと同様に実行されます。3 つのステージすべてについて、アプリケーションは OTG_HCCHAR1 レジスタの EPTYP フィールドを Control に設定することが期待されます。セットアップステージでは、アプリケーションは OTG_HCTSIZ1 レジスタの PID フィールドを SETUP に設定することが期待されます。

● インタラプト OUT トランザクション

典型的なインタラプト OUT 動作を [図 347](#) に示します。ここでは、以下のように仮定されています。

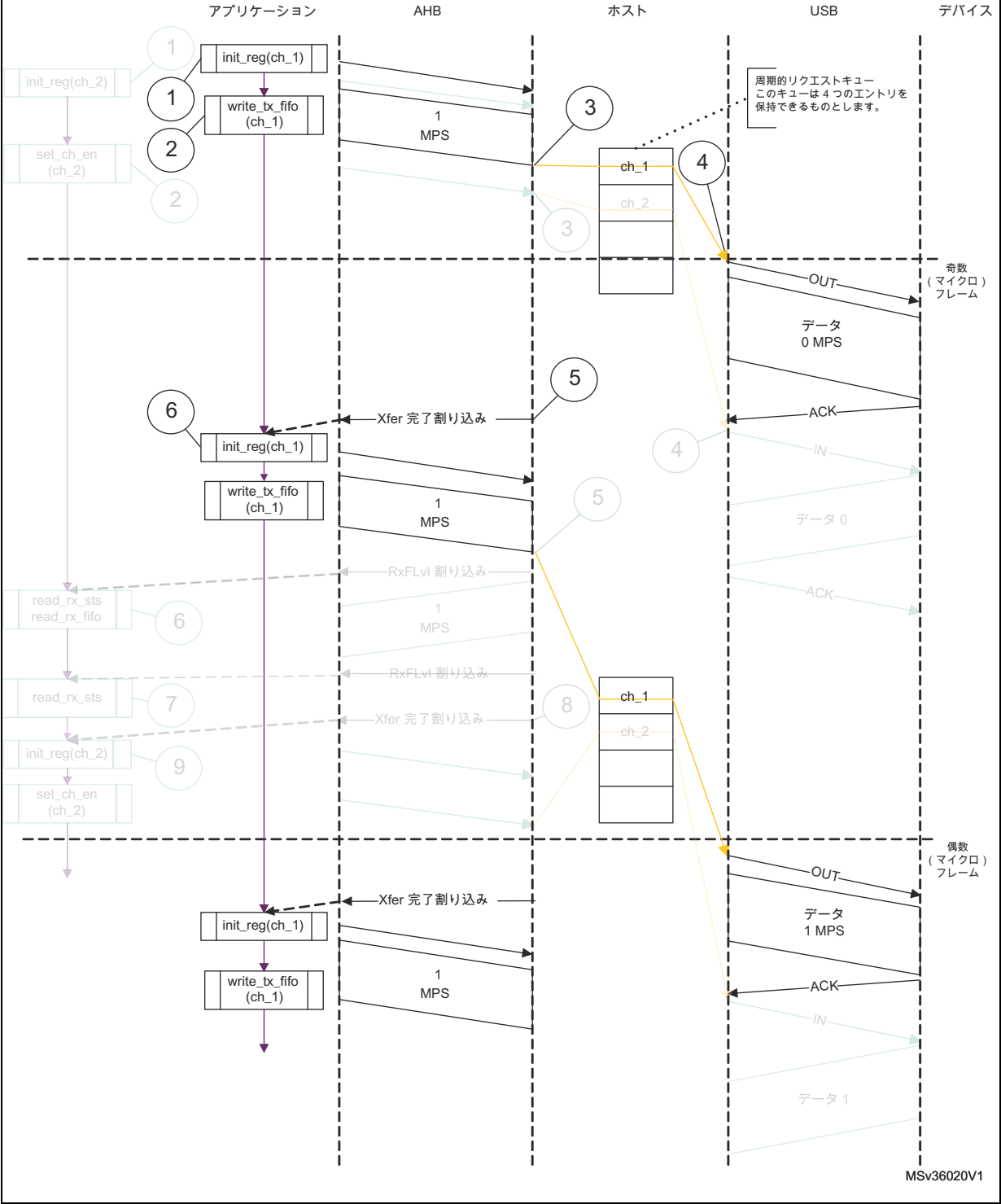
- アプリケーションは、奇数フレーム (転送サイズ = 1024 バイト) から始めて、フレームごとに 1 パケット (最大 1 パケットサイズ) の送信を試みています。
- 周期的送信 FIFO は 1 パケット (1 KB) を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャンネル 1 を初期化し、有効にします。アプリケーションは OTG_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
2. チャンネル 1 の最初のパケットを書き込みます。
3. 各パケットの最後のワードの書き込みとともに、OTG_FS ホストは、周期的リクエストキューにエントリを書き込みます。
4. OTG_FS ホストは、次の (奇数) フレームで OUT トークンの送信を試みます。
5. OTG_FS ホストは、最後のパケットが正常に送信されると、XFRC 割り込みを生成します。
6. XFRC 割り込みに応答して、次の転送のためにチャンネルを再初期化します。



図 347. 通常インタラプト OUT



- インタラプト OUT/IN トランザクションのための割り込みサービスルーチン

- a) インタラプト OUT

```
Unmask (NAK/TXERR/STALL/XFRC/FRMOR)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else
    if (STALL or FRMOR)
    {
        Mask ACK
        Unmask CHH
        Disable Channel
        if (STALL)
        {
            Transfer Done = 1
        }
    }
else
    if (NAK or TXERR)
    {
        Rewind Buffer Pointers
        Reset Error Count
        Mask ACK
        Unmask CHH
        Disable Channel
    }
else
    if (CHH)
    {
        Mask CHH
        if (Transfer Done or (Error_count == 3))
        {
            De-allocate Channel
        }
        else
        {
            Re-initialize Channel (in next b_interval - 1 Frame)
        }
    }
else
    if (ACK)
    {
        Reset Error Count
        Mask ACK
    }
```

アプリケーションは OTG_GINTSTS レジスタの NPTXFE 割り込みを使用して、送信 FIFO のスペースを見つけてます。

- b) インタラプト IN



```
Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    if (OTG_HCTSIZx.PKTCNT == 0)
    {
        De-allocate Channel
    }
    else
    {
        Transfer Done = 1
        Unmask CHH
        Disable Channel
    }
}
else
    if (STALL or FRMOR or NAK or DTERR or BBERR)
    {
        Mask ACK
        Unmask CHH
        Disable Channel
        if (STALL or BBERR)
        {
            Reset Error Count
            Transfer Done = 1
        }
        else
            if (!FRMOR)
            {
                Reset Error Count
            }
    }
else
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
        Unmask CHH
        Disable Channel
    }
else
    if (CHH)
    {
        Mask CHH
        if (Transfer Done or (Error_count == 3))
        {
            De-allocate Channel
        }
        else
            Re-initialize Channel (in next b_interval - 1 /Frame)
    }
```

```

    }
else
    if (ACK)
    {
        Reset Error Count
        Mask ACK
    }

```

● インタラプト IN トランザクション

ここでは、以下のように仮定されています。

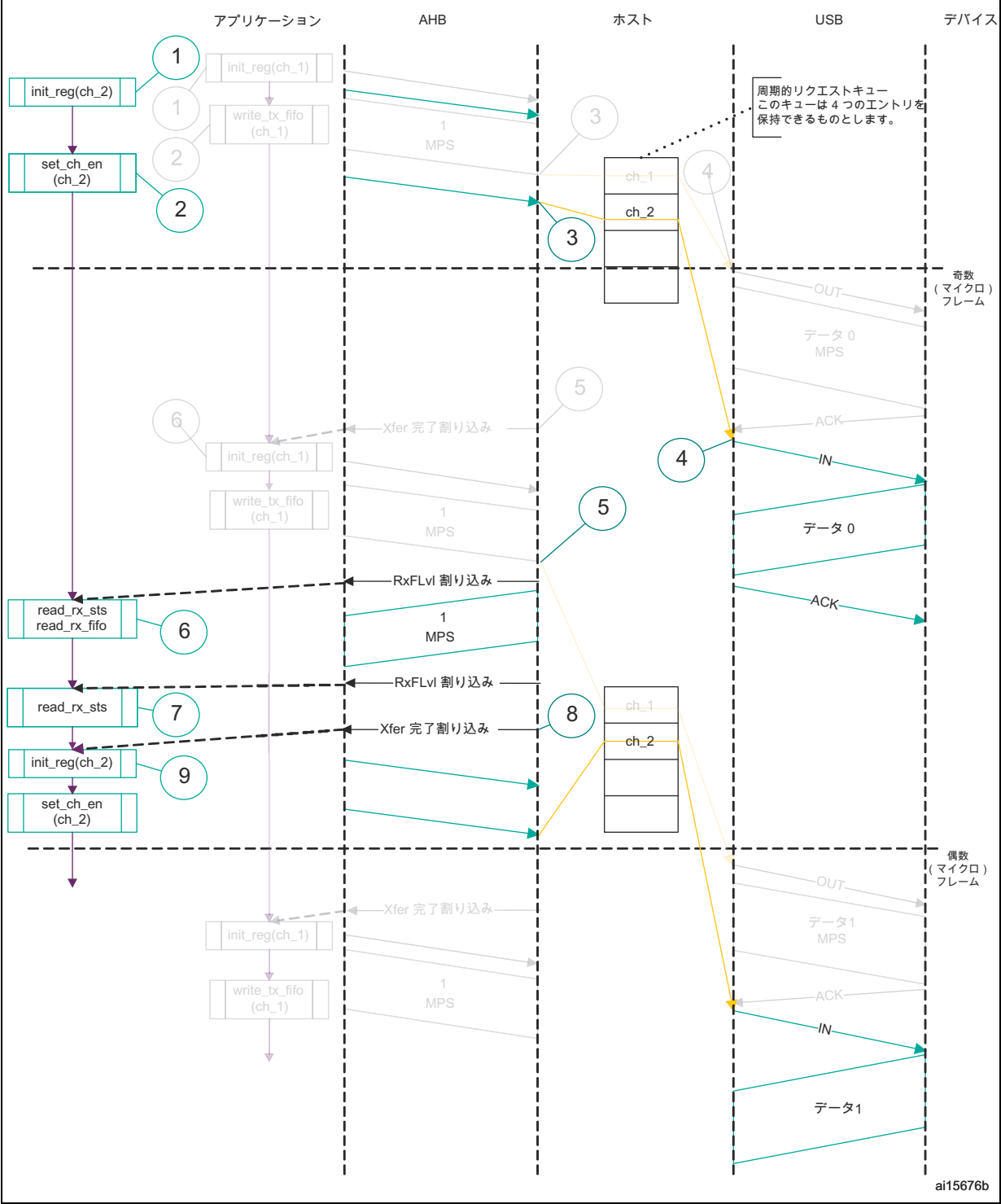
- アプリケーションは、奇数フレーム (転送サイズ = 1024 バイト) から始めて、フレームごとに 1 パケット (最大 1 パケットサイズ) の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード (1,031 バイト) を保持することができます。
- 周期的リクエストキューの深さ = 4。

● 通常のインタラプト IN 動作

動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。アプリケーションは OTG_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
2. OTG_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューを書き込みます。
3. OTG_FS ホストは、CHENA ビットがセットされた OTG_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
4. OTG_FS ホストは、次の (奇数) フレームで IN トークンの送信を試みます。
5. IN パケットが受信され、受信 FIFO に書き込まれると、OTG_FS ホストは RXFLVL 割り込みを生成します。
6. RXFLVL 割り込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割り込みをマスクし、すべてのパケットを読み出した後にマスクを解除しなければなりません。
7. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。アプリケーションは、受信パケットステータスを読み出して、受信パケットステータスが IN データパケットでないとき (GRXSTSR ≠ 0b0010 の PKTSTS) には無視する必要があります。
8. コアは、受信パケットステータスが読み出されると XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して OTG_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG_HCTSIZ2 の PKTCNT ビットが 0 でない場合は、次の転送 (ある場合) のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG_HCTSIZ2 の PKTCNT ビットが 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

図 348. 通常インタラプト IN



- アイソクロナス OUT トランザクション

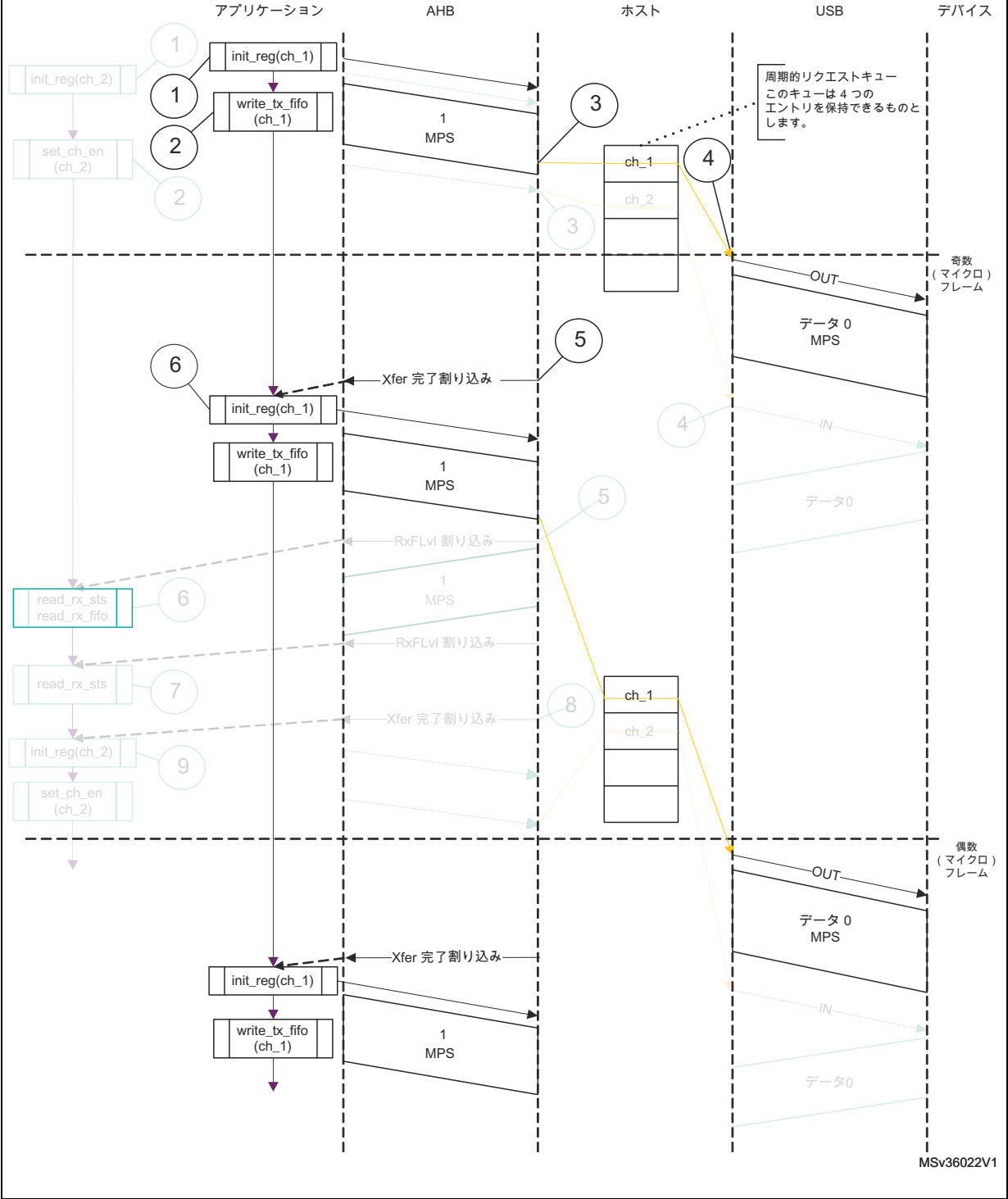
典型的なアイソクロナス OUT 動作を [図 348](#) に示します。ここでは、以下のように仮定されています。

- アプリケーションは、奇数フレームから始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の送信を試みています。（転送サイズ = 1024 バイト）。
- 周期的送信 FIFO は 1 パケット（1 KB）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャンネル 1 を初期化し、有効にします。アプリケーションは OTG_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
2. チャンネル 1 の最初のパケットを書き込みます。
3. 各パケットの最後のワードの書き込みとともに、OTG_FS ホストは、周期的リクエストキューにエントリを書き込みます。
4. OTG_FS ホストは、次のフレーム（奇数）で OUT トークンの送信を試みます。
5. OTG_FS ホストは、最後のパケットが正常に送信されると、XFRC 割り込みを生成します。
6. XFRC 割り込みに応答して、次の転送のためにチャンネルを再初期化します。
7. 非 ACK 応答の取り扱い

図 349. アイソクロナス OUT トランザクション



- アイソクロナス OUT/IN トランザクションのための割り込みサービスルーチン

コードサンプル: アイソクロナス OUT

```
Unmask (FRMOR/XFRC)
if (XFRC)
{
    De-allocate Channel
}
else
{
    if (FRMOR)
    {
        Unmask CHH
        Disable Channel
    }
    else
    {
        if (CHH)
        {
            Mask CHH
            De-allocate Channel
        }
    }
}
```

コードサンプル: アイソクロナス IN

```
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
{
    if (XFRC and (OTG_HCTSIZx.PKTCNT == 0))
    {
        Reset Error Count
        De-allocate Channel
    }
    else
    {
        Unmask CHH
        Disable Channel
    }
}
else
{
    if (TXERR or BBERR)
    {
        Increment Error Count
        Unmask CHH
        Disable Channel
    }
    else
    {
        if (CHH)
        {
            Mask CHH
            if (Transfer Done or (Error_count == 3))
            {
                De-allocate Channel
            }
        }
        else
        {
            {

```

```
        Re-initialize Channel  
    }  
}
```

● アイソクロナス IN トランザクション

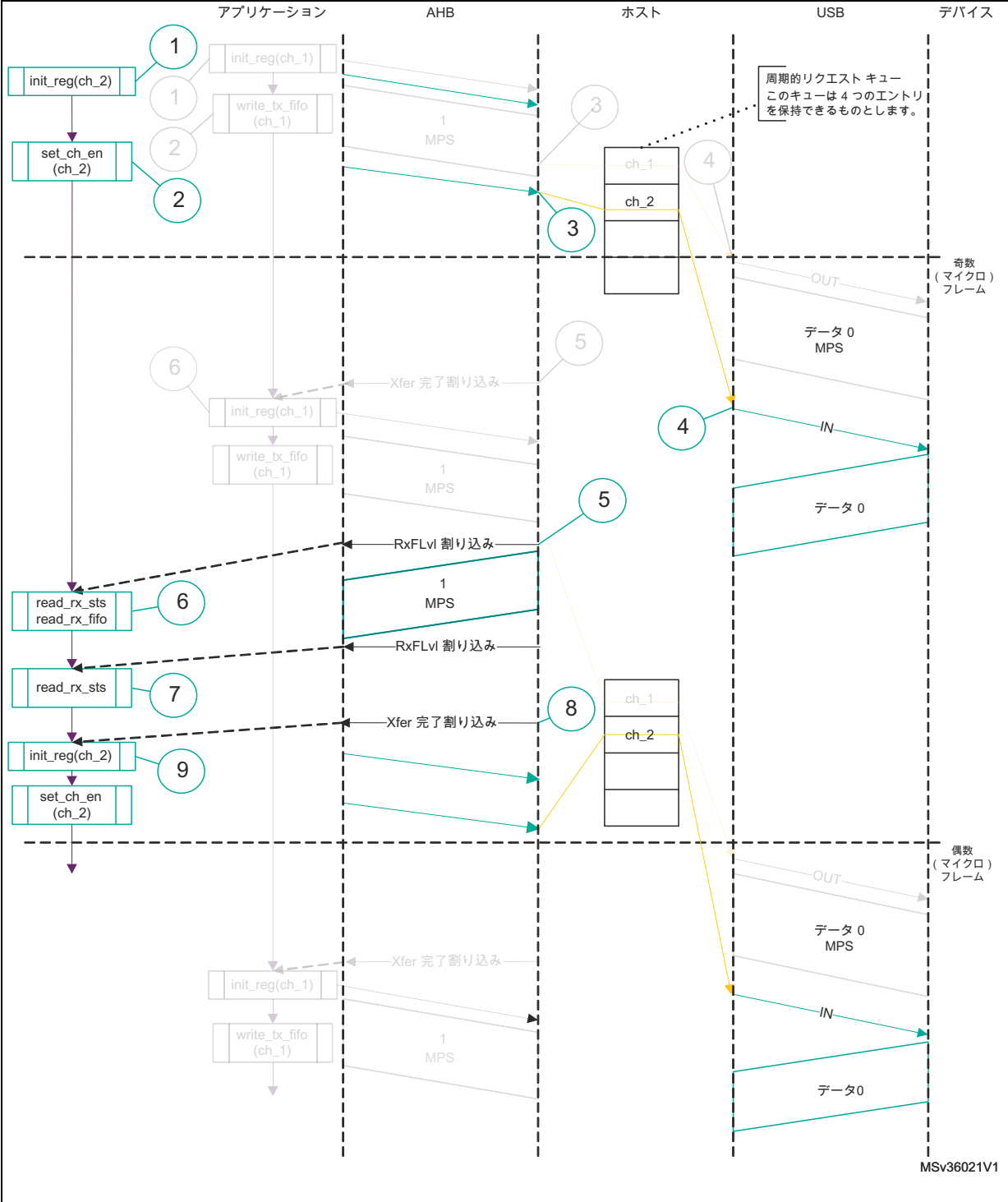
ここでは、以下のように仮定されています。

- アプリケーションは、次の奇数フレーム（転送サイズ = 1024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード（1,031 バイト）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。アプリケーションは OTG_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
2. OTG_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューを書き込みます。
3. OTG_FS ホストは、CHENA ビットがセットされた OTG_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
4. OTG_FS ホストは、次の奇数フレームで IN トークンの送信を試みます。
5. IN パケットが受信され、受信 FIFO に書き込まれると、OTG_FS ホストは RXFLVL 割り込みを生成します。
6. RXFLVL 割り込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割り込みをマスクし、すべてのパケットを読み出した後でマスクを解除しなければなりません。
7. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。今度は、アプリケーションは受信パケットステータスを読み出して、それが IN データパケットでなかった場合（OTG_GRXSTSR ≠ 0b0010 の PKTSTS ビット）、無視する必要があります。
8. コアは、受信パケットステータスが読み出されると XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して OTG_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG_HCTSIZ2 の PKTCNT ≠ 0 の場合、次の転送（ある場合）のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG_HCTSIZ2 の PKTCNT = 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

図 350. アイソクロナス IN トランザクション



● キューの深さの選択

周期的および非周期的リクエストキューの深さは、アクセスされる周期的/非周期的エンドポイントの数に合うように注意して選択してください。

非周期的リクエストキューの深さは、非周期的転送の性能に影響を与えます。キューが深いほど (FIFO のサイズが十分であれば)、コアは非周期的転送をより多くパイプライン化できます。キューのサイズが小さいと、コアはキューのスペースが空いたときしか新しいリクエストを入れることができません。

コアの周期的リクエストキューの深さは、周期的転送をスケジュールどおりに実行するために不可欠です。周期的キューの深さは、マイクロフレーム内でスケジュールされた周期的転送の数に基づいて選択してください。周期的リクエストキューの深さがマイクロフレーム内でスケジュールされた周期的転送の数より小さい場合、フレームオーバーラン条件が発生します。

● バブル条件の取り扱い

OTG_FS コントローラは、パケットバブルとポートバブルの 2 つのバブルを処理します。パケットバブルは、デバイスがチャネルの最大パケットサイズよりも多くのデータを送信した場合に発生します。ポートバブルは、コアが EOF2 (フレーム 2 の終わりの SOF にきわめて近い) でデバイスからデータを受信し続けると発生します。

OTG_FS コントローラがパケットバブルを検出すると、Rx バッファへのデータの書き込みを停止して、パケットの終わり (EOP) を待ちます。EOP を検出すると コントローラは Rx バッファにすでに書き込まれたデータを一扫して、アプリケーションに対するバブル割り込みを生成します。

OTG_FS コントローラがポートバブルを検出すると、Rx FIFO をフラッシュして、ポートを無効にします。コアは、ポートディセーブル割り込み (OTG_GINTSTS の HPRTINT、OTG_HPRT の PENCHNG) を生成します。この割り込みを受信すると、アプリケーションは OTG_HPRT の POCA ビットをチェックして、これが過電流条件 (ポートディセーブル割り込みのもう 1 つの原因) によるものではないことを確認してから、ソフトリセットを行う必要があります。コアは、ポートバブル条件を検出した後は、それ以上トークンを送信しません。

注： このセクションの内容は、USB OTG HS にのみ適用されます。

29.16.5 デバイスプログラミングモデル

USB リセット時のエンドポイントの初期化

1. すべての OUT エンドポイントの NAK ビットをセットします。
 - OTG_DOEPCTLx の SNAK = 1 (すべての OUT エンドポイントについて)
2. 以下の割り込みビットのマスクを解除します。
 - OTG_DAINMSK の INEP0 = 1 (コントロール 0 IN エンドポイント)
 - OTG_DAINMSK の OUTEP0 = 1 (コントロール 0 OUT エンドポイント)
 - OTG_DOEPMSK の STUPM = 1
 - OTG_DOEPMSK の XFRCM = 1
 - OTG_DIEPMSK の XFRCM = 1
 - OTG_DIEPMSK の TOM = 1
3. 各 FIFO の DATA FIFO RAM をセットアップします。
 - OTG_GRXFSIZ レジスタをプログラムして、コントロール OUT データとセットアップデータを受信できるようにします。閾値の設定が無効の場合、これは、少なくとも、コントロールエンドポイント 0 の最大パケットサイズ 1 個分 + 2 ワード (コントロール OUT データパケットのステータス用) + 10 ワード (セットアップパケット用) でなければなりません。

- OTG_DIEPTXF0 レジスタを、コントロール IN データを送信できるようにプログラムします (選択された FIFO 番号に応じて)。これは、少なくとも、コントロールエンドポイント 0 の最大パケットサイズ 1 個分以上でなければなりません。
- 4. SETUP パケットを受信するために、コントロール OUT エンドポイント 0 のエンドポイント固有レジスタの以下のフィールドをプログラムします。
 - OTG_DOEPTSIZE0 の STUPCNT = 3 (最大 3 つの連続 SETUP パケットを受信するため)

この時点で SETUP パケットを受信するために必要なすべての初期化が終了したことになります。

エニユメレーション完了時のエンドポイント初期化

1. エニユメレーション終了割り込み (OTG_GINTSTS の ENUMDNE ビット) 時には OTG_DSTS レジスタを読み出して、エニユメレーションスピードを決めます。
2. OTG_DIEPCTL0 の MPSIZ フィールドをプログラムして、最大パケットサイズを設定します。このステップでは、コントロールエンドポイント 0 を設定します。コントロールエンドポイントの最大パケットサイズは、エニユメレーションスピードに依存します。

この時点で、デバイスは SOF パケットを受信する準備ができ、コントロールエンドポイント 0 でコントロール転送を行うように設定されたことになります。

SetAddress コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetAddress コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. OTG_DCFG レジスタに SetAddress コマンドで受信したデバイスアドレスをプログラムします。
2. ステータス IN パケットを送信するように、コアをプログラムします。

SetConfiguration/SetInterface コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetConfiguration または SetInterface コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. SetConfiguration コマンドを受信したとき、アプリケーションは、新しい設定で有効なエンドポイントの特性で、エンドポイントのレジスタをプログラムする必要があります。
2. SetInterface コマンドを受信したとき、アプリケーションは、このコマンドの影響を受けるエンドポイントのレジスタをプログラムする必要があります。
3. 前の設定または代替設定ではアクティブであったエンドポイントが、新しい設定または代替設定では無効なことがあります。これらの無効なエンドポイントは、機能を停止させる必要があります。
4. アクティブな各エンドポイントの割り込みのマスクを解除し、OTG_DAINTRMSK レジスタですべての非アクティブなエンドポイントに対する割り込みをマスクします。
5. 各 FIFO のデータ FIFO RAM をセットアップします。
6. 必要なすべてのエンドポイントを設定した後、アプリケーションは、ステータス IN パケットを送信するようにコアをプログラムする必要があります。

この時点で、デバイスのコアは、任意のタイプのデータパケットを送信/受信できるように設定されたことになります。

エンドポイントのアクティブ化

このセクションでは、デバイスエンドポイントをアクティブ化するか、既存のデバイスエンドポイントを新しいタイプに設定するために必要な手順について説明します。

- 必要なエンドポイントの特性を OTG_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の以下のフィールドにプログラムします。
 - 最大パケットサイズ
 - USB アクティブエンドポイント = 1
 - エンドポイント開始データトグル (インタラプトおよびバルクエンドポイントの場合)
 - エンドポイントタイプ
 - Tx FIFO 番号
- エンドポイントがアクティブ化されると、コアは、そのエンドポイントに宛てたトークンのデコードを開始し、そのエンドポイントで受信された有効な各トークンについて有効なハンドシェイクを送出します。

エンドポイントの機能停止

このセクションでは、既存のエンドポイントの機能を停止させるために必要な手順について説明します。

- 非アクティブにすべきエンドポイントでは、OTG_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の USB のアクティブエンドポイントビットをクリアします。
- エンドポイントが非アクティブな状態になると、コアは、そのエンドポイントにアドレス指定されたトークンを無視するので USB 上でタイムアウトが発生します。

注： アプリケーションは、以下の条件を満たすように、デバイスコアがトラフィックを処理できるようにセットアップする必要があります。
OTG_GINTMSK レジスタの NPTXFEM および RXFLVLM をクリアする必要があります。

動作モデル

SETUP および OUT データ転送

このセクションでは、データ OUT 転送および SETUP トランザクション時の内部データフローとアプリケーションレベルの動作について説明します。

● パケットの読み出し

このセクションでは、受信 FIFO からパケット (OUT データおよび SETUP パケット) を読み出す方法について説明します。

- RXFLVL 割り込み (OTG_GINTSTS レジスタ) を受信すると、アプリケーションは、受信ステータスポップレジスタ (OTG_GRXSTSP) を読み出す必要があります。
- アプリケーションは、RXFLVLM = 0 (OTG_GINTMSK レジスタ) を書き込むことによって、受信 FIFO からそのパケットが読み出されるまで、RXFLVL 割り込み (OTG_GINTSTS レジスタ) をマスクすることができます。
- 受信パケットのバイトカウントが 0 でない場合、データのバイトカウントは受信データ FIFO からポップされ、メモリに格納されます。受信パケットのバイトカウントが 0 の場合、受信データ FIFO からデータはポップされません。

4. FIFO のパケット受信ステータスの読み出し結果は、以下のいずれかを示します。
 - a) グローバル OUT NAK のパターン
PKTSTS = グローバル OUT NAK、BCNT = 0x000、EPNUM = (0x0)、DPID = (0b00)。
これらのデータは、グローバル OUT NAK ビットが有効になっていることを示します。
 - b) SETUP パケットのパターン
PKTSTS = SETUP、BCNT = 0x008、EPNUM = コントロール EP 番号、DPID = DATA0。
これらのデータは、指定されたエンドポイントの SETUP パケットを受信 FIFO から読み出せることを示します。
 - c) セットアップステージ終了パターン
PKTSTS = セットアップステージ終了、BCNT = 0x0、EPNUM = コントロール EP 番号、DPID = (0b00)。
これらのデータは、指定されたエンドポイントのセットアップステージが完了し、データステージが開始されたことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定されたコントロール OUT エンドポイントでセットアップ割り込みをアサートします。
 - d) データ OUT パケットのパターン
PKTSTS = データ OUT、BCNT = 受信データ OUT パケットのサイズ ($0 \leq \text{BCNT} \leq 1024$)、EPNUM = パケットを受信した EPNUM、DPID = 実際のデータ PID。
 - e) データ転送完了パターン
PKTSTS = データ OUT 転送終了、BCNT = 0x0、EPNUM = データ転送が完了した OUT EP番号、DPID = (0b00)。
これらのデータは、指定された OUT エンドポイントの OUT データ転送が完了したことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定された OUT エンドポイントで転送完了割り込みをアサートします。
5. データペイロードが受信 FIFO からポップされた後、RXFLVL 割り込み (OTG_GINTSTS) がマスク解除されなければなりません。
6. 手順 1~5 は、アプリケーションが OTG_GINTSTS レジスタの RXFLVL による割り込みラインのアサーションを検出するたびに繰り返されます。空の受信 FIFO を読み出すと、定義されていないコア動作を引き起こすことがあります。


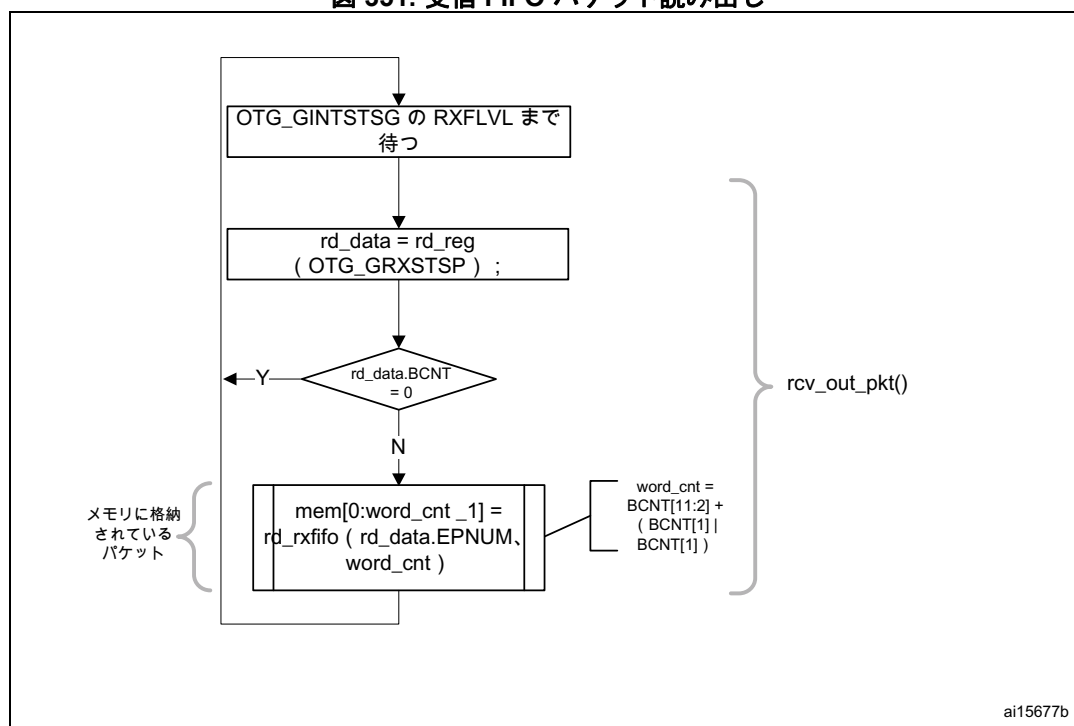
 351 上記の手順のフローチャートです。

図 351. 受信 FIFO パケット読み出し



SETUP トランザクション

このセクションでは、コアが SETUP パケットを処理する方法と、SETUP トランザクションを処理するアプリケーションのシーケンスについて説明します。

● アプリケーションの要件

1. SETUP パケットを受信するには、コントロール OUT エンドポイントの STUPCNT フィールド (OTG_DOEPTSIZE レジスタ) が、ゼロでない値にプログラムされなければなりません。アプリケーションが STUPCNT フィールドをゼロでない値にプログラムすると、NAK のステータスと OTG_DOEPCTLx レジスタの EPENA ビットの設定に関係なく、コアは、SETUP パケットを受信して、受信 FIFO に書き込みます。STUPCNT フィールドは、コントロールエンドポイントが SETUP パケットを受信するたびにデクリメントされます。SETUP パケットを受信する前に、STUPCNT フィールドが適切な値にプログラムされていなかった場合、コアは SETUP パケットを受信し、STUPCNT フィールドをデクリメントしますが、アプリケーションはコントロール転送のセットアップ ステージで受信した SETUP パケットの正しい数を判定できないことがあります。

– OTG_DOEPTSIZE レジスタの STUPCNT = 3

2. コントロールエンドポイントで 3 つまでの SETUP パケットを受信するためには、アプリケーションは、常に受信データ FIFO に余分のスペースを割り当てておく必要があります。
 - 確保すべきスペースは 10 ワード分です。最初の SETUP パケット用に 3 ワードが必要であり、セットアップステージ終了ワードのために 1 ワード、すべてのコントロールエンドポイントにわたって 2 つの余分な SETUP パケットを格納するために 6 ワードが必要です。
 - 8 バイトの SETUP データと 4 バイトの SETUP ステータス(セットアップパケットパターン) を格納するには、SETUP パケットあたり 3 ワードが必要です。コアは、このスペースを受信データの中に確保します。
 - FIFO は SETUP データ書き込み専用であり、このスペースをデータパケットのために使うことはありません。
3. アプリケーションは、受信 FIFO から SETUP パケットの 2 ワードを読み出す必要があります。
4. アプリケーションは受信 FIFO からセットアップステージ終了ワードを読み出して、破棄する必要があります。

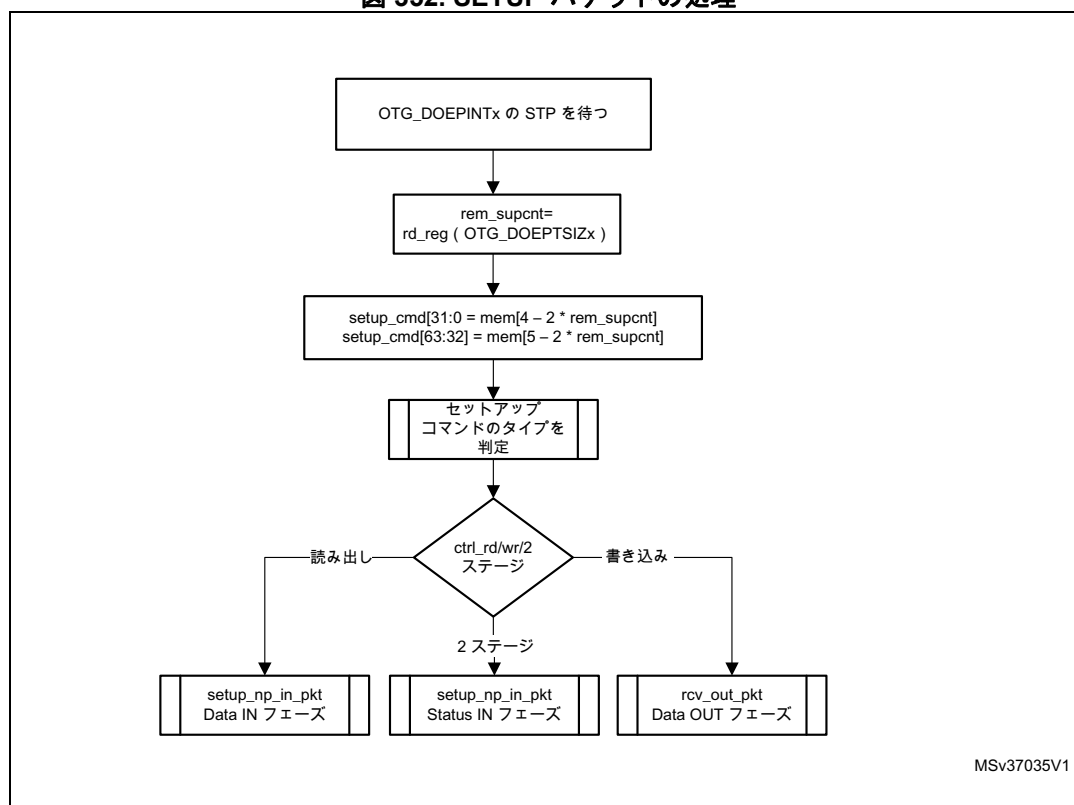
● 内部データフロー

1. SETUP パケットが受信されると、コアは受信 FIFO 内の空きスペースをチェックせずに、また、エンドポイントの NAK および STALL ビットの設定に関係なく、受信したデータを受信 FIFO に書き込みます。
 - コアは、SETUP パケットが受信されたコントロール IN/OUT エンドポイントの IN NAK および OUT NAK ビットを内部でセットします。
2. USB 上で受信された各 SETUP パケットについて、3 ワードのデータが受信 FIFO に書き込まれ、STUPCNT フィールドが 1 ずつデクリメントされます。
 - 最初のワードは、コアが内部的に使用する制御情報を含みます。
 - 2 番目のワードは、SETUP コマンドの最初の 4 バイトを含みます。
 - 3 番目のワードは、SETUP コマンドの最後の 4 バイトを含みます。
3. セットアップステージがデータ IN/OUT ステージに変わると、コアは、エントリ(セットアップステージ終了ワード)を受信 FIFO に書き込んで、セットアップステージの完了を示します。
4. AHB 側では、SETUP パケットはアプリケーションによってエンプティにされます。
5. アプリケーションが受信 FIFO からセットアップステージ終了ワードをポップすると、コアは、STUP 割り込み (OTG_DOEPINTx) でアプリケーションに割り込んで、受信した SETUP パケットを処理できることを示します。
6. コアは、コントロール OUT エンドポイントのエンドポイントイネーブルビットをクリアします。

● アプリケーションのプログラミングシーケンス :

1. OTG_DOEPSIZx レジスタをプログラムします。
 - STUPCNT = 3
2. RXFLVL 割り込み (OTG_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
3. STUP 割り込み (OTG_DOEPINTx) のアサートは、SETUP データ転送が正常に完了したことを示します。
 - この割り込み時、アプリケーションは、OTG_DOEPSIZx レジスタを読み出して、受信した SETUP パケットの数を確認し、最後に受信した SETUP パケットを処理する必要があります。

図 352. SETUP パケットの処理



● 3つを超える連続 SETUP パケットの処理

USB 2.0 仕様に従い、通常 SETUP パケットエラー時にはホストは同じエンドポイントに対して3つを超える連続 SETUP パケットを送信しません。ただし USB 2.0 仕様では、ホストが同じエンドポイントに送信できる連続 SETUP パケットの数を制限してはいません。この条件が発生すると、OTG_FS コントローラは、割り込み (OTG_DOEPINTx の B2BSTUP) を生成します。

● グローバル OUT NAK の設定

内部データフロー：

1. アプリケーションがグローバル OUT NAK (OTG_DCTL の SGONAK ビット) をセットすると、コアは SETUP パケットを除き、受信 FIFO へのデータの書き込みを停止します。受信 FIFO の使用可能なスペースの有無に関係なく、非アイソクロナス OUT トークンは NAK ハンドシェイクレスポンスを受信し、コアはアイソクロナス OUT データパケットを無視します。
2. コアは、グローバル OUT NAK パターンを受信 FIFO に書き込みます。アプリケーションがこのデータパターンを書き込むには、受信 FIFO に十分なスペースを確保する必要があります。
3. アプリケーションがグローバル OUT NAK パターンワードを受信 FIFO からポップすると、コアは、GONAKEFF 割り込み (OTG_GINTSTS) をセットします。
4. この割り込みを検出したアプリケーションは、コアがグローバル OUT NAK モードにあるとみなすことができます。アプリケーションは OTG_DCTL の SGONAK ビットをクリアすることによって、この割り込みをクリアできます。

アプリケーションのプログラミングシーケンス：

1. 受信 FIFO でのデータの受信も停止するには、アプリケーションは、以下のフィールドをプログラムすることによって、グローバル OUT NAK ビットをセットする必要があります。
 - OTG_DCTL レジスタの SGONAK = 1
2. OTG_GINTSTS の GONAKEFF 割り込みのアサートを待ちます。アサートされた場合、この割り込みは、コアが SETUP パケット以外のいかなるタイプのデータの受信も停止したことを示します。
3. アプリケーションは OTG_DCTL の SGONAK ビットをセットした後、コアが GONAKEFF 割り込み (OTG_GINTSTS) をアサートする前に、有効な OUT パケットを受信することができます。
4. アプリケーションは、OTG_GINTMSK レジスタの GONAKEFFM ビットに書き込むことによって、この割り込みを一時的にマスクできます。
 - OTG_GINTMSK レジスタの GONAKEFFM = 0
5. アプリケーションがグローバル OUT NAK モードを終了する準備ができたときには OTG_DCTL レジスタの SGONAK ビットをクリアする必要があります。これによって GONAKEFF 割り込み (OTG_GINTSTS) もクリアされます。
 - OTG_DCTL レジスタの CGONAK = 1
6. アプリケーションがこの割り込みをマスクするのが早すぎた場合は、以下のようにしてマスク解除を行う必要があります。
 - OTG_GINTMSK レジスタの GONAKEFFM = 1

● OUT エンドポイントの無効化

アプリケーションが有効にした OUT エンドポイントを無効にするには、このシーケンスを使用する必要があります。

アプリケーションのプログラミングシーケンス：

1. OUT エンドポイントを無効にする前に、アプリケーションは、コアのグローバル OUT NAK モードを有効にする必要があります。
 - OTG_DCTL レジスタの SGONAK = 1
2. OTG_GINTSTS の GONAKEFF 割り込みを待ちます。
3. 以下のフィールドをプログラムすることによって、必要な OUT エンドポイントを無効にします。
 - OTG_DOEPCTLx レジスタの EPDIS = 1
 - OTG_DOEPCTLx レジスタの SNAK = 1
4. EPDISD 割り込み (OTG_DOEPINTx) を待ちます。これは OUT エンドポイントが完全に無効にされたことを示します。EPDISD 割り込みがアサートされると、コアは、以下のビットもクリアします。
 - OTG_DOEPCTLx レジスタの EPDIS = 0
 - OTG_DOEPCTLx レジスタの EPENA = 0
5. アプリケーションは、無効にされていない他のエンドポイントからのデータ受信を開始するために、グローバル OUT NAK ビットをクリアする必要があります。
 - OTG_DCTL レジスタの SGONAK = 0

● 一般の非アイソクロナス OUT データ転送

このセクションでは、通常非アイソクロナス OUT データ転送（コントロール、バルク、またはインタラプト）について説明します。

アプリケーションの要件：

- OUT 転送をセットアップする前に、アプリケーションは OUT 転送の一部として受信されるすべてのデータを収容できるバッファをメモリ内で割り当てる必要があります。
- OUT 転送の場合、エンドポイントの転送サイズレジスタの転送サイズフィールドは、エンドポイントの最大パケットサイズの倍数でなければならず、ワードの境界に揃えられていなければなりません。
 - 転送サイズ [EPNUM] = $n * (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
 - パケットカウント [EPNUM] = n
 - $n > 0$
- OUT エンドポイント割り込み時には、アプリケーションは、エンドポイントの転送サイズレジスタを読み出して、メモリ内のペイロードのサイズを計算しなければなりません。受信したペイロードのサイズが、プログラムされた転送サイズより小さいこともあります。
 - メモリ内のペイロードのサイズ = アプリケーションがプログラムした初期転送サイズ - コアが更新した最終転送サイズ
 - このペイロードが受信された USB パケットの数 = アプリケーションがプログラムした初期パケット数 - コアが更新した最終パケット数

内部データフロー：

- アプリケーションがデータを受信するためには、エンドポイント固有レジスタの転送サイズおよびパケットカウントのフィールドを設定し、NAK ビットをクリアし、エンドポイントを有効にする必要があります。
- NAK ビットがクリアされると、コアは、データの受信を開始し、受信 FIFO にスペースがある限り、データを受信 FIFO に書き込みます。USB で受信された各データパケットについて、データパケットとそのステータスが受信 FIFO に書き込まれます。受信 FIFO にパケット（最大パケットサイズまたはショートパケット）が書き込まれるたびに、そのエンドポイントのパケットカウントフィールドが 1 ずつデクリメントされます。
 - 受信された OUT データパケットのデータ CRC が不良な場合、受信 FIFO から自動的に一掃されます。
 - USB 上のパケットに対して ACK を送信した後、コアは ACK を検出できないホストが再送信する非アイソクロナス OUT データパケットを破棄します。アプリケーションは、同じエンドポイント上では、同じデータ PID を持つ複数の連続データ OUT パケットを検出しません。この場合、パケットカウントはデクリメントされません。
 - 受信 FIFO にスペースがない場合、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれません。さらに、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
 - 上記の 3 つのケースのすべてにおいて、データは受信 FIFO に書き込まれないので、パケットカウントはデクリメントされません。
- パケットカウントが 0 になるか、エンドポイント上でショートパケットが受信されると、そのエンドポイントの NAK ビットがセットされます。NAK ビットがセットされると、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれず、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
- データが受信 FIFO に書き込まれた後、アプリケーションは受信 FIFO からデータを読み出して、エンドポイントあたり一度に 1 パケットずつ外部メモリに書き込みます。
- AHB 上で外部メモリへのパケットの書き込みが終わるたびに、書き込まれたパケットのサイズだけエンドポイントの転送サイズがデクリメントされます。

6. 以下の条件の 1 つで OUT エンドポイントの OUT データ転送完了パターンが受信 FIFO に書き込まれます。
 - 転送サイズが 0、およびパケットカウントが 0。
 - 受信 FIFO に書き込まれた最後の OUT データパケットがショートパケット。
($0 \leq \text{パケットサイズ} < \text{最大パケットサイズ}$)
7. アプリケーションがこのエントリ (OUT データ転送完了) をポップすると、エンドポイントの転送完了割り込みが生成され、エンドポイントイネーブルビットがクリアされます。

アプリケーションのプログラミングシーケンス：

1. OTG_DOEPTSLx レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
2. OTG_DOEPCTLx レジスタをエンドポイントの特性でプログラムし、EPENA ビットおよび CNAK ビットをセットします。
 - OTG_DOEPCTLx レジスタの EPENA = 1
 - OTG_DOEPCTLx レジスタの CNAK = 1
3. RXFLVL 割り込み (OTG_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
 - この手順は、転送サイズに応じて何度でも繰り返すことができます。
4. XFRC 割り込み (OTG_DOEPINTx) のアサートは、非アイソクロナス OUT データ転送が正常に完了したことを示します。
5. OTG_DOEPTSLx レジスタを読み出して、受信したデータペイロードのサイズを確認します。

● 一般のアイソクロナス OUT データ転送

このセクションでは、通常のアイソクロナス OUT データ転送について説明します。

アプリケーションの要件：

1. 非アイソクロナス OUT データ転送のアプリケーションの要件はすべて、アイソクロナス OUT データ転送にも適用されます。
2. アイソクロナス OUT データ転送の場合、転送サイズおよびパケットカウントフィールドは、常に、単一フレームで受信できる最大パケットサイズのパケット数を設定しなければなりません。アイソクロナス OUT データ転送は、複数のフレームにまたがってはいけません。
3. アプリケーションは、周期的フレームの終わり (OTG_GINTSTS の EOPF 割り込み) の前に、受信 FIFO からすべてのアイソクロナス OUT データパケット (データおよびステータス) を読み出す必要があります。
4. 次のフレームでデータを受信するには EOPF (OTG_GINTSTS) の後と SOF (OTG_GINTSTS) の前に、アイソクロナス OUT エンドポイントを有効にしなければなりません。

内部データフロー：

1. アイソクロナス OUT エンドポイントの内部データフローは、非アイソクロナス OUT エンドポイントの内部データフローと基本的に同じですが、少し異なっているところがあります。
2. エンドポイントイネーブルビットをセットし、NAK ビットをクリアすることによって、アイソクロナス OUT エンドポイントが有効にされたときには、偶数/奇数フレームビットも適切にセットされなければなりません。コアは、以下の条件が満たされた場合に限り、アイソクロナス OUT エンドポイント上で特定のフレームのデータを受信します。
 - EONUM (OTG_DOEPCTLx レジスタ) = FNSOF[0] (OTG_DSTS レジスタ)
3. アプリケーションが受信 FIFO からアイソクロナス OUT データパケット (データとステータス) を完全に読み出すと、コアは OTG_DOEPTSLx レジスタの RXDPID フィールドを、受信 FIFO から読み出された最後のアイソクロナス OUT データパケットのデータ PID で更新します。

アプリケーションのプログラミングシーケンス：

1. OTG_DOEPTSIZE レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
2. OTG_DOEPCTLx レジスタをエンドポイントの特性でプログラムし、エンドポイントイネーブル、ClearNAK、および偶数／奇数フレームの各ビットをセットします。
 - EPENA = 1
 - CNAK = 1
 - EONUM = (0 : 偶数 / 1 : 奇数)
3. RXFLVL 割り込み (OTG_GINTSTS) を待ち、受信 FIFO からのデータパケットをエンプティにします。
 - この手順は、転送サイズに応じて何度でも繰り返すことができます。
4. XFRC 割り込み (OTG_DOEPINTx) のアサートは、アイソクロナス OUT データ転送が正常に完了したことを示します。この割り込みは、必ずしもメモリ内のデータが良好であることを意味しません。
5. この割り込みは、アイソクロナス OUT 転送で必ず検出されるとは限りません。その代わり、アプリケーションは OTG_GINTSTS で INCOMPISOOUT 割り込みを検出することができます。
6. OTG_DOEPTSIZEx レジスタを読み出して、受信した転送データのサイズを確認し、フレームで受信したデータの有効性を確認します。アプリケーションは、以下の条件の 1 つが満たされた場合のみ、メモリに受信されたデータを有効として扱う必要があります。
 - RXDPID = DATA0 (OTG_DOEPTSIZEx レジスタ)、およびこのペイロードが受信された USB パケットの数 = 1
 - RXDPID = DATA1 (OTG_DOEPTSIZEx レジスタ)、およびこのペイロードが受信された USB パケットの数 = 2
 - このペイロードが受信された USB パケットの数 =
アプリケーションがプログラムした初期パケット数 – コアが更新した最終パケット数アプリケーションは、無効なデータパケットを破棄できます。

● 不完全アイソクロナス OUT データ転送

このセクションでは、アイソクロナス OUT データパケットがコアの内部でドロップされたときのアプリケーションのプログラミングシーケンスについて説明します。

内部データフロー：

1. アイソクロナス OUT エンドポイントの場合、XFRC 割り込み (OTG_DOEPINTx) は必ずアサートされるわけではありません。コアがアイソクロナス OUT データパケットをドロップした場合、アプリケーションは以下の状況で XFRC 割り込み (OTG_DOEPINTx) の検出に失敗することがあります。
 - 受信 FIFO が完全な ISO OUT データパケットを収容できない場合、コアは、受信した ISO OUT データをドロップします。
 - アイソクロナス OUT データパケットが CRC エラー付きで受信されたとき。
 - コアが受信したアイソクロナス OUT トークンが破損しているとき。
 - アプリケーションが受信 FIFO からデータを読み出すのに長時間かかっているとき。
2. コアが、すべてのアイソクロナス OUT エンドポイントへの転送を完了する前に周期的フレームの終わりを検出すると、不完全アイソクロナス OUT データ割り込み (OTG_GINTSTS レジスタの INCOMPISOOUT) をアサートして、アイソクロナス OUT エンドポイントのうち少なくとも 1 つで XFRC 割り込み (OTG_DOEPINTx) がアサートされていないことを示します。この時点で、不完全転送のエンドポイントは有効なままですが USB 上のこのエンドポイントでは、アクティブな転送は進行していません。

アプリケーションのプログラミングシーケンス：

1. INCOMPISOOUT 割り込み (OTG_GINTSTS) のアサートは、現在のフレームで、少なくとも1つのアイソクロナス OUT エンドポイントが転送を完了していないことを示します。
2. アイソクロナス OUT データがエンドポイントから完全に出されていないためにこの割り込みが発生した場合、アプリケーションは処理を進める前に、受信 FIFO からすべてのアイソクロナス OUT データ (データとステータス) を出力する必要があります。
 - すべてのデータが受信 FIFO から出されると、アプリケーションは XFRC 割り込み (OTG_DOEPINTx) を検出することができます。この場合、アプリケーションは、次のフレームでアイソクロナス OUT データを受信するには、エンドポイントを再び有効にする必要があります。
3. INCOMPISOOUT 割り込み (OTG_GINTSTS) を受信すると、アプリケーションは、すべてのアイソクロナス OUT エンドポイントの制御レジスタ (OTG_DOEPCTLx) を読み出して、現在のマイクロフレームにおいて転送を完了しなかったエンドポイントを確認する必要があります。次の両方の条件が満たされた場合、エンドポイント転送は完了しません。
 - EONUM ビット (OTG_DOEPCTLx レジスタ) = FNSOF[0] (OTG_DSTS レジスタ)
 - EPENA = 1 (OTG_DOEPCTLx レジスタ)
4. 現在のフレーム番号が変更されないように、上記の手順は SOF 割り込み (OTG_GINTSTS) が検出される前に実行されなければなりません。
5. 不完全転送のアイソクロナス OUT エンドポイントの場合、アプリケーションは、メモリ内のデータを破棄し OTG_DOEPCTLx レジスタの EPDIS ビットをセットすることによってエンドポイントを無効にする必要があります。
6. EPDISD 割り込み (OTG_DOEPINTx) を待ち、次のフレームで新しいデータを受信するために、エンドポイントを有効にします。
 - コアがエンドポイントを無効にするには若干の時間がかかるので、アプリケーションは不良なアイソクロナスデータを受信した後、次のフレームのデータを受信できないことがあります。

● 非アイソクロナス OUT エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

1. コアをグローバル OUT NAK モードにします。
2. 必要なエンドポイントを無効にします。
 - エンドポイントを無効にするときには、OTG_DOEPCTL の SNAK ビットをセットする代わりに STALL = 1 にセットしてください (OTG_DOEPCTL)。
STALL ビットは常に NAK ビットより優先されます。
3. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときは、STALL ビット (OTG_DOEPCTLx) をクリアしなければなりません。
4. アプリケーションが SetFeature.Endpoint Halt または ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションが制御エンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。

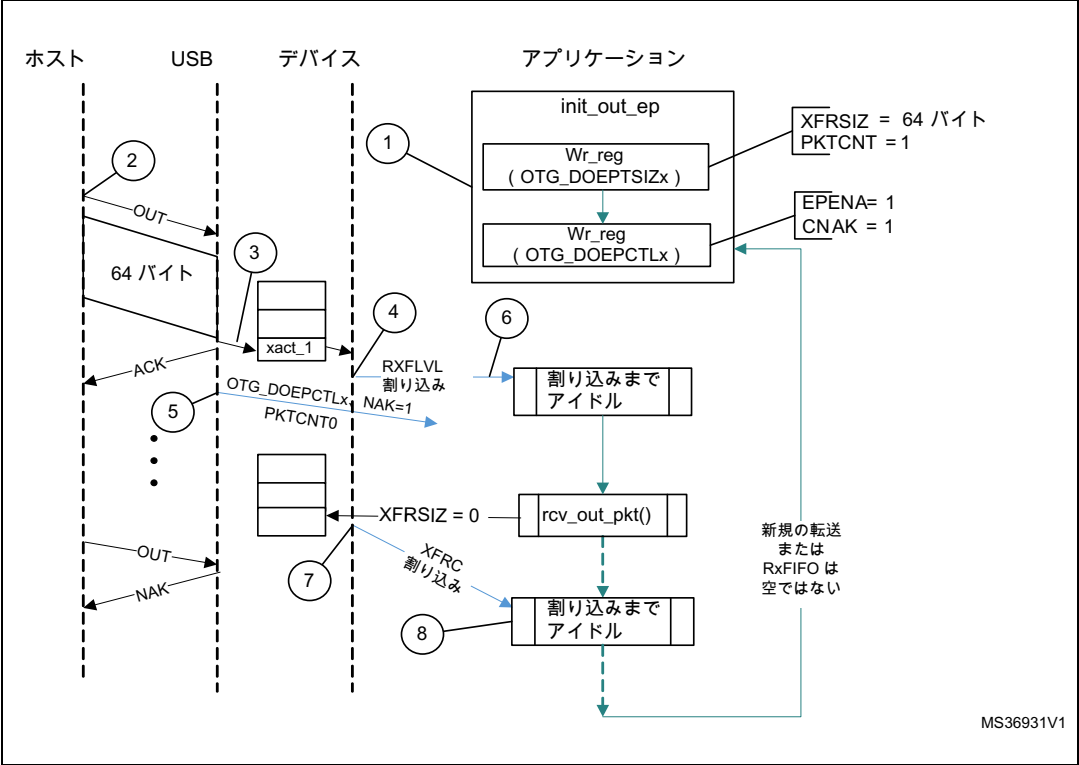
例

このセクションでは、いくつかの基本的な転送タイプとシナリオについて説明します。

● バルクOUTトランザクション

図 353 に USB から AHB への単一バルク OUT データパケットの受信と、このプロセスに伴うイベントを示します。

図 353. バルクOUTトランザクション



SetConfiguration/SetInterface コマンドの後、アプリケーションは CNAK = 1 および EPENA = 1 (OTG_DOEPCTLx レジスタ) をセットし、OTG_DOEPSIZx レジスタの適切な XFRSIZ および PKTCNT ビットをセットすることによって、すべての OUT エンドポイントを初期化します。

1. ホストは、データ (OUT トークン) のエンドポイントへの送信を試みます。
2. コアは、USB 上で OUT トークンを受信すると、スペースが使用可能なので、Rx FIFO にパケットを格納します。
3. Rx FIFO に完全なパケットを書き込んだ後、コアは RxFLVL 割り込み (OTG_GINTSTS) をアサートします。
4. USB パケットの PKTCNT 番号を受信すると、コアは、それ以上パケットが受信されないように、このエンドポイントの NAK ビットを内部でセットします。
5. アプリケーションは割り込みを処理して Rx FIFO からデータを読み出します。
6. アプリケーションがすべてのデータ (XFRSIZ に相当) を読み出すと、コアは、XFRSIZ 割り込み (OTG_DOEPINTx) を生成します。
7. アプリケーションは割り込みを処理して、XFRSIZ 割り込みビット (OTG_DOEPINTx) の設定を使用して、意図した転送が完了したかどうかを確認します。

IN データ転送

● パケットの書き込み

このセクションでは、専用の送信 FIFO が有効なときに、アプリケーションがデータパケットをエンドポイント FIFO に書き込む方法について説明します。

1. アプリケーションは、ポーリングまたは割り込みモードのいずれかを選択できます。
 - ポーリングモードでは、アプリケーションは、OTG_DTXFSTSx レジスタを読み出すことによってエンドポイント送信データ FIFO のステータスを監視して、データ FIFO に十分なスペースがあるかどうかを確認します。
 - 割り込みモードでは、アプリケーションは、TXFE 割り込み (OTG_DIEPINTx) を待ってから、OTG_DTXFSTSx レジスタを読み出して、データ FIFO 内に十分なスペースがあるかどうかを確認します。
 - 単一の非ゼロ長データパケットを書き込むには、パケット全体を書き込むためのスペースがデータ FIFO になければなりません。
 - ゼロ長パケットを書き込むには、アプリケーションは FIFO のスペースを考慮してはなりません。
2. 上記の方法の 1 つを使用して、アプリケーションが送信パケットを書き込むのに十分なスペースがあることを確認するときには、アプリケーションは、データをデータ FIFO に書き込む前に、まず、エンドポイント制御レジスタに書き込む必要があります。通常、アプリケーションは、エンドポイントイネーブルビットをセットする場合を除き、レジスタの内容を変更しないように、OTG_DIEPCTLx レジスタにリードモディファイライトを行う必要があります。

アプリケーションは、使用可能なスペースがあれば、同じエンドポイントに対する複数のパケットを送信 FIFO に書き込むことができます。周期的 IN エンドポイントの場合、アプリケーションは 1 つのマイクロフレームのパケットのみを書き込む必要があります。アプリケーションは、前のトランザクションの転送完了割り込みを受信した後でのみ、次の周期的トランザクションのパケットを書き込むことができます。

● IN エンドポイント NAK の設定

内部データフロー：

1. アプリケーションが特定のエンドポイントの IN NAK をセットすると、コアは、エンドポイントの送信 FIFO にデータがあるかどうかに関係なく、そのエンドポイントでのデータ送信を停止します。
2. 非アイソクロナス IN トークンは NAK ハンドシェイク応答を受信します。
 - アイソクロナス IN トークンはゼロデータ長のパケット応答を受信します。
3. コアは、OTG_DIEPCTLx の SNAK ビットに応答して、OTG_DIEPINTx の INEPNE 割り込み (IN エンドポイント NAK 有効) をアサートします。
4. この割り込みがアプリケーションによって検出されると、アプリケーションは、エンドポイントが IN NAK モードにあるとみなすことができます。この割り込みは、アプリケーションが OTG_DIEPCTLx の CNAK ビットをセットすることによってクリアできます。

アプリケーションのプログラミングシーケンス：

1. 特定の IN エンドポイントでのデータ送信を停止するには、アプリケーションは、IN NAK ビットをセットする必要があります。このビットをセットするには、以下のフィールドをプログラムする必要があります。
 - OTG_DIEPCTLx レジスタの SNAK = 1
2. OTG_DIEPINTx の INEPNE 割り込みのアサートを待ちます。この割り込みは、コアがエンドポイントでのデータ送信を停止したことを示します。
3. コアは、アプリケーションが NAK ビットをセットした後の NAK 有効割り込みがアサートされる前に、エンドポイントで有効な IN データを送信できます。
4. アプリケーションは OTG_DIEPMSK の INEPNEM ビットに書き込むことによって、この割り込みを一時的にマスクできます。
 - OTG_DIEPMSK レジスタの INEPNEM = 0
5. エンドポイント NAK モードを終了するには、アプリケーションは OTG_DIEPCTLx の NAK ステータスビット (NAKSTS) をクリアする必要があります。これによって INEPNE 割り込み (OTG_DIEPINTx) もクリアされます。
 - OTG_DIEPCTLx レジスタの CNAK = 1
6. アプリケーションがこの割り込みをマスクするのが早すぎた場合、以下のようにしてマスク解除を行う必要があります。
 - OTG_DIEPMSK レジスタの INEPNEM = 1

● IN エンドポイントディセーブル

以前に有効にされた特定の IN エンドポイントを無効にするには、以下のシーケンスを使用します。

アプリケーションのプログラミングシーケンス：

1. アプリケーションは、AHB でのデータ書き込みを停止して、IN エンドポイントを無効にする必要があります。
2. アプリケーションは、エンドポイントを NAK モードに設定する必要があります。
 - OTG_DIEPCTLx レジスタの SNAK = 1
3. OTG_DIEPINTx の INEPNE 割り込みを待ちます。
4. 無効にしなければならないエンドポイントについて OTG_DIEPCTLx レジスタの以下のビットをセットします。
 - OTG_DIEPCTLx レジスタの EPDIS = 1
 - OTG_DIEPCTLx レジスタの SNAK = 1
5. OTG_DIEPINTx レジスタの EPDISD 割り込みのアサートは、コアが指定されたエンドポイントを完全に無効にしたことを示します。割り込みのアサートとともに、コアは、以下のビットもクリアします。
 - OTG_DIEPCTLx レジスタの EPENA = 0
 - OTG_DIEPCTLx レジスタの EPDIS = 0
6. アプリケーションは、周期的 IN EP の OTG_DIEPTSIZx レジスタを読み出して、エンドポイント上のどれだけのデータが USB で送信されたかを計算する必要があります。
7. アプリケーションは、OTG_GRSTCTL レジスタの以下のフィールドを設定することによって、エンドポイント送信 FIFO 内のデータを一掃する必要があります。
 - TXFNUM (OTG_GRSTCTL) = エンドポイント送信 FIFO 番号
 - TXFFLSH (OTG_GRSTCTL) = 1

アプリケーションは、TXFFLSH ビットがコアによってクリアされるまで（すなわち、一掃動作の終了まで）、OTG_GRSTCTL レジスタにポーリングする必要があります。このエンドポイントで新しいデータを送信するために、アプリケーションは後で、エンドポイントを再び有効にできます。

● 一般の非周期的 IN データ転送

アプリケーションの要件：

1. IN 転送をセットアップする前に、アプリケーションは、IN 転送の一部として送信されるすべてのデータが単一バッファの一部であることを確認する必要があります。
2. IN 転送の場合、エンドポイント転送サイズレジスタの転送サイズフィールドは、最大パケットサイズの複数のパケットと単一のショートパケットから成るペイロードを表します。このショートパケットは、転送の最後に送信されます。
 - － 転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには：
転送サイズ [EPNUM] = $x * MPSIZ[EPNUM] + sp$
($sp > 0$) の場合、パケットカウント [EPNUM] = $x + 1$ 。
そうでない場合、パケットカウント [EPNUM] = x
 - － 単一のゼロ長データパケットを送信するには：
転送サイズ [EPNUM] = 0
パケットカウント [EPNUM] = 1
 - － 転送の終わりに少数の最大パケットサイズのパケットとゼロ長データパケットを送信するには、アプリケーションは転送を 2 つの部分に分ける必要があります。最初の部分では最大パケットサイズのデータパケットを送信し、2 番目の部分ではゼロ長データパケットのみを送信します。
最初の転送：転送サイズ [EPNUM] = $x * MPSIZ[epnum]$; パケットカウント = n ;
2 番目の転送：転送サイズ [EPNUM] = 0 ; パケットカウント = 1 ;
3. エンドポイントがデータ転送のために有効にされると、コアは、転送サイズレジスタを更新します。IN 転送の終了時に、アプリケーションは、転送サイズレジスタを読み出して、送信 FIFO にポストされたデータのうち、どれだけが USB で送信されたかを確認する必要があります。
4. 送信 FIFO にフェッチされたデータ = アプリケーションがプログラムした初期転送サイズ - コアが更新した最終転送サイズ
 - － USB で送信されたデータ = (アプリケーションがプログラムした初期パケットカウント - コアが更新した最終パケットカウント) * MPSIZ[EPNUM]
 - － USB 上でこれから送信されるデータ = (アプリケーションがプログラムした初期転送サイズ - USB で送信されたデータ)

内部データフロー：

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケットカウントフィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. アプリケーションは、要求されたデータをエンドポイントの送信 FIFO に書き込む必要もあります。
3. アプリケーションによってパケットが送信 FIFO に書き込まれるたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、データはアプリケーションによってメモリからフェッチされます。FIFO にデータを書き込んだ後、「FIFO 内のパケット数」はインクリメントされます（これは 3 ビットのカウンタであり、各 IN エンドポイントの送信 FIFO について、コアによって内部で維持されます。IN エンドポイント FIFO 内にコアによって一度に維持されるパケットの最大数は 8 です）。ゼロ長のパケットの場合、各 FIFO について、FIFO 内にデータがないことを示す個別のフラグがセットされます。



4. データが送信 FIFO に書き込まれると、コアは、IN トークンの受信時に読み出します。非アイソクロナス IN データパケットが ACK ハンドシェイクとともに送信されるたびに、エンドポイントのパケットカウンタは、0 になるまで 1 ずつデクリメントされます。パケットカウンタは、タイムアウト時にはデクリメントされません。
5. ゼロ長パケットの場合（内部ゼロ長フラグで示されます）、コアは IN トークンのゼロ長パケットを送出し、パケットカウンタフィールドをデクリメントします。
6. 受信された IN トークンのデータが FIFO になく、そのエンドポイントのパケットカウンタフィールドがゼロの場合、コアは、そのエンドポイントについて、「Tx FIFO がエンプティのとき IN トークンが受信された」という割り込み (ITTXFE 割り込み) を生成します。ただし、エンドポイントの NAK ビットがセットされていない場合に限りです。コアは、USB 上の非アイソクロナスエンドポイントに対して NAK ハンドシェイクで応答します。
7. コアは、内部で FIFO ポインタを巻き戻し、タイムアウト割り込みは生成されません。
8. 転送サイズが 0 であり、パケットカウンタが 0 のとき、エンドポイントの転送完了割り込み (XFRC) が生成され、エンドポイントイネーブルビットがクリアされます。

アプリケーションのプログラミングシーケンス：

1. OTG_DIEPTSIZx レジスタで転送サイズと対応するパケットカウンタをプログラムします。
2. OTG_DIEPCTLx レジスタをエンドポイントの特性でプログラムし、CNAK ビットおよび EPENA（エンドポイントイネーブル）ビットをセットします。
3. 非ゼロ長データパケットを送信するときには、アプリケーションは、OTG_DTXFSTSx レジスタ (x はそのエンドポイントに関連する FIFO 番号) をポーリングして、データ FIFO に十分なスペースがあるかどうかを確認する必要があります。アプリケーションは、データを書き込む前に、オプションで TXFE (OTG_DIEPINTx) を使用することができます。

● 一般の周期的 IN データ転送

このセクションでは、典型的な周期的 IN データ転送について説明します。

アプリケーションの要件：

1. **一般の非周期的 IN データ転送 (1085 ページ)** のアプリケーションの要件 1、2、3、および 4 は、要件 2 が少し変更されていることを除いて、周期的 IN データ転送にも適用されます。
 - アプリケーションは複数の最大パケットサイズのデータパケットの送信、または複数の最大パケットサイズのパケットに最後にショートパケットを加えた送信のみを行うことができます。転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには、以下の条件が満たされなければなりません。

$$\text{転送サイズ [EPNUM]} = x * \text{MPSIZ[EPNUM]} + \text{sp}$$
 (x は整数 ≥ 0 、および $0 \leq \text{sp} < \text{MPSIZ[EPNUM]}$)

(sp > 0) の場合、パケットカウンタ [EPNUM] = x + 1

そうでない場合、パケットカウンタ [EPNUM] = x;

 $\text{MCNT[EPNUM]} = \text{パケットカウンタ [EPNUM]}$
 - アプリケーションは転送の最後にゼロ長データパケットを送信することはできません。自分で単一のゼロ長パケットを送信することは可能です。単一のゼロ長データパケットを送信するには：
 - 転送サイズ [EPNUM] = 0

パケットカウンタ [EPNUM] = 1

 $\text{MCNT[EPNUM]} = \text{パケットカウンタ [EPNUM]}$

2. アプリケーションは一度に 1 フレームのデータ転送のみをスケジュールすることができます。
 - $(MCNT - 1) \times MPSIZ \leq XFERSIZ \leq MCNT \times MPSIZ$
 - $PKTCNT = MCNT$ (OTG_DIEPTSIZE レジスタ)
 - $XFERSIZ < MCNT \times MPSIZ$ の場合、転送の最後のデータパケットはショートパケットです。
 - 以下のことに注意してください。MCNT は OTG_DIEPTSIZE レジスタに、MPSIZ は OTG_DIEPTCTL レジスタに、PKTCNT は OTG_DIEPTSIZE レジスタに、また XFERSIZ は OTG_DIEPTSIZE レジスタにあります。
3. フレームで送信される完全なデータは、IN トークンが受信される前に、アプリケーションによって送信 FIFO に書き込まれなければなりません。IN トークンが受信されたときに、フレームあたりで送信されるデータの 1 ワードが送信 FIFO 内で欠落していても、コアは FIFO がエンプティである場合のように動作します。送信 FIFO がエンプティのとき：
 - アイソクロナス IN エンドポイントに対して USB 上でゼロデータ長のパケットが送信されます。
 - IN エンドポイントに割り込みをかけるため、USB 上で NAK ハンドシェイクが送信されます。

内部データフロー：

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケットカウントフィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. また、アプリケーションは、要求されたデータをエンドポイントの関連する送信 FIFO に書き込む必要があります。
3. アプリケーションがパケットを送信 FIFO に書き込むたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、アプリケーションメモリからデータがフェッチされます。
4. 周期的エンドポイントの IN トークンが受信されると、コアは、FIFO のデータ（ある場合）を送信します。フレームの完全なデータペイロード（専用 FIFO モードでは完全なパケット）が FIFO に存在しない場合、コアは、エンドポイントに対する Tx FIFO エンプティ割り込みがおきたときに IN トークンの受信を発生します。
 - アイソクロナス IN エンドポイントに対して、USB 上でゼロ長のデータパケットが送信されます。
 - IN エンドポイントに割り込みをかけるため、USB 上で NAK ハンドシェイクが送信されます。
5. エンドポイントのパケットカウントは、以下の条件のもとで 1 ずつデクリメントされます。
 - アイソクロナスエンドポイントに対して、ゼロ長または非ゼロ長データパケットが送信されたとき。
 - インタラプトエンドポイントに対して、ACK ハンドシェイクが送信されたとき。
 - 転送サイズとパケットカウントの両方が 0 のとき、エンドポイントに対して転送完了割り込みが生成され、エンドポイントイネーブルビットがクリアされます。
6. 「周期的フレームインターバル」(OTG_DCFG の PFIVL によって制御されます) で、コアが現在のフレームについてスケジュールされたアイソクロナス IN エンドポイント FIFO のいずれかが非エンプティであることを検出すると、コアは OTG_GINTSTS の IISOIXFR 割り込みを生成します。

アプリケーションのプログラミングシーケンス：

1. OTG_DIEPCTLx レジスタをエンドポイントの特性でプログラムして、CNAK および EPENA ビットをセットします。
2. 次のフレームで送信するデータを送信 FIFO に書き込みます。
3. OTG_DIEPINTx レジスタの ITTXFE 割り込みのアサートは、アプリケーションが送信するすべてのデータを送信 FIFO にまだ書き込んでいないことを示します。
4. この割り込みが検出されたときに、インタラプトエンドポイントがすでに有効だった場合、割り込みは無視されます。まだ有効になっていなかった場合は、次の IN トークン試行の際にデータを送信できるように、エンドポイントを有効にします。
5. OTG_DIEPINTx レジスタの ITTXFE 割り込みがない状態での XFRC 割り込み (OTG_DIEPINTx) のアサートは、アイソクロナス IN 転送が正常に完了したことを示します。OTG_DIEPTSIZx レジスタの読み出し結果は、転送サイズ = 0、かつパケットカウント = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。
6. ITTXFE 割り込み (OTG_DIEPINTx) の有無にかかわらず、XFRC 割り込み (OTG_DIEPINTx) のアサートは、インタラプト IN 転送が正常に完了したことを示します。OTG_DIEPTSIZx レジスタの読み出し結果は、転送サイズ = 0、かつパケットカウント = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。
7. 前述のどの割り込みもない状態での OTG_GINTSTS の不完全アイソクロナス IN 転送 (IISOIXFR) 割り込みのアサートは、コアが現在のフレームで少なくとも 1 つの周期的 IN トークンも受信しなかったことを示します。

● 不完全アイソクロナス IN データ転送

このセクションでは、アプリケーションが不完全なアイソクロナス IN データ転送で行う必要があることについて説明します。

内部データフロー：

1. アイソクロナス IN 転送は、以下の条件のいずれかがあてはまる場合に不完全として扱われます。
 - a) コアが、少なくとも 1 つのアイソクロナス IN エンドポイントで破損したアイソクロナス IN トークンを受信した場合。この場合、アプリケーションは、不完全アイソクロナス IN 転送割り込み (OTG_GINTSTS の IISOIXFR ビット) を検出します。
 - b) アプリケーションが完全なデータペイロードを送信 FIFO に書き込むのに時間がかかり、完全なデータペイロードが FIFO に書き込まれる前に IN トークンが受信された場合。この場合、アプリケーションは、OTG_DIEPINTx の Tx FIFO エンプティ割り込み時に IN トークンの受信を検出します。最終的には、周期的フレームの終わりでの不完全アイソクロナス IN 転送割り込み (OTG_GINTSTS の IISOIXFR) になるので、アプリケーションはこの割り込みを無視することができます。
コアは、受信した IN トークンにตอบสนองして、USB 上にゼロ長データパケットを送信します。
2. アプリケーションは、できるだけ速やかに、送信 FIFO へのデータペイロードの書き込みを停止する必要があります。
3. アプリケーションは、エンドポイントの NAK ビットとディセーブルビットをセットする必要があります。
4. コアは、エンドポイントを無効にし、ディセーブルビットをクリアし、エンドポイントのエンドポイントディセーブル割り込みをアサートします。

アプリケーションのプログラミングシーケンス：

1. 最終的には不完全アイソクロナス IN 転送割り込み (OTG_GINTSTS) になるので、アプリケーションは、アイソクロナス IN エンドポイントで OTG_DIEPINTx の Tx FIFO エンプティ割り込み時に受信された IN トークンを無視することができます。
2. 不完全アイソクロナス IN 転送割り込み (OTG_GINTSTS) のアサートは、少なくとも 1 つのアイソクロナス IN エンドポイントで不完全アイソクロナス IN 転送があったことを示します。
3. アプリケーションは、すべてのアイソクロナス IN エンドポイントのエンドポイント制御レジスタを読み出して、不完全 IN データ転送があるエンドポイントを検出する必要があります。
4. アプリケーションは、AHB 上でこれらのエンドポイントに関連する周期的送信 FIFO へのデータの書き込みを停止する必要があります。
5. OTG_DIEPCTLx レジスタの以下のフィールドをプログラムして、エンドポイントを無効にします。
 - OTG_DIEPCTLx レジスタの SNAK = 1
 - OTG_DIEPCTLx レジスタの EPDIS = 1
6. OTG_DIEPINTx のエンドポイントディセーブル割り込みのアサートは、コアがエンドポイントを無効にしたことを示します。
 - この時点で、アプリケーションは、次のマイクロフレームで新しい転送を行うために、関連する送信 FIFO 内のデータを一掃するか、エンドポイントを有効にすることによって、FIFO 内の既存のデータに上書きする必要があります。データを一掃するには、アプリケーションは OTG_GRSTCTL レジスタを使用する必要があります。

● 非アイソクロナス IN エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

アプリケーションのプログラミングシーケンス：

1. 停止する IN エンドポイントを無効にします。STALL ビットもセットします。
2. エンドポイントがすでに有効になっているときには、OTG_DIEPCTLx の EPDIS = 1 にします。
 - OTG_DIEPCTLx レジスタの STALL = 1 にします。
 - STALL ビットは常に NAK ビットより優先されます。
3. エンドポイントディセーブル割り込み (OTG_DIEPINTx) のアサートは、コアが指定されたエンドポイントを無効にしたことをアプリケーションに示します。
4. アプリケーションは、エンドポイントのタイプに応じて、非周期的または周期的 FIFO をフラッシュする必要があります。非周期的エンドポイントの場合、アプリケーションは、データを送信するために、停止する必要がない他の非周期的エンドポイントを再び有効にする必要があります。
5. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときには、OTG_DIEPCTLx の STALL ビットがクリアされなければなりません。
6. アプリケーションが SetFeature.Endpoint Halt コマンドまたは ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションが制御エンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。

特殊なケース：コントロール OUT エンドポイントの停止

コントロール転送のデータステージで、ホストが SETUP パケットで指定されたより多くの IN/OUT トークンを送信した場合、コアは、IN/OUT トークンを停止する必要があります。この場合、アプリ

ケーションは、コアが SETUP パケットで指定されたデータ量を転送した後、コントロール転送のデータステージで、OTG_DIEPINTx の ITTXFE 割り込みと

OTG_DOEPINTx の OTEPDIS 割り込みを有効にしなければなりません。その後、アプリケーションがこの割り込みを受信したとき、アプリケーションは、対応するエンドポイント制御レジスタの STALL ビットをセットし、この割り込みをクリアする必要があります。

29.16.6 最悪ケースの応答時間

OTG_FS コントローラがデバイスとして機能するとき、アイソクロナス OUT に続く任意のトークンで、最悪ケースの応答時間があります。この最悪ケース応答時間は、AHB クロック周波数に依存します。

コアのレジスタは AHB ドメインにあり、コアはこれらのレジスタを更新するまでは、別のトークンを受け入れません。アイソクロナストランザクションの場合、ハンドシェイクはなく、次のトークンがすぐに入ってくる可能性があるため、アイソクロナス OUT に続くトークンで最悪ケースが生じます。この最悪ケース値は、AHB クロックが PHY クロックと同じときには 7 PHY クロックです。AHB クロックが高速なほど、この値は小さくなります。

この最悪ケース条件が発生した場合、コアは、バルク/インタラプトトークンに NAK で応答し、アイソクロナスおよび SETUP トークンをドロップします。ホストは、これを SETUP のタイムアウト条件と解釈して、SETUP パケットを再試行します。アイソクロナス転送の場合、不完全アイソクロナス IN 転送割り込み (IISOIXFR) と不完全アイソクロナス OUT 転送割り込み (IISOXFR) は、アイソクロナス IN/OUT パケットがドロップされたことをアプリケーションに知らせます。

OTG_GUSBCFG の TRDT の値の選択

TRDT (OTG_GUSBCFG) の値は、MAC が IN トークンを受信した後、FIFO ステータスと PFC ブロックから最初のデータを取得するまでの時間 (PHY クロック数) です。この時間には、PHY クロックと AHB クロック間の同期遅延も含まれます。この場合の最悪ケース遅延は、AHB クロックが PHY クロックと同じときに発生します。この場合、遅延は 5 クロックです。

MAC が IN トークンを受信すると、この情報 (トークンの受信) は PFC (PFC は AHB クロックで動作します) によって AHB に同期されます。次に、PFC は、SPRAM からデータを読み出し、それらをデュアルクロックソースバッファに書き込みます。MAC は、データをソースバッファ (深さ 4) から読み出します。

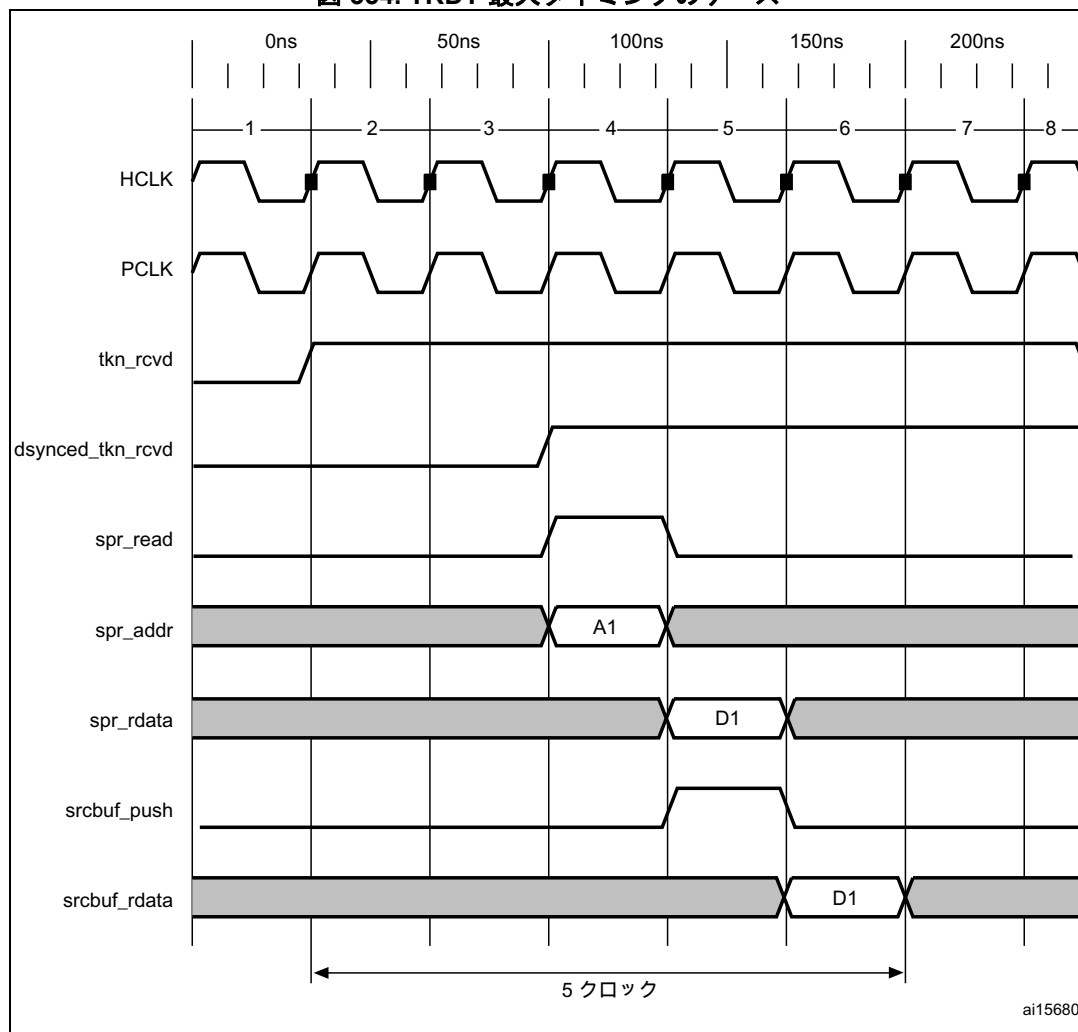
AHB が PHY より高い周波数で動作している場合、アプリケーションは TRDT (OTG_GUSBCFG) の値として、より小さな値を使用することができます。

 354 に以下の信号を示します。

- tkn_rcvd : MAC から PFC への、トークンが受信されたという情報
- dynced_tkn_rcvd : PCLK から HCLK ドメインへの二重同期 tkn_rcvd
- spr_read : SPRAM への読み出し
- spr_addr : SPRAM へのアドレス
- spr_rdata : SPRAM からのデータの読み出し
- srcbuf_push : ソースバッファへのプッシュ
- srcbuf_rdata : ソースバッファからのデータの読み出し MAC によるデータの検出

TRDT の値を計算するには、表 193 : TRDT 値 (FS) を参照してください。

図 354. TRDT 最大タイミングのケース



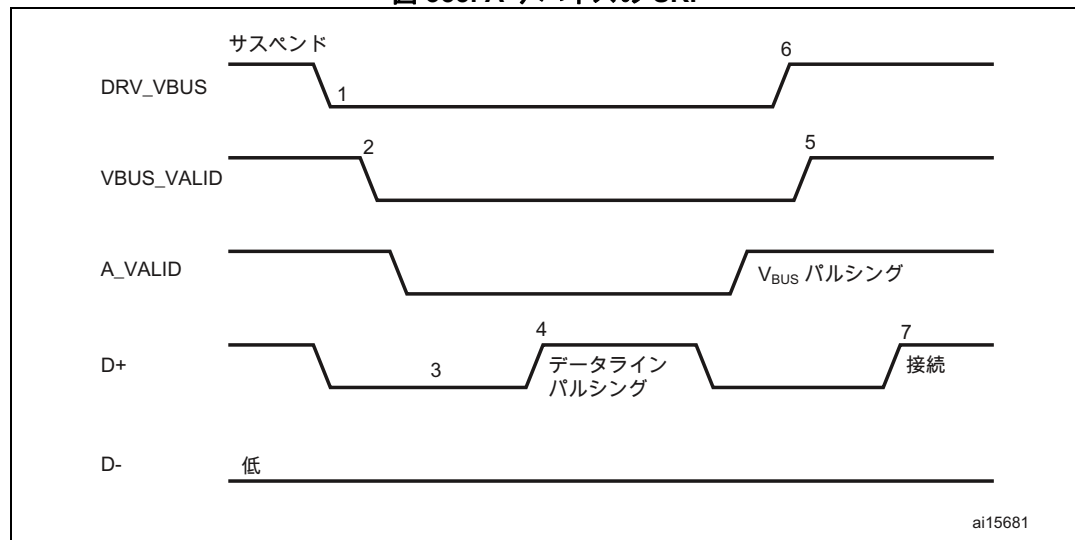
29.16.7 OTG プログラミングモデル

OTG_FS コントローラは、HNP と SRP をサポートする OTG デバイスです。コアが A プラグに接続されると、A デバイスと呼ばれます。コアが B プラグに接続されると、B デバイスと呼ばれます。ホストモードでは、OTG_FS コントローラは、電力を節約するために V_{BUS} をオフにします。SRP は、B デバイスが A デバイスに V_{BUS} 電源をオンにするように信号を送る手段です。デバイスは、データラインパルシングと V_{BUS} パルシングの両方を実行する必要がありますが、ホストは SRP のデータラインパルシングか V_{BUS} パルシングのどちらかを検出することができます。HNP は、B デバイスがネゴシエイトし、役割をホストに切り替える手段です。HNP 後のネゴシエイトモードでは、B デバイスはバスをサスペンドし、その役割をデバイスに戻します。

A デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG_FS コントローラは SRP を A デバイスとして検出できるようになります。

図 355. A デバイスの SRP



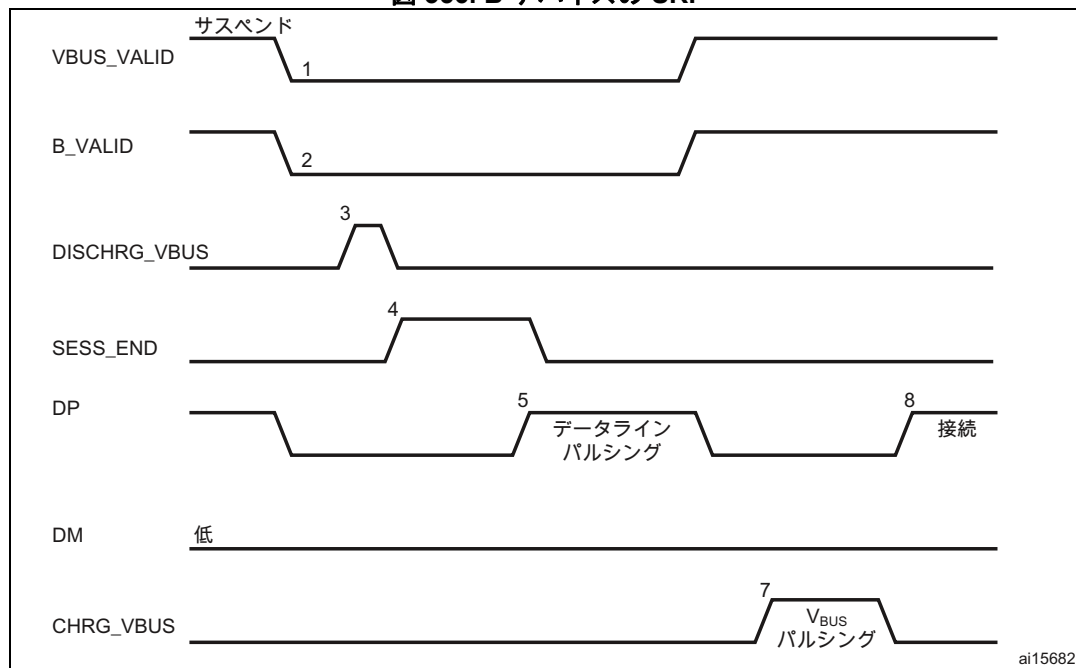
1. DRV_VBUS = PHY への V_{BUS} 駆動信号
VBUS_VALID = PHY からの V_{BUS} 有効信号
A_VALID = PHY への A ペリフェラル V_{BUS} レベル信号
D+ = データ正ライン
D- = データ負ライン

1. 節電のためには、アプリケーションは、バスがアイドル状態のときに、ホストポート制御およびステータスレジスタのポートサスペンドビットおよびポートパワービットに書き込むことによって、ポート電源をサスペンドおよびオフにします。
2. PHY は、VBUS_VALID 信号をネゲートすることによって、ポート電源のオフを示します。
3. デバイスは、 V_{BUS} 電源がオフになると、SRP を開始するために少なくとも 2 ms 以上 SE0 を検出しなければなりません。
4. SRP を開始するには、デバイスはデータラインプルアップ抵抗を 5~10 ms の間オンにします。OTG_FS コントローラはデータラインのパルシングを検出します。
5. デバイスは、 V_{BUS} を A デバイスセッション有効レベル以上（最低 2.0 V）に駆動します（ V_{BUS} パルシングに対して）。
OTG_FS コントローラは、SRP を検出すると、アプリケーションに割り込みをかけます。グローバル割り込みステータスレジスタのセッションリクエスト検出ビットがセットされます (OTG_GINTSTS の SRQINT)。
6. アプリケーションは、セッションリクエスト検出割り込みを処理して、ホストポート制御およびステータスレジスタのポートパワービットに書き込むことによって、ポート電源をオンにする必要があります。PHY は、VBUS_VALID 信号をアサートすることによって、ポート電源がオンになったことを示します。
7. USB に電源が投入されると、デバイスが接続され、SRP プロセスが完了します。

B デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG_FS コントローラは SRP を B デバイスとして扱えるようになります。SRP は、OTG_FS コントローラがホストからの新しいセッションをリクエストする手段です。

図 356. B デバイスの SRP



1. VBUS_VALID = PHY からの V_{BUS} 有効信号
B_VALID = PHY への B ペリフェラル有効セッション信号
DISCHRG_VBUS = PHY への放電信号
SESS_END = PHY へのセッション終了信号
CHRG_VBUS = PHY への V_{BUS} 充電信号
DP = データ正ライン
DM = データ負ライン

1. 節電のために、ホストはバスがアイドル状態のときはポート電源をサスペンドおよびオフにします。

OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。

OTG_FS コントローラは、PHY に V_{BUS} を放電するように通知します。

2. PHY はデバイスにセッションの終了を知らせます。これが SRP の初期条件です。OTG_FS コントローラは、SRP を初期化する前に、2 ms の SE0 を必要とします。

USB 1.1 フルスピードシリアルトランシーバの場合、アプリケーションは、BSVLD (OTG_GOTGCTL) がネゲートされた後、V_{BUS} が 0.2 V まで放電するまで待つ必要があります。この放電時間の値は、トランシーバのベンダから入手でき、ベンダごとに異なります。

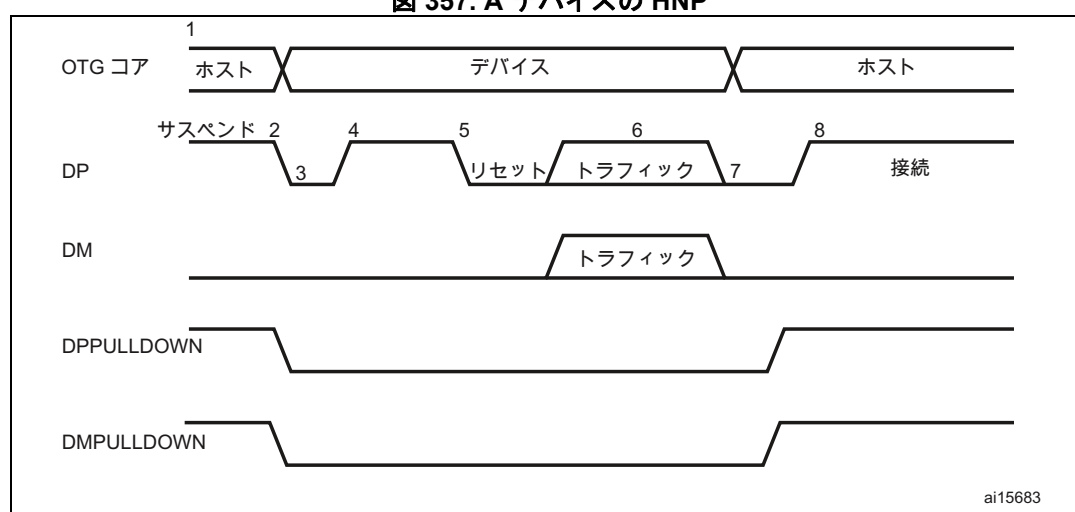
3. OTG_FS コアは PHY に V_{BUS} の放電を加速させることを通知します。
4. アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエストビットに書き込むことによって、SRP を開始します。OTG_FS コントローラは、データラインパルシングに続いて V_{BUS} パルシングを実行します。
5. ホストは、データラインパルシングまたは V_{BUS} パルシングから SRP を検出して、V_{BUS} をオンにします。PHY は、デバイスに V_{BUS} 電源オンを知らせます。

6. OTG_FS コントローラは、V_{BUS} パルシングを実行します。
ホストは、V_{BUS} をオンにして、SRP が成功したことを示すことによって、新しいセッションを開始します。OTG_FS コントローラは、OTG 割り込みステータスレジスタのセッションリクエスト成功ステータス変化ビットをセットすることによって、アプリケーションに割り込みをかけます。アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエスト成功ビットを読み出します。
7. USB に電源が投入されると、OTG_FS コントローラが接続され、SRP プロセスが完了します。

A デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を A デバイスから B デバイスに切り替えます。アプリケーションは、OTG_FS コントローラが HNP を A デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 357. A デバイスの HNP



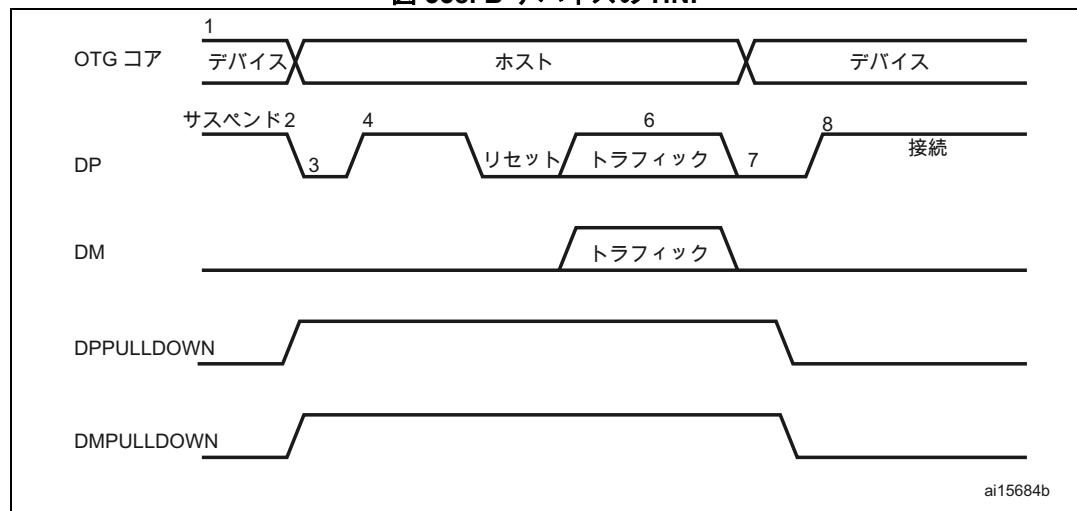
1. DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
1. OTG_FS コントローラは、SetFeature b_hnp_enable 記述子を B デバイスに送信して、HNP サポートを有効にします。B デバイスの ACK 応答は、B デバイスが HNP をサポートしていることを示します。アプリケーションは、OTG 制御およびステータスレジスタのセット HNP イネーブルビットをセットして、B デバイスが HNP をサポートしていることを OTG_FS コントローラに示す必要があります。
2. アプリケーションがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによって、サスペンドします。
3. B デバイスが USB サスペンドを検出すると、切断して、HNP の初期条件を示します。B デバイスは、ホストの役割に切り替える必要があるときにのみ、HNP を開始します。そうでないときには、バスはサスペンド状態を継続します。
OTG_FS コントローラは、OTG 割り込みステータスレジスタのホストネゴシエーション検出割り込みをセットして、HNP の開始を示します。
OTG_FS コントローラは、PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割を示します。PHY は、OTG_DP プルアップ抵抗を有効にして、B デバイスの接続を示します。
アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、デバイスモードの動作を確認する必要があります。

4. B デバイスは接続を検出し、USB リセットを発行し、データトラフィックのために OTG_FS コントローラをエニュメレーションします。
5. B デバイスはホストの役割を継続し、トラフィックを開始し、終了時にはバスをサスペンドします。
OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。
6. ネゴシエートモードでは、OTG_FS コントローラは、サスペンドを検出して切断し、ホストの役割に戻ります。OTG_FS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。
7. OTG_FS コントローラは、OTG 割り込みステータスレジスタのコネクタ ID ステータスレジスタ変更割り込みをセットします。アプリケーションは、OTG 制御およびステータスレジスタのコネクタ ID ステータスを読み出して、OTG_FS コントローラの動作が A デバイスであることを確認する必要があります。これは、HNP の完了をアプリケーションに示します。アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
8. B デバイスが接続され、HNP プロセスが完了します。

B デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を B デバイスから A デバイスに切り替えます。アプリケーションは、OTG_FS コントローラが HNP を B デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 358. B デバイスの HNP



1. DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
1. A デバイスは、SetFeature b_hnp_enable 記述子を送信して、HNP サポートを有効にします。OTG_FS コントローラの ACK 応答は、それが HNP をサポートしていることを示します。アプリケーションは、OTG 制御およびステータスレジスタのデバイス HNP イネーブルビットをセットして、HNP をサポートしていることを示す必要があります。
アプリケーションは、OTG 制御およびステータスレジスタの HNP リクエストビットをセットして、HNP を開始することを OTG_FS コントローラに示す必要があります。

2. A デバイスがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってサスペンドします。
OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。
OTG_FS コントローラは切断され、A デバイスはバス上で SE0 を検出して HNP を示します。OTG_FS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。
A デバイスは、SE0 を検出してから 3 ms 以内に OTG_DP プルアップ抵抗を有効にすることによって応答します。OTG_FS コントローラはこれを接続として検出します。
OTG_FS コントローラは、OTG 割り込みステータスレジスタのホストネゴシエーション成功ステータス変更割り込みをセットして、HNP ステータスを示します。アプリケーションは、OTG 制御およびステータスレジスタのホストネゴシエーション成功ビットを読み出して、ホストネゴシエーションの成功を確認する必要があります。アプリケーションは、コア割り込みレジスタ (OTG_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
3. アプリケーションはリセットビット (OTG_HPRT の PRST) をセットし、OTG_FS コントローラは USB リセットを発行し、データトラフィックのために A デバイスをエnumレーションします。
4. OTG_FS コントローラは、トラフィックを開始するというホストの役割を継続し、これが終了すると、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってバスをサスペンドします。
5. ネゴシエートモードでは、A デバイスは、サスペンドを検出すると、接続を切断し、ホストの役割に戻ります。OTG_FS コントローラは、PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割の前提を示します。
6. アプリケーションは、コア割り込みレジスタ (OTG_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
7. OTG_FS コントローラが接続され、HNP プロセスが完了します。

30 デバッグサポート (DBG)

30.1 概要

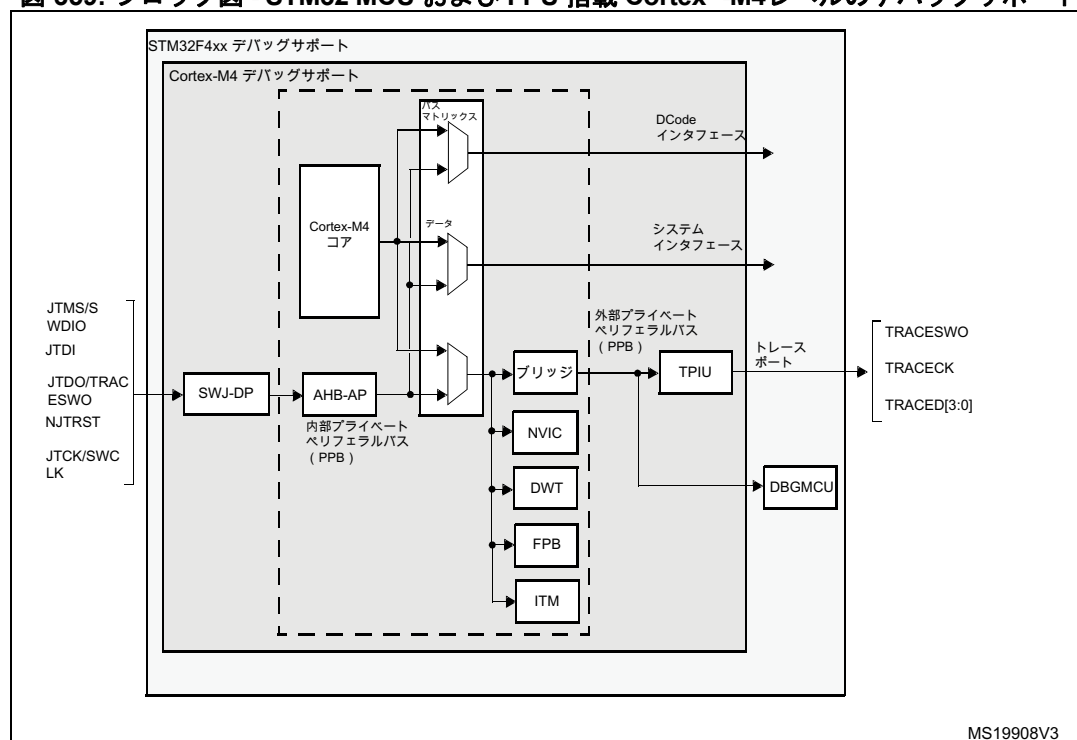
STM32F412xx は、FPU 搭載 Cortex®-M4 コアを中心にして構築されており、高度なデバッグ機能のためのハードウェア拡張機能を含んでいます。デバッグ拡張機能によって、コアは、特定の命令フェッチ時（ブレークポイント）またはデータアクセス時（ウォッチポイント）に停止することができます。停止したとき、コアの内部状態とシステムの外部状態を調査することができます。調査が終わったら、コアとシステムを復元して、プログラム実行を再開することができます。

デバッグ機能は、STM32F412xx MCU への接続時とデバッグ時にデバuggによって使用されます。

デバッグ用に、次の 2 つのインタフェースを使用できます。

- シリアルワイヤ
- JTAG デバッグポート

図 359. ブロック図 - STM32 MCU および FPU 搭載 Cortex®-M4レベルのデバッグサポート



注： FPU 搭載 Cortex®-M4 コアに内蔵されているデバッグ機能は、ARM® CoreSight Design Kit のサブセットです。

ARM® FPU 搭載 Cortex®-M4 コアは、統合的なオンチップデバッグサポート機能を提供します。この機能は以下の要素で構成されます。

- SWJ-DP : シリアルワイヤ/JTAG デバッグポート
- AHP-AP : AHB アクセスポート
- ITM : 計測トレースマクロセル (Instrumentation Trace Macrocell)
- FPB : フラッシュパッチブレークポイント (Flash Patch Breakpoint)
- DWT : データウォッチポイントトリガ
- TPUI : トレースポートインタフェースユニット (Trace Port Interface Unit : 対応するピンが配置される大型パッケージで使用)
- ETM : エンベデッドトレースマクロセル (Embedded Trace Macrocell : 対応するピンが配置される大型パッケージで使用)

また、STM32F412xx専用の以下のデバッグ機能も内蔵されています。

- 柔軟性の高いデバッグピンの割り当て
- MCU デバッグボックス (低電力モードのサポート、ペリフェラルクロックの制御など)

注 : **ARM® FPU 搭載 Cortex®-M4 コアでサポートされているデバッグ機能の詳細は、FPU 搭載 Cortex®-M4-r0p1 Technical Reference Manual および CoreSight Design Kit-r0p1 TRM を参照してください (セクション 30.2 : ARM® リファレンス資料を参照)。**

30.2 ARM® リファレンス資料

- FPU 搭載 Cortex®-M4 r0p1 Technical Reference Manual (TRM)
(1 ページの関連資料を参照)
- ARM® Debug Interface V5
- ARM® CoreSight Design Kit revision r0p1 Technical Reference Manual

30.3 SWJ デバッグポート (シリアルワイヤと JTAG)

STM32F412xx コアには、シリアルワイヤ/JTAG デバッグポート (SWJ-DP) が組み込まれています。これは、JTAG-DP (5 ピン) インタフェースと SW-DP (2 ピン) インタフェースを組み合わせた ARM® 標準の CoreSight デバッグポートです。

- JTAG デバッグポート (JTAG-DP) は、AHP-AP ポートに 5 ピンの標準 JTAG インタフェースを提供します。
- シリアルワイヤデバッグポート (SW-DP) は、AHP-AP ポートに 2 ピン (クロック + データ) のインタフェースを提供します。

SWJ-DP では、SW-DP の 2 個の JTAG ピンは、JTAG-DP の 5 個の JTAG ピンの一部と多重化されています。

30.4.1 SWJ デバッグポートピン

5 個のピンが SWJ-DP 用の STM32F412xx からの出力として使用されます (汎用入出力のオルタネート機能)。これらのピンはすべてのパッケージで使用できます。

表 196. SWJ デバッグポートピン

SWJ-DP ピン名	JTAG デバッグポート		SW デバッグポート		ピン 割り当て
	タイプ	説明	タイプ	デバッグ割り当て	
JTMS/SWDIO	I	JTAG テストモード選択	IO	シリアルワイヤデータ入出力	PA13
JTCK/SWCLK	I	JTAG テストクロック	I	シリアルワイヤクロック	PA14
JTDI	I	JTAG テストデータ入力	-	-	PA15
JTDO/TRACESWO	O	JTAG テストデータ出力	-	非同期トレースが有効な場合は TRACESWO	PB3
NJTRST	I	JTAG テスト nReset	-	-	PB4

30.4.2 柔軟性の高い SWJ-DP ピンの割り当て

RESET (SYSRESETn または PORESETn) 後、SWJ-DP 用に使用される 5 個のピンはすべて、デバッグホストによってすぐに使用可能な専用ピンとして割り当てられます (なお、デバッグホストによって明示的にプログラミングされた場合を除いて、トレース出力は割り当てられません)。

ただし、STM32F412xx MCU は SWJ-DP ポートの一部またはすべてを無効にすることができますので、汎用 IO (GPIO) に使用する関連ピンを解放する可能性もあります。SWJ-DP ポートピンを無効にする方法の詳細については、[セクション 7.3.2: I/O ピンマルチプレクサとマッピング](#)を参照してください。

表 197. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ IO ピンの割り当て				
	PA13 / JTMS / SWDIO	PA14 / JTCK / SWCLK	PA15 / JTDI	PB3 / JTDO	PB4 / NJTRST
全 SWJ (JTAG-DP + SW-DP)、リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効					解放

注： APB ブリッジの書き込みバッファがフルのとき、GPIO_AFR レジスタへの書き込みには APB 1 サイクル分が追加で必要になります。これは、コアの nTRST および TCK 入力信号でのクリーンレベルを保証するために、JTAGSW ピンの無効化が 2 サイクルで行われるからです。

- サイクル 1：コアへの JTAGSW 入力信号は 1 または 0 に (nTRST、TDI、および TMS では 1 に、TCK では 0 に) 接続されます。
- サイクル 2：GPIO コントローラが SWJTAG 入出力ピンの制御 (方向、プルアップ/ダウン、シュミットトリガの有効化などの制御) 信号を受け取ります。

30.4.3 JTAG ピンでの内部プルアップ／プルダウン

JTAG 入力ピンは、デバッグモード機能を制御するためにフリップフロップに直結されます。したがって、JTAG 入力ピンをフロート状態にしないことが必要です。これらのフリップフロップの一部のクロックに直結される SWCLK/TCK ピンについては、特に注意が必要です。

入出力レベルを正しく制御するため、デバイスには内部プルアップ／プルダウンが JTAG 入力ピンに内蔵されています。

- NJTRST : 内部プルアップ
- JTDI : 内部プルアップ
- JTMS/SWDIO : 内部プルアップ
- TCK/SWCLK : 内部プルダウン

JTAG 入出力がユーザソフトウェアによって解放されると、GPIO コントローラが再び制御権を獲得します。リセット状態では、GPIO 制御レジスタは入出力をこれと同等の状態に設定します。

- NJTRST : AF 入力プルアップ
- JTDI : AF 入力プルアップ
- JTMS/SWDIO : AF 入力プルアップ
- JTCK/SWCLK : AF 入力プルダウン
- JTDO : AF 出力フローティング

ソフトウェアはこれらの入出力を標準の GPIO 信号として使用することができます。

注 : *JTAG IEEE 規格では、TDI、TMS、およびnTRST にプルアップを追加することを推奨していますが、TCK に関しては特別な推奨はありません。ただし、TCK に関しては、デバイスは内蔵プルダウンを必要とします。*

プルアップとプルダウンを内蔵しているため、外部抵抗を追加する必要はありません。

30.4.4 シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法

シリアルワイヤ DP を使って GPIO を解放するには、ユーザソフトウェアが GPIO_MODER レジスタで GPIO (PA15、PB3、および PB4) コンフィギュレーションモードを変更する必要があります。これによって PA15、PB3、および PB4 が解放され、GPIO として使用できるようになります。

デバッグ時には、ホストは次の動作を行います。

- システムリセット中に、すべての SWJ ピンが割り当てられます (JTAG-DP + SW-DP)。
- システムリセット中に、デバッグホストは、JTAG-DP から SW-DP に切り替える JTAG シーケンスを送信します。
- システムリセット中に、さらに、デバッグはベクタリセットにブレークポイントを設定します。
- システムリセットは解除され、コアは停止します。
- これ以降のすべてのデバッグ通信は、SW-DP を使用して行われます。残りの JTAG ピンは、ユーザソフトウェアによって GPIO として再割当てできます。

注： ユーザソフトウェアの設計に関しては、次の点に注意してください。

デバッグピンを解放するには、リセット後にユーザソフトウェアがピンを解放するまでの一定の期間、デバッグピンは、まず、入力プルアップ (nTRST、TMS、TDI)、プルダウン (TCK)、または出カトライステート (TDO) に設定されることに注意してください。

デバッグピン (JTAG、SW、または TRACE) が配置されると、IOPORT コントローラの対応する入出力ピンの設定を変更しても効果はありません。

30.5 JTAG TAP 接続

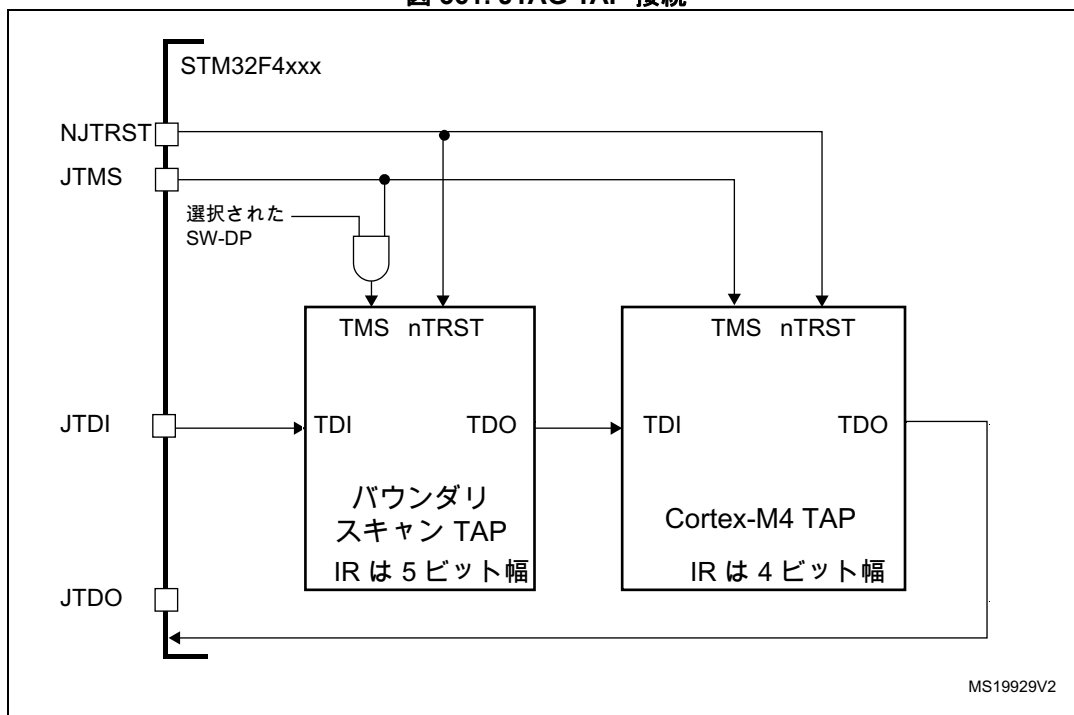
MCU は、バウンダリスキャン TAP (IR は 5 ビット幅) と FPU 搭載 Cortex®-M4 TAP (IR は 4 ビット幅) という、シリアル接続された 2 個の JTAG TAP を内蔵しています。

デバッグ目的で FPU 搭載 Cortex®-M4 の TAP にアクセスするには、

1. まず、バウンダリスキャン TAP の BYPASS 命令をシフトする必要があります。
2. 次に、IR シフトごとに、スキャンチェーンには 9 (=5+4) ビットが格納され、未使用の TAP 命令を BYPASS 命令によってシフトインする必要があります。
3. データシフトごとに、BYPASS モードにある未使用の TAP は、データスキャンチェーンに 1 個のデータビットを追加します。

注： 重要：専用の ARM® JTAG シーケンスを使用してシリアルワイヤが選択されると、バウンダリスキャン TAP は自動的に無効になります (JTMS は強制的にハイレベルになります)。

図 361. JTAG TAP 接続



30.6 ID コードとロック機構

MCU には、内部にいくつかの ID コードがあります。ツール設計者は、外部 PPB メモリマップのアドレス 0xE0042000 に配置されている MCU デバイス ID コードを使用して、デバッグをロックすることを強く推奨します。

30.6.1 MCU デバイス ID コード

MCU には MCU ID コードが内蔵されています。この ID は、ST 社製 MCU の部品番号とダイのリビジョンを識別します。これは DBG_MCU 部品の一部であり、外部 PPB バスに配置されます ([セクション 30.16 \(1116 ページ\)](#) を参照)。このコードにアクセスするには、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアを使用します。アクセスは MCU がシステムリセット中でも可能です。

デバッガ/プログラマツールでは、DEV_ID(11:0) のみを識別に使用してください。

DBGMCU_IDCODE

アドレス : 0xE004 2000

32 ビットアクセスのみサポートされます。読み出し専用。

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

ビット 31:16 **REV_ID(15:0)** リビジョン識別子

このフィールドは、デバイスのリビジョンを示します。

0x1001 = リビジョン Z

0x2000 = リビジョン B

0x3000 = リビジョン C

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DEV_ID(11:0)** : デバイス識別子

デバイス ID は 0x441 です。

30.6.2 バウンダリスキャン TAP

JTAG ID コード

BSC (バウンダリスキャン) の TAP は、次の JTAG ID コードを内蔵しています : 0x0645 8041

30.6.3 FPU 搭載 Cortex®-M4 TAP

ARM® FPU 搭載 Cortex®-M4 の TAP は JTAG ID コードを内蔵しています。この ID コードは ARM® のデフォルトであり、変更されていません。このコードは、JTAG デバッグポートからのみアクセスできます。

このコードは 0x4BA0 0477 です (FPU 搭載 Cortex®-M4 r0p1 に対応、[セクション 30.2 : ARM® リファレンス資料](#)を参照)。

30.6.4 FPU 搭載 Cortex®-M4 JEDEC-106 ID コード

ARM® FPU 搭載 Cortex®-M4 は JEDEC-106 ID コードを内蔵しています。これは内部 PPB バスのアドレス 0xE00F FFD0_0xE00F FFE0 に配置された 4KB ROM テーブルに置かれています。

このコードは、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアによってアクセスできます。

30.7 JTAG デバッグポート

標準的な JTAG ステートマシンは、4 ビット命令レジスタ (IR) と 5 個のデータレジスタを搭載しています (全詳細については、FPU 搭載 Cortex®-M4r0p1 *Technical Reference Manual (TRM)* を参照し、参考として [セクション 30.2 : ARM® リファレンス資料](#) を参照してください)。

表 198. JTAG デバッグポートのデータレジスタ

IR[3:0]	データレジスタ	詳細
1111	BYPASS [1 ビット]	
1110	IDCODE [32 ビット]	ID コード 0x4BA0 0477 (ARM® FPU 搭載 Cortex®-M4 r0p1 の ID コード)
1010	DPACC [35 ビット]	デバッグポートアクセスレジスタ デバッグポートを初期化し、デバッグポートレジスタへのアクセスを可能にします。 – 入力データ転送時 : ビット 34:3 = DATA[31:0] = 書き込みリクエスト用に転送する 32 ビットデータ ビット 2:1 = A[3:2] = デバッグポートレジスタの 2 ビットアドレス ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0) – 出力データ転送時 : ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : 010 = OK/FAULT 001 = WAIT その他 = 予約済み A[3:2] ビットの説明については、 表 199 を参照してください。

表 198. JTAG デバッグポートのデータレジスタ (続き)

IR[3:0]	データレジスタ	詳細
1011	APACC [35 ビット]	<p>アクセスポートアクセスレジスタ</p> <p>アクセスポートを初期化し、アクセスポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> 入力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 書き込みリクエスト用にシフトインする 32 ビットデータ ビット 2:1 = A[3:2] = 2 ビットアドレス (サブアドレス AP レジスタ) ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0) 出力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : <ul style="list-style-type: none"> 010 = OK/FAULT 001 = WAIT その他 = 予約済み <p>次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。</p> <ul style="list-style-type: none"> シフトされた値 A[3:2] DP SELECT レジスタの現在値
1000	ABORT [35 ビット]	<p>アボートレジスタ</p> <ul style="list-style-type: none"> ビット 31:1 = 予約済み ビット 0 = DAPABORT : DAP アボートを生成するには 1 を書き込みます

表 199. シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ

アドレス	A[3:2] の値	説明
0x0	00	予約済みであり、リセット値に保持する必要があります。
0x4	01	<p>DP CTRL/STAT レジスタ : 次の目的で使用されます。</p> <ul style="list-style-type: none"> システム/デバッグパワーアップのリクエスト AP アクセス用の転送動作の設定 プッシュ比較/プッシュ検証動作の制御 ステータスフラグ (オーバーラン、パワーアップの確認応答) の読み出し
0x8	10	<p>DP SELECT レジスタ : 現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウの選択に使用されます。</p> <ul style="list-style-type: none"> ビット 31:24 (APSEL) : 現在の AP を選択します。 ビット 23:8 : 予約済み ビット 7:4 (APBANKSEL) : 現在の AP でアクティブな 4 ワードレジスタウィンドウを選択します。 ビット 3:0 : 予約済み
0xC	11	<p>DP RDBUFF レジスタ : 一連の動作のあとで (新しい JTAG-DP 動作をリクエストせずに)、デバッガが最終結果を得られるようにします。</p>

30.8 SW デバッグポート

30.8.1 SW プロトコルの概要

この同期式シリアルプロトコルでは、次の 2 個のピンを使用します。

- SWCLK : ホストからターゲットへのクロック
- SWDIO : 双方向

このプロトコルでは、2 バンクのレジスタ (DPACC レジスタと APACC レジスタ) の読み出し／書き込みが可能です。

ビットは、ワイヤ上を LSB ファーストで転送されます。

SWDIO を双方向管理するには、ボード上でラインがプルアップされている必要があります (ARM® の推奨値は 100 KΩ です)。

プロトコルで SWDIO の方向が変化するたびに、ラインがホストからもターゲットからも駆動されないターンアラウンド時間が挿入されます。このターンアラウンド時間のデフォルトは 1 ビット時間ですが、SWCLK 周波数の設定によって調整できます。

30.8.2 SW プロトコルシーケンス

各シーケンスは 3 つのフェーズで構成されます。

1. ホストによって送信されるパケットリクエスト (8 ビット)
2. ターゲットによって送信される確認応答 (3 ビット)
3. ホストまたはターゲットによって送信されるデータ転送フェーズ (33 ビット)

表 200. パケットリクエスト (8 ビット)

ビット	名前	説明
0	Start	“1”である必要があります。
1	APnDP	0 : DP アクセス 1 : AP アクセス
2	RnW	0 : 書き込みリクエスト 1 : 読み出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 199 を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。プルアップの効果で、ターゲットには「1」と読み出される必要があります。

DPACC レジスタと APACC レジスタの詳細については、FPU 搭載 Cortex®-M4 r0p1 TRM を参照してください。

パケットリクエストの後には、必ずホストもターゲットもラインを駆動しないターンアラウンド時間 (デフォルトでは 1 ビット) が続きます。



表 201. ACK 応答 (3 ビット)

ビット	名前	説明
0~2	ACK	001 : FAULT 010 : WAIT 100 : OK

読み出しトランザクションの場合や、受信した ACK 応答が“WAIT” または“FAULT” の場合にのみ、ACK 応答の後にターンアラウンド時間が続く必要があります。

表 202. データ転送 (33 ビット)

ビット	名前	説明
0~31	WDATA または RDATA	書き込み／読み出しデータ
32	Parity	32 データビットの 1 ビットパリティ

読み出しトランザクションの場合にのみ、データ転送の後にターンアラウンド時間が続く必要があります。

30.8.3 SW-DP ステートマシン (リセット、アイドル状態、ID コード)

SW-DP のステートマシンには、SW-DP を識別する内部 ID コードがあります。これは JEP-106 規格に準じています。この ID コードは、ARM® のデフォルトコードであり、0x2BA01477 (FPU 搭載 Cortex®-M4 r0p1 に対応) がセットされています。

注： **SW-DP ステートマシンは、ターゲットがこの ID コードを読み出すまで非アクティブであることに注意してください。**

- パワーオンリセット後、または DP が JTAG から SWD に切り替えられた後、またはラインが 50 サイクルを超えてハイレベルにあった後では、SW-DP ステートマシンは RESET 状態になります。
- リセット状態のあと、ラインが 2 サイクル以上の間ローレベルであれば、SW-DP ステートマシンはアイドル状態になります。
- リセット状態のあとは、まずアイドル状態に入り、次に DP-SW ID CODE レジスタの読み出しアクセスを行う**必要があります**。そうしないと、ターゲットは、他のトランザクションに対して ACK 応答の“FAULT”を発行します。

SW-DP ステートマシンの詳細については、FPU 搭載 Cortex®-M4 r0p1 TRM および CoreSight Design Kit r0p1 TRM を参照してください。

30.8.4 DP と AP の読み出し／書き込みアクセス

- DP への読み出しアクセスはポストされません。つまり、ターゲットは、ACK 応答が“OK”の場合はただちに応答し、ACK 応答が“WAIT”の場合は遅れて応答します。
- AP への読み出しアクセスはポストされます。つまり、アクセスの結果は次の転送時に返されます。次のアクセスが AP アクセスでない場合、結果を得るには DP-RDBUFF レジスタを読み出す必要があります。
AP 読み出しアクセスが成功したかどうかを判断するため、DP-CTRL/STAT レジスタの READOK フラグは、AP 読み出しアクセスまたは RDBUFF 読み出しリクエストのたびに更新されます。

- SW-DP は、DP と AP の両方の書き込みに使用できる書き込みバッファを実装しているため、たとえ他のトランザクションが未処理であっても、書き込み動作を受け付けることができます。書き込みバッファがフルのとき、ターゲットの ACK 応答は“WAIT”です。例外として、IDCODE 読み出し、CTRL/STAT 読み出し、または ABORT 書き込みは、書き込みバッファがフルであっても受け付けられます。
- 非同期クロックドメイン SWCLK と HCLK によって、書き込みを内部的に有効にするには、書き込みトランザクション後（パリティビット後）に SWCLK の 2 サイクルが余分に必要となります。これらのサイクルは、ラインをローレベルに駆動している間（アイドル状態）に適用してください。
これは、パワーアップリクエストのために CTRL/STAT の書き込みを行う際に特に重要です。パワーアップを必要とする次のトランザクションがただちに発生すると、そのトランザクションは失敗します。

30.8.5 SW-DP レジスタ

これらのレジスタへのアクセスは、APnDP = 0 のときに開始されます。

表 203. SW-DP レジスタ

A[3:2]	読み出し／書き込み	SELECT レジスタの CTRLSEL ビット	レジスタ	注
00	読み出し	-	IDCODE	製造者コードは ST 社のコードではありません。0x2BA01477（SW-DP を識別）。
00	書き込み	-	ABORT	-
01	読み出し／書き込み	0	DP CTRL/STAT	用途は以下のとおりです。 – システム／デバッグパワーアップのリクエスト – AP アクセス用の転送動作の設定 – ブッシュ比較／ブッシュ検証動作の制御 – ステータスフラグ（オーバーラン、パワーアップの確認応答）の読み出し
01	読み出し／書き込み	1	WIRE CONTROL	物理的なシリアルポートプロトコルの設定（ターンアラウンド時間など）を行います。
10	読み出し		READ RESEND	元の AP 転送を反復しなくても、破壊されたデバッグ転送からの読み出しデータの復旧を可能にします。
10	書き込み		SELECT	現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウを選択します。
11	読み出し／書き込み		読み出しバッファ	AP アクセスはポストされるため、この読み出しバッファは効果的です（AP 読み出しリクエストの結果は、次の AP トランザクションで取得できる）。 この読み出しバッファは、新しいトランザクションを開始することなく、前回の読み出しの結果として AP から出力されるデータをキャプチャします。

30.8.6 SW-AP レジスタ

これらのレジスタへのアクセスは、APnDP = 1 のときに開始されます。

次の項目の組み合わせとして、多くの AP レジスタ（AHB-AP を参照）をアドレス指定します。

- シフトされた値 A[3:2]
- DP SELECT レジスタの現在値

30.9 AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効

機能：

- システムアクセスはプロセッサステータスから独立しています。
- SW-DP または JTAG-DP が AHB-AP にアクセスします。
- AHB-AP はバスマトリックスの AHB マスタです。したがって、AHB-AP は ICode バスを除くすべてのデータバス（Dcode バス、システムバス、内部／外部の PPB バス）にアクセスできます。
- ビットバンドトランザクションがサポートされます。
- AHB-AP トランザクションは FPB を迂回します。

32 ビット AHB-AP レジスタは 6 ビット幅（最大 64 ワードまたは 256 バイト）であり、以下の構成となっています。

- c) ビット [7:4] = DP_SELECT レジスタのビット [7:4] APBANKSEL
- d) ビット [3:2] = SW-DP 用の 35 ビットパケットリクエストの 2 ビットアドレス A[3:2]

FPU 搭載 Cortex®-M4 の AHB-AP は、9 個の 32 ビットレジスタを内蔵しています。

表 204. FPU 搭載 Cortex®-M4 AHB-AP レジスタ

アドレス オフセット	レジスタ名	注
0x00	AHB-AP 制御およびステータス ワード	AHB インタフェースを通じて転送を設定および制御します (サイズ、hprot、現在の転送のステータス、アドレスインクリ メントタイプ)。
0x04	AHB-AP 転送アドレス	-
0x0C	AHB-AP データ読み出し／書き込 み	-
0x10	AHB-AP バンクデータ 0	転送アドレスレジスタを書き換えずに、4 個のアラインドデー タワードを直接配置します。
0x14	AHB-AP バンクデータ 1	
0x18	AHB-AP バンクデータ 2	
0x1C	AHB-AP バンクデータ 3	
0xF8	AHB-AP デバッグ ROM アドレス	デバッグインタフェースのベースアドレス
0xFC	AHB-AP ID レジスタ	-

詳細については、FPU 搭載 Cortex®-M4 r0p1 TRM を参照してください。

30.10 コアデバッグ

コアデバッグはコアデバッグレジスタを通じてアクセスされます。これらのレジスタへのデバッグアクセスには、*Advanced High-performance Bus* (AHB-AP) ポートを使用します。プロセッサは、内部の *プライベートペリフェラルバス* (PPB) を介してこれらのレジスタに直接アクセスできます。

コアデバッグは 4 個のレジスタから構成されています。

表 205. コアデバッグレジスタ

レジスタ	説明
DHCSR	32 ビットのデバッグ停止制御／ステータスレジスタ： レジスタプロセッサの状態についてのステータス情報を提供し、コアデバッグを有効にし、プロセッサの停止とステップ実行を行います。
DCRSR	17 ビットのコアデバッグレジスタセレクトレジスタ： データの転送先または転送元となるプロセッサレジスタを選択します。
DCRDR	32 ビットのコアデバッグレジスタデータレジスタ： DCRSR (セレクト) レジスタによって選択されたプロセッサとの間でレジスタの読み出しおよび書き込みに使用するデータを保持します。
DEMCR	32 ビットのコアデバッグ例外／モニタ制御レジスタ： ベクタキャッチとデバッグモニタの制御を行います。このレジスタには、TRACE を使用できるようにする <i>TRCENA</i> というビットがあります。

注： **重要：**これらのレジスタは、システムリセットによってはリセットされません。パワーオンリセットによってのみリセットされます。

詳細については、*FPU 搭載 Cortex®-M4 r0p1 TRM* を参照してください。

リセット時に停止させるには、以下の手順が必要です。

- デバッグ例外／モニタ制御レジスタのビット 0 (VC_CORRESET) を有効にします。
- デバッグ停止制御／ステータスレジスタのビット 0 (C_DEBUGEN) を有効にします。

30.11 システムリセット中のデバッグホスト接続機能

MCU のリセットシステムは、次のリセットソースから構成されます。

- POR (パワーオンリセット) : パワーアップのたびに RESET をアサートします。
- 内部ウォッチドッグリセット
- ソフトウェアリセット
- 外部リセット

FPU 搭載 Cortex®-M4 では、デバッグ部のリセット (一般に PORRESETn) とそれ以外のリセット (SYSRESETn) が区別されます。

こうすれば、リセットベクタをフェッチするときにコアを停止させるようコアデバッグレジスタをプログラミングすることで、デバッグはシステムリセット中に接続が可能になります。これによってホストはシステムリセットを解除でき、コアは命令を実行しなくても、ただちに停止します。さらに、システムリセット中にデバッグ機能をプログラミングできます。

注 : システムリセット中にデバッグホストを接続する (リセットベクタにブレークポイントを設定する) ことを強く推奨します。

30.12 FPB (フラッシュパッチブレークポイント)

FPB ユニットは、

- ハードウェアブレークポイントを実装します。
- コード空間からシステム空間に、コードとデータをパッチします。この機能によって、コードメモリ空間にあるソフトウェアバグの訂正が可能になることがあります。

ソフトウェアパッチとハードウェアブレークポイントは同時に使用できません。

FPB は以下から構成されます。

- コード空間からのリテラルロードと照合し、システム空間内の対応する領域に再配置する 2 個のリテラルコンパレータ。
- コード空間からの命令フェッチと照合する 6 個の命令コンパレータ。命令コンパレータを使用すると、システム空間内の対応する領域に再配置したり、コアへのブレークポイント命令を生成したりできます。

30.13 DWT (データウォッチポイントトリガ)

DWT ユニットは 4 個のコンパレータから構成されており、次の用途に設定できます。

- ハードウェアウォッチポイント
- ETM へのトリガ
- PC サンプラ
- データアドレスサンプラ

DWT は、プロファイル情報を提供することもできます。このため、次の値を得るためのカウンタにアクセスできます。

- クロックサイクル数
- フォールドされた命令数
- ロードストアユニット (LSU) の動作数
- スリープサイクル数
- CPI (命令あたりのクロック数)
- 割込みオーバーヘッドの回数

30.14 ITM (計測トレースマクロセル)

30.14.1 概要

ITM は、アプリケーション駆動のトレースソースであり、*printf* 方式のデバッグをサポートすることでオペレーティングシステム (OS) やアプリケーションのイベントをトレースし、システムの診断情報を発行します。ITM がパケットとして発行するトレース情報は、次のように生成できます。

- **ソフトウェアトレース**: ソフトウェアは、ITM スティムラスレジスタに直接書き込みを行って、パケットを発行できます。
- **ハードウェアトレース**: DWT がパケットを生成し、ITM がそれを発行します。
- **タイムスタンプ**: タイムスタンプはパケットを基準にして発行されます。ITM は、タイムスタンプを生成するための 21 ビットカウンタを内蔵しています。カウンタは、FPU 搭載 Cortex®-M4 クロックまたはシリアルワイヤ ビューア (SWV) 出力のビットクロックレートによって駆動されます。

ITM によって発行されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから (TPIU を参照)、完全なパケットシーケンスをデバッガホストに出力します。

ITM をプログラミングまたは使用するには、その前に、デバッグ例外/モニタ制御レジスタの TRCEN ビットを有効にする必要があります。

30.14.2 タイムスタンプパケット、同期およびオーバーフローパケット

タイムスタンプパケットは、タイムスタンプ情報や一般的な制御および同期情報をコード化します。そのために使用される 21 ビットのタイムスタンプカウンタ (プリスケアラ付きの場合もあります) は、タイムスタンプパケットが発行されるたびにリセットされます。このカウンタは、CPU クロックまたは SWV クロックによって駆動できます。

同期パケットは 6 バイトの値 0x80_00_00_00_00_00 から構成され、TPIU には 00 00 00 00 00 80 として LSB ファーストで発行されます。

同期パケットは、タイムスタンプ制御のためのパケットで、DWT トリガのたびに発行されます。



このため、DWT 制御レジスタの CYCCNTENA (ビット 0) をセットして、ITM をトリガするように DWT を設定する必要があります。さらに、ITM トレース制御レジスタのビット 2 (SYNCENA) もセットする必要があります。

注 : **SYNENA ビットがセットされていない場合、DWT は TPIU への同期トリガを生成し、TPIU 同期パケットのみを送信し、ITM 同期パケットは送信しません。**

オーバーフローパケットは、データの書き込み時に FIFO がフルであったことを示す特殊なタイムスタンプパケットで構成されます。

表 206. 主な ITM レジスタ

アドレス	レジスタ	詳細
@E0000FB0	ITM ロックアクセス	他の ITM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
@E0000E80	ITM トレース制御	ビット 31-24 = 常に 0
		ビット 23 = ビジー
		ビット 22-16 = トレースデータのソースを識別する 7 ビットの ATB ID
		ビット 15-10 = 常に 0
		ビット 9:8 = TSPrescale = タイムスタンププリスケラ
		ビット 7-5 = 予約済み
		ビット 4 = SWOENA = SWV 動作 (タイムスタンプカウンタを SWV クロックによって駆動) を有効にします。
		ビット 3 = DWTENA : DWT スティムラスを有効にします。
		ビット 2 = SYNCENA : DWT が同期トリガを生成して TPIU が同期パケットを発行できるようにするには、このビットを 1 にする必要があります。
		ビット 1 = TSENA (タイムスタンプ有効)
@E0000E40	ITM トレース特権	ビット 0 = ITMENA : ITM のグローバル有効ビット
		ビット 3 : ポート 31:24 のトレースを有効にするマスク
		ビット 2 : ポート 23:16 のトレースを有効にするマスク
		ビット 1 : ポート 15:8 のトレースを有効にするマスク
@E0000E00	ITM トレース有効	ビット 0 : ポート 7:0 のトレースを有効にするマスク
		各ビットは、トレースを生成するために対応するスティムラスポートを有効にします。
@E0000000 ~ E000007C	スティムラスポートレジスタ 0-31	選択されたスティムラスポート (32 個まで) に 32 ビットデータを書き込んでトレース出力します。



設定例

TPIU に単純な値を出力するには、

- TPIU を設定し、DBGMCU_CR を設定することで TRACE I/O を割り当てます ([セクション 30.17.2: TRACE ピンの割当て](#) および [セクション 30.16.3: デバッグ MCU 設定レジスタ](#) を参照)。
- ITM ロックアクセスレジスタに 0xC5ACCE55 を書き込んで、ITM レジスタへの書き込みアクセスをロック解除します。
- ITM トレース制御レジスタに 0x00010005 を書き込んで、同期を有効にした状態で ITM を有効にし、ATB ID を 0x00 以外の値にします。
- ITM トレース有効レジスタに 0x1 を書き込んで、スティムラサポート 0 を有効にします。
- ITM トレース特権レジスタに 0x1 を書き込んで、スティムラサポート 7:0 をマスク解除します。
- スティムラサポートレジスタ 0 に出力値を書き込みます。これはソフトウェアで行うことができます (printf 機能を使用)。

30.15 ETM (組み込みトレースマクロセル)

30.15.1 概要

ETM はプログラム実行の再構築を有効にします。データはデータウォッチポイントおよびトレース (DWT) コンポーネントまたは計測トレースマクロセル (ITM) を使用してトレースされます。一方、命令は組み込みトレースマクロセル (ETM) を使用してトレースされます。

ETMは、情報をパケットとして送信し、埋め込みリソースによってトリガされます。これらのリソースは個別にプログラミングする必要があります。トリガソースの選択にはトリガイベントレジスタ (0xE0041008) を使用します。イベントは、単一イベント (アドレスコンパレータからのアドレスマッチ) または 2 つのイベント間の論理式とすることができます。トリガソースは、DWT モジュールの 4 番目のコンパレータの 1 つで、次のイベントを監視できます。

- クロックサイクルマッチング
- データアドレスマッチング

トリガリソースの詳細については、[セクション 30.13: DWT \(データウォッチポイントトリガ\)](#) を参照してください。

ETM によって送信されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから ([セクション 30.17: TPIU \(トレースポートインタフェースユニット\)](#) を参照)、完全なパケットシーケンスをデバッグホストに出力します。

30.15.2 信号プロトコル、パケットタイプ

これについては、ARM® IHI 0014N ドキュメントの第 7 章「ETMv3 信号プロトコル」を参照してください。



30.15.3 主な ETM レジスタ

レジスタの詳細については、ARM® IHI 0014N 仕様の第 3 章を参照してください。

表 207. 主な ETM レジスタ

アドレス	レジスタ	詳細
0xE0041FB0	ETM ロックアクセス	他の ETM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
0xE0041000	ETM 制御	このレジスタは、ETM の一般的な動作（たとえば、トレースを有効にする方法）を制御します。
0xE0041010	ETM ステータス	このレジスタは、トレースおよびトリガロジックの現在のステータスに関する情報を提供します。
0xE0041008	ETM トリガイイベント	このレジスタは、トリガを制御するイベントを定義します。
0xE004101C	ETM トレース有効制御	このレジスタは、どのコンパレータが選択されるかを定義します。
0xE0041020	ETM トレース有効イベント	このレジスタは、トレース有効化イベントを定義します。
0xE0041024	ETM トレース開始/停止	このレジスタは、トレースをそれぞれ開始および停止するためにトリガソースによって使用されるトレースを定義します。

30.15.4 設定例

TPIU に単純な値を出力するには、

1. TPIU を設定し、I/O_TRACEN を有効にしてデバッグ設定レジスタで TRACE I/O を割り当てます。
2. ETM ロックアクセスレジスタに 0xC5ACCE55 を書き込んで、ETM レジスタへの書き込みアクセスをロック解除します。
3. 制御レジスタに 0x00001D1E を書き込みます（トレースを設定します）。
4. トリガイイベントレジスタに 0000406F を書き込みます（トリガイイベントを定義します）。
5. トレース有効イベントレジスタに 0000006F を書き込みます（開始/停止するイベントを定義します）。
6. トレース開始/停止レジスタに 00000001 を書き込みます（トレースを有効にします）。
7. ETM 制御レジスタに 0000191E を書き込みます（設定の終了）。

30.16 MCU デバッグコンポーネント (DBGMCU)

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

- 低電力モード
- ブレークポイントにおける、タイマ、ウォッチドッグ、および I2C のクロック制御
- TRACE ピンの割当て制御

30.16.1 低電力モードのデバッグサポート

低電力モードに入るには、WFI または WFE 命令を実行する必要があります。

MCU はいくつかの低電力モードを実装しており、CPU クロックを無効にしたり、CPU の消費電力を低減したりすることができます。

デバッグセッション中には、コアは FCLK や HCLK をオフにすることはできません。デバッグ時のデバッグ接続に必要なため、これらをアクティブな状態に保つ必要があります。MCU は、ユーザが低電力モードでソフトウェアをデバッグするための特殊な手段を備えています。

このため、デバッグホストは、最初にいくつかのデバッグ設定レジスタをセットして、低電力モード動作を変更する必要があります。

- SLEEP モードでは、DBGMCU_CR レジスタの DBG_SLEEP ビットをデバッグによって事前にセットする必要があります。これによって、HCLK には FCLK と同じクロックが供給されます (システムクロックはソフトウェアによって事前に設定されています)。
- STOP モードでは、DBG_STOP ビットをデバッグによって事前にセットする必要があります。これによって、内部 RC オシレータが、STOP モードで FCLK と HCLK にクロックを供給できます。

30.16.2 タイマ、ウォッチドッグ、bxCAN、および I²C のデバッグサポート

ブレークポイントにおいては、以下に示すタイマのカウンタやウォッチドッグの動作方法を選択する必要があります。

- ブレークポイントの中でもカウントを継続できます。この動作は、たとえば、PWM がモータを制御しているときに一般的に必要です。
- ブレークポイントの中でカウントを停止できます。この動作はウォッチドッグ用に必要です。

bxCAN の場合、ブレークポイントにおいては受信レジスタの更新をブロックするように選択できます。

I²C の場合、ブレークポイントにおいては SMBUS タイムアウトをブロックするように選択できます。

30.16.3 デバッグ MCU 設定レジスタ

このレジスタを使用して、デバッグ中に MCU を設定できます。次の設定が可能です。

- 低電力モードのサポート
- タイマおよびウォッチドッグカウンタのサポート
- TRACE ピンの割当て

この DBGMCU_CR は、アドレス 0xE0042004 にある外部 PPB バスに配置されます。

このレジスタは PORESET によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

デバッグホストがこれらの機能をサポートしない場合でも、ユーザソフトウェアによってこれらのレジスタへの書き込みが可能です。

DBGMCU_CR レジスタ

アドレス : 0xE004 2004

32 ビットアクセスのみサポートされます。

POR リセット : 0x0000 0000 (システムリセットではリセットされません)



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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:5 **TRACE_MODE[1:0]** および **TRACE_IOEN** : TRACE ピン割り当て制御

- TRACE_IOEN=0 の場合 :
TRACE_MODE=xx : TRACE ピンは割り当てられません (デフォルト)。
- TRACE_IOEN=1 の場合 :
 - TRACE_MODE=00 : 非同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=01 : TRACEDATA サイズが 1 の同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=10 : TRACEDATA サイズが 2 の同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=11 : TRACEDATA サイズが 4 の同期モードの TRACE ピン割り当てです。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG_STANDBY** : デバッグ STANDBY モード

- 0 : (FCLK=オフ、HCLK=オフ) デジタル部全体が電源オフになります。
ソフトウェアから見て、STANDBY モードから抜けることは、リセットベクタのフェッチと同じになります (ただし、いくつかのステータスビットは MCU が STANDBY モードから再開していることを示す)。
- 1 : (FCLK=オン、HCLK=オン) この場合、デジタル部は電源オフ状態ではなく、FCLK と HCLK は引き続きアクティブ状態の内部 RC オシレータから供給されます。さらに、MCU は STANDBY モード中にシステムリセットを生成するため、STANDBY モードから抜けることはリセットからのフェッチと同じになります。

ビット 1 **DBG_STOP** : デバッグ STOP モード

- 0 : (FCLK=オフ、HCLK=オフ) STOP モードでは、クロックコントローラがすべてのクロック (HCLK と FCLK を含む) を無効にします。STOP モードから抜けると、クロック設定はリセット後の場合と同じになります (CPU は 8 MHz の内部 RC オシレータ (HSI) から供給されます)。したがって、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります。
- 1 : (FCLK=オン、HCLK=オン) この場合、STOP モードに入ると、FCLK と HCLK は STOP モードでもアクティブ状態の内部 RC オシレータから供給されます。STOP モードから抜けるとき、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります (DBG_STOP=0 の場合と同様です)。

ビット 0 **DBG_SLEEP** : デバッグ SLEEP モード

- 0 : (FCLK=オン、HCLK=オフ) SLEEP モードでは、FCLK はソフトウェアによって事前に設定されたシステムクロックによって駆動され、HCLK は無効にされます。
SLEEP モードでは、クロックコントローラの設定はリセットされず、事前にプログラミングされた状態のままです。したがって、SLEEP モードから抜けるときに、ソフトウェアでクロックコントローラを再設定する必要はありません。
- 1 : (FCLK=オン、HCLK=オン) この場合、SLEEP モードに入ると、HCLK には FCLK と同じクロック (ソフトウェアによって事前に設定されたシステムクロック) が供給されます。

30.16.4 デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)

DBGMCU_APB1_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB1 ペリフェラルと関係のあるレジスタで、アドレス 0xE004 2008 にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます（システムリセットではリセットされません）。システムリセット中に、デバッグによって書き込みが可能です。

アドレス : 0xE004 2008

32 ビットアクセスのみサポートされます。

パワーオンリセット (POR) : 0x0000 0000（システムリセットではリセットされません）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	DBG_CAN2_STOP	DBG_CAN1_STOP	DBG_I2CFMP_SMBUS_TIMEOUT	DBG_I2C3_SMBUS_TIMEOUT	DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.
					rw	rw	rw	rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **DBG_CAN2_STOP** : コア停止時にデバッグ CAN2 は停止

- 0 : 通常モードと同じ動作です。
- 1 : CAN2 受信レジスタは停止状態です。

ビット 25 **DBG_CAN1_STOP** : コア停止時にデバッグ CAN1 は停止

- 0 : 通常モードと同じ動作です。
- 1 : CAN1 受信レジスタは停止状態です。

ビット 24 **DBG_I2CFMP_SMBUS_TIMEOUT** : コア停止時に FMPI2C SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 23 **DBG_I2C3_SMBUS_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 22 **DBG_I2C2_SMBUS_TIMEOUT** : コア停止時に I2C2 SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 21 **DBG_I2C1_SMBUS_TIMEOUT** : コア停止時に I2C1 SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DBG_IWDG_STOP** : コア停止時にデバッグ独立型ウォッチドッグは停止

- 0 : コアが停止しても独立型ウォッチドッグカウンタのクロックは継続されます。
- 1 : コア停止時に独立型ウォッチドッグカウンタのクロックは停止します。

ビット 11 **DBG_WWDG_STOP** : コア停止時にデバッグウィンドウ型ウォッチドッグは停止

- 0 : コアが停止してもウィンドウ型ウォッチドッグカウンタのクロックは継続されます。
- 1 : コア停止時にウィンドウ型ウォッチドッグカウンタのクロックは停止します。

- ビット 10 **DBG_RTC_STOP** : コア停止時は RTC 停止
- 0 : コアが停止した場合も RTC カウンタのクロックは継続されます。
 - 1 : コア停止時に RTC カウンタのクロックは停止します。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8:0 **DBG_TIMx_STOP** : コア停止時に TIMx カウンタは停止 (x=2, 7, 12..14)
- 0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。
 - 1 : コア停止時に当該タイマカウンタのクロックは停止します。

30.16.5 デバッグ MCU APB2 フリーズレジスタ (DBGMCU_APB2_FZ)

DBGMCU_APB2_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB2 ペリフェラルと関係のあるレジスタで、

アドレス 0xE004 200C にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

アドレス : 0xE004 200C

32 ビットアクセスのみサポートされます。

POR : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_TIM9_STOP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM8_STOP	DBG_TIM1_STOP
														rw	rw

- ビット 31:19 予約済みであり、リセット値に保持する必要があります。
- ビット 18:16 **DBG_TIMx_STOP** : コア停止時に TIMx カウンタは停止 (x=9..11)
- 0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。
 - 1 : コア停止時に当該タイマカウンタのクロックは停止します。
- ビット 15:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1:0 **DBG_TIMx_STOP** : コア停止時に TIMx カウンタは停止 (x=1/8)
- 0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。
 - 1 : コア停止時に当該タイマカウンタのクロックは停止します。

30.17 TPIU (トレースポートインタフェースユニット)

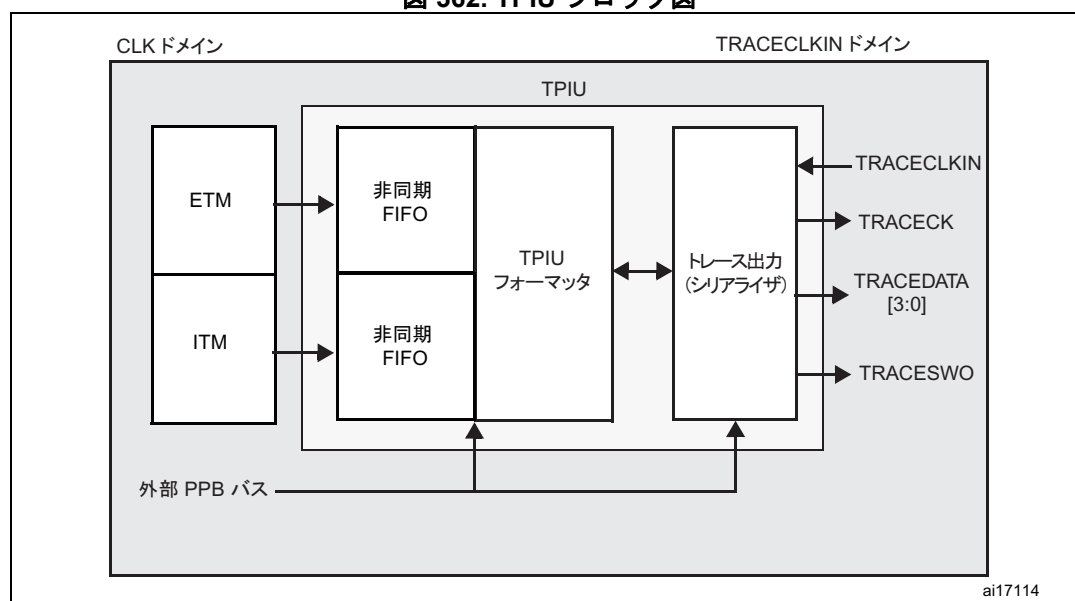
30.17.1 概要

TPIU は、ITM と ETM からオンチップトレースデータ間のブリッジとしての役割を果たします。

出力データストリームによってカプセル化されたトレースソース ID は、トレースポートアナライザ (TPA) によってキャプチャされます。

コア には、特に低コストのデバッグ向けの単純な TPIU (特別バージョンの CoreSight TPIU から構成) が内蔵されています。

図 362. TPIU ブロック図



30.17.2 TRACE ピンの割当て

- 非同期モード
非同期モードは、1 個のピンを追加で必要とし、すべてのパッケージで使用できます。非同期モードは、JTAG モードではなくシリアルワイヤモードを使用する場合にのみ使用できます。

表 208. 非同期 TRACE ピンの割当て

TPUI ピン名	TRACE 同期モード		ピン割り当て
	タイプ	説明	
TRACESWO	O	TRACE 非同期データ出力	PB3

- 同期モード
同期モードは、データトレースサイズに応じて 2~6 本のピンを追加で必要とし、大型パッケージでのみ使用できます。さらに、JTAG モードとシリアルワイヤモードでも使用でき、非同期トレースよりも優れた帯域幅出力機能を提供します。

表 209. 同期 TRACE ピンの割当て

TPUI ピン名	TRACE 同期モード		ピン割り当て
	タイプ	説明	
TRACECK	O	TRACE クロック	PE2
TRACED[3:0]	O	TRACE 同期データ出力 1、2、または 4 とすることができます。	PE[6:3]、PF[7:6]、PD3、 PG[14:13]

TPUI TRACE ピンの割り当て

デフォルトでは、これらのピンは割り当てられません。これらのピンを割り当てるには、**MCU デバッグコンポーネント設定レジスタ**の TRACE_IOEN および TRACE_MODE ビットをセットします。この設定はデバッグホストで行う必要があります。

さらに、割り当てるピン数は、トレースの設定（非同期／同期）によって異なります。

- 非同期モード：1 本のピンが追加で必要です。
- 同期モード：データトレースポートレジスタのサイズ（1、2、または 4）に応じて、2~5 本のピンが追加で必要です。
 - TRACECK
 - TRACED(0)、ポートサイズが 1、2、または 4 に設定された場合
 - TRACED(1)、ポートサイズが 2 または 4 に設定された場合
 - TRACED(2)、ポートサイズが 4 に設定された場合
 - TRACED(3)、ポートサイズが 4 に設定された場合

TRACE ピンを割り当てるには、デバッグホストは、デバッグ MCU 設定レジスタ (DBGMCU_CR) の TRACE_IOEN ビットと TRACE_MODE[1:0] ビットをプログラミングする必要があります。デフォルトでは、TRACE ピンは割り当てられません。

このレジスタは、外部 PPB に配置され、PORESET によってリセットされます（システムリセットではありません）。このレジスタは、システムリセット中にデバッグによる書き込みができます。



表 210. 柔軟性の高い TRACE ピン割り当て

DBGMCU_CR レジスタ		ピン割り当て:	TRACE IO ピンの割当て ⁽¹⁾					
TRACE_ IOEN	TRACE_ MODE [1:0]		JTDO/TRAC ESWO	TRACE CK	TRACE D[0]	TRACE D[1]	TRACE D[2]	TRACE D[3]
0	XX	トレースなし (デフォルト状態)	解放 ⁽²⁾	-				
1	00	非同期トレース	TRACESWO	-	-	解放 (GPIO として使用可能)		
1	01	同期トレース、 1 ビット	解放 ⁽²⁾	TRACECK	TRACED[0]	-	-	-
1	10	同期トレース、 2 ビット		TRACECK	TRACED[0]	TRACED[1]	-	-
1	11	同期トレース、 4 ビット		TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

1. データシートオルタネート機能配置表を参照してください。

2. シリアルワイヤモードが使用されているときには解放されます。しかし、JTAG が使用されると、JTDO に割り当てられます。

注: デフォルトでは、TPIU の TRACECLKIN 入力クロックは GND に接続されています。このクロックは、TRACE_IOEN ビットがセットされてから 2 クロックサイクル後に HCLK に割り当てられます。

デバッグは、TPIU の SPP_R (選択ピンプロトコル) レジスタの PROTOCOL[1:0] ビットに書き込むことによって、トレースモードをプログラミングする必要があります。

- PROTOCOL = 00 : トレースポートモード (同期モード)。
- PROTOCOL = 01 または 10 : シリアルワイヤ (マンチェスタまたは NRZ) モード (非同期モード)。デフォルト状態は 01 です。

次に、TPIU の CPSPS_R (カレント同期化ポートサイズレジスタ) のビット [3:0] に書き込むことによって TRACE ポートサイズも設定します。

- 0x1 : 1 ピンの場合 (デフォルト状態)
- 0x2 : 2 ピンの場合
- 0x8 : 4 ピンの場合

30.17.3 TPUI フォーマット

フォーマットプロトコルは、データを 16 バイトフレームで出力します。

- 7 バイトのデータ
- 8 バイトの混合バイトであり、以下の構成になっています。
 - 1 ビット (LSB) : データバイト (“0”) または ID バイト (“1”) であることを示します。
 - 7 ビット (MSB) : データまたはソース ID トレースの変更とすることができます。
- 1 バイトの補助ビット : 各ビットは 8 つの混合バイトのいずれかに対応します。
 - 対応するバイトがデータであった場合、このビットはデータのビット 0 を示します。
 - 対応するバイトが ID 変更であった場合、このビットはその ID 変更が有効になるタイミングを示します。

注: 詳細については、ARM® CoreSight Architecture Specification v1.0 (ARM® IHI 0029B) を参照してください。

30.17.4 TPUI フレーム同期パケット

TPUI は 2 種類の同期パケットを生成できます。

- フレーム同期パケット (フルワード同期パケット)
これはワード 0x7F_FF_FF_FF (LSB ファーストで発行) から構成されます。ID ソースコード 0x7F が使用されていない場合、このシーケンスは他のタイミングでは発生できません。
このパケットはフレーム *間* で定期的に出力されます。
連続モードでは、同期フレームが検出されると、TPA がこれらすべてのフレームを破棄する必要があります。
- ハーフワード同期パケット
これはハーフワード 0x7F_FF (LSB ファーストで発行) から構成されます。
このパケットはフレーム *間または内* で定期的に出力されます。
これらのパケットは連続モードでのみ生成され、TPA は TRACE ポートが IDLE モードである (TRACE はキャプチャされない) ことを検出できるようになります。TPA によって検出されたパケットは、破棄する必要があります。

30.17.5 同期フレームパケットの送信

コアの TPUI には、同期カウンタレジスタは実装されていません。したがって、同期トリガは DWT によってのみ生成できます。DWT 制御レジスタ (SYNCTAP[11:10] ビット) と DWT カレント PC サンプラサイクルカウントレジスタの説明を参照してください。

TPUI フレーム同期パケット (0x7F_FF_FF_FF) は次の場合に発行されます。

- 毎回の TPUI リセット解除後。このリセットは、TRACECLKIN クロックの立ち上がりエッジに同期して解除されます。つまり、DBGMCU_CFG レジスタの TRACE_IOEN ビットがセットされると、このパケットが送信されます。この場合、ワード 0x7F_FF_FF_FF の後に、フォーマットされたパケットは続きません。
- 毎回の DWT トリガ時 (DWT は事前に設定されているものとします)。次の 2 つの場合があります。
 - ITM の SYNENA ビットがリセットされた場合、ワード 0x7F_FF_FF_FF だけが発行され、フォーマットされたストリームが続くことはありません。
 - ITM の SYNENA ビットがセットされた場合、TPUI によってフォーマット (トレースソース ID を追加) された ITM 同期パケットが続きます (0x80_00_00_00_00_00)。

30.17.6 同期モード

トレースデータの出力サイズは 4、2、または 1 ピンから選択できます TRACED(3:0)

出力クロックはデバッグに出力されます (TRACECK)。

なお、TRACECLKIN は内部で駆動され、TRACE が使用される場合にのみ HCLK に接続されます。

注： *同期モードでは、必ずしも安定したクロック周波数を提供する必要はありません。*

TRACE I/O (TRACECK を含む) は TRACCLKIN (HCLK と同等) の立ち上がりエッジで駆動されます。したがって、TRACECK の出力周波数は HCLK/2 になります。

30.17.7 非同期モード

これは 1 ピン（非同期出力ピン TRACESWO）のみを使用してトレースを出力する低コストな方法ですが、明らかにその帯域幅は制限されます。

SW-DP ピンを使用する場合、TRACESWO ピンは JTDO ピンと多重化されます。このように、この機能はすべてのパッケージで使用できます。

この非同期モードでは、TRACECLKIN に一定の周波数が要求されます。標準的な UART（NRZ）のキャプチャ機構では、5 % の精度が必要です。マンチェスタエンコード方式では、10 % までの誤差が許されます。

30.17.8 TRACECLKIN 接続

TRACECLKIN 入力は内部で HCLK に接続されます。つまり、非同期トレースモードでは、アプリケーションは、CPU 周波数が安定しているタイムフレームのみを使用できます。

注： **重要：非同期トレースを使用する場合は、以下のことに注意してください。**
MCU のデフォルトクロックは、内部 RC オシレータです。リセット中の周波数はリセット解除後の周波数とは異なります。これは、RC 較正はシステムリセット中のデフォルト動作であり、システムリセット解除のたびに更新されるからです。
したがって、トレースポートアナライザ（TPA）は、システムリセット中に（TRACE_IOEN ビットによって）トレースを有効にするべきではありません。なぜなら、同期フレームパケットは、リセット解除後に送信されるトレースパケットとは異なるビット時間で発行されるからです。

30.17.9 TPIU レジスタ

TPIU APB レジスタは、デバッグ例外およびモニタ制御レジスタ（DEMCR）の TRCENA ビットがセットされた場合にのみ読み出し／書き込みが可能です。そうでない場合、これらのレジスタはゼロとして読み出されます（このビットの出力は TPIU の PCLK を有効にします）。

表 211. 重要な TPIU レジスタ

アドレス	レジスタ	説明
0xE0040004	現在のポートサイズ	トレースポートサイズの選択： ビット 0：ポートサイズ = 1 ビット 1：ポートサイズ = 2 ビット 2：ポートサイズ = 3、サポートされません ビット 3：ポートサイズ = 4 1 ビットのみセットする必要があります。デフォルトでは、ポートサイズは 1 ビットです。(0x00000001)
0xE00400F0	選択ピンプロトコル	トレースポートプロトコルの選択： ビット 1:0= 00：同期トレースポートモード 01：シリアルワイヤ出力 - マンチェスタ（デフォルト値） 10：シリアルワイヤ出力 - NRZ 11：予約済み

表 211. 重要な TPIU レジスタ (続き)

アドレス	レジスタ	説明
0xE0040304	フォーマッタおよびフラッシュ制御	ビット 31-9 = 常に“0” ビット 8 = Trigin = 常に“1”で、トリガが指定されたことを示します ビット 7-4 = 常に 0 ビット 3-2 = 常に 0 ビット 1 = EnFCont同期トレースモード (Select_Pin_Protocol レジスタのビット 1:0 = 00) では、このビットは強制的に“1”にされ、フォーマッタは自動的に連続モードで有効にされます。非同期モード (Select_Pin_Protocol レジスタのビット 1:0 <> 00) では、このビットの書き込みによって、フォーマッタを有効／無効にできます。 ビット 0 = 常に 0 このレジスタのデフォルト値は 0x102 です。 注: 同期モードでは、TRACECTL ピンがチップ外に配置されないため、フォーマッタは常に連続モードで有効になります。このように、フォーマッタは、トレースパケットの転送元を識別するための制御パケットを挿入します。
0xE0040300	フォーマッタおよびフラッシュステータス	FPU 搭載 Cortex®-M4 では使用されず、常に 0x00000008 が読み出されます。

30.17.10 設定例

- デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットをセットします。
- TPIU カレントポートサイズレジスタに希望する値を書き込みます (デフォルトは 0x1 で 1 ビットのポートサイズ)。
- TPIU フォーマッタおよびフラッシュ制御レジスタに 0x102 (デフォルト値) を書き込みます。
- TPIU 選択ピンプロトコルレジスタに書き込んで同期／非同期モードを選択します。例: 非同期 NRZ モード (UART に類似) の場合は 0x2 を書き込みます。
- DBGMCU 制御レジスタに 0x20 (IO_TRACEN ビット) を書き込み、TRACE I/O を非同期モードとします。このとき、TPIU 同期パケット (FF_FF_FF_7F) が発行されます。
- ITM を設定し、ITM スティムラスレジスタに書き込んで値を出力します。

30.18 DBG レジスタマップ

次の表にデバッグレジスタの一覧を示します。

表 212. DBG レジスタマップとリセット値

アドレス	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE004 2000	DBGMCU_IDCODE	REV_ID																Res.	Res.	Res.	Res.	DEV_ID											
	リセット値 ⁽¹⁾	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0xE004 2004	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_MODE[1:0]	TRACE_IOEN	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP	
	リセット値																									0	0	0			0	0	0
0xE004 2008	DBGMCU_APB1_FZ	Res.	Res.	Res.	Res.	Res.	DBG_CAN2_STOP	DBG_CAN1_STOP	DBG_I2CFMP_SMBUS_TIMEOUT	DBG_I2C3_SMBUS_TIMEOUT	DBG_I2C2_SMBUS_TIMEOUT	DBG_CAN2_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP	
	リセット値						0	0	0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0
0xE004 200C	DBGMCU_APB2_FZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_TIM9_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM8_STOP	DBG_TIM1_STOP		
	リセット値												0	0	0															0	0	0	0

1. リセット値は製品によって異なります。詳細については、[セクション 30.6.1 : MCU デバイス ID コード](#)を参照してください。



31 デバイス電子署名

電子署名は、フラッシュメモリ領域に格納され、JTAG/SWD または CPU を使用して読み出すことができます。電子署名には、出荷時にプログラミングされた識別データが含まれています。このデータを使用すれば、ユーザファームウェアやその他の外部デバイスは、そのインタフェースを STM32F4xx マイクロコントローラの特性に自動的に整合させることができます。

31.1 ユニークデバイス ID レジスタ（96 ビット）

このユニークデバイス識別子は、以下の用途に最適です。

- シリアル番号として使用
- 内部フラッシュメモリをプログラムする前に、このユニーク ID をソフトウェア暗号プリミティブやプロトコルと組み合わせて使用する際に、フラッシュメモリ内のコードのセキュリティを高めるためのセキュリティキーとして使用
- セキュアなブートプロセスなどの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

96 ビットのユニークデバイス識別子は、さまざまな方法で 1 バイト/ハーフワード/ワード単位で読み出し、カスタムアルゴリズムを使用して連結することもできます。

ベースアドレス : 0x1FFF 7A10

アドレスオフセット : 0x00

読み出し専用 = 0xFFFF XXXX (X は出荷時にプログラムされます)

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

ビット 31:0 **U_ID[31:0]** : 31:0 ユニーク ID ビット

アドレスオフセット : 0x04

読み出し専用 = 0xFFFF XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **U_ID[63:32]** : 63:32 ユニーク ID ビット

アドレスオフセット : 0x08

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 U_ID[95:64] : 95:64 ユニーク ID ビット

31.2 フラッシュサイズ

ベースアドレス : 0x1FFF 7A22

アドレスオフセット : 0x00

読み出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 F_ID[15:0] : フラッシュメモリサイズ

このビットフィールドは、デバイスのフラッシュメモリサイズを KB 単位で示します。

31.3 パッケージデータレジスタ

ベースアドレス : 0x1FFF 7BF0
アドレスオフセット : 0x00
読み出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	PKG[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					r	r	r								

- ビット 15:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10:8 **PKG[2:0]** : パッケージタイプ
- 0x111 : UFBGA144/LQFP144
 - 0x110 : 予約済み
 - 0x101 : 予約済み
 - 0x100 : LQFP100
 - 0x011 : UFBGA100
 - 0x010 : WLCSP64
 - 0x001 : LQFP64
 - 0x000 : UFQFPN48
- ビット 7:0 予約済みであり、リセット値に保持する必要があります。

索引

A

ADC_CCR	335
ADC_CR1	324
ADC_CR2	326
ADC_CSR	334
ADC_DR	334
ADC_HTR	330
ADC_JDRx	333
ADC_JOFRx	330
ADC_JSQR	333
ADC_LTR	331
ADC_SMPR1	329
ADC_SMPR2	329
ADC_SQR1	331
ADC_SQR2	332
ADC_SQR3	332
ADC_SR	322

C

CAN_BTR	934
CAN_ESR	933
CAN_FA1R	943
CAN_FFA1R	943
CAN_FiRx	944
CAN_FM1R	942
CAN_FMR	941
CAN_FS1R	942
CAN_IER	931
CAN_MCR	924
CAN_MSR	926
CAN_RDHxR	940
CAN_RDLxR	940
CAN_RDTxR	939
CAN_RF0R	930
CAN_RF1R	930
CAN_RlRx	938
CAN_TDHxR	937
CAN_TDLxR	937
CAN_TDTxR	936
CAN_TlRx	935
CAN_TSR	927
CKGATENR	158
CRC_DR	81
CRC_IDR	82

D

DBGMCU_APB1_FZ	1119
DBGMCU_APB2_FZ	1121
DBGMCU_CR	1117
DBGMCU_IDCODE	1104
DFSDM_CHyAWSCDR	368
DFSDM_CHyCFGR1	365
DFSDM_CHyCFGR2	367
DFSDM_CHyDATINR	370
DFSDM_CHyWDATR	369
DFSDM_FLTxAWCFR	383
DFSDM_FLTxAWHTR	381
DFSDM_FLTxAWLTR	382
DFSDM_FLTxAWSR	383
DFSDM_FLTxCNVTIMR	385
DFSDM_FLTxCR1	371
DFSDM_FLTxCR2	373
DFSDM_FLTxEXMAX	384
DFSDM_FLTxEXMIN	384
DFSDM_FLTxFCR	379
DFSDM_FLTxICR	377
DFSDM_FLTxISR	375
DFSDM_FLTxJCHGR	378
DFSDM_FLTxJDATAR	380
DFSDM_FLTxRDATAR	381
DMA_HIFCR	212
DMA_HISR	210
DMA_LIFCR	211
DMA_LISR	209
DMA_SxCR	213
DMA_SxFCR	218
DMA_SxM0AR	217
DMA_SxM1AR	217
DMA_SxNDTR	216
DMA_SxPAR	217

E

EXTI_EMR	231
EXTI_FTSR	233
EXTI_IMR	231
EXTI_PR	235
EXTI_RTSR	232
EXTI_SWIER	234

F

FLASH_ACR	72
-----------	----



FLASH_CR	75
FLASH_KEYR	73
FLASH_OPTCR	76
FLASH_OPTKEYR	73
FLASH_SR	74
FMPI2C_CR1	690
FMPI2C_CR2	693
FMPI2C_ICR	702
FMPI2C_ISR	700
FMPI2C_OAR1	696
FMPI2C_OAR2	697
FMPI2C_PECR	703
FMPI2C_RXDR	704
FMPI2C_TIMEOUTR	699
FMPI2C_TIMINGR	698
FMPI2C_TXDR	704
FSMC_BCR1..4	269
FSMC_BTR1..4	272
FSMC_BWTR1..4	274

G

GPIOx_AFRH	179
GPIOx_AFRL	178
GPIOx_BSRR	176
GPIOx_IDR	176
GPIOx_LCKR	177
GPIOx_MODER	174
GPIOx_ODR	176
GPIOx_OSPEEDR	175
GPIOx_OTYPER	174
GPIOx_PUPDR	175

I

I2C_CCR	738
I2C_CR1	728
I2C_CR2	730
I2C_DR	733
I2C_OAR1	696, 732
I2C_OAR2	697, 732
I2C_SR1	733
I2C_SR2	737
I2C_TIMEOUTR	699
I2C_TIMINGR	698
I2C_TRISE	739
I2Cx_CR2	693
IWDG_KR	591
IWDG_PR	592
IWDG_RLR	593
IWDG_SR	593

O

OTG_CID	999
OTG_DAIINT	1022
OTG_DAIINTMSK	1023
OTG_DCFG	1017
OTG_DCTL	1018
OTG_DIEPCTL0	1025
OTG_DIEPCTLx	1027
OTG_DIEPEMPMSK	1025
OTG_DIEPINTx	1033
OTG_DIEPMSK	1021
OTG_DIEPTSIZ0	1035
OTG_DIEPTSIZx	1037
OTG_DIEPTXF0	996
OTG_DIEPTXFx	1004
OTG_DOEPCTL0	1029
OTG_DOEPCTLx	1031
OTG_DOEPINTx	1034
OTG_DOEPMSK	1022
OTG_DOEPTSIZ0	1036
OTG_DOEPTSIZx	1038
OTG_DSTS	1020
OTG_DTXFSTSx	1038
OTG_DVBUSDIS	1024
OTG_DVBUSPULSE	1024
OTG_GAHBCFG	982
OTG_GCCFG	998
OTG_GINTMSK	991
OTG_GINTSTS	987
OTG_GLPMCFG	1000
OTG_GOTGCTL	978
OTG_GOTGINT	981
OTG_GRSTCTL	985
OTG_GRXFSIZ	995
OTG_GRXSTSP	994
OTG_GRXSTSR	994
OTG_GUSBCFG	983
OTG_HAIINT	1009
OTG_HAIINTMSK	1009
OTG_HCCHARx	1012
OTG_HCFG	1005
OTG_HCINTMSKx	1015
OTG_HCINTx	1014
OTG_HCTSIZx	1016
OTG_HFIR	1006
OTG_HFNUM	1007
OTG_HNPTXFSIZ	996
OTG_HNPTXSTS	997
OTG_HPRT	1010
OTG_HPTXFSIZ	1004
OTG_HPTXSTS	1008
OTG_PCGCTL	1039

P

PWR_CR	102
PWR_CSR	104

Q

QUADSPI_PIR	305
QUADSPI_PSMAR	304
QUADSPI_PSMKR	304
QUADSPI_ABR	303
QUADSPI_AR	302
QUADSPI_CCR	300
QUADSPI_CR	294
QUADSPI_DCR	297
QUADSPI_DLR	299
QUADSPI_DR	303
QUADSPI_FCR	299
QUADSPI_LPTR	305
QUADSPI_SR	298

R

RCC_AHB1ENR	136
RCC_AHB1LPENR	143
RCC_AHB1RSTR	128
RCC_AHB2ENR	137-138
RCC_AHB2LPENR	144-145
RCC_AHB2RSTR	130-131
RCC_APB1ENR	138
RCC_APB1LPENR	146
RCC_APB2ENR	141
RCC_APB2LPENR	149
RCC_BDCR	151
RCC_CFGR	123
RCC_CIR	126
RCC_CR	119
RCC_CSR	152
RCC_PLLCFGR	121, 155
RCC_SSCGR	154
RNG_CR	393
RNG_DR	395
RNG_SR	393
RTC_ALRMAR	627
RTC_ALRMBR	628
RTC_ALRMBSSR	636
RTC_BKxR	637
RTC_CALIBR	626
RTC_CALR	632
RTC_CR	620
RTC_DR	619
RTC_ISR	622
RTC_PRER	624

RTC_SHIFTR	630
RTC_SSR	629
RTC_TR	618
RTC_TSDR	631
RTC_TSSSR	632
RTC_TSTR	631
RTC_WPR	629
RTC_WUTR	625

S

SDIO_ARG	890
SDIO_CLKCR	888
SDIO_DCOUNT	895
SDIO_DCTRL	893
SDIO_DLEN	893
SDIO_DTIMER	892
SDIO_FIFO	901
SDIO_FIFOCNT	901
SDIO_ICR	896
SDIO_MASK	898
SDIO_POWER	888
SDIO_RESPCMD	891
SDIO_RESPx	892
SDIO_STA	895
SPI_CR1	835
SPI_CR2	837
SPI_CRCPR	840
SPI_DR	840
SPI_I2SCFGR	842
SPI_I2SPR	844
SPI_RXCR	841
SPI_SR	838
SPI_TXCR	841
SYSCFG_CFGR	187
SYSCFG_CFGR2	186
SYSCFG_CMPCR	186
SYSCFG_EXTICR1	184
SYSCFG_EXTICR2	184
SYSCFG_EXTICR3	185
SYSCFG_EXTICR4	185
SYSCFG_MEMRMP	182

T

TIM2_OR	524
TIM5_OR	525
TIMx_ARR	520, 560, 571, 587
TIMx_BDTR	461
TIMx_CCER	455, 518, 559, 570
TIMx_CCMR1	450, 514, 556, 567
TIMx_CCMR2	453, 517
TIMx_CCR1	459, 521, 561, 572



TIMx_CCR2	460, 521, 561
TIMx_CCR3	460, 522
TIMx_CCR4	461, 522
TIMx_CNT	458, 520, 560, 571, 586
TIMx_CR1	440, 505, 549, 564, 583
TIMx_CR2	441, 507, 585
TIMx_DCR	463, 523
TIMx_DIER	445, 510, 552, 565, 585
TIMx_DMAR	464, 523
TIMx_EGR	448, 513, 555, 566, 586
TIMx_PSC	458, 520, 560, 571, 587
TIMx_RCR	459
TIMx_SMCR	443, 508, 551
TIMx_SR	447, 511, 553, 565, 586

U

USART_BRR	786
USART_CR1	787
USART_CR2	789
USART_CR3	790
USART_DR	786
USART_GTPR	792
USART_SR	783

W

WWDG_CFR	600
WWDG_CR	599
WWDG_SR	600

32 改版履歴

表 213. 文書改版履歴

日付	版	変更内容
2015 年 11 月 24 日	1	初版リリース。
2016 年 3 月 23 日	2	<p>更新：</p> <ul style="list-style-type: none"> 表 1：レジスタ境界アドレス 表 24：STM32F412xx の RCC レジスタマップとリセット値 図 2：メモリマップ セクション 6.3：RCC レジスタ セクション 9.5.5：DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0~7) セクション 14.3.4：シリアルチャネルトランシーバ セクション 30.6.1：MCU デバイス ID コード セクション 30.16.2：タイマ、ウォッチドッグ、bxCAN、および I²C のデバッグサポート <p>追加：</p> <ul style="list-style-type: none"> 表 138：プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 100 \text{ MHz}$ または $f_{PCLK} = 50 \text{ MHz}$)、16 倍のオーバーサンプリング 表 139：プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 100 \text{ MHz}$ または $f_{PCLK} = 50 \text{ MHz}$)、8 倍のオーバーサンプリング
2016 年 5 月 26 日	3	<p>更新：</p> <ul style="list-style-type: none"> 表 89：RNG レジスタマップとリセット値 セクション 30.6.1：MCU デバイス ID コード
2016 年 6 月 10 日	4	<p>更新：</p> <ul style="list-style-type: none"> セクション 6.3.13：RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR) セクション 6.3.23：RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR) セクション 6.3.24：RCC 専用クロック設定レジスタ (RCC_DCKCFGR) セクション 6.3.26：RCC 専用クロック設定レジスタ (RCC_DCKCFGR2) セクション 14：デルタシグマモジュレータのデジタルフィルタ (DFSDM) セクション 23.4.8：FMPI2C マスタモード セクション 23.7.2：制御レジスタ 2 (FMPI2C_CR2) <p>追加：</p> <ul style="list-style-type: none"> セクション 6.3.25：RCC クロックゲート有効レジスタ (CKGATENR)

重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点で有効なST の販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してST は一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST 製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

ST およびST ロゴはSTMicroelectronics の商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2017 STMicroelectronics - All rights reserved

